



D5.4

Final report on Multimedia Summarisation, Analysis and Production

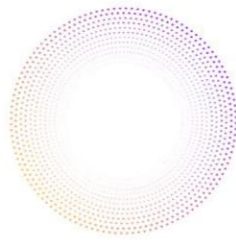
Project Title	AI4Media - A European Excellence Centre for Media, Society and Democracy
Contract No.	951911
Instrument	Research and Innovation Action
Thematic Priority	H2020-EU.2.1.1. - INDUSTRIAL LEADERSHIP - Leadership in enabling and industrial technologies - Information and Communication Technologies (ICT) / ICT-48-2020 - Towards a vibrant European network of AI excellence centres
Start of Project	1 September 2020
Duration	48 months



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 951911

info@ai4media.eu

www.ai4media.eu



Deliverable title	Final report on Multimedia Summarisation, Analysis and Production
Deliverable number	D5.4
Deliverable version	1.0
Previous version(s)	-
Contractual date of delivery	August 31, 2024
Actual date of delivery	September 9, 2024
Deliverable filename	AI4Media_D5_4-final.pdf
Nature of deliverable	Report
Dissemination level	Public
Number of pages	322
Work Package	WP5
Task(s)	T5.1, T5.2, T5.3, T5.4, T5.5, T5.6, T5.7
Partner responsible	AUTH
Author(s)	Evangelos Charalampakis, Ioannis Pitas (AUTH)
Editor	Ioannis Pitas (AUTH)
Project Officer	Evangelia Markidou

Abstract	This document presents the final research outcomes of AI4Media research activities performed in WP5 up to M48. For each WP5 task, the relevant contributions are presented, along with relevant publications, links to software and relevance with AI4media use cases. The document concludes with a short presentation of future research directions.
Keywords	artificial intelligence, media, content analysis, content production, video analysis, video summarisation, key-frame extraction, information retrieval, symbolic reasoning, machine learning, deep learning, learning from scarce data, data-efficient learning, few-shot learning, domain adaptation, semi-supervised learning, clustering, representation learning, multimodal learning, dictionary learning, music similarity analysis, music mixes generation, audio provenance analysis, audio phylogeny analysis, natural language processing, large language models

Copyright

© Copyright 2024 AI4Media Consortium

This document may not be copied, reproduced, or modified in whole or in part for any purpose without written permission from the AI4Media Consortium. In addition to such written permission to copy, reproduce, or modify this document in whole or part, an acknowledgement of the authors of the document and all applicable portions of the copyright notice must be clearly referenced.

All rights reserved.





Contributors

NAME	ORGANIZATION
Ioannis Pitas	AUTH
Evangelos Charalampakis	AUTH
Ioanna Valsamara	AUTH
Christos Papaioannidis	AUTH
Nicu Sebe	UNITN
Marco Formentini	UNITN
Alberto Messina	RAI
Maurizio Montagnuolo	RAI
Stefano Scotta	RAI
Antonios Liapis	UM
Roberto Gallotta	UM
Hannes Fassold	JR
Ioannis Patras	QMUL
Christos Tzelepis	QMUL
Ioannis Maniadis Metaxas	QMUL
Vasileios Mezaris	CERTH
Evlampios Apostolidis	CERTH
Konstantinos Tsigos	CERTH
Julie Tores	3IA
Giuseppe Amato	CNR
Lucia Vadicamo	CNR
Luca Ciampi	CNR
Gabriele Lagani	CNR
Nicola Messina	CNR
Fabrizio Falchi	CNR
Fabrizio Sebastiani	CNR
Alejandro Moreo	CNR
Silvia Corbara	CNR
Patrick Aichroth	FhG-IDMT
Milica Gerhard	FhG-IDMT
Thomas Kölmer	FhG-IDMT
Jakob Abeßer	FhG-IDMT





Peer Reviews

NAME	ORGANIZATION
Vasileios Mezaris	CERTH
Roberto Iacoviello	RAI

Revision History

Version	Date	Reviewer	Modifications
0.1	6/6/2024	Ioannis Pitas	First draft sent to partners for contributions.
0.2	16/7/2024	Vasileios Mezaris, Roberto Iacovello	Version sent for internal review.
0.3	6/9/2024	Ioannis Pitas, Filareti Tsalakanidou	Updated version addressing internal review comments.
1.0	9/9/2024	Ioannis Pitas, Filareti Tsalakanidou	Final version ready for submission.

The information and views set out in this report are those of the author(s) and do not necessarily reflect the official opinion of the European Union. Neither the European Union institutions and bodies nor any person acting on their behalf.





Table of Abbreviations and Acronyms

Abbreviation	Meaning
AA	Authorship Attribution
ACC	Accuracy of Consensus Clustering
ADE	Average Displacement Error
AI	Artificial Intelligence
ALADIN	ALign And DIstill Network
AP	Average Precision
API	Application Programming Interface
AR	Adversarial Reprogramming
ARI	Adjusted Rand Index
ASFF	Adaptive Spatial Feature Fusion
AV	Authorship Verification
AVS	Ad-Hoc Video Search
BCE	Binary Cross Entropy
BN	Batch Normalization
BP	Back Propagation
CAV	Concept Activation Vector
CBIR	Content-Based Image retrieval
CC	Consensus Clustering
CDS	Categorical Distribution Sampler
CJ	Color Jitter
CLIP	Contrastive Language-Image Pre-training
CNN	Convolutional Neural Network
CSVD	Complementary Scene Video Detection
CSVR	Complementary Scene Video Retrieval
DAL	dense absolute localization
DCN	Deep Convolutional Network
DCNN	Deep Convolutional Neural Network
DCT	Discrete Cosine Transforms
DDSP	Differentiated Digital Signal Processing Synthesizer
DIR	Detection and Identification Rate
DL	Deep Learning
DNN	Deep Neural Network
DnS	Distill-and-Select
DPP	Determinantal Point Process
DR	Dropout
DRL	Deep Reinforcement Learning
DSVD	Duplicate Scene Video Detection





DSVR	Duplicate Scene Video Retrieval
FA	Face Alignment
FC	Fully Connected
FCN	Fully Convolutional Network
FDE	Final Displacement Error
FDN	Feedback-Delay Network
FER	Facial Expression Recognition
FID	Frechet Inception Distance
FiLM	Feature-wise Linear Modulation
FIR	Finite Impulse Response
FMF	Face Management Framework
FMoD	Frame Moments Descriptor
FMs	Foundation Models
FP	Forward Pass
FPN	Feature Pyramid Network
FR	Full-Reference
FRA	Facial Region Awareness
FSOD	Few Shot Object Detection
FT	Fine Tuning
GAN	Generative Adversarial Network
GCP	Global Covariance Pooling
GD	Gradient Descent
GHF	Global Hypercolumn Features
GPU	Graphics Processing Unit
GRU	Gated Recurrent Unit
GT	Ground Truth
HED	Holistically-nested Edge Detection
HNSW	Hierarchical Navigable Small World
HPCA	Hebbian Principal Component Analysis
IAA	Inter-annotator agreement
IoU	Intersection-over-Union
ISVD	Incident Scene Video Detection
ISVR	Incident Scene Video Retrieval
KD	Knowledge Distillation
KIS	Know-Item Search
KL	Kullback Leiber
KNN	K-Nearest Neighbor
LLM	Large Language Model
LMM	Large Multimodal Model
LORA	Low-Rank Matrix Adaptation





LPIPS	Learned Perceptual Image Patch Similarity
LR	Logistic Regression
LSC	lifelong search challenges
LSTM	Long Short-Term Memory
LULN	Learning with Unknown Label Noise
MAD	Multilingual Aligned news Dataset
MAE	Mean Absolute Error
MAP	Mean Average Precision
MC	Monte Carlo (MC)-Dropout
MIDI	Musical Instrument Digital Interface
MIR	Music Information Retrieval
MJP	Masked Jigsaw Puzzle
ML	Machine Learning
MLP	Multi-Layer Perceptron
MLS	Moving Least Squares
MMGR	Multimodal Gesture Recognition
MPEG	Moving Picture Experts Group
MSE	Mean Squared Error
MSS	Mult-Scale Spectral
MT	Machine Translation
NCC	Normalised Cross Correlation
NLP	Natural Language Processing
NMI	Normalized Mutual Information
NMS	Non-Maximum Suppression
NOG	Nearest Orthogonal Gradient
NPK	Non Parametric KNN
NR	No-Reference
OLR	Optimal Learning Rate
P-BFT	Practical Byzantine Fault Tolerant
PCA	Principal Component Analysis
PCBM	Post-hoc Concept Bottleneck Model
PCG	Procedural Content Generation
PE	Positional Embedding
PJ	Planckian Jitter
PMC	Parametric Model Classifier
PnP	Perspective-n-Point
PoQI	Proof of Quality Inference
POS	Part-Of-Speech
PSA	Positive Sample Augmentation
PSNR	Peak Signal-to-Noise Ratio





PST	Pest Sticky Traps
QA	Questions and Answers
R-CNN	Recurrent Convolutional Neural Network
REST	Representational State Transfer
RGB	Red Green Blue
RMLP	Recurrent Multi-Layer Perceptron
RNN	Recurrent Neural Network
ROI	Region-Of-Interest
ROS	Robot Operating System
RPN	Region Proposal Network
SAV	Same Author Verification
SD	Stable Diffusion
SFT	Spatial Feature Transform
SGD	Stochastic Gradient Descent
SM	Softmax
SMR	State Machine Replication
SN	Selector Network
SOTA/SoA	State of the Art
SQuAD	Stanford Question Answering Dataset
SR	Super-Resolution
SSIM	Structure Similarity Index
SSL	Semi/Self-Supervised Learning
SSR	Sample Selection and Relabelling
STR	Surrogate Text Representations
SVC	Smart Video Cropping
SVD	Singular Value Decomposition
TBR	Temporal Binary Representation
TFA	Two-stage Fine-tuning Approach
TL	Transfer Learning
TSC	Target-dependent Sentiment Classification
TSD	Temporal Shuffle-Dropout
UAV	Unmanned Aerial Vehicle
UDA	Unsupervised Domain Adaptation
UI	User Interface
UMAP	Uniform Manifold Approximation and Projection
VAE	Variational Auto-Encoder
VBS	Video Browsing Showdown
ViT	Vision Transformer
VP	Variational Predictability
VQA	Video Question Answering





VSR	Video Super-Resolution
WCT	Whitening and Coloring Transform
WE	Word Embedding





Contents

1	Executive Summary	28
2	Introduction	30
2.1	Efficient media analysis and summarization (Task 5.1)	30
2.2	Media content production (Task 5.2)	31
2.3	Learning with scarce data (Task 5.3)	31
2.4	Language analysis in Media (Task 5.4)	32
2.5	Computationally Demanding Learning (Task 5.5)	33
2.6	Music Annotation and Audio Provenance Analysis (Task 5.6)	33
2.7	Research on Large Language Models for the media industry (Task 5.7)	34
3	Media analysis and summarization methods	35
3.1	Overview	35
3.2	Selecting a diverse set of aesthetically-pleasing and representative video thumbnails using reinforcement learning	35
3.2.1	Introduction	35
3.2.2	Methodology	35
3.2.3	Experimental Results	36
3.2.4	Relevance to AI4Media use cases and media industry applications	38
3.2.5	Relevant Publications	38
3.2.6	Relevant software/datasets/other outcomes	38
3.3	Facilitating the production of well-tailored video summaries for sharing on social media	38
3.3.1	Introduction	38
3.3.2	Methodology	38
3.3.3	Relevance to AI4Media use cases and media industry applications	40
3.3.4	Relevant Publications	41
3.3.5	Relevant software/datasets/other outcomes	41
3.4	Using language-guided attention for text-driven video summarization	41
3.4.1	Introduction	41
3.4.2	Methodology	42
3.4.3	Experimental Results	43
3.4.4	Relevance to AI4Media use cases and media industry applications	44
3.4.5	Relevant Publications	45
3.5	Faster than real-time detection of shot boundaries, sampling structure and dynamic keyframes in video	45
3.5.1	Introduction	45
3.5.2	Methodology	46
3.5.3	Initial qualitative evaluation	47
3.5.4	Relevance to AI4Media use cases and media industry applications	47
3.5.5	Relevant Publications	47
3.5.6	Relevant software/datasets/other outcomes	47
3.6	Escaping local minima in deep reinforcement learning for video summarization	47
3.6.1	Introduction	48
3.6.2	Methodology	48
3.6.3	Experimental Results	50
3.6.4	Relevance to AI4Media use cases and media industry applications	50
3.6.5	Relevant Publications	50





3.7	Visual Feature Reprogramming for Neural Video Summarization	51
3.7.1	Introduction	51
3.7.2	Methodology	51
3.7.3	Experimental Results	53
3.7.4	Relevance to AI4Media use cases and media industry applications	54
3.7.5	Relevant Publications	54
3.8	Lightweight Human Gesture Recognition Using Multimodal Features	54
3.8.1	Introduction	54
3.8.2	Methodology	55
3.8.3	Experimental Results	56
3.8.4	Relevance to AI4Media use cases and media industry applications	57
3.8.5	Relevant Publications	57
3.9	Proof of Quality Inference (PoQI): An AI Consensus Protocol for Decentralized DNN Inference Frameworks	57
3.9.1	Introduction	57
3.9.2	Methodology	58
3.9.3	Experimental Results	59
3.9.4	Relevance to AI4Media use cases and media industry applications	60
3.9.5	Relevant Publications	60
3.10	Human face labelling	61
3.10.1	Introduction	61
3.10.2	Methodology	61
3.10.3	Experimental results	62
3.10.4	Relevance to AI4Media use cases and media industry applications	63
3.10.5	Relevant Publications	63
3.11	People@Places and ToDY: Two Datasets for Scene Classification in Media Production and Archiving	64
3.11.1	Introduction	64
3.11.2	People@Places: Dataset for bustle and shot type classification	65
3.11.3	ToDY: Dataset for time of day and season	68
3.11.4	Experimental Results	69
3.11.5	Conclusion	71
3.11.6	Relevance to AI4Media use cases and media industry applications	71
3.11.7	Relevant Publications	71
3.11.8	Relevant software/datasets/other outcomes	71
3.12	Deep Learning to detect objectification in films and visual media	71
3.12.1	Introduction	71
3.12.2	Methodology	71
3.12.3	Experimental Results	73
3.12.4	Conclusion	74
3.12.5	Relevance to AI4Media use cases and media industry applications	74
3.12.6	Relevant Publications	74
3.12.7	Relevant software/datasets/other outcomes	74
4	Media content production	75
4.1	Overview	75
4.2	Photoconsistent and Trajectory Guided Novel-View Synthesis Tool for UAV Cinematog- raphy Based on Autoregressive Transformers	75
4.2.1	Introduction	75





4.2.2	Methodology	75
4.2.3	Experimental Results	76
4.2.4	Conclusion	77
4.2.5	Relevance to AI4Media use cases and media industry applications	77
4.2.6	Relevant Publications	78
4.2.7	Relevant software/datasets/other outcomes	78
4.3	Real-time object geopositioning from monocular target detection/tracking for aerial cinematography	78
4.3.1	Introduction	78
4.3.2	Methodology	78
4.3.3	Experimental Results	79
4.3.4	Conclusion	81
4.3.5	Relevance to AI4Media use cases and media industry applications	82
4.3.6	Relevant Publications	82
4.3.7	Relevant software/datasets/other outcomes	82
4.4	Forecasting in Multimedia	82
4.4.1	Introduction	82
4.4.2	Methodology	82
4.4.3	Experimental Results	84
4.4.4	Conclusion	84
4.4.5	Relevance to AI4Media use cases and media industry applications	85
4.4.6	Relevant Publications	85
4.4.7	Relevant software/datasets/other outcomes	85
4.5	3D, 4D and other Modalities	85
4.5.1	Introduction	86
4.5.2	Experimental Results	96
4.5.3	Conclusion	99
4.5.4	Relevance to AI4Media use cases and media industry applications	100
4.5.5	Relevant Publications	100
4.5.6	Relevant software/datasets/other outcomes	100
4.6	Image and Video Quality Enhancement	100
4.6.1	Introduction	100
4.6.2	Methodology	100
4.6.3	Experimental Results	104
4.6.4	Conclusion	105
4.6.5	Relevance to AI4Media use cases and media industry applications	106
4.6.6	Relevant Publications	106
4.6.7	Relevant software/datasets/other outcomes	106
4.7	Expressive Piano Performance Rendering from symbolic data	106
4.7.1	Introduction	106
4.7.2	Methodology	107
4.7.3	Experimental Results	109
4.7.4	Conclusion	111
4.7.5	Relevance to AI4Media use cases and media industry applications	111
4.7.6	Relevant Publications	111
4.7.7	Relevant software/datasets/other outcomes	111
4.8	Differentiable Piano Synthesizer	111
4.8.1	Introduction	111
4.8.2	Methodology	112





4.8.3	Experimental results	114
4.8.4	Conclusion	116
4.8.5	Relevance to AI4Media use cases and media industry applications	116
4.8.6	Relevant Publications	117
4.8.7	Relevant software/datasets/other outcomes	117
5	Learning from scarce data	118
5.1	Overview	118
5.2	Few-shot Object Detection as a Semi-supervised Learning Problem	118
5.2.1	Introduction	118
5.2.2	Methodology	119
5.2.3	Experimental Results	120
5.2.4	Relevance to AI4Media use cases and media industry applications	122
5.2.5	Relevant Publications	122
5.2.6	Relevant software/datasets/other outcomes	123
5.3	Bioinspired learning approaches to data scarcity	123
5.3.1	Introduction	123
5.3.2	Methodology	123
5.3.3	Experimental results	124
5.3.4	Relevance to AI4Media use cases and media industry applications	124
5.3.5	Relevant Publications	126
5.3.6	Relevant software/datasets/other outcomes	126
5.4	Domain Adaptation and Counting techniques	126
5.4.1	Introduction	126
5.4.2	Methodology	127
5.4.3	Experimental Results	128
5.4.4	Relevance to AI4Media use cases and media industry applications	128
5.4.5	Relevant Publications	128
5.4.6	Relevant software/datasets/other outcomes	130
5.5	Augmentation for Self-supervised and semi-supervised learning	130
5.5.1	Introduction	130
5.5.2	Methodology	130
5.5.3	Experimental results	133
5.5.4	Relevance to AI4Media use cases and media industry applications	137
5.5.5	Relevant Publications	137
5.5.6	Relevant software/datasets/other outcomes	137
5.6	MaskCon: Masked Contrastive Learning for Coarse-Labeled Dataset	137
5.6.1	Introduction and methodology	137
5.6.2	Experimental results	139
5.6.3	Conclusion	139
5.6.4	Relevance to AI4Media use cases and media industry applications	139
5.6.5	Relevant publications	140
5.6.6	Relevant software/datasets/other outcomes	140
5.7	Self-Supervised Video Similarity Learning	140
5.7.1	Introduction and methodology	140
5.7.2	Experimental results	143
5.7.3	Conclusion	144
5.7.4	Relevance to AI4Media use cases and media industry applications	144
5.7.5	Relevant publications	144





5.7.6	Relevant software/datasets/other outcomes	144
5.8	Efficient Data Utilization for enhanced DNN Inference Reliability	144
5.8.1	Introduction	144
5.8.2	Methodology	145
5.8.3	Experimental results	146
5.8.4	Relevance to AI4Media use cases and media industry applications	148
5.8.5	Relevant Publications	148
5.9	Representation learning for knowledge distillation: teaching representations in triplets	148
5.9.1	Introduction	148
5.9.2	Methodology	149
5.9.3	Experimental Results	150
5.9.4	Relevance to AI4Media use cases and media industry applications	152
5.9.5	Relevant Publications	153
5.10	Self-Supervised Facial Representation Learning with Facial Region Awareness	154
5.10.1	Introduction	154
5.10.2	Methodology	154
5.10.3	Experimental results	155
5.10.4	Conclusion	156
5.10.5	Relevance to AI4Media use cases and media industry applications	156
5.10.6	Relevant publications	156
5.10.7	Relevant software/datasets/other outcomes	156
5.11	Self-Supervised Representation Learning with Cross-Context Learning between Global and Hypercolumn Features	157
5.11.1	Introduction and methodology	157
5.11.2	Experimental results	160
5.11.3	Conclusion	160
5.11.4	Relevance to AI4Media use cases and media industry applications	160
5.11.5	Relevant publications	160
5.11.6	Relevant software/datasets/other outcomes	160
5.12	SSR: An Efficient and Robust Framework for Learning with Unknown Label Noise	160
5.12.1	Introduction	160
5.12.2	Methodology	162
5.12.3	Experimental results	164
5.12.4	Conclusion	164
5.12.5	Relevance to AI4Media use cases and media industry applications	165
5.12.6	Relevant publications	165
5.12.7	Relevant software/datasets/other outcomes	165
5.13	Adaptive Soft Contrastive Learning	165
5.13.1	Introduction	165
5.13.2	Methodology	166
5.13.3	Experimental results	168
5.13.4	Conclusion	168
5.13.5	Relevance to AI4Media use cases and media industry applications	169
5.13.6	Relevant publications	169
5.13.7	Relevant software/datasets/other outcomes	169
5.14	DivClust: Controlling Diversity in Deep Clustering	169
5.14.1	Introduction	169
5.14.2	Methodology	169
5.14.3	Experimental results	171





5.14.4	Conclusion	171
5.14.5	Relevance to AI4Media use cases and media industry applications	172
5.14.6	Relevant publications	172
5.14.7	Relevant software/datasets/other outcomes	172
5.15	Efficient Unsupervised Visual Representation Learning with Explicit Cluster Balancing	172
5.15.1	Introduction	172
5.15.2	Methodology	172
5.15.3	Experimental results	174
5.15.4	Conclusion	174
5.15.5	Relevance to AI4Media use cases and media industry applications	175
5.15.6	Relevant publications	175
5.15.7	Relevant software/datasets/other outcomes	175
5.16	Few-shot Object Detection as a Semi-Supervised Learning Problem	175
5.16.1	Introduction	175
5.16.2	Methodology	176
5.16.3	Experimental Results	176
5.16.4	Conclusion	176
5.16.5	Relevance to AI4Media use cases and media industry applications	177
5.16.6	Relevant Publications	177
5.16.7	Relevant software/datasets/other outcomes	177
5.17	Deep Learning for Image Retrieval: An Overview	177
5.17.1	Introduction	177
5.17.2	Literature overview	177
5.17.3	Relevance to AI4Media use cases and media industry applications	181
5.17.4	Relevant Publications	181
5.18	Solutions to large scale Video Browsing and Retrieval	181
5.18.1	Introduction	181
5.18.2	Methodology	181
5.18.3	Experimental Results	184
5.18.4	Relevance to AI4Media use cases and media industry applications	184
5.18.5	Relevant Publications	184
5.18.6	Relevant software/datasets/other outcomes	185
5.19	DnS: Distill-and-Select for Efficient and Accurate Video Indexing and Retrieval	185
5.19.1	Introduction and methodology	185
5.19.2	Experimental results	187
5.19.3	Conclusion	188
5.19.4	Relevance to AI4Media use cases and media industry applications	189
5.19.5	Relevant publications	189
5.19.6	Relevant software/datasets/other outcomes	189
6	Language analysis in Media	190
6.1	Overview	190
6.2	MAD-TSC: A Multilingual Aligned News Dataset for Target-dependent Sentiment Classification	190
6.2.1	Introduction	190
6.2.2	Methodology	190
6.2.3	Experimental Results	191
6.2.4	Conclusion	192
6.2.5	Relevance to AI4Media use cases and media industry applications	193





6.2.6	Relevant Publications	193
6.2.7	Relevant software/datasets/other outcomes	193
6.3	Same or Different? Diff-Vectors for Authorship Analysis	193
6.3.1	Introduction	193
6.3.2	Methodology	194
6.3.3	Experimental results	195
6.3.4	Conclusion	196
6.3.5	Relevance to AI4Media use cases and media industry applications	197
6.3.6	Relevant publications	197
6.3.7	Relevant software/datasets/other outcomes	197
7	Computationally Demanding Learning	198
7.1	Overview	198
7.2	Orthogonal SVD Covariance Conditioning and Latent Disentanglement	198
7.2.1	Introduction and methodology	198
7.2.2	Experiments on Latent Disentanglement	201
7.2.3	Conclusion	202
7.2.4	Relevant publications	203
7.2.5	Relevant software/datasets/other outcomes	203
7.2.6	Relevance to AI4Media use cases and media industry applications	203
7.3	Masked Jigsaw Puzzle: A Versatile Position Embedding for Vision Transformers	203
7.3.1	Introduction	203
7.3.2	Methodology	204
7.3.3	Experimental results	205
7.3.4	Conclusion	206
7.3.5	Relevance to AI4media use cases and media industry applications	207
7.3.6	Relevant publications	207
7.3.7	Relevant software/datasets/other outcomes	207
7.4	4K Video Super-Resolution Detection	207
7.4.1	Introduction	207
7.4.2	Methodology	208
7.4.3	Conclusion	209
7.4.4	Relevance to AI4media use cases and media industry applications	210
7.4.5	Relevant software/datasets/other outcomes	210
8	Music Annotation and Audio Provenance Analysis	211
8.1	Overview	211
8.2	How reliable are posterior class probabilities in automatic music classification?	211
8.2.1	Introduction	211
8.2.2	Methodology	211
8.2.3	Experimental Results	212
8.2.4	Conclusion	213
8.2.5	Relevance to AI4media use cases and media industry applications	214
8.2.6	Relevant Publications	214
8.2.7	Relevant software/datasets/other outcomes	214
8.3	Free-form Text to Music Search Retrieval and Music Tagging	214
8.3.1	Introduction	214
8.3.2	Methodology	215
8.3.3	Experimental Results	215





8.3.4	Conclusion	215
8.3.5	Relevance to AI4media use cases and media industry applications	216
8.4	Audio Provenance Analysis in Heterogeneous Media Content Sets	216
8.4.1	Introduction and methodology	216
8.4.2	Experimental Results	218
8.4.3	Conclusion	219
8.4.4	Relevance to AI4media use cases and media industry applications	219
8.4.5	Relevant Publications	220
8.4.6	Relevant software/datasets/other outcomes	220
9	Research on Large Language Models for the media industry	221
9.1	Overview	221
9.2	LLMs for media content editorial segmentation	221
9.2.1	Challenge	221
9.2.2	Related Work	222
9.2.3	Objectives	223
9.2.4	Methodology	224
9.2.5	Metrics	227
9.2.6	Learning Paradigms	229
9.2.7	Test and Validation	230
9.2.8	Potential impact on AI research/media industry/society	236
9.2.9	Assets released to the community	237
9.2.10	Conclusions/future work	237
9.3	Evaluating LMMs on common sense and factuality	241
9.3.1	Challenge	241
9.3.2	Objectives	241
9.3.3	State of the Art	242
9.3.4	Methodology	243
9.3.5	Experimental results	250
9.3.6	Assets released to the community	251
9.3.7	Potential impact on AI research/media industry/society	252
9.3.8	Conclusions/future work	253
9.4	Use of LLMs for co-creative human-computer interfaces for game design	253
9.4.1	Challenge	254
9.4.2	Related Work	254
9.4.3	Game Domain: Dungeon Despair	256
9.4.4	Objectives	256
9.4.5	Methodology	257
9.4.6	Experimental results	263
9.4.7	Assets released to the community	265
9.4.8	Potential impact on AI research/media industry/society	266
9.4.9	Conclusions/future work	267
10	Conclusion	269





List of Tables

2	Performance comparison of RL-DiVTS with a baseline (random-picking) approach, and a set of SoA video thumbnail selection and summarization methods.	37
3	Comparison of RL-DiVTS and ARL-VTS, in terms of training time and amount of learnable parameters.	37
4	Performance (F-Score (%)) of SUM-GAN-AEE and AC-SUM-GAN on SumMe and TVSum; the last row reports AC-SUM-GAN's performance for augmented training data.	39
5	Performance comparison (F-Score (%)) with state-of-the-art unsupervised approaches after using augmented training data. The reported scores for the listed methods are from the corresponding papers.	41
6	Video aspect ratio transformation performance (IoU (%)) on the RetargetVid dataset.	41
7	Performance comparison (F-Score (%)) on the VRT data.	44
8	The performance (F-Score (%)) of different configurations of the developed network architecture on the VRT data, that relate to different options about the number of heads for the global and local attention mechanisms of the network.	45
9	Comparison of various deep unsupervised video summarization methods on the TVSum and SumMe datasets, using the F-score metric (percentage, higher is better). Best results are in bold.	50
10	Performance comparisons in the canonical setting, in terms of F1-Score (%). All approaches utilize GoogleNet for unimodal video feature extraction and integrate attention mechanisms in their methods. Top 5 performances are ranked.	53
11	Transfer learning method comparison utilizing Fine-tuning and adversarial reprogramming for the SumMe and TVSum datasets, with GoogleNet features. Reported performance is in terms of F1-Score (%). The TL format is "source" → "target" dataset. The trainable parameter number is reported in millions (M). Best average (AVG.) performance is in bold	54
12	Comparison on both evaluation protocols (P-I and P-II) using the AUTH-GESTURE dataset [1].	56
13	Comparison on both evaluation protocols (P-I and P-II) using the UAV-GESTURE dataset [2].	57
14	Accuracy (%) comparison between the PoQI Consensus Protocol and conventional centralized aggregation methods across different datasets, assuming all nodes act honestly, highlighting results obtained from one node.	60
15	Per DNN node accuracy (%) comparison between the PoQI Consensus Protocol and conventional aggregation methods, assuming a subset of faulty nodes acting arbitrarily	61
16	Neural network architecture for the face gender estimation task.	62
17	Comparison of different gender estimation tools applied to television streams.	63
18	Definition of bustle and shot type classes.	66
19	Definition of time of day classes.	69
20	Comparison on Places365 validation (365 classes).	70
21	Performance for bustle and shot type. Toolchain refers to the toolchain in Section 3.11.2, E2E refers to an end-to-end trained classifier.	70
22	Top-1 accuracy for time of day and season classification using EfficientNetB3. The pretraining column specifies the base model being used, ToD+ refers to the time of day annotations after manual revision.	70
23	F1-score on the binary task of objectification detection for models trained with easy or with hard negatives and tested on easy or all negative samples, with standard deviations.	73
24	Metrics comparison based on viewpoints spacing	76
25	Results on SDD. K is the number of predictions generated by the models.	85





26	Quantitative results for depth forecasting after $t+k$ on Cityscapes test set, both at short-term and mid-term predictions, i.e. at $k=5$ and $k=10$ respectively.	86
27	91
28	Florence 4D expression dataset: summary of released data	92
29	Reconstruction accuracy for the <i>unseen individuals and seen motions</i> protocol. We report the Chamfer distance (lower is better). Results for the best and second best performing methods are given in bold and underlined, respectively. Our approach scored the second best accuracy.	97
30	Reconstruction accuracy for the <i>seen individuals and unseen motions</i> protocol. We report the Chamfer distance (lower is better). Results for the best and second best performing methods are given in bold and underlined, respectively. Our approach results in the third best performance.	97
31	Inference time for different configurations of our model using a three-frames buffer. Every test was performed on an Nvidia2080Ti. For the other models it must be noted that they used a 17 frame input sequence to output a frame.	97
32	Reconstruction error (mm) on expression-independent (left) and identity-independent (right) splits	98
33	Absolute accuracy and relative performances of our baseline model over the different data domains and using both labelling versions of NEFER.	99
34	Quantitative comparison between the proposed approach and other state-of-the-art methods for Constant Rate Factor (CRF) 42 on DFD dataset. Best and second best results are in bold and underlined, respectively. \uparrow = higher values are better, \downarrow = lower values are better. 104	
35	Evaluation of image restoration over compression artifacts with GAN using LANBIQUE with different captioning metrics (best results highlighted in bold). For each metric we denote higher(\uparrow) or lower(\downarrow) is better. JPEG q indicates a JPEG compressed image with $QF=q$ (e.g. 10), while (REC q) indicates the corresponding reconstruction using [3]. Captions created from reconstructed images obtain a better score for every metric.	105
36	Evaluation using No-Reference and Full-Reference metrics on MS-COCO. For each metric we denote higher(\uparrow) or lower(\downarrow) is better. JPEG q indicates a JPEG compressed image with $QF=q$ (e.g. 10), while (REC q) indicates the corresponding reconstruction using [3]. NIQE and BRISQUE rate better GAN images than the ORIGINAL. SSIM always rate restored images worse than compressed. PSNR shows negligible improvement. [4] and CIDEr have been used by LANBIQUE-NC respectively as language model and language metric.	105
37	Number of samples from different shares of the data, and additional images needed.	121
38	Results for baselines, soft-sampling and prediction experiments. Average precision (AP) for IoU 50% and 75% as well as average AP are provided for all and novel classes.	122
39	Tiny ImageNet accuracy (top-5) and 95% confidence intervals obtained with a linear classifier on top of various layers, for the various sample efficiency regimes. Results obtained with supervised backprop (BP), VAE-based semi-supervised approach (VAE), Hebbian PCA (HPCA), and HPCA plus Fine Tuning (HPCA+FT) are compared. It is possible to observe that, in regimes where the number of available samples is low (roughly between 1% and 5% of the total available samples), HPCA performs better than BP and VAE approaches in almost all the cases, leading to an improvement up to almost 3% (on layer 3, in the 4% regime) w.r.t. non-Hebbian approaches. HPCA+FT helps to further boost accuracy. 125	
40	Evaluation on downstream tasks. Self-supervised training was performed on IMAGENET at (224×224) and testing performed on the downstream datasets resized to (224×224) . 133	





41	Comparison of several data augmentation approaches for small object detection with FPN, STDnet and CenterNet networks on the small object testing subset of UAVDT. The training phase was conducted by simulating a low instance small object scenario —25% of the UAVDT training videos.	136
42	Results on CIFAR100 dataset.	140
43	State-of-the-art comparison via retrieval mAP (%) and detection μ AP (%) on three evaluation datasets. Bold and <u>underline</u> indicate the best and second best approach, respectively. Missing values are either due to unavailability or unfair comparison due to leak of evaluation data during training.	143
44	Alex-Net Classification Accuracy of Multiple Datasets	147
45	Minimum number of data required to ensure reliable DNN (AlexNet [5]) inferences for each dataset and the percentage in relation to the training dataset size.	148
46	Comparison between the teacher model (ResNet 101) and the student model (ResNet 50) trained under the examined scenarios for different datasets.	151
47	Comparison between the teacher model (ResNet 34) and the student model (ResNet 18) trained under the examined scenarios for different datasets.	151
48	Comparison of different datasets with their respective teacher (T) and student (S) models in the image retrieval task.	152
49	Comparisons on facial expression recognition. We report the Top-1 accuracy on test set. Text denotes text supervision. †: our reproduction using the official codes.	156
50	Comparisons on face alignment. †: our reproduction using the official codes.	157
51	Linear and KNN evaluation results on IN-1K with ResNet-50 backbone. All methods are evaluated with the single-crop setting. Top-1 and Top-5 validation accuracy are reported. †: our reproduction using the official codes. *: results cited from [6].	159
52	Results on CIFAR10/CIFAR100 datasets with synthetic noise.	164
53	Results on ImageNet-1K dataset.	168
54	Results combining DivClust with CC for various diversity targets D^T . We <u>underline</u> DivClust results that outperform the single-clustering baseline CC, and note with bold the best results for each metric across all methods and diversity levels. We emphasize that the NMI in this table measures the similarity between the single clustering produced by each method and the ground truth classes.	171
55	Avg. inter-clustering similarity scores D^R for clustering sets produced by DivClust combined with CC for various diversity targets D^T . The objective of DivClust is that $D^R < D^T$. 171	171
56	Linear & k-NN classification on ImageNet. We report linear and k-NN classification accuracy on ImageNet, along each method’s pretraining batch size and epochs. *TWIST follows standard pretraining with filtered self-labeled training.	174
57	Linear classification with ViT. We report linear classification accuracy on ImageNet for various epochs.	175
58	mAP comparison of our proposed students and re-ranking method against several video retrieval methods on four evaluation datasets. † indicates that the runs are implemented with the same features extracted with the same process as ours. * indicates that the corresponding results are on different dataset split.	188
59	Performance in mAP, storage in KiloBytes (KB) and time in Seconds (Sec) requirements of our proposed students and re-ranking method and several video retrieval implemented with the same features. * indicates that the corresponding results are on different dataset split. 189	189
60	$F1_m$ results for the eight languages included in MAD-TSC. SPC is applied on top of models pretrained specifically for each target language (TG) or with a multilingual corpus (ML) using SPC.	192





61	$F1_m$ results for machine translation languages included in MAD-TSC, compared to the results obtained when without machine translation for English-only (72.3) and monolingual models (fourth row copied from Table 60). Notations: EN - English, TG - target language. The original train/test sets were used if no subscripts are present. DL (DeepL) and $M2M$ [7] subscripts give the machine translation model used. All results are reported with language-specific pretrained models. TSC models are trained with SPC.	192
62	Intrinsic evaluation of DVs: results on closed-set SAV, using vanilla accuracy as the evaluation measure on dataset <code>arXiv</code> . Boldface indicates the best method. The first two methods are DV-based, while the last 2 methods are based on standard representations. Symbols * and ** denote the method (if any) whose score is <i>not</i> statistically significantly different from the best one at $\alpha = 0.05$ (*) or at $\alpha = 0.001$ (**) according to a paired sample, two-tailed t-test. No symbols * and ** appear in this particular table since all differences are statistically significant.	196
63	Comparisons on gradient leakage by analytic attack [8] with ImageNet-1K validation set, where we test (1) ViT-S, DeiT-S and our model in the setting (a); (2) ViT-S, DeiT-S and our model in the setting (b) (<i>i.e.</i> , MJP with $\gamma=0.27$); (3) ablation on without (w/o) using \mathbf{E}_{unk} in setting (a); and (4) Our model in setting (c).	206
64	Explained variance versus PCA projected dimensionality.	206
65	Quantitative metrics for BVI-DVC-SR and BSC-4K datasets	208
66	Subjective Comparison with RAI’s dataset. Mean Opinion Score (MOS) is used, a subjective quality metric rated by human observers.	209
67	Accuracy Metrics for all studied SR and detection methods. * denotes SR methods that are in the training set	209
68	Accuracy values for both datasets and network architectures in %. The accuracy values for single models are provided as mean over all single models with the standard deviation in parentheses.	213
69	Performance Metrics for Different Models	216
70	Example of a <i>scenelet</i> . The orange text is the segment label (corresponding to an edit unit in the programme’s timeline). The green and cyan text represent the audio and video classification as detected by two state-of-the-art zero-shot audio and image classifiers, respectively. The yellow text labels the speaker as per the output of speaker diarization. The plain text is the audio transcription.	225
71	Examples of learned purposes and criteria.	230
72	Legenda for graphic labels in Figures 75, 76 and 77	231
73	Implementation details	232
74	Experimental datasets	234
75	CMMDataset detailed composition.	235
76	Average metrics values on CMMDataset. In green, yellow and red the best, second best and worst values.	237
77	Average metrics values on CMMDocDataset. In green, yellow and red the best, second best and worst values.	237
78	Average metrics values on ANTS Dataset. In green, yellow and red the best, second best and worst values.	238
79	Average metrics values on YT Dataset. In green, yellow and red the best, second best and worst values.	238
80	Assets delivered to the community from LLM research on editorial segmentation.	239
81	Relevant amounts of information about the raw data extrapolated from the video of each city.	246
82	Prompts used to generate the frames descriptions with llava and to generate the questions with llama.	247





83	Accuracy in a 0-shot setting for 7 different CLIP-like models.	251
84	Accuracy in a 0-shot setting for the LLaVA model.	251
85	User requests for test case T5. Each request is submitted sequentially via LLMaker. . .	260
86	Example of metrics values for “ <i>Mad Tinkerer</i> ” in “ <i>Steam Engine Room</i> ” at different levels of context.	262
87	Results for different prompting methods on all test cases averaged from 10 independent runs. Fails measure the number of instances of 10 runs that failed, while Responses and Time (per Request) are averaged from 10 runs and include the 95% Confidence Interval. Responses and Time values with * indicate significantly outperforming all other configurations on this Test Case.	264
88	Summary table from 250 generation tests per context level; results include both a vanilla SD model and a fine-tuned SD model for the task at hand. The number under each column indicates the times a context level yields significantly higher value in the row’s metric, compared to the other context levels. The best context level per metric appears in bold.	266
89	Sprites generated for the entity “Mischievous Imp” using the fine-tuned SD model and the vanilla SD model. The two rooms tested are “Submerged Arena” (top two rows) and “Hieroglyphic Hallway” (bottom two rows) and are similarly generated via the respective SD model.	267
90	Assets delivered to the community from LLMAKER activities.	268





List of Figures

1	The RL-DiVTS network architecture. Orange/gray boxes indicate pretrained/trainable components and white boxes correspond to reward functions. Dashed lines represent iterative processes during a training epoch.	36
2	Processing steps of the proposed Frame Picking mechanism. Dashed lines indicate iterative processes during an episode.	37
3	Instances of the updated and extended UI.	40
4	The utilized language-guided attention mechanism.	42
5	The network architecture of the extended version of PGL-SUM. The extracted representations from the input text are given as input to the added local language-guided multihead attention mechanisms.	43
6	a) Video Summarization pipeline. Training affects exclusively the Regression Head's parameters. b.1) Adversarial Reprogramming pipeline. A trainable adversarial program is applied on each RGB video frame. b.2) RGB video frame reprogramming by weighted addition. c.1) Re-SUM pipeline. A trainable adversarial program is applied on each feature vector output of the pre-trained Feature Extractor. c.2) Visual feature reprogramming by weighted addition.	52
7	The overall architecture of the proposed method.	55
8	Correlation between the output of the RMLP layers and the face gender categories. . .	63
9	Pretrained on fine-grained places categories, the backbone of the network is used to train classification heads for supercategories, bustle and shot type. Bustle/shot type annotations are created automatically (manually corrected for the validation set). . . .	66
10	Dataset creation process for bustle and shot type annotations.	67
11	Location of webcams in the Skyfinder dataset (left), visualization of the times of day used (right).	69
12	In modern film media, the unequal characterization of gender on screen frequently evokes concepts of objectification, such as (A) unequal gaze (<i>Pulp Fiction</i> , 1994), (B) Nudity and submissive postures (<i>Pulp Fiction</i> , 1994), (C) animalisation or infantilisation (<i>Marley and Me</i> , 2008), and (D) transparent clothing, camera framing, domestic gender roles, and voyeurism (<i>Gone Girl</i> , 2014).	72
13	Distribution of visual factors annotated for each level of objectification (HN = Hard negative, NS = Not sure, S = Sure). The percentage of the dataset for each level of objectification as well as the average number of concepts per clip are also shown. (Best viewed in colors)	73
14	Comparison between prediction and ground truth with different gap length.	77
15	Sample output of the proposed 2D detection and tracking software.	79
16	Angular error in estimation over hexacopter-to-target distance.	80
17	Hexacopter-to-target distance error estimation over actual distance to subjects.	80
18	Object of interest actual coordinates and algorithm-generated coordinates comparison. Pins represent the real-world positions of 4 aerial cinematography targets. Red and orange dots are erroneous positional estimations that happen when the target is > 120 meters away from the camera. Yellow and green dots are acceptable or accurate position estimations that are produced with targets 80 to 120 meters away from the camera. . .	81
19	Schematic representation of the proposed Parallel Double Sampling (PDS) module. . .	87
20	Schematic representation of the proposed GCN architecture. <i>Top</i> : Generator architecture; <i>Bottom</i> : Discriminator architecture.	89
21	Plutchik's wheel of emotions [9], illustrating expression relations.	90





22	Sample frames from a generated sequence: (top) the expression passes from neutral to apex and to neutral again; (bottom) the expression passes from neutral to apex for expr. 1, then to apex for expr. 2, and finally to neutral again.	92
23	Examples frames from generated sequences: (top) apex frames of nine expression sequences for subject <i>DAZ_MCH20</i> ; (middle) <i>angry</i> expression for a male (<i>DAZ_M_CH020</i>) and a female (<i>DAZ_F_CH073</i>) subject; (bottom) For subject <i>DAZ_F_CH046</i> the transitions <i>happy-pain</i> , and <i>confident-frown</i> are shown.	93
24	Four samples from the NEFER dataset. First row: happiness; Second row: fear; third row: disgust; fourth row: surprise. Subtle movements are almost invisible with RGB but are emphasized in event frames.	94
25	Examples of detected faces and estimated landmarks on real event videos of NEFER. Better viewed in color on a PC screen. Bounding boxes are shown in green, landmarks are shown in yellow.	95
26	Training pipeline of the piano symbolic performance rendering	108
27	Subjective evaluation results of the piano symbolic performance rendering	110
28	Full architecture of DDSP-Piano-v1.	112
29	Full architecture of DDSP-Piano-v2.	113
30	Subjective evaluation results of DDSP-Piano-v1	115
31	We address the issue of partial annotations in few-shot object detection in a two-stage fine-tuning (TFA) framework. Base setup of the framework (left), extended with soft-sampling to reduce the impact of negative samples caused by missing annotations (middle) and predicting additional annotations (right).	119
32	Some samples of predictions over the target domain. In the four rows, we report some samples of True Positives, True Negatives, False Positives, and False Negatives concerning the best model, i.e., ResNet50 + UDA, for each of the considered source domains (one for each column).	129
33	Downsampling Generative Adversarial Network (DS-GAN) architecture. The generator is trained with HR objects to synthesize small objects. A discriminator between real and fake small objects forces the generator to produce synthetic objects that are increasingly similar to real-world small objects.	132
34	Real HR samples (left), and real LR samples (right).	134
35	FID (a) and classification accuracy (b) for different subsampling methods on the LR testing subset of UAVDT.	135
36	$AP_s^{@[.5,.95]}$ for small object detection in UAVDT for different percentage of training videos with the FPN and STDnet architectures.	136
37	Contrastive learning sample relations using MaskCon (ours) and other learning paradigms when only coarse labels are available. MaskCon are closer to the fine ones.	138
38	Through our experiments, we determine the minimum amount of data required to provide reliable inferences while maintaining high performance. Our method demonstrates that achieving optimal inference accuracy in Big data environments does not require processing the entire dataset; instead, it efficiently delivers reliable inferences using the fewest necessary data points.	145
39	AlexNet classification accuracy scores and the number of samples plot on the F-MNIST dataset [10].	147
40	A visual explanation of a deep metric learning framework using triplet loss.	149





41	Our method achieves knowledge distillation by minimizing the discrepancy between the feature representations of the teacher and the student, while simultaneously learning a representation $(\mathbf{e}_a, \mathbf{e}_p, \mathbf{e}_n)$ that brings “positive” samples closer to an anchor point and pushes “negative” samples further away in the metric space. To facilitate the transfer of structural knowledge and obtain optimal representations, our method uses a triplet-based knowledge distillation loss (TBKD) that combines both the distillation and the triplet loss.	150
42	Overview of the proposed FRA framework. \odot denotes cosine similarity. For each input image \mathbf{x} , its augmented views \mathbf{x}_1 and \mathbf{x}_2 are passed into two network branches to produce the global embeddings \mathbf{z}_1 and \mathbf{z}_2 . In addition, we produce a set of heatmaps \mathbf{M}_1 and \mathbf{M}_2 indicating the local facial regions, via the correlation between the pixel features and “facial mask embeddings” computed from a set of learnable positional embeddings. Then we aggregate the feature map to obtain the local facial embeddings $\{\mathbf{z}_1^m\}$ and $\{\mathbf{z}_2^m\}$. The semantic consistency loss is applied to global embeddings and facial embeddings to maximize the similarity across augmented views. To learn such heatmaps, i.e., <i>facial mask embeddings</i> , we treat the facial mask embeddings as facial region clusters and propose a semantic relation loss to align the cluster assignments of each pixel feature over the facial region clusters between the online and momentum network.	154
43	Overview of the proposed CGH framework. We adopt a knowledge distillation framework where the teacher is the exponential moving average of the student. A heavily corrupted view \mathbf{x}_1 is fed into the student E_s to obtain both a hypercolumn embedding \mathbf{z}_1^h and a global embedding \mathbf{z}_1^g while a weakly augmented view \mathbf{x}_2 is passed to the teacher E_t to obtain a hypercolumn embedding \mathbf{z}_2^h and a global embedding \mathbf{z}_2^g . The embeddings are used to measure the similarity relationships between the augmented views $\mathbf{x}_1, \mathbf{x}_2$ and the samples in the memory bank – this leads to a similarity distribution. We enforce two instance relations alignments: “ <i>global-hypercolumn alignment</i> ” and “ <i>hypercolumn-global alignment</i> ”, which are detailed in the text.	158
44	Different “tigers”.	161
45	A toy example of SSR with a noisy animal dataset.	162
46	Structure of ASCL. When we remove the adaptive relabelling step (indicated in light grey), ASCL can be considered as a general contrastive learning framework such as MoCo.	166
47	Overview of DivClust. Assuming clusterings A and B , the proposed diversity loss L_{div} calculates their similarity matrix S_{AB} and restricts the similarity between cluster pairs to be lower than a similarity upper bound d . In the figure, this is represented by the model adjusting the cluster boundaries to produce more diverse clusterings. Best seen in color.	170
48	Illustration of ExCB’s balancing operator \mathcal{B} for two clusters c_1 (red) and c_2 (blue). $\mathcal{B}(z; s)$ adjusts sample-cluster cosine similarities z according the relative cluster sizes, as measured in s . For smaller clusters the similarities are increased ($z^B > z$), whereas for larger clusters the similarities are decreased ($z^B < z$). The impact, as seen in the figure, is that the boundary between clusters shifts, undersized (oversized) clusters are assigned more (fewer) samples, and clusters become more balanced.	173
49	Comparison between regular and proposed ensembling architectures for FSOD systems.	176
50	An example of image retrieval from an image database (Tiny ImageNet) [11]. Given the query image (left), the images on the right are retrieved.	179
51	A deep CBIR framework.	179





55	Overview of the proposed framework. It consists of three networks: a coarse-grained student S^c , a fine-grained student S^f , and a selector network SN . Processing is split into two phases, <i>Indexing</i> and <i>Retrieval</i> . During indexing (blue box), given a video database, three representations needed by our networks are extracted and stored in a video index, i.e., for each video, we extract a 3D tensor, a 1D vector, and a scalar that captures video self-similarity. During retrieval (red box), given a query video, we extract its features, which, along with the indexed ones, are processed by the SN . It first sends all the 1D vectors of query-target pairs to S^c for an initial similarity calculation. Then, based on the calculated similarity and the self-similarity of the videos, the selector network judges which query-target pairs have to be re-ranked with the S^f , using the 3D video tensors. Straight lines indicate continuous flow, i.e., all videos/video pairs are processed, whereas dashed lines indicate conditional flow, i.e., only a number of selected videos/video pairs are processed. Our students are trained with Knowledge Distillation based on a fine-grained teacher network, and the selector network is trained based on the similarity difference between the two students.	187
56	The covariance conditioning of the SVD meta-layer during the training process in the tasks of decorrelated BN (<i>left</i>) and GCP (<i>Right</i>). The decorrelated BN is based on ResNet-50 and CIFAR100, while ImageNet and ResNet-18 are used for the GCP. . . .	199
57	Illustration of the benefit of orthogonality in latent disentanglement. As revealed in [12, 13], the interpretable directions of latent codes are the eigenvectors of weight or gradient matrices. For non-orthogonal matrices, the principle eigenvector is of the most importance, which would make this direction correspond to many semantic attributes. The other eigenvectors might fail to capture any semantic information. By contrast, the eigenvectors of orthogonal matrices are equally important. The network with the orthogonal weight/gradient is likely to learn more disentangled representations.	200
58	Overview of the EigenGAN architecture.	201
59	Latent traversal on AnimeFace [14]. The EigenGAN has entangled attributes in the identified interpretable directions, while our methods achieve better disentanglement and each direction corresponds to a unique attribute.	201
60	Subtle semantic attributes mined by our method.	202
61	Low-dimensional projection of position embeddings from DeiT-S [15]. (a) The 2D UMAP projection, it shows that reverse diagonal indices have the same order as the input patch positions. (b) The 3D PCA projection, it also shows that the position information is well captured with PEs. Note that the embedding of index 1 (<i>highlighted in red</i>) corresponds to the [CLS] embedding that does not embed any positional information.	204
62	(a) The original input patches; (b) Totally random shuffled input patches; (c) Partially random shuffled input patches; (d) An overview of the proposed MJP. Note that we show the random shuffled patches and its corresponding <i>unknown</i> position embedding in green and the rest part in blue. DAL means the self-supervised <i>dense absolute localization</i> regression constraint.	204
63	Visual comparisons on image recovery with gradient updates [8]. Our proposed DeiT-S+MJP model significantly outperforms the original ViT-S [16] and DeiT-S [15] models.	206
64	Proposed architecture of SR detection module for upscaling detection and recognition.	210
65	Reliability diagrams for all datasets and models	213
66	Audio Provenance Analysis workflow proposed in [17]	217





67	Partial audio matching focuses on identifying reused or recurring segments, sometimes just a few seconds long, within datasets or streams without any prior knowledge of the segments' existence, duration, or frequency of reuse. The image illustrates a dataset containing six audio items, where partial matching successfully detected three different recurring segments, despite having no prior knowledge of the quantity or length of the recurring content.	217
68	Complete audio phylogeny analysis system with transformation prediction via DNN classifier, dissimilarity calculation, and tree reconstruction	218
69	Reconstructed phylogeny trees results for own approach.	219
70	Reconstructed phylogeny trees results for method from [18].	219
71	Reconstructed phylogeny trees results for method from [19].	219
72	Reconstructed phylogeny trees results for own approach with extended set of transformations	220
73	The concept of transmodality.	222
74	Average values of the metrics evaluated on the 1000 simulated reference segmentations against their perturbed versions. In (a) we added/removed segments while in (b) we changed the position of the points of segment change applying a Gaussian perturbation with standard deviation θ to them.	229
78	Segmentation examples (ground truth).	234
79	CMM dataset segments.	240
80	Examples of egocentric raw videos captured in different cities from the City Videos collection.	243
83	Example of frames extracted from a video over which the following questions is asked: <i>"Is it true or false that the statue of the seated figure, possibly a Buddha, appears in the video before the blue plastic chair with a simple design?"</i>	246
84	Platform screenshots.	248
85	Example of the interface shown to the annotators.	249
87	VISIONE software interface, developed by ISTI-CNR in collaboration with RAI as a demonstrator of UC3. This video search and browsing tool can greatly benefit from LMMs capable of understanding long-range temporal dependencies and answering complex factual questions concerning events happening in hours-long videos.	252
88	A screenshot of our <i>Dungeon Despair</i> demo video game. In the "Encounter" tab, the user can see the room, the heroes party (left) and the enemies (right), as well as hovering over their sprites to learn more about each of them. In the "Events" tab, the user can see the combat history as well as any other additional messages. In the "Level" tab, a preview of the map is shown, with information about encounters and other dangers the heroes party may face. Finally, in the "Actions" tab, a series of possible attacks are displayed for the user to choose and progress through the encounter.	257
89	A screenshot of our chat-based level design interface, LLMAKER. On the upper left pane, the preview of the currently selected room. On the lower left pane, the generated level layout, with rooms (larger squares) and corridors (smaller squares). On the right pane, the chat area with the conversation between designer and LLM.	259
90	Pipeline diagram for the generation of an entity leveraging both semantics (room name and description) and image context. We show interim outputs and final output for "Faerie Queen's Guard": <i>"Elite warriors sworn to protect the faerie queen, armed with enchanted blades and shields, and capable of flight"</i> in the "Airship Docking Bay": <i>"A vast hangar housing airships of various sizes, bustling with activity as crews prepare for departure"</i> . The full prompt, with Compel weighting, for the in-painting step is <i>"darkest dungeon, (full body)+++ faerie queen's guard: (elite warriors sworn to protect the faerie queen, armed with enchanted blades and shields, and capable of flight)++, set in airship docking bay: a vast hangar housing airships of various sizes, bustling with activity as crews prepare for departure, masterpiece++, highly detailed+"</i> .	262





1. Executive Summary

Deliverable D5.4 “Final report on Multimedia Summarization, Analysis and Production” is the final deliverable of Work-Package 5 (WP5) “Content-centered AI” of the AI4Media project. WP5 develops novel scientific approaches for content-centered AI, targeting issues in media content production/processing and mostly relying on Deep Neural Networks (DNNs). Its scope broadly covers AI for textual, visual, and audio media, multimedia production, enhancement, and summarization. D5.4 contains results of WP5 activities concerning all tasks of WP5, namely, T5.1 “Media analysis and summarisation”, T5.2 “Media content production”, T5.3 “Learning with scarce data”, T5.4 “Language analysis in Media”, T5.5 “Computationally demanding Learning”, T5.6 “Music Annotation and Audio Provenance Analysis” and T5.7 “Research on Large Language Models for the media industry”.

The deliverable sums up the research carried out in WP5 since the submission of previous deliverables (D5.2 and D5.3) and up to M48. This work has led to several papers published or submitted for publication to well-known, relevant scientific journals and conferences. D5.4 presents the developed methods in their scientific context, the obtained evaluation results, as well as any relevant publications, public software, and datasets. The presented work is clearly aligned with AI4Media use-cases identified in WP8, since WP5 aims at research with a direct application focus. The deliverable concludes with a short discussion of the results of WP5.

T5.1 focuses on AI-based analysis and summarisation of media data, such as images or video. The work presented in this deliverable mainly consists of (a) novel methods (both visual-based and multimodal) for video summarization and thumbnail selection, (b) media analysis through action recognition, video shot detection, representation learning for face labeling and knowledge distillation, and, (c) three new datasets for media analysis.

T5.2 covers a wide range of topics relevant to multimedia content production, including image and video content enhancement techniques, generation of playable video, automated cinematography, generation of synthetic musical mixes, and more. Specifically, in T5.2 work was performed in developing a software for automated target detection for UAV cinematography and tools for enhancing realism in music scores.

T5.3 addresses the limitations of deep learning related to training data scarcity, extending AI applicability to a wider set of media, context, and use cases. Work performed in the final period mainly consists of (a) bioinspired approaches to tackle data scarcity issues, (b) software for automated neural annotation of visual content, (c) multiple novel works on self-supervised representation learning, (d) multimedia content retrieval and (e) few-shot detection.

T5.4 focuses on Language analysis in media and develops methods to improve Natural Language Processing performance and/or to adapt language models to specialized domains. Specifically, the work presented in this deliverable concerns a multilingual dataset with aligned sentiments, and a thorough experimentation of the use of contrastive vectorial representations of text in authorship analysis.

T5.5 investigates techniques to facilitate computationally demanding learning. Methods developed for T5.5 were (a) algorithms that enable DNNs to discover semantic attributes through their training, and (b) two new datasets and a baseline for Super-Resolution methods evaluation.

T5.6 focuses on advanced audio analysis for automatic music annotation and audio provenance analysis mainly relying on DNNs. Work presented in this deliverable mainly consists of: (a) improved algorithms for music classification with enhanced prediction realism, (b) methods for fine-tuning DNNs for music tagging and retrieval, and (c) introduction of two new approaches for the audio provenance analysis task.

T5.7 focuses on new research exploring different aspects of LLM use in the media industry. An internal open call was organized where AI4Media beneficiaries were able to submit proposals for LLM-focused mini projects. An internal evaluation committee evaluated the submitted proposals and selected three of them for funding. The three selected mini projects investigated different aspects of LLMs for the media industry: (a) use of LLMs for co-creative human-computer interfaces for game design, (b)





evaluation of common sense, factuality and biases of LLMs used in Question Answering applications, and (c) application of LLMs for automated editorial content segmentation, based on a trans-modal approach that merges visual, aural, and textual information into a unified textual domain for LLMs to process.

In summary, the work presented in this deliverable has resulted in:

- 52 conference articles (CVPR, ECCV, ACL,) and 27 journal articles (TPAMI, IJCV, ...),
- 33 articles and datasets available in AI4Media's Zenodo collection, and
- 28 open-source software and tools publicly available (e.g., in GitHub).

The remainder of this deliverable is structured as follows. In Section 2, we introduce each task of WP5 and we provide concise descriptions of the presented contributions of each partner, while detailed descriptions of contributions are given for each task in Section 3 (Task 5.1), Section 4 (Task 5.2), Section 5 (Task 5.3), Section 6 (Task 5.4), Section 7 (Task 5.5), Section 8 (T5.6), Section 9 (T5.7). All the methods presented in this deliverable can be applied to media-related areas and applications. Following the description of each method, we additionally present their relevance to WP8 Use Cases. Finally, Section 10 concludes the deliverable by discussing the work covered in this period of WP5.





2. Introduction

AI4Media work-package 5 (WP5) is one of the main research work-packages of the project, with a clear focus on developing novel approaches for content-centered AI. It has the following objectives:

1. Addressing AI issues in content production and processing in textual, visual, and audio media, multimedia production, enhancement, and summarization.
2. Addressing limitations of Deep Learning related to training data scarcity, extending the potential applicability of AI to a wider set of media.
3. Applying Deep Neural Networks (DNNs) to improve tools for analyzing content provenance and reuse.
4. Investigating AI methods with the potential to revolutionize multimedia content production by automating several processes.
5. Achieving improvements in the field of summarization, specifically addressing high-resolution visual data and audio as special cases.
6. Investigating the use of Large Language Models (LLMs) for media industry applications.

This document reports on the activities carried out in all Tasks of WP5, after the submission of previous deliverables and up to M48 of the project. Specifically, the works reported per-Task cover the periods:

- **T5.1:** from M13 to M48
- **T5.2:** from M19 to M48
- **T5.3:** from M13 to M48
- **T5.4:** from M25 to M48
- **T5.5:** from M37 to M48
- **T5.6:** from M13 to M48
- **T5.7:** from M39 to M48

2.1. Efficient media analysis and summarization (Task 5.1)

Task 5.1 (T5.1) “Efficient media analysis and summarization” is a set of hard computational problems, marked by high application relevance in several media domains. Modern AI can provide scientific tools for handling similar problems, with existing methods being able to handle image, video, text, and other data modalities. T5.1 focuses on AI-based media analysis with a special focus on summarization of media data, such as images or video.

For T5.1, CERTH proposed an algorithm for selecting aesthetically pleasing and representative video thumbnails based on diversity using reinforcement learning agents. Moreover, CERTH introduced a Web-based AI tool that automatically generates video summaries that encapsulate the story flow of a full-length video. Finally, CERTH extended and applied their state-of-the-art supervised video summarization algorithm to be able to incorporate information about both visual and textual data.

AUTH introduced a novel loss term regularizer for escaping local minima in DRL agent training for video summarization, and a method that utilizes adversarial training to repurpose pre-trained high-performing video summarizers. For general media analysis, AUTH presented a gesture recognition method that utilized both 2D skeleton sequences and visual cues from the raw video for enhanced accuracy. Moreover, as decentralized and distributed media analysis is an emerging trend, AUTH applied a novel consensus protocol on multimedia datasets, to investigate the fusion of Practical Byzantine Fault Tolerant algorithm within the concept of decentralized DNN inference tasks.





RAI proposed a human face labeling neural method, that uses face embeddings extracted by the RAI Face Management Framework as its input.

JR proposed a novel unified method for video shot detection, sampling structure detection, and dynamic key-frame extraction in a unified way, that runs four times faster than real-time. Also, JR introduced two new datasets: People@Places, an extension of the Places365 dataset, and ToDY, an extension of the Skyfinder dataset, while also providing baselines for both and the toolchains that were used for their creation.

UCA proposed a novel video-interpretation task to detect character objectification in films, by creating a new dataset, ObyGaze12, annotated by experts for objectification concepts.

Progress achieved in these areas is detailed in Section 3 of this document.

2.2. Media content production (Task 5.2)

Task 5.2 (T5.2) “Media content production” of AI4Media investigated multiple aspects of automatic media content production, focusing on the creation, adaptation, and enhancement of media content. The task examines both the pure synthesis of media content exploiting computational methods such as Deep Generative Models as well as methodologies that help in the acquisition and streaming of such content to end-user devices. Research activities in T5.2 cover a range of topics relevant to content production, including but not limited to: cinematography planning, with emphasis on UAV media production; procedural content generation and sound synthesis of musical instruments based on synthetic music sounds.

Specifically, AUTH developed two software tools based on the Robot Operating System (ROS) for automatic content production. The first tool is an end-to-end solution designed to generate novel views from known, but unvisited, viewpoints of a UAV camera, addressing the image extrapolation problem. The second tool integrates monocular visual target detection and tracking with a basic ground intersection model to enable efficient and automated UAV cinematography shot planning.

IRCAM proposed a generative approach that attempts to transform MIDI music scores into human-like performances without supervision on the performance features and reliance on score markings. Moreover, they extended their work on their Differentiable Digital Signal Processing Piano to handle polyphonic MIDI input and reproduce particular properties of the non-digital piano sound.

Progress achieved in these areas is detailed in Section 4 of this document.

2.3. Learning with scarce data (Task 5.3)

Despite their high accuracy, DNNs typically require a lot of high-quality data to be properly trained, making their deployment difficult in cases where large domain-specific datasets are not readily available. Of course, fully supervised learning is the hardest scenario, since all training examples have to be correctly annotated. **Task 5.3 (T5.3)** “Learning with scarce data” of AI4Media aimed to advance the state-of-the-art in methods attempting to facilitate DNN learning from multimedia content in the face of data scarcity. Unsupervised domain adaptation, semi-supervised learning, few-shot learning, data augmentation and unsupervised representation learning approaches fall in this category. They share a common theme of reducing the need for massive, domain-specific, fully and manually annotated training datasets. Methods of this type can increase the applicability of DNNs in real-world scenarios, with T5.3 also partially relating to WP3; notably to transfer learning and learning to count.

JR proposed approaching the few-shot object detection problem through a Semi-Supervised Learning scenario. UPB also attempted to tackle the young few-shot detection problem, by approaching it from a DNN ensemble learning perspective.

CNR explored a bioinspired learning approach to tackle data scarcity. They developed a semi-supervised learning approach that combines Hebbian learning with SGD on object recognition tasks with Deep Convolutional Neural Networks (DCNNs). Then, they developed a scalable solution for Hebbian





synaptic updates and performed exhaustive experimentation on large-scale datasets and architectures that have been out of reach for Hebbian algorithms so far. Moreover, CNR proposed a video search system, VISIONE, that relies on AI to automatically analyze and annotate visual content, hence, providing users with various functionalities to easily search for targeted videos. Furthermore, they extended their work on Unsupervised Domain Adaptation (UDA). They developed novel UDA AI tools for estimating the number of pests in images of sticky chromotropic traps, an object localizer for microscope images of biological structures, and a violence detector for videos. In the context of content-centered AI, CNR co-organized CLEF 2022 (the 13th Conference and Labs of the Evaluation Forum), which took place in Bologna, Italy, in September 2022 ([link](#)).

UNIFI proposed several approaches to tackle learning scenarios with limited access to annotations. Specifically, they studied effective color space augmentation in self-supervised learning, semi-supervised learning for fine-grained classification and finally, they introduced a pipeline for data augmentation based on synthetic object generation.

QMUL also produced several research results in T5.3. They proposed a novel learning scheme, MaskCon, that is aimed at reducing annotation effort, by learning fine-grained representations with a coarsely-labelled dataset. Together with CERTH, they adopted a DNN video similarity architecture and trained it in a self-supervised way, to eliminate the need for video annotations while performing at state-of-the-art level in retrieval and detection benchmarks. QMUL and CERTH also proposed a knowledge distillation approach for efficient and accurate retrieval from videos, by using as a teacher again a pre-trained high-performing DNN architecture for video similarity. Next, QMUL attempted to tackle the human face understanding problem by proposing a facial region awareness (FRA) learning framework that tries to learn consistent global and local facial representations by self-supervised training. Moreover, QMUL proposed a novel self-supervised framework that enforces the consistency of instance relations between low and high-level semantics in contrastive learning settings. In these kinds of contrastive learning problems, QMUL tackled the class-collision deficiency by introducing meaningful inter-sample relations. Furthermore, QMUL introduced a novel problem setting that is learning from annotated data with unknown label noise, and then, they provided a novel selection mechanism that identifies clean samples with correct labels and a relabelling mechanism for the rest. Since clustering has been a staple in machine learning research, QMUL developed a diversity-enforcing clustering loss component that can be used to train models to produce multiple clusterings of controlled diversity with each other, and which explore different partitionings of a given dataset. Finally, they also proposed a framework that uses a novel online cluster balancing method that achieves cluster discrimination in visual representation learning, without requiring large batch size.

Furthermore, AUTH introduced an efficient data utilization strategy for enhanced DNN inference reliability. Additionally, AUTH worked on media analysis issues for image retrieval. To this end, it first produced a survey for Deep Image Retrieval. Then AUTH proposed a method for DNN compression via knowledge distillation based on triplet-based losses.

Progress of T5.3 activities is detailed in Section 5 of this document.

2.4. Language analysis in Media (Task 5.4)

Pre-trained word embeddings (WE) have been the standard way of initializing Natural Language Processing (NLP) neural models. **Task 5.4 (T5.4)** “Language analysis in Media” focuses on automatic language analysis in the media sector and develops methods to improve Natural Language Processing performance and adapt language models to specialized domains that can be directly useful in media organizations and consumers.

Some of the main challenges in this field are: (1) the ever-growing number of new topics and public personalities that emerge in the news and that need to be algorithmically detected; (2) the fine-grained opinions expressed in those documents that need to be accounted for when performing document retrieval. For T5.4, CEA introduced MAD-TSC, the first large multilingual aligned dataset, for target-dependent





sentiment classification. MAD-TSC aligns sentiments expressed toward given entities in a given context, across different languages.

CNR focused instead on a different aspect of Natural Language Processing, namely, the vectorial representations of texts that are given as input (a) to supervised learning algorithms for training a text classifier, and (b) to the text classifiers themselves once they have been trained. CNR presented the first systematic comparison between “standard” vectorial representations of texts and “contrastive” vectorial representations of texts, where the latter are such that a vector represents not one but TWO texts; in other words, a contrastive representation of two texts focuses on representing the DIFFERENCES between these two texts and is geared towards training a classifier that predicts if two texts belong to the same class or not. Since these contrastive representations were first discussed in the field of authorship analysis, CNR’s systematic exploration targets this field; however, since contrastive representations are agnostic with respect to the meaning of the classes they deal with, CNR’s investigation is also “implicitly” relevant to other types of text classification, such as text classification by topic.

The methods explored for T5.4 are detailed in Section 6.

2.5. Computationally Demanding Learning (Task 5.5)

In **Task 5.5 (T5.5)**, “Computationally Demanding Learning”, ways of efficiently handling DNN scaling to larger architecture size and, particularly, larger image resolutions were originally explored. T5.5 scope has been expanded in D5.3 to include efficient training methods and mathematical computations for DNNs.

UNITN proposed a method that enforces matrix factorization (e.g., SVD) layers in DNNs to produce disentangled variable representations, hence, providing DNNs the ability to discover precise semantic attributes. Moreover, UNITN studied a novel positional embedding methodology for Transformers, namely the Masked Jigsaw Puzzle (MJP) positional embedding. MJP enhances Transformer accuracy on image classification benchmarks, while simultaneously improving robustness and privacy preservation under typical gradient attacks.

BSC together with RAI analyzed the performance of diverse Super-Resolution (SR) methods and created two new datasets and a framework for evaluating the performance of different SR methods.

Progress of T5.5 research activities is detailed in Section 7 of this document.

2.6. Music Annotation and Audio Provenance Analysis (Task 5.6)

AI-enabled music analysis is a topic of high industrial relevance that requires special attention. **Task 5.6 (T5.6)** “Music Annotation and Audio Provenance Analysis” dealt with automated music annotation and music similarity analysis, as well as with audio partial matching/reuse detection and audio phylogeny analysis, mainly using novel DNN-based methods. Music similarity analysis refers to the task of quantifying similarity between different music tracks and is particularly significant for the music replacement problem, i.e., when we search for a song as similar as possible to the query track. On the other hand, automated music annotation refers to methods that permit automatic production/extraction of annotation metadata for music tracks (e.g., for training DNNs in a supervised manner). Audio phylogeny implies the automatic detection of processing history relationships between audio items, while partial audio matching involves the detection and temporal localization of arbitrary partial matches between different audio items.

For T5.6, FHG-IDMT examined the reliability of confidence values of DNN outputs in automatic music classification tasks and implemented an algorithm based on Deep Learning techniques to improve estimation reliability and realism. Moreover, FHG-IDMT developed a method of fine-tuning pre-trained DNNs to novel music-relevant domains, to achieve increased accuracy in both the music tagging and the music information retrieval tasks. Finally, FHG-IDMT introduced two novel tasks, essential for an effective audio provenance analysis framework: Provenance Clustering and Provenance Graph Building.





By attending to these new tasks, its approach to the audio provenance analysis outperformed the state-of-the-art while maintaining computational efficiency.

Progress of T5.6 activities is detailed in Section 8 of this document.

2.7. Research on Large Language Models for the media industry (Task 5.7)

Recently, there has been an explosion of Large Language Model (LLM) research. Following this trend, the new **Task 5.7 (T5.7)** “Research on Large Language Models for the media industry” is focused on new research exploring different aspects of LLM use in the media industry. An internal open call was organized where AI4Media beneficiaries were able to submit proposals for LLM-focused mini-projects. An internal evaluation committee evaluated the submitted proposals and selected three of them for funding. Each mini-project received funding of up to 50,000 Euros (coming from the unspent mobility budget).

Specifically, RAI approached the challenging problem of editorial media segmentation. Editorial segmentation of media content is a complex process including cultural and social aspects, operational purposes and other task-specific criteria. Such aspects are difficult to isolate and rigorously define. Editorial segmentation focuses on finding relevant parts (e.g., short clips or larger segments) in multimedia data that can have an independently exploitable nature on publication platforms and that can be identified following multiple segmentation criteria. RAI created a framework that, differently from previous approaches, introduces multimodality at the core of the proposed solution for this task, and grounds its development on a general theoretical/algorithmical formulation.

CNR attempted to address the task of understanding long-range temporal dependencies in untrimmed multicultural video using LLMs. Through a novel benchmark that incorporates true/false questions concerning multiple time-spanning events within a video, they have provided a robust framework for evaluating the current capabilities and limitations of state-of-the-art LMMs on the processing of challenging raw egocentric video data. This benchmark, coupled with automated sentence generation and reliable human labeling, offers a comprehensive evaluation tool that can reveal the deficiencies in existing models’ abilities to handle complex video understanding tasks.

Finally, UM developed LLMAKER, an innovative tool for co-creative video game content design empowered by LLMs. LLMAKER helps the designer and system interaction and is entirely based on natural language, with the LLM translating user queries into properly formatted requests to a back-end system via function calling. UM also proposed a pipeline using stable diffusion models to generate the graphical assets that represent the content being idealized by the user.

The outcomes of the projects are described in Section 9 and ,finally, Section 10 draws conclusions from the presented works.





3. Media analysis and summarization methods

3.1. Overview

Task 5.1 (T5.1) “Efficient media analysis and summarization” focuses on a set of hard computational problems, marked by high application relevance in several domains. Modern AI can provide scientific tools for handling similar problems, with existing methods being able to handle image, video, text, and other data modalities. T5.1 focuses on AI-based media analysis with a special focus on summarization of media data, such as images or video.

Given the broad scope of Task 5.1, the outcomes presented in the following subsections are categorized as follows: first, works that focus on various aspects of automatic video summarization are presented; these are followed by research on general media content analysis.

3.2. Selecting a diverse set of aesthetically-pleasing and representative video thumbnails using reinforcement learning

Contributing partner: CERTH

3.2.1. Introduction

Over the last years there is a tremendous growth of videos over the Web. To facilitate users’ navigation in data collections, most video sharing platforms and social networks represent each video, in their data browsing interfaces, using one or a few thumbnails. However, manually selecting good thumbnails is a tedious and time-consuming process, as it requires a careful inspection of the entire content by a human editor. To accelerate this process, several methods have been proposed over the last years. Early approaches were based on rules about the optimal video thumbnail and extracted low-level (e.g., luminance) and mid-level features (e.g., appearance of faces) to assess frames’ alignment with these rules [20, 21, 22]. More recent methods focused on specific characteristics of the video frames, such as their representativeness and aesthetic quality, and were based either on traditional feature extraction and clustering algorithms [23, 24, 25], or on the use of deep network architectures [26, 27, 28]. Finally, a few multimodal approaches take into account the users’ intentions, expressed as textual queries [29, 30, 31].

Contrary to existing approaches that use similar thumbnail selection criteria [26, 28, 23], we propose a new method (called RL-DiVTS) that considers also the frames’ diversity during the selection and evaluation of video thumbnails. Moreover, instead of assessing frames’ representativeness using Autoencoders [26], Generative Adversarial Networks (GANs) [28], or data clustering algorithms [23], our method uses a tailored reward function. Finally, the proposed method is the first to learn the video thumbnail selection task based on reinforcement learning and a set of reward functions.

3.2.2. Methodology

An overview of the RL-DiVTS network architecture is shown in Figure 1. Given a video of T frames, at training time the Thumbnail Selector assesses the aesthetic quality and importance of each frame with the help of two estimators. The Aesthetic Estimator is a Fully Convolutional Network (FCN) proposed in [32], trained on the AVA dataset [33]. The assessment is done on a per frame basis and results in a sequence of scores that quantify the aesthetic quality of each video frame ($\mathbf{a} = \{a_t\}_{t=1}^T$ with $a_t \in [0,1]$). The evaluation of the frames’ importance is performed by modeling their temporal dependence. The Importance Estimator extracts one feature vector per frame using the pool5 layer of a model of GoogleNet [34] trained on ImageNet [35], and passes the extracted feature vectors ($\mathbf{X} = \{\mathbf{x}_t\}_{t=1}^T$) to a bi-directional LSTM (Long Short-Term Memory) network that models the frames’ temporal dependence and assigns a score to each frame that represents its importance ($\mathbf{i} = \{i_t\}_{t=1}^T$ with $i_t \in [0,1]$). The computed scores about



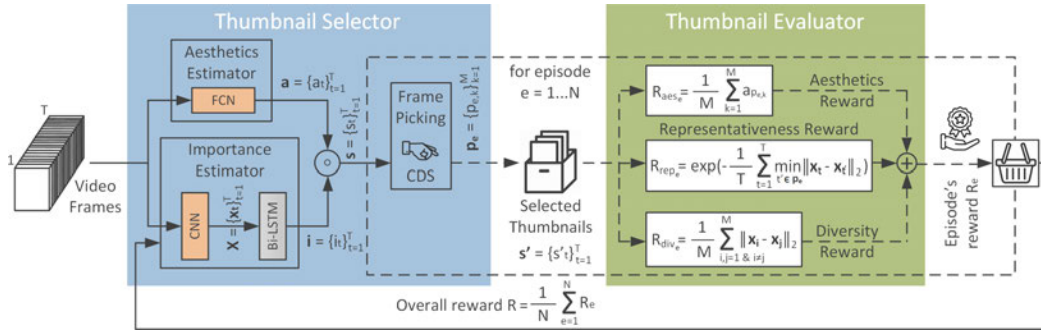


Figure 1. The RL-DiVTS network architecture. Orange/gray boxes indicate pretrained/trainable components and white boxes correspond to reward functions. Dashed lines represent iterative processes during a training epoch.

the frames' aesthetic quality and importance are then fused via their Hadamard product (denoted as \circ in Figure 1), resulting to a new sequence of scores ($\mathbf{s} = \{s_t\}_{t=1}^T$) that is used by the Frame Picking mechanism.

To promote the selection of diverse frames, we introduce a Categorical Distribution Sampler (CDS) that selects frames sequentially by sampling from an appropriate distribution. At the first step, this distribution is based on $\mathbf{f}_1 = \{f_t\}_{t=1}^T$ (computed as $\mathbf{f}_1 = N(\mathbf{s})$, where $N()$ denotes min-max normalization) and the sampling process results in the first picked frame (p_1) and a log probability of picking this sample from the distribution (lp_1). At each subsequent step m (with $m \in [2, M]$), this distribution is based on $\mathbf{f}_m = N(\mathbf{f}_{m-1} \circ (1 - \mathbf{u}_{p_{m-1}}))$, where $\mathbf{u}_{p_{m-1}}$ denotes the row of the frames' (cosine) similarity matrix that corresponds to the picked frame at step $m-1$ (see Figure 2) and the Hadamard product within $N()$ effects a re-weighting, i.e., denotes the selection of frames that are visually-similar to the already picked ones. After the end of the M steps, the Frame Picking mechanism defines a set of picked frames $[p_1, \dots, p_M]$ and a set of log probabilities $[lp_1, \dots, lp_M]$; the latter are used to compute the expected reward in the context of episodic reinforcement learning.

The output of the frame selection process for the e^{th} episode (see $\mathbf{p}_e = \{p_{e,k}\}_{k=1}^M$ in Figure 1) is assessed by the Thumbnail Evaluator, in terms of aesthetic quality, representativeness and diversity, using the reward functions in Eq. 1, 2 and 3, respectively. The overall reward for the current episode is then formed by the weighted sum in Eq. 4 (denoted as \oplus in Figure 1), where D projects R_{rep_e} in the same scale with the other rewards. Finally, the average reward across all the N episodes is the feedback of the Thumbnail Evaluator for the current training sample. To train RL-DiVTS, we use the episodic REINFORCE algorithm [36].

$$R_{aes_e} = \frac{1}{M} \sum_{k=1}^M a_{p_{e,k}} \quad (1)$$

$$R_{rep_e} = \exp\left(-\frac{1}{T} \sum_{t=1}^T \min_{t' \in \mathbf{p}_e} \|\mathbf{x}_t - \mathbf{x}_{t'}\|_2\right) \quad (2)$$

$$R_{div_e} = \frac{1}{M} \sum_{i,j=1}^M \|\mathbf{x}_i - \mathbf{x}_j\|_2 \quad (3)$$

$$R_e = \alpha \cdot R_{aes_e} + \beta \cdot D \cdot R_{rep_e} + \gamma \cdot R_{div_e} \quad (4)$$

3.2.3. Experimental Results

We assessed the performance of RL-DiVTS using the publicly-available datasets and evaluation protocol of [26]. The OVP dataset is composed of 50 videos (up to 3.5 min. long) with diverse content (e.g., documentaries, lecture videos). The YouTube dataset contains 50 videos (up to 9.5 min. long) of different types (e.g., news, TV-shows). Each video has been annotated by 5 users in the form of key-frames. For each

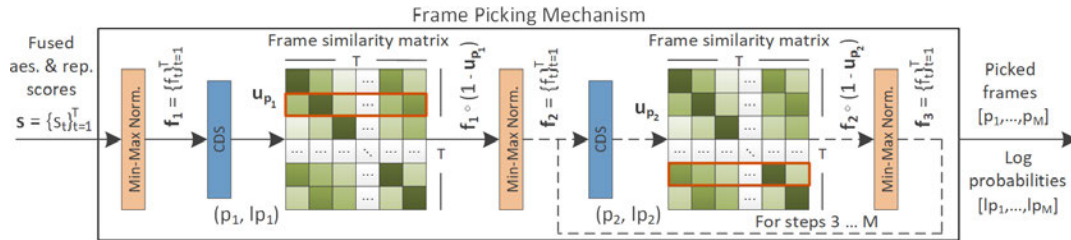


Figure 2. Processing steps of the proposed Frame Picking mechanism. Dashed lines indicate iterative processes during an episode.

video we considered the 3 most selected key-frames among all annotators as its ground-truth thumbnails, and we estimated their similarity with the automatically-selected ones using the Structural Similarity Index (SSIM); we called it a match if SSIM score > 0.7 . For evaluation, we applied the “top-3 matching” approach of [26], that measures the overlap between the top-3 machine- and human-selected thumbnails per video.

We compared RL-DiVTS against a baseline that selects video thumbnails randomly, and a set of SoA video thumbnail selection and summarization methods from the literature. The results of this comparison are shown in Table 2. These results show that RL-DiVTS performs consistently well on both datasets, being by far the top-performing one on OVP and the second best-performing one (slightly below the best one) on YouTube. Moreover, it is more suitable for thumbnail selection, compared to the examined summarization methods. Finally, compared to our previous ARL-VTS method [28], RL-DiVTS brings a noticeable performance improvement on both datasets. Moreover, it exhibits significant gains w.r.t. training time and memory footprint. The results in Table 3 demonstrate that replacing the GAN-based Representativeness Evaluator of ARL-VTS by a reward function, reduced the needed training time by more than 16 and 23 times for the OVP and YouTube videos, respectively. Moreover, this replacement removed the most computationally-demanding module of ARL-VTS, as indicated by the significantly reduced number of learnable parameters of RL-DiVTS.

	OVP	YouTube
Baseline (Random)	8.63 \pm 2.50	4.41 \pm 1.77
AC-SUM-GAN [37]	7.87 \pm 3.41	7.33 \pm 0.70
CA-SUM [38]	7.60 \pm 2.85	8.00 \pm 3.56
Hecate-VTS [23]	11.72	16.47
ReconstSum [26]	12.18	18.25
ARL-VTS [28]	12.50 \pm 3.37	7.83 \pm 1.49
RL-DiVTS (proposed)	25.33 \pm 3.97	17.50 \pm 2.57

Table 2. Performance comparison of RL-DiVTS with a baseline (random-picking) approach, and a set of SoA video thumbnail selection and summarization methods.

	Training time (sec/epoch)		# Param. (in Millions)
	OVP	YouTube	
ARL-VTS [28]	38.41	62.43	28.36
RL-DiVTS	2.33	2.70	12.60

Table 3. Comparison of RL-DiVTS and ARL-VTS, in terms of training time and amount of learnable parameters.



3.2.4. Relevance to AI4Media use cases and media industry applications

The developed method can (i) support the production of highly-summarized versions of a given video and facilitate content curation (Use Case 3: AI in Vision - High Quality Video Production & Content Automation), and (ii) advance both the re-organization of media collections and the content moderation, by providing condensed representations (thumbnails) of the videos for use in browsing interfaces (Use Case 7: AI for (Re-)organisation and Content Moderation).

3.2.5. Relevant Publications

- E. Apostolidis, G. Balaouras, V. Mezaris, I. Patras, "Selecting a Diverse Set of Aesthetically-pleasing and Representative Video Thumbnails using Reinforcement Learning", IEEE Int. Conf. on Image Processing (ICIP 2023), Kuala Lumpur, Malaysia, Oct. 2023. <https://zenodo.org/records/10006049>

3.2.6. Relevant software/datasets/other outcomes

- The code for implementing RL-DiVTS, is available at <https://github.com/e-apostolidis/RL-DiVTS>

3.3. Facilitating the production of well-tailored video summaries for sharing on social media

Contributing partner: CERTH

3.3.1. Introduction

Social media users crave short videos that attract the viewers' attention and can be ingested quickly. Therefore, for sharing on social media platforms, video creators often need a trimmed-down version of their original full-length video. However, different platforms impose different restrictions on the duration and aspect ratio of the video that they accept, e.g., on Facebook's feed videos up to 2 min. appear in a 16:9 ratio, whereas Instagram and Facebook stories usually allow for 20 sec. and are shown in a 9:16 ratio. This makes the generation of tailored versions of video content for sharing on multiple platforms, a tedious task. To tackle this problem, we introduce a web-based tool that harnesses the power of AI to automatically generate video summaries that encapsulate the flow of the story and the essential parts of the full-length video, and are already adapted to the needs of different social media platforms in terms of video length and aspect ratio.

3.3.2. Methodology

The proposed solution (available at <https://idt.iti.gr/summarizer>) is an extension of the web-based service for video summarization, presented in [39]. It is composed of a front-end user interface (UI) that allows interaction with the user, and a back-end component that analyses the video and produces the video summary. The front-end and back-end communication is carried out via REST calls that initiate the analysis, periodically request its status, and, after completion, retrieve the video summary for presentation to the user. Our solution extends our previous technology [39] by i) using an advanced AI-based method for video summarization, ii) integrating an AI-based approach for spatially cropping the video given a target aspect ratio, and iii) supporting customized values for the target duration and aspect ratio of the generated video summary.





Table 4. Performance (F-Score (%)) of SUM-GAN-AAE and AC-SUM-GAN on SumMe and TVSum; the last row reports AC-SUM-GAN’s performance for augmented training data.

Method	SumMe	TVSum
SUM-GAN-AAE [42] (used in [39])	48.9	58.3
AC-SUM-GAN [37]	50.8	60.6
AC-SUM-GAN _{aug} (used now)	52.0	61.0

3.3.2.1. Front-end UI The UI of the proposed solution (see the top part of Figure 3) allows the user to submit a video (that is either available online or locally stored in the user’s device) for summarization, and choose the duration and aspect ratio of the produced summary. This choice can be made either by selecting among presets for various social media channels, or in a fully-custom manner. After initiating the analysis, the user can monitor its progress (see the middle part of Figure 3) and submit additional requests while the previous ones are being analyzed. When the analysis is completed, the original video and the produced summary are shown to the user through an interactive page containing two video players that support all standard functionalities (see the bottom part of Figure 3); through the same page, the user is able to download the produced video summary. Further details about the supported online sources, the permitted file types, and the management of the submitted and produced data can be found in [39].

3.3.2.2. Back-end component The submitted video is initially fragmented to shots using a pre-trained model of the method from [40]. Following, **video summarization** is performed using a pre-trained model of AC-SUM-GAN [37], a top-performing unsupervised video summarization method [41]. This method embeds an Actor-Critic model into a Generative Adversarial Network and formulates the selection of important video fragments as a sequence generation task. At training time, the Actor-Critic model utilizes the Discriminator’s feedback as a reward, to progressively explore a space of states and actions, and learn a value function (Critic) and a policy (Actor) for key-fragment selection. As shown in Table 4, AC-SUM-GAN performs much better than SUM-GAN-AAE [42] (used in [39]), on the SumMe [43] and TVSum [44] benchmark datasets for video summarization. Both methods learn the task using a summary-to-video reconstruction mechanism and the received feedback from an adversarially-trained Discriminator. We argue that the advanced performance of AC-SUM-GAN relates to the use of this feedback as a reward for training an Actor-Critic model and learning a good policy for key-fragment selection, rather than using it as part of a loss function to train a bi-directional LSTM for frame importance estimation. The proposed solution uses a model of AC-SUM-GAN that has been trained using augmented data. Following the typical approach in the literature [41], we extended the pool of training samples of the SumMe and TVSum datasets, by including videos of the OVP and YouTube [45] datasets. This data augmentation process resulted in improvements on both benchmarking datasets (see the last row of Table 4) and to a very competitive performance against several state-of-the-art unsupervised methods from the literature that have been assessed under the same evaluation settings (see Table 5).

To minimize the possibility of losing semantically-important visual content or resulting in visually unpleasant results during **video aspect ratio transformation** (that would be highly possible when using naive approaches, such as fixed cropping of a central area of the video frames, or padding of black borders to reach the target aspect ratio), the proposed solution integrates an extension of the smart video cropping (SVC) method of [54]. The latter starts by computing the saliency map for each chosen frame for inclusion in the video summary. Then, to select the main part of the viewers’ focus, the integrated method applies a filtering-through-clustering procedure on the pixel values of each predicted saliency map. Finally, it infers a single point as the center of the viewer’s attention and computes a crop window for each frame based on the displacement of this point. The applied extension on [54], relates to the use of a state-of-the-art saliency prediction method [55], which resulted in improved performance on





Video summarizer

This service lets you submit videos (up to 10 minutes long) in various formats and generate summaries for use in various social media channels.

Usage instructions
Browser compatibility
Version history

https://www.youtube.com/watch?v=V_tGrD6Pelo

...OR upload local video (mp4, webm, mov, wmv, ogv, mkv, avi)

OPTIONAL: Enter your email to get a 24-hours-active link to the summarization results

I want to generate a summary for posting it on:

Twitter
 Facebook (feed)
 Facebook (stories)
 Instagram (feed)
 Instagram (stories)
 YouTube
 TikTok
 Custom

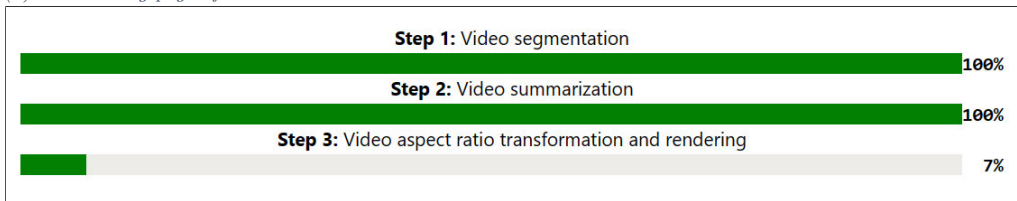
Summary characteristics:

Video length up to seconds

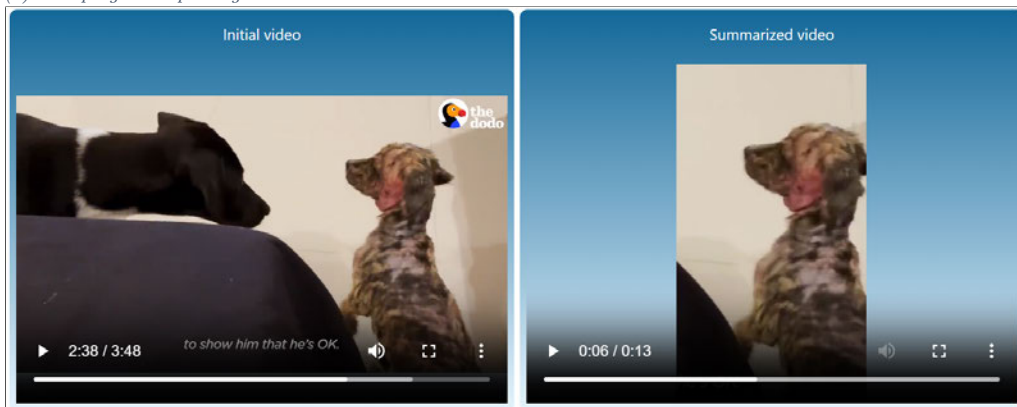
Aspect ratio

Submit

(a) The landing page of the UI.



(b) The progress-reporting bars.



(c) The video players of the page showing the analysis results.

Figure 3. Instances of the updated and extended UI.

the RetargetVid dataset [56]. As shown in Table 6, the averaged Intersection-over-Union (IoU) scores for all video frames have been increased by more than 2 percentage points.

3.3.3. Relevance to AI4Media use cases and media industry applications

The developed method can support the production of different summarized versions of a given video based on the needs of the targeted audiences and according to the specifications of different distribution





Table 5. Performance comparison (F-Score (%)) with state-of-the-art unsupervised approaches after using augmented training data. The reported scores for the listed methods are from the corresponding papers.

Method	SumMe	TVSum
ACGAN [46]	47.0	58.9
RSGN _{unsup} [47]	43.6	59.1
3DST-UNet [48]	49.5	58.4
DSR-RL-GRU [49]	48.5	59.2
ST-LSTM [50]	52.0	58.1
CAAN [51]	50.9	59.8
SUM-GDA _{unsup} [52]	50.2	60.5
SUM-FCN _{unsup} [53]	51.1	59.2
AC-SUM-GAN _{aug}	52.0	61.0

Table 6. Video aspect ratio transformation performance (IoU (%)) on the RetargetVid dataset.

	Method	Worst	Best	Mean
1:3 target aspect ratio	SVC (used in [54])	51.7	53.8	52.9
	SVC _{ext} (used now)	53.8	57.6	55.6
3:1 target aspect ratio	SVC (used in [54])	74.4	77.0	75.3
	SVC _{ext} (used now)	76.3	78.0	77.6

channels (Use Case 3: AI in Vision - High Quality Video Production & Content Automation).

3.3.4. Relevant Publications

- E. Apostolidis, K. Apostolidis, V. Mezaris, "Facilitating the Production of Well-tailored Video Summaries for Sharing on Social Media", Proc. 30th Int. Conf. on MultiMedia Modeling (MMM 2024), Amsterdam, NL, Springer LNCS vol. 14557, pp. 271-278, Jan.-Feb. 2024. <https://zenodo.org/records/13143903>

3.3.5. Relevant software/datasets/other outcomes

- The proposed solution is available at <https://idt.iti.gr/summarizer>.

3.4. Using language-guided attention for text-driven video summarization

Contributing partner: CERTH

3.4.1. Introduction

A generic video summary is a condensed version of the full-length video that conveys the whole story and features the most important scenes. Nevertheless, the importance of different parts of a video is often subjective, and users should have the option of customizing the summary according to their needs, by using textual descriptions to specify what is important to them. Most of the existing models for fully automatic generic summarization [41] have not exploited the use of textual descriptions about the content of the video summary, which can serve as an effective prior for saliency. In AI4Media, we proposed a text-driven method for video summarization, by extending a previous state-of-the-art supervised method



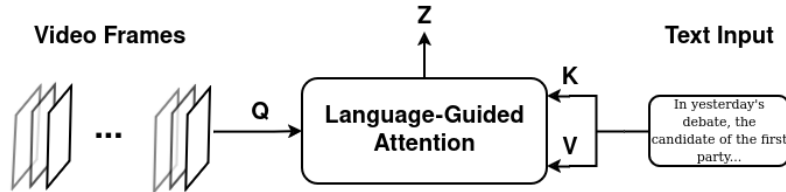


Figure 4. The utilized language-guided attention mechanism.

of CERTH for visual-based video summarization. We adapted and extended the network architecture of PGL-SUM [57] in order to integrate attention mechanisms that incorporate information about both the visual and the textual data. Experimental evaluations and comparisons using data provided by the VRT partner of AI4Media, indicated the competitive performance of our method against other approaches, and the potential of using language-guided attention mechanisms as proposed.

3.4.2. Methodology

The proposed method is an extension of CERTH's PGL-SUM method for visual-based video summarization [57]. PGL-SUM uses global and local multi-head attention mechanisms to discover different modelings of the frames' dependence at different levels of granularity, and estimate the frames' importance. Moreover, the utilized attention mechanisms integrate a component that encodes the temporal position of video frames, which is of major importance when producing a video summary. Experiments on two benchmarking datasets (SumMe and TVSum) demonstrated the effectiveness of PGL-SUM compared to other attention-based methods, and its competitiveness against other state-of-the-art supervised summarization approaches.

To take into account a textual description about the content of the video summary, we extended the network architecture of PGL-SUM, inspired by the CLIP-It! method for language-guided video summarization [58]. In particular, we replaced some of the multi-head attention mechanisms of PGL-SUM by attention mechanisms that fuse information across the visual and textual modalities and infer dependencies across both of them. As depicted in Figure 4, the integrated language-guided attention mechanism, uses the deep representations of the textual description (obtained by the text encoder of a pretrained model of CLIP [59]) as Key and Value, and the deep representations of the visual content of the video frames (obtained by the visual encoder of the same model of CLIP) as Query. Following the processing pipeline of the attention mechanisms of PGL-SUM, the context vectors \mathbf{Z} in the output of each language-guided attention are computed as: $\mathbf{Z} = \text{softmax}(\hat{\mathbf{Q}}\hat{\mathbf{K}}^T)\hat{\mathbf{V}}$, where $\hat{\mathbf{Q}}$, $\hat{\mathbf{K}}$ and $\hat{\mathbf{V}}$ are the obtained embeddings after passing Q, K and V through a triplet of linear layers, respectively.

So, after the applied replacements, the network architecture of the extended version of PGL-SUM became as the one shown in Figure 5. Given a video of T frames and a textual description of S sentences about the content of the video summary, the extended PGL-SUM model initially produces two sets of deep feature representations ($\mathbf{X} = \{\mathbf{x}_t\}_{t=1}^T$ and $\mathbf{Y} = \{\mathbf{y}_s\}_{s=1}^S$) of size D ($\mathbf{x}_t = \{x_{t,i}\}_{i=1}^D$ and $\mathbf{y}_s = \{y_{s,i}\}_{i=1}^D$) for the visual and textual content respectively, using a pretrained CLIP model. These representations form the input to the trainable part of the architecture and follow two different processing paths. One of these paths includes a global multi-head attention mechanism that takes into account only the visual representations and aims to discover different modelings of the frames' dependencies according to the entire frame sequence. The other processing path includes a segmentation step that splits the originally extracted set of deep feature vectors for the video frames (\mathbf{X}) into M (with M equal to 2) consecutive and non-overlapping segments. Each one of these segments (\mathbf{Z}_i , with $i \in [1, M]$) contains the deep feature vectors of the video frames that lie within the segment (\mathbf{Z}_i). Each set of these feature vectors, along the deep representations of the input text (\mathbf{Y}) are then forwarded to a different local language-guided



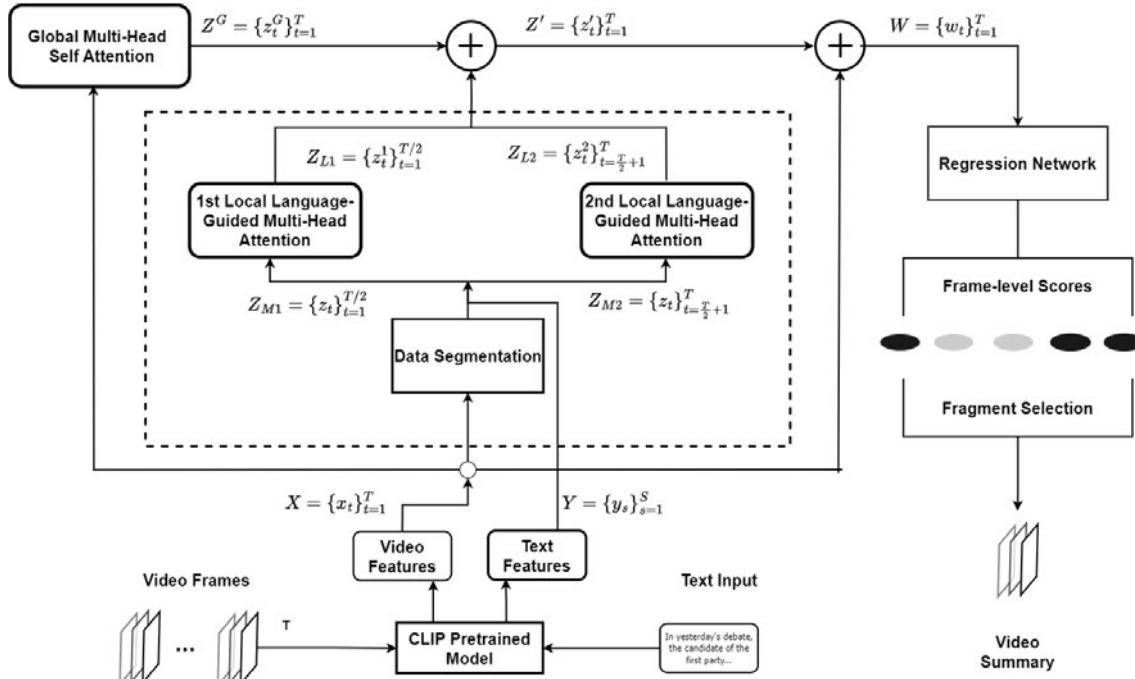


Figure 5. The network architecture of the extended version of PGL-SUM. The extracted representations from the input text are given as input to the added local language-guided multihead attention mechanisms.

multi-head attention mechanism that focuses on the corresponding part of the video. Each of these attention mechanisms produces a new representation of the feature vectors of the frames that lie within the associated segment of the video (Z_i^L , with $i \in [1, M]$). Having available the generated representations from the global (Z^G) and the multiple local multi-head attention mechanisms (Z_i^L , with $i \in [1, M]$), a feature addition process (represented by the \oplus symbol at the left in Figure 5) is applied and produces a new representation for each video frame, that carries information about each frame's importance according to its global and local dependence and its association with the input text ($Z' = \{z_t'\}_{t=1}^T$). The resulting set of representations is then added to the original deep representations of the video frames ($X = \{x_t\}_{t=1}^T$) via a residual skip connection that aims to facilitate back-propagation (this addition is represented by the \oplus symbol at the right in Figure 5). The output of this operation ($W = \{w_t\}_{t=1}^T$) is forwarded to a dropout layer that is followed by a normalization layer. The resulting representation is given as input to the Regressor Network, which produces a set of frame-level scores that indicate the frames' importance.

At training time, we compute the Mean Squared Error between the produced scores about the frames' importance and the ground-truth annotations (also representing frame-level importance). The computed training loss is then back-propagated to compute the gradients and update all the different trainable parts of the architecture. At inference time, the provided importance scores are used to compute fragment-level importance and select the key-fragments of the video and form the video summary given a time budget about its length by solving the Knapsack problem, similarly to most works in the literature [41].

3.4.3. Experimental Results

We assessed the performance of our method using data provided by the VRT partner of AI4Media and the key-fragment-based evaluation protocol proposed in [60], that quantifies the overlap between a machine-generated and a user-defined summary using the F-Score (as percentage). The VRT dataset is



Table 7. Performance comparison (F-Score (%)) on the VRT data.

Method	F-Score (%)
Random Summarizer	20.5
CLIP-It! variant	25.5
PGL-SUM	26.5
PGL-SUM-ext	26.9
PGL-SUM-ext fine-tuned	29.8

composed of 45 “video-script-summary” triplets. The videos are of varying visual content (e.g. interviews, documentaries, news, talk shows, sports), the summaries are summarized versions of the full-length videos with a length that spans from 10% to 45% of the full-length video duration, while the scripts are short descriptions of the visual content of the video summary (could be used as a voice narration while someone is watching the video summary). For training and evaluation, the dataset was divided into 5 non-overlapping splits following the 5-fold cross validation approach; in each split 80% of the videos were used for training and 20% for testing.

The results of our evaluations are presented in Table 7. The performance of a random summarizer on a given video was measured as proposed in [61]. In particular, we initially assigned randomly-created importance scores to the video frames based on a uniform distribution of probabilities. Then, we computed fragment-level scores based on the temporal fragmentation of the video, and formed the summary using the Knapsack algorithm and a predefined time budget about the length of the summary. Random summarization was performed 100 times and we report the average score over these runs. The reported scores in Table 7 show that the performance of the visual-based PGL-SUM method, after being trained on the VRT data, is slightly better than the performance of a variant of the CLIP-It! method (that uses a Regressor Network instead of a Transformer-based scorer) on the same data. Moreover, we observe small gains in performance after replacing each one of the local attention mechanisms of PGL-SUM with the language-guided attention mechanism in Figure 4. Finally, a fine-tuning of a few hyper-parameters of the extended version of PGL-SUM leads to noticeable improvements in the summarization performance.

To fine-tune the developed network architecture, we initially examined different options about the use of the textual data in the attention mechanisms of PGL-SUM. Our experiments indicated that the incorporation of this information only in the local attention mechanisms is the best option, as it led to higher performance. Then, we investigated different options about the number of local attention mechanisms (in the range [1,4]), keeping the number of heads in both global and local mechanisms equal to four. The results of this study showed that the use of two local attention mechanisms is the optimal choice. Based on this finding, we then considered various options about the number of heads in the different attention mechanisms of the network. The results reported in Table 8 indicate that the use of four heads for both global and local attention mechanisms is the best choice as it leads to the highest summarization performance. So, the configuration of the best-performing model of the network architecture is formed as follows: one global 4-head attention mechanism, two local 4-head language-guided attention mechanisms, and no positional encoding.

3.4.4. Relevance to AI4Media use cases and media industry applications

The developed method can (i) assist the summarization of the developed news stories according to user-specified descriptions about the content of the summary (Use Case 2: AI for News - The Smart News Assistant), and (ii) support the production of summarized versions of a given video according to a user-specified script about the summary, and facilitate content curation (Use Case 3: AI in Vision - High Quality Video Production & Content Automation).





Table 8. The performance (F-Score (%)) of different configurations of the developed network architecture on the VRT data, that relate to different options about the number of heads for the global and local attention mechanisms of the network.

	Local	1	2	4	8
Global					
1		26.0	29.5	25.6	26.9
2		25.4	29.2	28.5	26.2
4		25.1	29.2	29.8	27.4
8		23.3	28.3	27.9	27.1

3.4.5. Relevant Publications

We plan to perform a more extended experimentation with additional datasets for video summarization (e.g. SumMe [43], TVSum [44], BLiSS [62], MultiSum [63], Instruct-V2Xum [64], LfVS-T [65]), and publish the outcomes of our study.

3.5. Faster than real-time detection of shot boundaries, sampling structure and dynamic keyframes in video

Contributing partner: JR

3.5.1. Introduction

In order to perform high-level computer vision tasks (like object detection and tracking) on video content, first some fundamental preprocessing tasks have to be performed in advance. Specifically, the content has to be split into individual shots (shot boundary detection), which are usually separated by hardcuts or short dissolves.

It is also crucial to detect the sampling structure of the video. The sampling structure can be progressive, interlaced (each frame contains two fields of half width which are from consecutive timepoints), or 3:2 pulldown (the standard method for converting progressive film content with 24 frames per second to interlaced video content with 60 fields per second). For example, for an interlaced video it is not advisable to use the whole frame, as it will exhibit combing artifacts if motion is present in the scene (as the two fields are from different timepoints). For interlaced video, the information about the field order (which can be upper field first or lower field first) is also desired.

Finally, for extracting an image dataset for training neural networks from an video an algorithm for the extraction of dynamic keyframes is needed. Dynamic keyframes are non-uniformly spaced frames which adapt to the variation in the video. In video segments with high variation (e.g. fast motion scene) more keyframes will be extracted, whereas in a static video segment the spacing between the keyframe will be much larger.

Despite the practical importance of these fundamental video analysis tasks, not a lot of research is devoted to this area (although there are a few patents). For example, for sampling structure detection only a few works (like [66] and [67]) have been proposed in the literature. Addressing this, we propose a novel method which does shot detection, sampling structure detection and dynamic keyframe extraction in an unified way. Due to the unified approach and sparse and selective calculation of the content-based measures, it is able to run four times faster than real-time.





3.5.2. Methodology

All components of our proposed algorithm rely on the following measures:

- $AMM(I, J)$ is the average magnitude of the motion vectors of the motion field calculated between the images I and J . For calculating the motion field, we employ the *Dense Inverse Search* optical flow algorithm from [68]. It runs very fast employing only the CPU and is quite robust against brightness variations. It works well also for large motion in the scene (e.g. sports videos), which is especially important for the shot detection component.
- $SWR(I, J)$ is the image dissimilarity between the *reference* image I and the warped (motion-compensated) J . The warped image is generated by calculating the motion field between both images and warping the image J with the motion field. When the motion compensation works properly, then the warped image should be identical to the reference image. We employ the normalised cross correlation (NCC) similarity measure in order to be invariant against brightness variations due to flicker or camera flashlights.
- $ACT(I, J)$ is the geometric average of $AMM(I, J)$ and $SWR(I, J)$ and measures the *activity* between the images I and J . If the images are very similar and there is not a lot a motion between them, $ACT(I, J)$ will be nearly zero, whereas in the opposite case its value will be high.

The activity between consecutive video frames $ACT(I_t, I_{t+1})$ is calculated always. All other measures $ACT(I_t, I_s)$ are calculated sparsely and selectively, only if they are beneficial to verify a certain hypothesis (e.g. the hypothesis that the current shot is interlaced). We employ a framework where measures are calculated on-demand and cached, in order to ensure that the shot detector and the dynamic keyframes detector do not calculate the same measure twice.

The **shot detector** comprises two phases. The first phase (fast check) does for each frame a check whether it is possible that at this frame a hardcut or short dissolve (consisting of up to 4 frames) occurs. As the first phase is done for every frame, it must be very fast. The second phase (deep check) is only invoked if the first phase decides that at this frame it is possible that a short dissolve occurs. It tests for each $K \in 1, \dots, 4$ whether the hypothesis of a K -frame dissolve at this frame is valid or not (note that $K = 1$ denotes a hardcut). The second phase is not invoked very often and therefore can be computationally much more expensive without impacting the overall runtime negatively. A hypothesis for a K -frame dissolve is verified if the activity $ACT(I_t, I_{t-j})$ between the last frame I_t before the dissolve and its predecessors I_{t-j} is significantly smaller than the activity $ACT(I_t, I_{t+K})$ between the last frame before the dissolve and the first frame after it. This makes sense, as the activity $ACT(I_t, I_{t+K})$ will be high if the frames I_t and I_{t+K} are from different shots.

The **sampling structure detector** utilizes a combination of inter-frame and intra-frame activity measures. Each frame I_t is split into its upper field I_t^u and lower field I_t^l , then we calculate the three basic measures $v_0 = ACT(I_t^u, I_t^l)$, $v_1 = ACT(I_t^u, I_{t+1}^l)$ and $v_2 = ACT(I_t^l, I_{t+1}^u)$. Each sampling structure type has now a very characteristic pattern in the relation of the measures v_0 , v_1 and v_2 . *Progressive content* is characterized by a near-zero value of v_0 , whereas the values v_1 and v_2 are non-zero and approximately equal. *Interlaced content* is characterized by nonzero values of v_0 , v_1 and v_2 . Furthermore, the values v_1 and v_2 are *not* approximately equal, because one corresponds to fields which are significantly further apart in time. For 3:2 *pulldown*, the pattern is more complex, as it depends also on the position of the frame within a *pulldown unit* consisting of 5 frames. By analyzing several frames of the shot statistically, we can determine now whether its sampling structure is progressive, interlaced or 3:2 pulldown.

The principle of the **dynamic keyframe detector** is straightforward. Within a shot, we are accumulating the activity values $ACT(I_t, I_{t+q})$ between consecutive frames. If the accumulated sum is higher than a certain threshold, then we trigger a keyframe for the current frame and set the accumulated sum back to zero.





3.5.3. Initial qualitative evaluation

An initial evaluation of the algorithm has been done with respect to quality (detection capability, robustness, false positives) and runtime. Regarding runtime, the detector is able to process 2K (2048 x 1536) content roughly four times faster than real-time (~11 milliseconds per frame). For 4K video content, the algorithm is roughly three times faster than real-time (~14 milliseconds per frame). The detector implementation uses multiple CPU threads (4 CPU threads), but it does not employ GPU acceleration currently. Regarding quality, the evaluation shows that the developed shot boundary detector algorithm is extremely robust even for challenging content with large camera or object motion, flashlights, flicker, low contrast and the like. A major reason for the robustness of the algorithm is likely the usage of motion compensation backed by a high-quality optical flow algorithm and of a brightness-invariant similarity measure (normalized cross correlation).

A qualitative evaluation of the sampling structure detector on diverse progressive, interlaced and pulldown content shows that the algorithm is able to detect reliably the sampling structure as well as the field order (for interlaced content). Due to the usage of the same robust features like the shot detector, it is also very robust against fast camera or object motion, brightness variations, noise, low contrast and the like. It is able to detect the correct sampling structure also for video content for which it is difficult to discern whether the content is progressive or interlaced due to low amount of motion present in the scene. One example for this type of content are videos from weather panorama cameras, which usually have only minimal horizontal camera panning and often are also of low contrast due to cloudy weather or fog.

Finally, a qualitative evaluation of the dynamic keyframe detector shows that it adapts very well to the dynamic present in the video. So for video content where this is a high amount of motion present (like sports videos), it extracts keyframes in shorter intervals, whereas for content with low motion it extracts the keyframes in larger intervals. Typically, a keyframe is extracted every 8 – 30 frames.

3.5.4. Relevance to AI4Media use cases and media industry applications

This algorithm can be used in all AI4Media use cases where the video has to be split up first into individual shots before the actual processing. The method is also useful to reduce the representation of the video to its keyframes, which is a much more compact representation, for example, for subsequent training of a neural network.

3.5.5. Relevant Publications

- H. Fassold, "Faster than real-time detection of shot boundaries, sampling structure and dynamic keyframes in video", International Conference on Imaging, Signal Processing and Communication 2024 (ICISPC 2024)
Zenodo record: <https://zenodo.org/records/12169764>

3.5.6. Relevant software/datasets/other outcomes

A demo video which demonstrates the algorithm has been generated. A download link to the demo video is given in the Google Cloud at

<https://drive.google.com/file/d/17eFQeMtCusaQZjEb9wf3JX0qyAtrI8Rs/view?usp=sharing>

3.6. Escaping local minima in deep reinforcement learning for video summarization

Contributing partners: AUTH





3.6.1. Introduction

Video summarization is a very important media processing task. State-of-the-art deep neural unsupervised video summarization methods mostly fall under the adversarial reconstruction framework. This employs a Generative Adversarial Network (GAN) structure and Long Short-Term Memory (LSTM) autoencoders during its training stage [69]. It is composed of two main components: the *Summarizer* and the *Discriminator*. The Summarizer contains the *Selector*, the *Encoder* and the *Decoder*. It serves the role of the *Generator*, constructing training data points for the Discriminator (which is a binary classifier), under a GAN setting. These interacting components are LSTM networks and are trained concurrently, using back-propagation and any variant of gradient descent. The Selector generates importance scores that indicate each video frame's appropriateness for inclusion in the summary. Accordingly, given the chosen key-frames, the Autoencoder (Encoder-Decoder) tries to reconstruct the entire, original input video sequence, whilst the Discriminator is trained to distinguish between summary-based reconstructions and original videos. After training, the only component necessary to produce the summary of a new video is the Selector. This fundamental approach concentrates on the ability of the summary to recreate the initial video, but [69] also integrated a Determinantal Point Process (DPP) regularizer [70] during training, in order to obtain more visually diverse key-frames.

Various algorithms have built upon the original method from [69], such as [71], [42] and [72]. The approach most relevant to this paper is [72], which embedded a DRL Actor-Critic agent into the training process. The Actor receives as its initial input state the State Generator's output, i.e., the vector of scalar importance scores for all original video fragments (non-overlapping segments of multiple consecutive video frames), and gradually modifies it; these modifications stem from the actions performed by the agent. The Discriminator's output is exploited as a reward guiding this DRL task. After training has been completed, the Actor and the Selector are the only neural modules required for key-frame extraction in new, test videos. They form a pipeline, with the Actor transforming the output of the Selector into an optimized set of importance scores.

Although this DRL-enhanced variant of the adversarial reconstruction framework has led to state-of-the-art results in unsupervised key-frame extraction, it is well-known that DRL may suffer from entrapment in suboptimal local loss minima [73]. Thus, this paper proposes a new regularizer for escaping local minima, under the guise of a novel loss term introduced into the training process of the Actor-Critic model in any DRL-based baseline method for key-frame extraction. Adding this regularizer during training augments the quality of the DRL agent by compelling it to escape the local minimum it normally tends to converge during training, thus allowing its optimization to reach a better solution. This regularizer may easily be added to the pool of loss functions used for training the overall framework, with its gradient signal specifically influencing the Actor-Critic module. Notably, it is entirely different from common ways of modifying the DRL learning objective for achieving increased exploration during training (e.g., by policy entropy maximization in Soft Actor-Critic [74]).

A quantitative assessment using common protocols on two publicly available datasets, TVSum and SumMe, reveals favorable findings and non-negligible increases over the baseline.

3.6.2. Methodology

The proposed method (\mathcal{L}_{elm}) is a potential addition to any DRL-based deep neural key-frame extraction method that relies on Actor-Critic agents. It is a training-stage regularizer that can be added to the original pool of loss functions influencing the optimization of the Actor-Critic models. It operates by segmenting training into two phases. The first phase is exactly identical to the complete baseline method's training stage, proceeding for K epochs without \mathcal{L}_{elm} . During the second phase, the final trained baseline Actor-Critic model/agent from the K -th epoch of the first phase is exploited as a reference "frozen" model. This second training phase proceeds for another K epochs, but this time with the proposed loss term





\mathcal{L}_{elm} punishing at each iteration the similarity between the reference frozen Actor-Critic model and the current one. This similarity is computed within \mathcal{L}_{elm} in terms of the model parameters. This additional optimization objective forces the agent's training process to search for a different local loss minimum than the one found during the traditional first phase (the first K epochs, without the \mathcal{L}_{elm} regularizer).

This integrated compulsion towards diversity in the parametric structure of the agent, i.e., the difference between the final solution and the previously found reference model, leads at the end of the second training phase to an Actor with better summarization performance. This comes at zero overhead in terms of inference runtime during the test stage. The obvious drawback of an approximately double required time interval compared to baseline (since the baseline architecture is trained for K epochs, i.e., the first phase, while the proposed method needs training for $2K$ epochs, i.e., the first and the second phase) is irrelevant to the actual deployment of a pretrained summarization model.

The proposed method can be applied generally, as an add-on to any DRL-based deep neural key-frame extraction framework, but was actually implemented and evaluated on top of a DRL-enhanced version of the adversarial reconstruction framework [69], namely AC-SUM-GAN [72].

3.6.2.1. ELM Loss The proposed regularizer \mathcal{L}_{elm} can be applied by doubling the number of epochs the baseline DRL-based summarization model is trained. During the initial K epochs, training proceeds as usual. At the end of the K -th epoch, the parameters of the final trained baseline Actor and Critic models are stored as two reference vectors. Subsequently, training resumes with an identical copy of the neural architecture and proceeds for a second phase of K epochs. The only difference from the first phase is that the proposed ELM Loss term is subtracted from the computed Actor Loss and Critic Loss. Thus, during the second phase:

$$\mathcal{L}_{actor,elm} = \mathcal{L}_{actor} - \lambda \mathcal{L}_{elmA} \quad (5)$$

and

$$\mathcal{L}_{critic,elm} = \mathcal{L}_{critic} - \lambda \mathcal{L}_{elmC}, \quad (6)$$

where $\mathcal{L}_{actor,elm}/\mathcal{L}_{critic,elm}$ is the loss function updating the Actor/Critic, respectively, during the novel second training phase, while $\mathcal{L}_{elmA}/\mathcal{L}_{elmC}$ is the version of the proposed regularizer for updating the Actor/Critic, respectively. $\lambda > 0$ is a coefficient adjusting how much the proposed loss term is taken into account by the optimization process, against the task-specific \mathcal{L}_{actor} and \mathcal{L}_{critic} loss terms.

\mathcal{L}_{elm} is computed as the distance between the parameter vector of the current training iteration's agent during the on-going second phase from the respective stored/frozen parameter vector of the reference baseline agent (obtained previously, at the end of the first training phase). This is done separately for the Actor and for the Critic, resulting in the differentiation between \mathcal{L}_{elmA} and \mathcal{L}_{elmC} . Such a distance can be calculated individually for each of the agent's neural layers; these partial distances can then be summed to form the loss value. Below, the difference between \mathcal{L}_{elmA} and \mathcal{L}_{elmC} is ignored for purposes of clearer presentation, since they only deviate to one another with respect to where the stored reference parameter vector came from (the reference Actor/Critic, correspondingly). Thus, the following general definition of \mathcal{L}_{elm} holds:

$$\mathcal{L}_{elm} = \sum_{i=1}^n (1 - S_C(\mathbf{w}_i^C, \mathbf{w}_i^R)), \quad (7)$$

where S_C is the cosine similarity between vectors, $\mathbf{w}_i^C/\mathbf{w}_i^R$ is the parameter vector of the i -th layer of the current/reference agent, respectively, and n is the number of layers in the agent. This formulation does not penalize an agent that is identical to the reference one, but essentially rewards (by reducing the loss value in Eqs. (5) and (6)) one that is different from the reference one in terms of cosine distance. As previously noted, Eq. (7) is obviously implemented differently for the Actor and for the Critic, since these two agents correspond to different stored reference parameter vectors.





3.6.3. Experimental Results

The proposed method was implemented on top of AC-SUM-GAN and evaluated using two publicly available datasets: TVSum [75] and SumMe [43]. The protocol used for evaluation is the key-fragment-based approach with the F-score metric [76], as executed in [72]. The dataset was divided into 5 random splits, while 80% of the videos were used for training and 20% for testing.

The proposed training-stage method, implemented on top of the baseline AC-SUM-GAN, is compared in terms of summarization performance (measured in F-Score) against several unsupervised key-frame extraction approaches in Table 9. To achieve a fair comparison, the performance of the baseline AC-SUM-GAN is reported not only for 100 epochs (as in [72]), but also for 200 epochs, since the proposed method requires double the typical number of training epochs. As it can be seen, adding \mathcal{L}_{elm} to the pool of loss functions leads to non-negligible test-stage gains in F-Score with regard to the directly comparable AC-SUM-GAN-200-epochs competitor, which is the second best performer.

<i>Method</i>	<i>TVSum</i>	<i>SumMe</i>
Online Motion-AE [77]	51.5%	37.7%
SUM-FCN _{unsup} [78]	52.7%	41.5%
DR-DSN [79]	57.6%	41.4%
EDSN [80]	57.3%	42.6%
Unpaired VSN [81]	55.6%	47.5%
PCDL [82]	58.4%	42.7%
ACGAN [83]	58.5%	46.0%
SUM-GAN-sl [71]	58.4%	47.8%
SUM-GAN-AAE [42]	58.3%	48.9%
CSNet [84]	58.8%	51.3%
AC-SUM-GAN [72]	60.6%	50.8%
AC-SUM-GAN (200 epochs)	61.4%	54.4%
Proposed ([72] - $\lambda\mathcal{L}_{elm}$)	62.0%	55.8%

Table 9. Comparison of various deep unsupervised video summarization methods on the TVSum and SumMe datasets, using the F-score metric (percentage, higher is better). Best results are in bold.

3.6.4. Relevance to AI4Media use cases and media industry applications

This method could be useful in UC3 (AI in Vision - High quality Video Production and Content Automation) since it provides means for creating video summaries without human intervention, which greatly speeds up the data gathering phase while reducing the cost of the process. Moreover, this work tackles a problem that is a well known hurdle in Deep Learning applications in general (and especially DRL) that is local minima entrapment.

3.6.5. Relevant Publications

- P. Alexoudi, I. Mademlis and I.Pitas, "Escaping local minima in Deep Reinforcement Learning for video summarization", ACM International Conference on Multimedia Retrieval (ICMR), 2023. Zenodo record: <https://zenodo.org/records/10572182>





3.7. Visual Feature Reprogramming for Neural Video Summarization

Contributing partner: AUTH

3.7.1. Introduction

In recent years, DNNs have played an important role in video summarization. A significant challenge in supervised video summarization stems from the scarcity of annotated training video data due to the arduous and costly video annotation, as is well documented in [85] and [75]. This challenge can be typically tackled by employing transfer learning techniques, which involve fine-tuning a Deep Neural Network (DNN), that is pre-trained on a separate large annotated “source” dataset (e.g., ImageNet [86]), using a small amount of annotated video data samples from a novel dataset of interest [87].

In this section, we present *Re-Summarization* (**Re-SUM**), a novel video summarization method, that utilizes adversarial reprogramming (AR) [88][89] to repurpose common two-stage video summarizers. Contrary to previous works in the field, Re-SUM applies adversarial reprogramming on the *intermediate video frame features* extracted by the feature extraction DNN. Instead of learning a noise vector, called *program*, that augments the input RGB video frames, Re-SUM learns a program that is used to refine the already extracted video frame features. The program-augmented video-features are then utilized by a subsequent Transformer regressor to accurately predict video frame importance scores. Most importantly, Re-SUM enables the Transformer to produce accurate predictions, by learning a simple vector of parameters, that are equal in number to the width of the output layer of the feature extraction DNN ($\approx 1K$), instead of learning millions of parameters, as is typically required in regular DNN training or Transfer Learning approaches. Summed up, the advantages of our methodology, are:

Memory Efficiency Re-SUM is able to produce video summaries *without re-training any of the employed DNNs*, even in the case where the Transformer regressor is *randomly initialized and completely untrained*. Hence, even a complex summarizer, trained on unknown data, can be exploited to perform on a new video dataset without storing extra instances of the full model, as is required in fine-tuning.

Computational Efficiency Applying AR to the raw video inputs requires backward passes through the entire video summarization two-stage pipeline. Re-SUM makes these calculations redundant since it is applied after the video frame feature extraction phase.

Knowledge Preservation Re-SUM introduces a completely modular perturbation that is applied in the inference stage of the video summarization pipeline, after the training phase. Consequently, by simply removing the program application function from the inference, the DNN performance on the original domain it was trained on is always readily accessible.

3.7.2. Methodology

Suppose that we have a neural pipeline $S(\cdot) = T(\|_f f_{CNN}(\cdot; \theta_C^{(I)}); \theta_T^{(A)})$ with its feature extractor f_{CNN} and the frame score regressor T trained asynchronously and separately. $\theta_C^{(I)}$ are the parameters for the CNN feature extractor, trained on a database for image classification, minus the final classification layer. $\theta_T^{(A)}$ denotes the parameters of the Transformer, trained on an unknown, source video summarization dataset \mathbf{A} with c video samples. $(\mathbf{X}_{A_i}, \mathbf{Y}_{A_i})$ denotes the i -th sample in dataset \mathbf{A} , with \mathbf{X}_{A_i} a video sample of N RGB video frames $[\mathbf{x}_{A_{i0}} \mathbf{x}_{A_{i1}} \dots \mathbf{x}_{A_{iN-1}}]$ with $\mathbf{x}_{A_{ij}} \in \mathbb{R}^{k \times k \times 3}$ and \mathbf{Y}_{A_i} its corresponding ground truth frame importance vector.

In this scenario, the objective of Re-SUM is to reprogram $S(\cdot)$ to perform well on a target video summarization dataset \mathbf{B} with r video samples $(\mathbf{X}_{B_i}, \mathbf{Y}_{B_i})$. Each \mathbf{X}_{B_i} is a sequence of N RGB video frames $[\mathbf{x}_{B_{i0}} \mathbf{x}_{B_{i1}} \dots \mathbf{x}_{B_{iN-1}}]$, with $\mathbf{x}_{B_{ij}} \in \mathbb{R}^{k \times k \times 3}$ and \mathbf{Y}_{B_i} their corresponding ground truth frame importance vector. The primary challenge in applying the adversarial program to the input video is that, initially, every video frame needs to get passed through the pre-trained $f_{CNN}(\cdot; \theta_C^{(I)})$ for the video frame



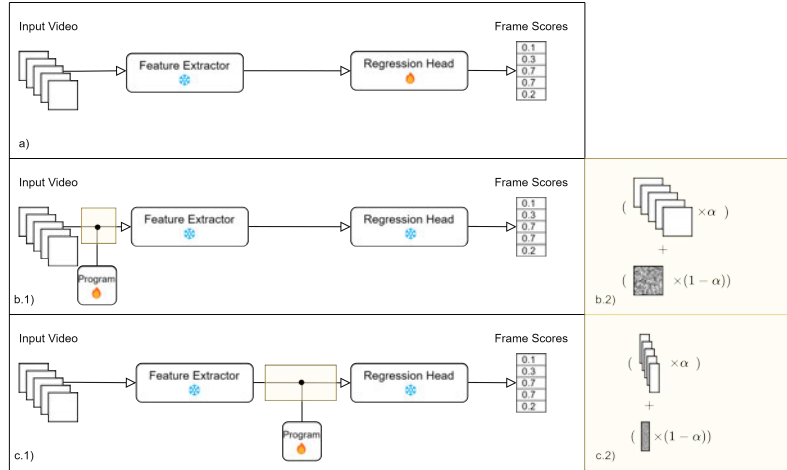


Figure 6. a) Video Summarization pipeline. Training affects exclusively the Regression Head’s parameters. b.1) Adversarial Reprogramming pipeline. A trainable adversarial program is applied on each RGB video frame. b.2) RGB video frame reprogramming by weighted addition. c.1) Re-SUM pipeline. A trainable adversarial program is applied on each feature vector output of the pre-trained Feature Extractor. c.2) Visual feature reprogramming by weighted addition.

feature vectors to be acquired. It is obvious, that in the case of typical AR, a back-propagation through the whole video summarization pipeline is needed for the optimization of the adversarial parameters \mathbf{W} . Consequently, if the choice of the feature extraction DNN was a task-agnostic high-performing large Neural architecture, optimization complexity would suffer an upscaling of extreme magnitude, due to the required gradient computations.

Re-SUM mitigates this issue by learning to reprogram only the task-specific part of the pipeline T for target dataset \mathbf{B} , without taking into account neither the feature extractor f_{CNN} inputs nor its parameters $\theta_C^{(I)}$. This is achieved by learning a new adversarial program:

$$\mathbf{W} \in \mathbb{R}^d, \quad (8)$$

that is applied to the **output** of f_{CNN} as shown in Figure 6. The learnable parameters \mathbf{W} are static and of one dimension. Since, f_{CNN} outputs for both source and target datasets have identical dimensions, padding isn’t needed. As a result A_{pad} will not be used and A'_{add} can be introduced:

$$A'_{add}(\mathbf{X}_{B_i}; \mathbf{W}) = \|\|_j((1-\alpha)\mathbf{W} + \alpha f_{CNN}(\mathbf{x}_{B_{ij}}; \theta_C^{(I)})), \quad j=0, \dots, N-1, \quad (9)$$

with $f_{CNN}(\mathbf{x}_{B_{ij}}; \theta_C^{(I)}) = \mathbf{h}_{B_{ij}}$ the intermediate feature vector for the video frame $\mathbf{x}_{B_{ij}}$. Finally, based on eq. 9 the optimization problem of Re-SUM is represented as:

$$\mathbf{W}^{(B)} = \underset{\mathbf{W}}{\operatorname{argmin}} \frac{1}{r} \sum_{i=0}^{r-1} (T(A'_{add}(\mathbf{X}_{B_i}; \mathbf{W}); \theta_T^{(A)}) - \mathbf{Y}_{B_i})^2, \quad j=0, \dots, N-1, \quad (10)$$

and an estimation of the importance score vector for a full video \mathbf{X}_{B_i} from the target dataset \mathbf{B} after $\mathbf{W}^{(B)}$ has converged, is:

$$\hat{\mathbf{Y}}_{B_i} = T(\|\|_j A'_{add}(\mathbf{X}_{B_{ij}}; \mathbf{W}^{(B)}); \theta_T^{(A)}). \quad (11)$$

Following [90] it is also possible for Re-SUM to reprogram an untrained Transformer head $T(\cdot; \theta_T)$, with θ_T the learnable parameters of T that are untrained and randomly initialized [91]. The loss function

Table 10. Performance comparisons in the canonical setting, in terms of F1-Score (%). All approaches utilize GoogleNet for unimodal video feature extraction and integrate attention mechanisms in their methods. Top 5 performances are ranked.

Method	TVSum	rank	SumMe	rank
CLIP-It! _{gnet} [92]	64.2	1	51.6	2
PGL-SUM [93]	61.0		55.6	1
M-AVS [94]	61.0		44.4	
DSNet _{a-b} [95]	62.1	5	50.2	5
DSNet _{a-f} [95]	61.9		51.2	3
VASNet [96]	61.4		49.7	
DASP [97]	63.6	2	45.5	
ST_{med}	63.5	3	49.4	
Re-SUM_{med}	63.1	4	50.5	4

presented either in the trained 10 or in the untrained scenario, can be denoted as $L'_{MSE}(\mathbf{W})$. Then, the chain rule yields:

$$\frac{dL'_{MSE}}{d\mathbf{W}} = \frac{\partial L'_{MSE}}{\partial T} \cdot \frac{\partial T}{\partial A_{add}} \cdot \frac{\partial A_{add}}{\partial \mathbf{W}} \quad (12)$$

We can observe in eq. 12, that, by using Re-SUM, f_{CNN} does **not** participate in the gradient computation for the Re-SUM loss function minimization. As a result, redundant gradient computations are completely omitted since the optimization process is dependant only on the important task-specific parameters θ_T of the video frame score regressor T .

3.7.3. Experimental Results

All aspects of the evaluation process follow established common protocols, utilizing 2 public benchmark datasets: TVSum [75] and SumMe [85]. To apply the proposed Re-SUM method on the Transformer architectures, we had to aggregate GoogleNet’s last projection layer output $\mathbf{h} \in \mathbb{R}^{1024}$ for each video frame, with a learnable adversarial perturbation of the same dimensionality. Consequently, in the training phase we only optimized a global learnable program $\mathbf{W} \in \mathbb{R}^{1024}$ that consisted of approximately 1K parameters. These experiments resulted in video summarizers Re-SUM_{tiny}, Re-SUM_{small}, Re-SUM_{med}, Re-SUM_{large}, that are randomly initialized Transformer regressors, augmented with our optimized adversarial module.

In addition, the best performing models obtained by the typical supervised learning process (ST_{med}) and by Re-SUM (Re-SUM_{med}) were compared against unimodal, state-of-the-art, video summarization methods that utilize the same feature extraction CNN as the one utilized in the proposed method. The results reported in Table 10 show that even the performance of the reprogrammed untrained Transformer regressor (Re-SUM_{med}) is highly competitive to state-of-the-art methods (top 4). This observation is particularly intriguing given the fact that Re-SUM_{med} utilizes only the trainable parameters \mathbf{W} of the program, which are many orders of magnitude fewer compared to competitors.

Moreover, in order to demonstrate the effectiveness of Re-SUM, we compare it against the typical fine-tuning transfer learning (TL) approach. In this setting, we assume a DNN video summarizer ST_{med} , pre-trained to an unknown “source” dataset of high quality and quantity, that we want to re-use on a small, novel “target” dataset. Then, for the transfer learning case, we followed two different approaches: a) we fine-tuned the entire architecture on the “target” dataset, and b) we fine-tuned only the final linear layers of the architecture. For the Re-SUM case, we simply applied AR on the ST_{med} that was pre-trained on the “source” dataset to repurpose it for the “target” one.



Table 11. Transfer learning method comparison utilizing Fine-tuning and adversarial reprogramming for the SumMe and TVSum datasets, with GoogLeNet features. Reported performance is in terms of F1-Score (%). The TL format is “source” → “target” dataset. The trainable parameter number is reported in millions (M). Best average (AVG.) performance is in bold.

	ST_{med} -NoFT	ST_{med} -lastFT (0.13)	ST_{med} -FullFT (6.2)	Re-SUM $_{med}$ (0.001)
TVSum → SumMe (re-test) TVSum	43.13	47.27	48.49	50.80
SumMe → TVSum (re-test) SumMe	58.90	60.40	60.92	62.82
		42.97	41.72	49.94

This procedure was repeated with the roles of the datasets reversed, i.e., SumMe initially played the role of the “source” dataset and TVSum the role of the “target” dataset, and, subsequently, we re-did the experiments with the reverse “source” - “target” scheme. The results of this study are presented in Table 11 where Re-SUM’s superiority is apparent.

3.7.4. Relevance to AI4Media use cases and media industry applications

This method is useful in UC3 (AI in Vision - High quality Video Production and Content Automation) since it provides means for creating video summaries by re-purposing high-performing video summarizers pre-trained on an unknown source dataset, by training a modular lightweight feature perturbation (number of parameters $\approx 1K$).

3.7.5. Relevant Publications

- E. Charalampakis, C. Papaioannidis, and I. Pitas, "Visual Feature Reprogramming for Neural Video Summarization", Under review

3.8. Lightweight Human Gesture Recognition Using Multimodal Features

Contributing partners: AUTH

3.8.1. Introduction

Human gesture recognition is a very important tool in human-computer or human-robot interaction. In many cases, such algorithms may need to be executed on systems with limited computational capabilities, due to size or weight constraints. Therefore, such approaches aim to recognize gestures by exclusively processing the available skeleton sequences, discarding all visual appearance information contained within the input RGB content. However, this approach is not optimal, since the extracted skeletons are only a rough representation of the human body, as they consist only of a specific, predefined set of body joints (e.g., shoulders, wrists, knees, etc.). Certain gestures that involve finer motions, such as finger movements or very small displacements (slight body joint rotations/translations) cannot be distinguished effectively between each other. While methods that include palm/finger joints in the 2D/3D skeletons [98] could offer a sufficient solution, their predictions can only be trusted when the person performing the gestures is really close to the camera and his/her fingers are clearly visible. In real-world scenarios, this is rarely the case.

This paper proposes a gesture recognition method, called Multimodal Gesture Recognition (*MMGR*), that analyzes both 2D skeleton sequences and visual information obtained from the input RGB videos to



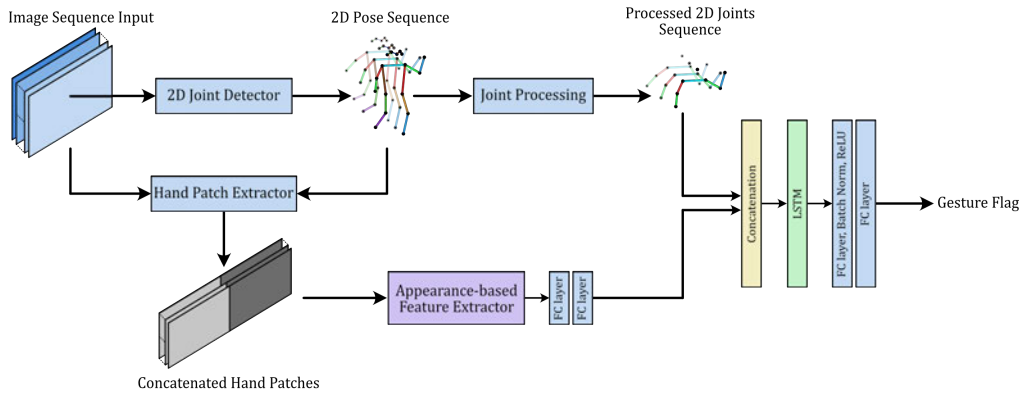


Figure 7. The overall architecture of the proposed method.

output accurate gesture predictions. This is achieved by introducing a novel architecture that augments existing skeleton-based ones with an *appearance-based feature extractor*, which analyzes only specific regions of the input video frames. This additional module is specifically designed to provide rich information to the overall DNN, obtained by the visual cues, while adding minimum computational overhead. It relies on a neural implementation of the Frame Moments Descriptor (FMod) and its local variant LMod [99, 100, 101], which can effectively encode local visual information by efficiently capturing the most informative input appearance statistics. Thus, the proposed appearance-based feature extractor is a plug-and-play module *without any learnable parameters* that can be used to enhance the performance of existing lightweight skeleton-based methods.

Experimental evaluation on two gesture recognition datasets shows that the proposed method increases the gesture recognition accuracy compared to the baseline. Moreover, integrating the proposed FMod-powered appearance-based feature extractor within the overall gesture recognition architecture is more effective than using a learnable CNN feature extractor in its place.

3.8.2. Methodology

3.8.2.1. Gesture recognition using multi-modal features Let $\mathcal{S} = \{\mathbf{S}_1, \dots, \mathbf{S}_L\} \in \mathbb{R}^{L \times W \times H \times 3}$ be an input RGB image sequence, where L is the length of the sequence and W, H are the image width and height respectively, depicting a person that performs a series of gestures. The ultimate goal of the proposed method presented in Figure 7 is to effectively analyze this sequence to extract rich features which allow the accurate recognition of all the performed gestures.

In this direction, two types of features are extracted. First, a CNN is utilized to extract the person's 2D skeleton from each RGB image in the input sequence, calculating a 2D skeleton sequence $\mathcal{P} = \{\mathbf{P}_1, \dots, \mathbf{P}_L\} \in \mathbb{R}^{L \times N \times 2}$ that comprises the corresponding 2D pixel coordinates of a pre-defined set of N body joints (e.g., shoulders, wrists, knees, etc.). However, since gestures may be performed using only specific body parts (e.g., arms or hands), the extracted 2D skeleton sequence is further processed to obtain $\tilde{\mathcal{P}} = \{\tilde{\mathbf{P}}_1, \dots, \tilde{\mathbf{P}}_L\} \in \mathbb{R}^{L \times M \times 2}$, $M < N$, which contains only the 2D pixel coordinates of the relevant M body joints. This ensures attention is focused solely on the body joints of interest, thus reducing the burden on the overall DNN by avoiding the processing of redundant features that may introduce noise (e.g., due to imperfect 2D skeleton predictions). Therefore, if f_{CNN} denotes the 2D skeleton extraction CNN, the calculation of $\tilde{\mathcal{P}}$ proceeds as:

$$\mathbf{P}_i = f_{CNN}(\mathbf{S}_i) \quad (13)$$

and

$$\tilde{\mathbf{P}}_i = \text{joint_proc}(\mathbf{P}_i). \quad (14)$$

Note that since in the scope of this work all gestures are performed by the person's upper body, $\tilde{\mathcal{P}}$ considers only the wrists, elbows, shoulders, and head joints, resulting in $M=7$ body joints.

Besides providing important spatiotemporal information, the first type of features $\tilde{\mathcal{P}}$ also serve an additional purpose. That is, they are utilized to define the regions-of-interest (ROIs) on the input RGB images. Therefore, after $\tilde{\mathbf{P}}_i$ is obtained, the 2D pixel coordinates of the body joints extracted from each input image are utilized to define a rectangular ROI of fixed size on the corresponding image that encloses all of them. In the context of this work, the "head" body joint is excluded from this process, essentially resulting in a hand patch extraction process.

The extracted hand patch sequence \mathcal{H} is subsequently fed to the appearance-based feature extractor to calculate the second type of features $\tilde{\mathcal{H}}$ that encode rich visual information. To this end, the proposed appearance-based feature extractor utilizes at its core a neural implementation [101] of the local variant [100] of FMoD [99], which is denoted as f_{FMoD} . FMoD operates by iteratively dissecting an input frame into segments and calculating statistical properties for each compartment.

With both types of features ($\tilde{\mathcal{P}}$ and $\tilde{\mathcal{H}}$) available, the proposed method proceeds to the final gesture recognition step. To this end, the two types of features are fused before given to the final gesture recognition DNN, denoted as f_G . Finally, the fused features \mathcal{F} are given as input to the gesture recognition DNN f_G , which consists of a simple LSTM and two fully-connected layers, in order to predict the final gesture label \hat{g} :

$$\hat{g} = f_G(\text{fusion}(\tilde{\mathcal{P}}, \tilde{\mathcal{H}})). \quad (15)$$

3.8.3. Experimental Results

The proposed method is evaluated using two gesture recognition datasets. The first dataset is the AUTH-GESTURE dataset [1], which consists of 4930 videos (80/20 split for training and testing) of six gestures (Cross arms, Extend one arm to the side, Palms together, Raise one arm upwards, Thumps up, V shape). The second dataset is the UAV-Gesture dataset [2], which is composed of 119 UAV-captured videos, containing 13 gestures performed by 10 subjects in total.

Table 12. Comparison on both evaluation protocols (P-I and P-II) using the AUTH-GESTURE dataset [1].

Model	Accuracy	
	P-I	P-II
<i>DDNet</i> [102]	74.18 %	52.41 %
<i>RGR</i> [103]	75.83%	62.51%
<i>MMGR</i> _{$\tilde{\mathcal{P}}$}	77.08%	64.94%
<i>MMGR</i> _{CNN}	77.17%	67.06%
<i>MMGR</i>	77.74%	70.37%

The proposed method has been evaluated following two protocols. The first one (P-I) assumes that the entire input RGB image sequence \mathcal{S} depicts a single and complete gesture, while the second protocol (P-II) assumes that at least the last 80% of the RGB images in the sequence correspond to the gesture of interest, simulating a more realistic scenario.

The comparison results for the AUTH-GESTURE dataset are presented in Table 12. It can be seen that the proposed *MMGR* method outperforms all competing methods in both evaluation protocols. Importantly, the proposed method increases the gesture recognition accuracy by at least 7% when compared to



Table 13. Comparison on both evaluation protocols (P-I and P-II) using the UAV-GESTURE dataset [2].

Model	Accuracy	
	P-I	P-II
<i>DDNet</i> [102]	91.51 %	69.03 %
<i>RGR</i> [103]	92.62 %	66.35%
<i>MMGR</i> _{\bar{p}}	92.25%	62.94%
<i>MMGR</i> _{CNN}	90.77%	69.87%
<i>MMGR</i>	93.73%	74.06%

RGR and *DDNet* in the more realistic evaluation protocol (P-II), thus offering a more reliable solution for real-world applications. Moreover, when evaluated on the UAV-GESTURE dataset, it again demonstrates increased gesture recognition performance in both evaluation protocols, as it can be seen in Table 13.

Overall, the comparison results presented in Tables 12 and 13 indicate that the multimodal features extracted by the proposed method allow a very simple gesture recognizer to produce accurate predictions. Finally, since *MMGR* outperforms both *MMGR*_{CNN} and *MMGR* _{\bar{p}} variants, it is shown that the proposed appearance-based feature extractor is able to encode rich visual information that is necessary for accurate gesture recognition.

3.8.4. Relevance to AI4Media use cases and media industry applications

This method matches perfectly with UC3 (AI in Vision - High quality Video Production and Content Automation) since it yielded a fast and automatic video analysis, specifically for lightweight gesture recognition. By the utilization of two similar, alas distinct modalities (human skeletons and RGB images) fast and automatic high-accuracy gesture recognition can be achieved.

3.8.5. Relevant Publications

- A. Christidis, C. Papaioannidis, I. Mademlis, and I. Pitas, "Lightweight Human Gesture Recognition Using Multimodal Features", 2024 Signal Processing for Consumer Behavior Analysis Workshop (EUSIPCO 2024), Zenodo record: <https://zenodo.org/records/13384444>

3.9. Proof of Quality Inference (PoQI): An AI Consensus Protocol for Decentralized DNN Inference Frameworks

Contributing partners: AUTH

3.9.1. Introduction

Decentralized or distributed DNN training and inference are emerging trends in the global media world, where many media asset management systems are interconnected. However, in such environments, individual DNN nodes can be attacked can compromised. In the realm of machine learning systems, achieving consensus among networked DNN nodes is a fundamental yet challenging task. This work presents Proof of Quality Inference (PoQI), a novel consensus protocol designed to integrate deep learning inference under the basic format of the Practical Byzantine Fault Tolerant (P-BFT) algorithm. PoQI is applied to Deep Neural Networks (DNNs) to infer the quality and authenticity of produced estimations by evaluating the trustworthiness of the DNN node's decisions. In this manner, PoQI enables DNN inference nodes to reach





a consensus on a common DNN inference history in a fully decentralized fashion, rather than relying on a centralized inference decision-making process. Through P-BFT adoption, our method ensures byzantine fault tolerance, permitting DNN nodes to reach an agreement on inference validity swiftly and efficiently.

Many distributed or decentralized deep neural network (DNN) methodologies operate under the assumption of reliable communication links among participating nodes, facilitated either through interconnected networks or by presuming the unwavering honesty of nodes irrespective of circumstances. In numerous cases, conventional aggregation techniques such as averaging [104] or simple majority voting [105] are employed to integrate individual DNN node outputs into a cohesive system-wide outcome. However, employing conventional aggregation methods like simple majority voting in decentralized systems with unreliable nodes presents significant challenges [106]. The presumption of consistent node honesty may be unfounded, opening avenues for malicious actors to infiltrate and manipulate transmitted data, compromising the system's integrity. Consequently, the resulting aggregation may be distorted by these faulty nodes, leading to inaccuracies. Moreover, since simple majority voting lacks built-in fault tolerance properties suitable for such environments [107], the aggregated results may not faithfully represent the network's true consensus. Ultimately, relying on majority voting in decentralized systems with unreliable nodes undermines system robustness and jeopardizes decision-making integrity.

Motivated by this, in this work, we introduce a novel consensus protocol called Proof of Quality Inference (PoQI) to investigate the fusion of the P-BFT algorithm within the concept of decentralized DNN inference tasks. By requiring DNN nodes to reach a consensus solely on the final layer probability distributions, we significantly reduce the amount of information that needs to be exchanged between them. Consequently, DNN nodes operating in our system can achieve consensus and locally maintain a universally accepted DNN inference history without relying on any centralized DNN aggregation, thus making the system resilient to Byzantine failures. Our experiments on classification tasks demonstrate that PoQI is capable of achieving accurate results comparable to traditional centralized aggregation schemes.

3.9.2. Methodology

3.9.2.1. PoQI: Proof of Quality Inference Let $\mathcal{G} = \{\mathcal{A}, \mathcal{E}\}$ be a graph consisting of N collaborating AI agents described in a set $\mathcal{A} = \{\alpha_1, \alpha_2, \dots, \alpha_N\}$, that are employed to perform a DNN inference task, e.g., data classification of the form $\hat{\mathbf{y}} = \mathbf{f}(\mathbf{x}; \boldsymbol{\theta})$ where \mathbf{x} is a data sample and $\boldsymbol{\theta}$ are the DNN parameters. \mathcal{E} is as a set of fixed communication links allowing them to communicate with each other. It is assumed that all nodes have obtained access to the same test sample \mathbf{x} , while their goal is to produce a single prediction \hat{y} out of $\hat{y}_{ij} \forall i \in [0, N]$ and $j \in \mathcal{C}$, where \mathcal{C} is a set of valid classes for the specific data classification task of the form $\mathcal{C} = \{c_1, \dots, c_c\}$. Furthermore, it is assumed that each DNN node produces a softmax classification inference result so that $\hat{\mathbf{y}}_i = \mathbf{f}_i(\mathbf{x}_i; \boldsymbol{\theta}_i)$ and $\sum_{i=1}^C \hat{\mathbf{y}}_i = 1$. For the nodes to coordinate on a single inference prediction, we propose a novel consensus protocol to be formed as a single inference rule and provide coordination in the individualized nodes decisions under a fully decentralized structure, thus eliminating the need of any kind of centralized coordination. The *Proof of Quality Inference (PoQI)* protocol can be thought of as a new hybrid consensus mechanism, where the core process is achieved by adapting the traditional BFT SMR approach, where $N \leq 2f + 1$ DNN nodes are needed to tolerate f faulty nodes who may fail during execution or who may behave maliciously by transmitting tampered data to the neighboring nodes. The normally operating DNN nodes are considered to be the *honest* ones. The PoQI protocol is designed to tackle Byzantine failures within the context of decentralized inference. In our systematic design, a classical Byzantine failure scenario is assumed, where malicious DNN nodes may attempt to disrupt the entire DNN Inference process by influencing honest DNN nodes with their incorrect inference predictions. Specifically, each DNN node processes the same data sample \mathbf{x} , and must collaborate with the other $N - 1$ DNN nodes to achieve a consensus regarding both the sample label and the sequential order of DNN classifications.

The system operates within synchronous assumptions, ensuring that the delivery schedule of DNN predictions through messages, is reliably maintained. Efforts are made to minimize communication delays





within this framework. Each DNN node is responsible for broadcasting predictions, to all neighboring DNN node, thereby ensuring that every DNN node receives inference predictions in the same order. Consequently, each DNN node maintains a comprehensive record of its own DNN inference prediction history. Thus, PoQI is tasked with ensuring the following properties:

- **Validity:** If an individual honest DNN node i broadcasts a prediction \hat{y}_i , then every honest DNN node eventually receives \hat{y}_i .
- **Agreement:** If an individual honest DNN node i decides an inference \hat{y}_i , then every other honest DNN node must also produce the same inference decision \hat{y}_i .
- **Integrity:** A prediction \hat{y} for sample \mathbf{x} appears at most once in the delivery sequence of any honest DNN node.
- **Total Order:** The ordered sequence of predictions \hat{y}_i and \hat{y}_{i+1} for samples $\mathbf{x}_i, \mathbf{x}_{i+1}$ must be the same for all honest DNN nodes.

As previously stated, PoQI operates as a state machine replication protocol, consisting of three primary sub-operations: *view change*, *normal operation*, and *conflict decision agreement*. The view change operation orchestrates the primary election process, where a primary DNN node initiates the consensus process, by disseminating its DNN inference prediction regarding the given input sample \mathbf{x} to all other DNN nodes. During normal operation, the core execution of the PoQI protocol takes place, wherein the proposed decision of the primary undergoes evaluation for universal acceptance or rejection. If universally accepted, the primary DNN node is responsible for conveying the network's final decision. Conversely, if the proposed primary decision faces universal rejection, a view-change process ensues. Additionally, in instances where the view change process fails to elect a universally accepted primary node, a conflict decision agreement mechanism is activated. This mechanism is employed to address scenarios arising from inherent characteristics of the DNN models themselves rather than from malicious behavior. In such conflict decision scenarios, the assurance of total ordering during the specific decision period is not guaranteed.

DNN nodes operate through a sequence of actions known as *views* $v \in \mathcal{V}$ where $\mathcal{V} = \{v_1, \dots, v_k\}$. Given that, PoQI protocol operates in consensus rounds, each defined as one execution of the normal consensus process, regardless if it is successful or not. Views describe the consensus rounds that are required, in order for the DNN network to reach a consensus about the label of a given sample \mathbf{x} . It is defined as an index of the form $v \in \mathcal{V}$, containing a sequence of testing pairs whose DNN inference predictions have been scheduled in the time interval t . At each view, one DNN node is operating as *primary* node while the rest $N-1$ nodes are operating as *validators*. In the remainder of this work and for simplicity reasons, each view refers to a single inference prediction of the form (\mathbf{x}, y) . Our goal is that every honest DNN node in N maintains an identical DNN inference history set defined as $\hat{\mathcal{Y}} = \{\hat{y}_{ij}, \forall i \in \mathcal{V} \text{ and } j \in \mathcal{C}\}$.

3.9.3. Experimental Results

In our experimental design, we assume a decentralized network comprising several DNN nodes that communicate with each other. Each DNN node contains a pre-trained Convolutional Neural Network (CNN) model, tailored to its unique task or domain. In the experiments, we aim to explore the collective intelligence and collaborative potential of the PoQI consensus protocol. We compare the results with conventional centralized DNN aggregation methods like majority voting and weighted averaging. Lastly, to assess the BFT property of our protocol, we conduct experiments to determine both the maximum number of misbehaving nodes our system can effectively handle and the behavior of majority voting and weighted average in such settings.

Table 14 presents results from benchmark datasets, aiming to approximate outcomes achieved by centralized methods, under the assumption that all DNN nodes act honestly. In our second set of experiments in Table 15, we introduce a subset of faulty DNN nodes attempting to disrupt the consensus process by transmitting randomized DNN inference results. In the case of Cifar-10 with 0 faulty agents we can observe that majority voting is stable, producing the same results on every DNN node, thus the system





Table 14. Accuracy (%) comparison between the PoQI Consensus Protocol and conventional centralized aggregation methods across different datasets, assuming all nodes act honestly, highlighting results obtained from one node.

Model	Dataset		
	F-MNIST	Cifar-10	SVHN
ResNet 20	90.63	92.18	90.90
ResNet 32	90.99	92.65	91.38
VGG 11	87.85	91.53	88.28
VGG 16	90.35	93.68	93.18
MobileNet v2	91.02	92.57	90.83
ShuffleNet v2	-	89.96	89.69
RepVGG	-	94.51	93.56
Weighted Average	92.51	95.12	94.09
Majority Voting	92.01	95.05	93.75
PoQI	92.33	95.27	94.12

as a whole is in agreement and works as it is supposed to work. However, in the rest of the experiments, we can observe that even with a 1 faulty DNN node that is arbitrarily sending randomized DNN inference results, the weighted average is completely failing while the majority voting is not stable anymore. This means that there is at least one data sample, on which DNN nodes are no longer in agreement and they produce randomized results. This instability on majority voting proves that is not fault-tolerant at all, and even with one faulty agent, it can not be used anymore to establish a commonly accepted agreement on the system. On the other hand, our protocol is able to coordinate the decision-making process of the DNN nodes, even in the presence of faulty nodes, since the decision for each sample is performed by the primary DNN node and every honest DNN node must and will comply with his decision. We assume that the faulty nodes are acting completely arbitrarily so their results are not reported.

3.9.4. Relevance to AI4Media use cases and media industry applications

This work contributes to UC7 "AI for Content Organization and Content Moderation" and UC1 "AI against Disinformation" by proposing a decentralized inference strategy that can be integrated with advanced deep learning techniques for content analysis. Drawing inspiration from societal practices, a decentralized decision-making approach is suggested where individual neural agents are enabled to make autonomous decisions, sharing and aggregating information with other agents within a network. By incorporating advanced AI capabilities and this decentralized inference strategy, media companies can manage visual content efficiently and cost-effectively, ensuring its relevance and safety while preventing media data and decision process tampering. In essence, this strategy empowers individual AI agents and additionally enhances security, and promotes collaboration among multiple neural agents.

3.9.5. Relevant Publications

- D. Papaioannou, V. Mygdalis, and I. Pitas, "Proof of Quality Inference (PoQI): An AI Consensus Protocol for Decentralized DNN Inference Frameworks", 2024 IEEE International Workshop on Distributed Intelligent Systems (DistInSys 2024)
Zenodo record: <https://zenodo.org/records/13384377>





Table 15. Per DNN node accuracy (%) comparison between the PoQI Consensus Protocol and conventional aggregation methods, assuming a subset of faulty nodes acting arbitrarily

Dataset	Faulty Nodes	Method	Accuracy (%)						
			N1	N2	N3	N4	N5	N6	N7
Cifar-10	0	Weighted Average	95.12	95.12	95.12	95.12	95.12	95.12	95.12
		Majority Voting	95.05	95.05	95.05	95.05	95.05	95.05	95.05
		PoQI	95.27	95.27	95.27	95.27	95.27	95.27	95.27
Cifar-10	1	Weighted Average	16.40	15.87	15.92	16.24	15.35	16.11	-
		Majority Voting	94.63	94.86	94.76	95.02	94.72	94.56	-
		PoQI	94.99	94.99	94.99	94.99	94.99	94.99	-
SVHN	1	Weighted Average	15.27	15.41	15.37	15.33	-	15.13	15.52
		Majority Voting	93.21	93.36	93.17	93.12	-	93.04	93.77
		PoQI	93.42	93.42	93.42	93.42	-	93.42	93.42
SVHN	3	Weighted Average	-	11.14	11.40	-	11.16	11.36	-
		Majority Voting	-	92.56	93.12	-	92.94	91.82	-
		PoQI	-	93.18	93.18	-	93.18	93.18	-

3.10. Human face labelling

Contributing partner: RAI

3.10.1. Introduction

Content labeling is the process of annotating or tagging raw data to add context and meaning. In this context, face labelling includes tasks such as distinguishing between different people in images and video streams, or estimating other features of which gender [108], here intended as apparent biological sex - i.e., the biological sex that would be mostly associated to a person by a set of observers - is a very important one. In fact, the guarantee of gender equality in the media is one of the main pillars of public service media. The analysis and reporting of how representatives of both sexes participate in radio and television programmes is becoming increasingly important. For this purpose, national and international Government policies have been put in place [109, 110]. The choice of the most appropriate method is crucial, as it has a direct impact on the overall performance and efficacy. To this end, deep neural network models have been shown to be highly effective [111, 112, 113, 114, 115, 116, 117, 118].

3.10.2. Methodology

We built a recurrent multi-layer perceptron (RMLP) architecture, using the face embeddings extracted by the RAI Face Management Framework (FMF) as input to the network [119]. The output is a label indicating the predicted gender class.

Table 16 illustrates the architecture of the network, which has a total of 877,602 trainable parameters. The backbone of the architecture is a fully connected layer (FC), followed by a rectifier layer (ReLU), and coupled with a cascade combination of batch normalisation (BN), dropout (DR), FC and ReLU layers. In [120], the authors demonstrated that using normalisation followed by drop-down helps to improve the training efficiency of a neural network. Then, there are seven blocks, which again consist of a sequence of BN → DR → FC → ReLU layers, repeated twice and once in even blocks and odd blocks,





Table 16. Neural network architecture for the face gender estimation task.

Block	Description	#Params
Input	FMF embedding \rightarrow Normalize	-
Backbone	FC \rightarrow ReLU \rightarrow BN \rightarrow DR \rightarrow FC \rightarrow ReLU	526,336
Neck (1)	BN \rightarrow DR \rightarrow FC \rightarrow ReLU	132,352
Neck (2)	BN \rightarrow DR \rightarrow FC \rightarrow ReLU (x2)	132,608
Neck (3)	BN \rightarrow DR \rightarrow FC \rightarrow ReLU	33,408
Neck (4)	BN \rightarrow DR \rightarrow FC \rightarrow ReLU (x2)	33,536
Neck (5)	BN \rightarrow DR \rightarrow FC \rightarrow ReLU	8,512
Neck (6)	BN \rightarrow DR \rightarrow FC \rightarrow ReLU (x2)	8,576
Neck (7)	BN \rightarrow DR \rightarrow FC \rightarrow ReLU	2,208
Head	FC \rightarrow SM	66
Total Parameters:		877,602

respectively. The architecture ends with an FC layer coupled with softmax activation function (SM). Our implementation differs from [115] in the input normalisation and the different hyperparameters, specifically batch size, learning rate, weight decay and learning rate scheduling.

Figure 8 shows the correlation between the output of each of the RMLP layers and the gender classes, mapped to a 2D feature space using the t-SNE algorithm for dimension reduction [121]. Each point on the graphs represents a face detected within some keyframes taken from RAI’s television channels and manually labelled as woman (blue markers) or man (orange markers). The top left graph plots the input FMF normalised embeddings. Neighbouring points represent with good approximation faces belonging to the same individual. The other graphs show (top to bottom, left to right) the output of the starting layers (i.e., network backbone, block 0, actual size of the embeddings 512), and the outputs of the middle layers (i.e., network neck, blocks 1 to 7, actual size of the embeddings 256, 256, 128, 128, 64, 64, 32, respectively). It is interesting to note that the network does a satisfactory job of distinguishing between the two classes from the intermediate blocks.

3.10.3. Experimental results

In accordance with [115], we adopted a 2-step approach to build the neural network model. First, we trained the model using the IMDB-WIKI dataset [122], one of the largest datasets of face images from IMDB and Wikipedia with age and gender labels. Then, we fine tuned the model using the Adience dataset [123], and a 5-fold strategy for refinement and validation. Both datasets were preprocessed to filter out uninformative samples, i.e., images containing zero or more than one face, and images whose ground truth was absent. Differently from [115], we did not make any assumption on the face quality, and retained all the images found by the FMF system. We believe this better reflects the application context in which we operate, which is characterised by great variability of the processed content, such as image quality (for example, low resolution, sampling artefacts and noise in older archive material) and size (e.g., long shots, very long shots).

In order to strengthen the experimentation in a real life application scenario, we collected more than 7,000 frames captured from RAI’s programmes of various television genres, including in-depth journalism, talk shows and entertainment. Only frames depicting one face were considered and manually annotated as either woman (2,530 in total) or man (4,815 in total). Table 17 shows the results in terms of weighted precision, recall and F-score of the RMLP architecture compared to three state of the art libraries for gender estimation. We calculated the weighted form in order to take into account the imbalance between the two classes in the test set. This can result in an F-score that is not between precision and recall.



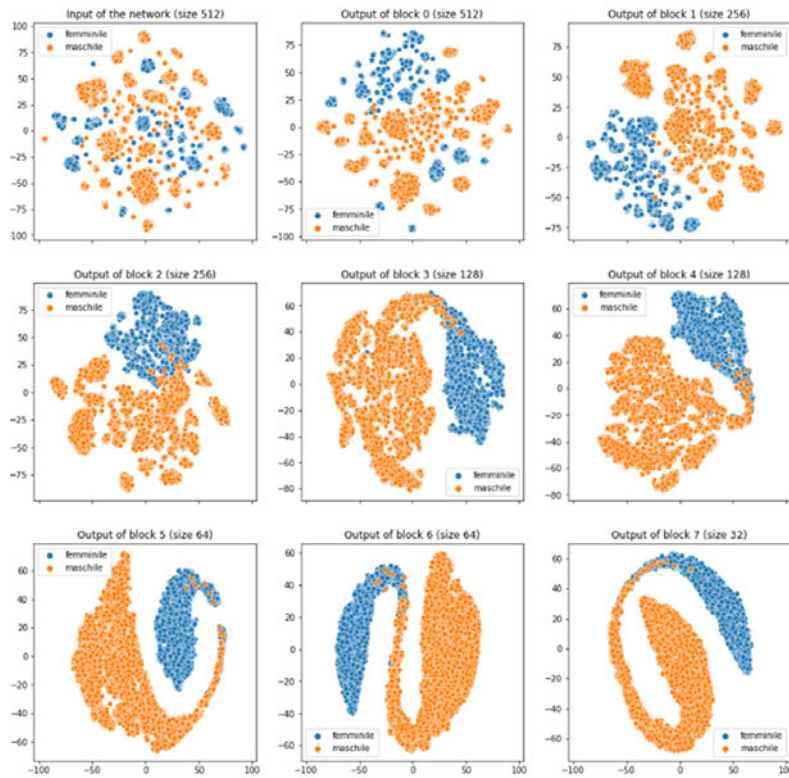


Figure 8. Correlation between the output of the RMLP layers and the face gender categories.

Table 17. Comparison of different gender estimation tools applied to television streams.

Library	Precision	Recall	F-score
InsightFace [124]	0.962	0.962	0.962
FaceLib [125]	0.924	0.924	0.923
DeepFace [126]	0.942	0.939	0.938
RMLP	0.979	0.979	0.979

The RMLP architecture goes beyond the state of the art, demonstrating its ability to be one of the key elements for building advanced analytical tools and data insights.

3.10.4. Relevance to AI4Media use cases and media industry applications

This method is applicable to UC3 (AI in Vision - High quality Video Production and Content Automation) since it yielded a fast and automatic way for video annotation. Recognising the growing importance of information on the participation of women and men in television programming, it is clear that media workflows could greatly benefit from the developed tool.

3.10.5. Relevant Publications

- M. Montagnuolo, F. Negro, A. Messina, A. Bruccoleri and R. Iacoviello, "Who's in My Archive? An End-to-End Framework for Automatic Annotation of TV Personalities", 2023 IEEE International



3.11. People@Places and ToDY: Two Datasets for Scene Classification in Media Production and Archiving

Contributing partner: JR

3.11.1. Introduction

While research has moved from image classification to object detection, segmentation and other more advanced topics, performing classifications of images or entire shots of videos is still a practically relevant task in describing visual content in order to make it findable. This task occurs when describing newly arriving content for production purposes (e.g., news) or annotating large amounts of otherwise sparsely documented content in media archives. Locations are among the three most frequently used search facets in video archive search [127]. For many purposes in visual content creation, place categories (i.e., street, shopping mall) are needed rather than named locations. Automatically labeling images or video shots with such location categories is a typical classification problem, and the Places365 dataset [128] is a very well known resource for this task. However, in a practical setting, there are other key properties of the scene, that are relevant to judge whether a shot is usable or not.

First, it is important to know whether the scene is “empty”, or there are people or vehicles visible. We call this property “bustle”, i.e., whether there are traces of people being active in that scene or not. While it has always been an important query criteria to explicitly look for a quiet or busy view of the scene, the recent COVID-19 pandemic has made that a much requested feature, as depending on the level of restrictions valid at that time, news reports require either empty or populated street scenes.

Second, the shot type (sometimes called shot size) is a key cinematographic property, which determines the importance of a subject, and the context in which a particular shot can be used. The shot type is typically defined by the height ratio of the depicted persons in relation to the view.

Third, for outdoor shots the time of day and the season are important properties. A news editor searching outdoor shots of a building (e.g., house of parliament) wants to find shots that match the season of the story, as well as day or nighttime. For more scenic views, a sunrise or sunset shot is often requested.

Although these are not uncommon properties of content, there are hardly any datasets covering these properties – in particular, datasets with sizes useful for applying deep learning. We propose an automatic workflow to add relevant annotations to these datasets, performing manual annotations where required. Our contribution is now as follows:

- We propose the People@Places dataset, based on Places365, adding bustle (6 classes) and shot type (9 classes) annotations.
- We propose the ToDY (time of day/year) dataset, based on Skyfinder [129], adding time of day (5 classes) and season (4 classes) annotations.
- We provide a baseline for the classification tasks on these datasets, using an efficient state of the art approach.
- We provide the toolchains that were used to create the two datasets, which can be used to replicate this approach for other datasets.

Related work. We review related work on location, shot type and time of day/season classification, and for the detectors used for automatic dataset annotation. To the best of our knowledge, there is no existing work on bustle classification. The closest tasks seem to be people counting or crowd estimation, but those differ as we consider both persons and vehicles, while we are not interested in the exact numbers.

For *location type classification*, many traditional classification architectures, such as the VGG or ResNet families have been applied. Global covariance pooling is proposed in [130] to capture richer



features and improve generalization. One variant of this approach, iterative matrix square root normalized covariance pooling network (iSQRT-COV-Net) used to be the best performing method on Places365, while RS-VGG16 [131] is a recent method proposing a compact model derived from VGG16. In the last few months, vision transformer models such as ViT have taken the lead [132]. A recent extension using large transformer models (86-632M parameters), and self-supervision using masked autoencoders (MAE) is to the best of our knowledge currently the best performing model for classification on Places365.

Like other computer vision tasks, *shot type* (sometimes referred to as *shot size*) *classification* is primarily addressed with deep learning approaches, either approached using CNNs directly for classification [133], using general semantic segmentation [134] or focusing on separating the subject from the background and feeding the regions into a two-stream network [135]. One issue with shot type classification is that the datasets used in many works are not accessible, as they rely on materials from motion picture films that cannot be distributed due to copyright restrictions.

The *classification of time of day* and *season* is a topic that seems to be somewhat neglected. An early work, [136] proposes a system for season classification, but relies on color histograms and the amount of exposed skin of the depicted persons rather than on training samples. The TRECVID semantic indexing task [137] included daytime/nighttime as concepts, and the task was addressed both with traditional machine learning as well as early deep learning methods. However, except for the limitation to only two classes, the resolution and quality of this dataset is quite limited. The Youtube-8M dataset [138] covers some of the relevant classes (sunset, sunrise, night, autumn and winter), while the rest of the times of day and seasons are missing. Some vocabularies from the broadcast domain cover time of day (e.g., EBU LocationTime¹) or season (e.g., TV-Anytime Weather [139]), but no annotated images are provided in this context.

For annotating the dataset for bustle and shot type with vehicles, persons and size of the (partial) persons in the image, we employ object detection, face detection and human pose detection. We employ YoloV4-CSP [140], which combines the CSP-Net proposed in YoloV4 [141] with an efficient model scaling strategy [140], a combination which provides us a highly accurate detector with a low inference time. RetinaFace [142] was chosen as one of the top performing methods on the challenging WIDER Face [143] hard split. For human pose detection, we employ the ROMP algorithm [144]. We chose this method because it is one of the top performing methods on a very realistic (and consequently difficult) dataset named 3D Poses in the Wild [145]. Furthermore, in contrast to other methods (like [146] which performs also quite well on this dataset), it is a computationally efficient single-stage method which does the pose detection for all persons occurring in the image simultaneously.

3.11.2. People@Places: Dataset for bustle and shot type classification

We amend the Places365-Standard dataset (high resolution images) with per image annotations for bustle and shot type. For bustle, we define six classes from entirely unpopulated to populated, resulting from discussions with domain experts from media production and archiving. The classification treats few large persons or vehicles separately, in order to address cases where those are in the focus of the image. Otherwise the classes use a combination of the number and size of objects, expressed by the image area covered together by these objects (see Table 18).

For shot types, there are a number of taxonomies that differ in the level of detail. All of them use the size of the main person depicted in the shot as reference. We use the IPTC NewsCodes scene types², and the lists proposed by Arijon [147], Galvane [148] and Rao et al. [135] as sources, but decided to go for a finer classification (see Table 18). As the annotations of the Places365 test split are not provided (as part of a benchmark), we work with the training and validation splits in this paper, to which we have full access.

The dataset creation process is semi-automatic, where automatic annotation is performed for the entire dataset, and manual verification is performed for the validation split. The process for creating the

¹https://www.ebu.ch/metadata/ontologies/ebucore/ebucore_LocationTimeType.html

²<https://cv.iptc.org/newscodes/scene/>



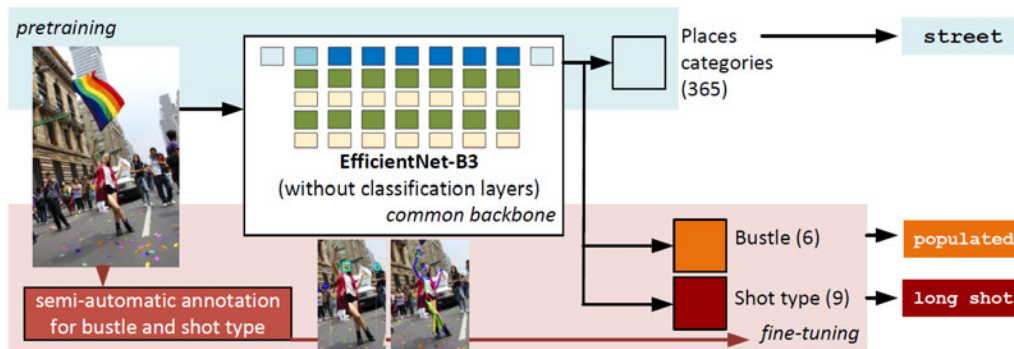


Figure 9. Pretrained on fine-grained places categories, the backbone of the network is used to train classification heads for supercategories, bustle and shot type. Bustle/shot type annotations are created automatically (manually corrected for the validation set).

Class	Definition
Bustle	
unpopulated	no persons or vehicles
few people	< 3 persons, no vehicles, area < 10%
few vehicles	< 3 vehicles, no persons, area < 20%
few large	< 3 people/vehicles, any area
medium	< 11 people/vehicles, area < 30%
populated	more people/vehicles or covering larger area
Shot type	
extreme close-up	detail of face
close-up	head
medium close-up	cut under chest
tight medium shot	cut under waist
medium shot	cut under crotch
medium full shot	cut under knee
full shot	person fully visible
long shot	person 1/3 of frame height
extreme long shot	person <1/3 of frame height

Table 18. Definition of bustle and shot type classes.

annotations is shown in Figure 10. The bustle classes depend on the presence of persons and vehicles, thus object detections for these classes are used. While person detections give a coarse indication about the size of the depicted persons, it is not clear which part of the person is visible. Human pose estimation and face detection are used to complement this information. In detail, the process consists of the following steps.

Object detection. We run YOLOv4 CSP [140] (trained on MS COCO) over all the images, considering all detections with a score ≤ 0.1 as no occurrence. From the remaining detections, those with a score ≥ 0.5 are kept as reliable. Detections between these thresholds are considered uncertain, and the images are excluded. From the detections, persons and vehicles (i.e., the classes bicycle, car, motorcycle, airplane, bus, train, truck, boat) are kept. Based on the criteria defined in Table 18, the bustle annotation is created. In addition, the tallest person is selected and output as annotation.



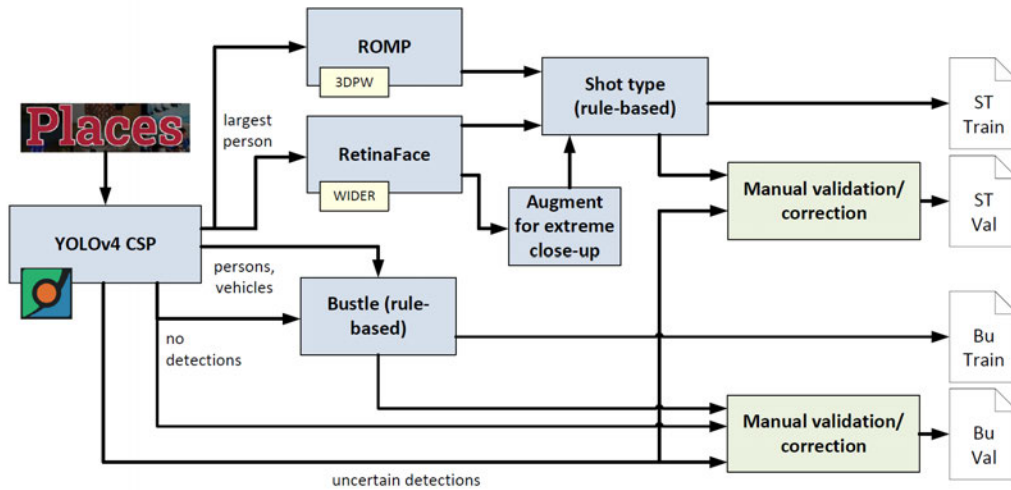


Figure 10. Dataset creation process for bustle and shot type annotations.

Face detection. Face detection is performed using RetinaFace [142], with a model trained on WIDER Face [143], on all images that contain person detections. Multiple faces may overlap the tallest person, and it is not always straight forward to identify the correct one. We keep the face region with the largest size of the intersecting area, weighted by the detection confidence, i.e. $sc_f = (F \cap P)c_f$, where F is the face region, P is the person region and c_f is the confidence reported by the face detector.

Human pose detection. We use the ROMP [144] human pose detector (trained on 3DPW [145]), applied to a cropped out image of the tallest detected person (resp. the visible part of it). We obtain a 2D skeleton (SMPL [149] with 54 points), of which we use 10 points (pelvis, left/right foot, head, left/right hip, thorax, left/right knee, spine).

Person size estimation. In order to filter unreliable detections, we filter pose and face detections for which $\max(w_D, h_D) \geq \tau \min(w_P, h_P)$, where w and h denote width and height, D denotes the pose/face detection bounding box and P denotes the person detection bounding box. τ is set to 0.1 for faces, and 0.6 for poses. If a reliable pose is found, we use it for person size estimation. We use the legs only if they appear to be stretched, i.e. head and at least one foot are on different sides of a horizontal line through the pelvis point, and the hip to feet distance is larger than the thorax to pelvis distance. If the legs are used, we check if feet and hip are on different sides of the knee (at least for one leg), otherwise we ignore the feet. If head to feet is visible, this determines the person size, otherwise we estimate the size of the part of the body not considered reliable to get the overall size measurement. We use ratios of body proportions from [150], a compact visualisation can be found on Wikipedia³. This is also done if only the face detection is usable. If neither pose nor face are available, we use the person detection to determine long and extreme long shots from the person height, if the person bounding box does not extend to the lower image border.

Augmentation for extreme close-up. As we found that extreme close-ups are rare in the dataset, we augment it by sampling cropped images from all close-up shots. If the larger side of a face bounding box is at least s_{\min} pixels, we determine a randomly sized bounding box with $w \in [s_{\min}, 0.75w_D]$ and $h \in [s_{\min}, 0.75h_D]$, with $s_{\min}=175$.

Verification (validation split only). For verification, we import the set of images into the CVAT annotation tool⁴. Each image’s bustle and shot type annotation is initialized from the automatic annotation. A single annotator reviewed and corrected around 1,300 images. The accuracy of the

³<https://en.wikipedia.org/wiki/Drawing>

⁴<https://github.com/openvinotoolkit/cvat>



automatically created annotations against the manually checked ones is provided in Table 21.

Data sampling. From the training set we randomly sample 100K images per class. For validation, we sample 100 images per class from the manually corrected set (images used for the bustle and shot type tasks may partly overlap which is not an issue since they are treated as independent classification problems).

The annotations for bustle and shot type as well as the code of the toolchain used to create it are provided at <https://github.com/wbailer/PeopleAtPlaces>.

3.11.3. ToDY: Dataset for time of day and season

In order to build a dataset, we need a large scale outdoor dataset. We amend the Skyfinder dataset, which is a subset of the Archive of Many Outdoor Scenes (AMOS) dataset [129], consisting of about 1,500 weather webcam images per camera from 53 webcams, each covering one or multiple years. The images come with location (see Figure 11 left for a plot), date and time metadata, image timestamps (in UTC), basic weather conditions and a number of derived attributes. We aim to label each of the images with time of day and season based on the available metadata. The time of day classes and their definitions are listed in Table 19, the season classes are the meteorological seasons [151], i.e., spring, summer, fall and winter.

As the location of the webcams from which the images were collected are known, as well as the dates and times when the images were taken, we can derive the season from the date and the hemisphere, and we can determine the time of the day based on the sun's position. We calculate the sun's elevation over/under the horizon at the location and time of the image, using the PyEphem⁵ library. Note that this calculation will assume a horizon in a flat landscape, not considering any mountains or buildings. We are aware of this limitation, but still assume that the calculated position will be a useful approximation of the real situation.

There are multiple definitions of dusk and dawn, and we use the one for civil dusk/dawn [152], which defines begin of dusk/end of dawn when the sun is 6° below the horizon. While the begin of sunrise/end of sunset is clearly defined with the upper tip of the sun disk being just/still visible, there is not such a clear definition of the end of sunrise/begin of sunset. As the visual effect of sunrise/sunset extends beyond the point where the sun is fully visible, we chose to set this mark at the sun being 3° above the horizon. A visualization of those definitions is shown in Figure 11 (right). In addition, it needs to be considered whether a location is sufficiently far north/south, so that polar night or day occur, and thus no sunset/sunrise happens.

Based on this information, we derive season and time of day images for each image in the dataset. However, we observe three main issues with the data: (i) noisy images, in particular during nighttime, (ii) incomplete images (due to data loss when transmitting the image from the camera) and (iii) inaccurate timestamps. In order to estimate the noise level, we use the mask for the sky region provided for the Skyfinder dataset, as the sky region does hardly contain structures with strong gradients. We split the image into 8×8 patches, and we calculate the standard deviation of all patches containing at least 80% sky, and determine the noise level as the median of the standard deviations in these patches. In order to handle incomplete images, we calculate a RGB histogram of the image, and remove all images where one value covers more than 50% of the pixels of the image.

The time provided in the metadata should match the time stamp of the downloaded image file, when corrected by the UTC offset. However, even with a tolerance of 15 minutes, this does not hold for about 2/3 of the images. This is in particular a problem for classifying twilight, sunset and sunrise, as this inaccuracy may change the correct class. As we cannot tell which of the two times is correct, we decided to manually check the images. We import the set of images into the CVAT annotation tool⁶, and initialize the time of day with the automatically determined value. About 10K images have been manually checked and the annotations have been corrected when necessary.

The toolchain also supports augmentation of the data by cropping versions of the images with a smaller portion of sky region. From the sky annotations of the dataset, a horizon line is determined

⁵<https://rhodesmill.org/pyephem/>

⁶<https://github.com/openvinotoolkit/cvat>



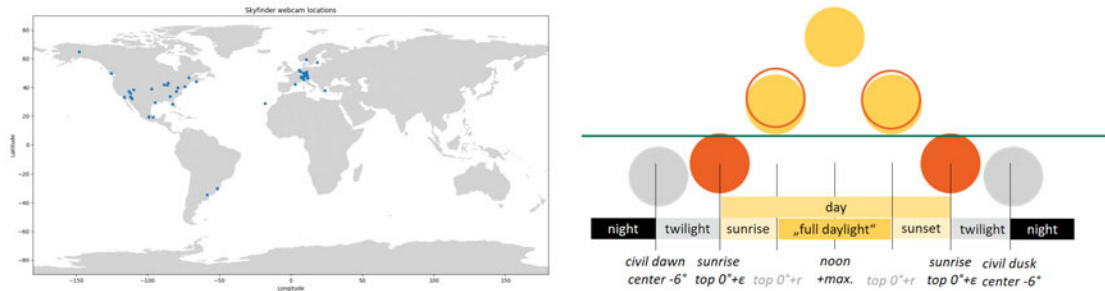


Figure 11. Location of webcams in the Skyfinder dataset (left), visualization of the times of day used (right).

Class	Definition
night	night time
twilight	before sunrise/after sunset, using the definition of civil twilight
sunrise	sun above horizon, until fully above horizon
sunset	sun above horizon, after being fully above horizon
fulldaylight	sun completely above horizon
day	day time, i.e. fulldaylight, sunrise or sunset (not used as a separate class, can be derived from the other classes)

Table 19. Definition of time of day classes.

as the 0.9 quantile of lowest sky pixels in each column. Then images with the same aspect ratio as the original image but different fractions of the height above this horizon line are sampled. As the annotations are global, they are still valid for the modified images.

We split the resulting season and time of day annotations into balanced training and test sets. This results in 2,790 training files and 311 validation files per class for season, and 986 training files and 110 validation files per class for time of day.

The annotations for time of day and season as well as the code of the toolchain used to create it are provided at <https://github.com/wbailer/ToDY>.

3.11.4. Experimental Results

We use EfficientNet-B3 [153] as the baseline model for location type classification and as a common backbone for all tasks (see Figure 9). EfficientNet is a family of DNNs that differ in terms of number of parameters and performance. According to [153], the B3 variant provides a good tradeoff, and variants with better performance will have a significantly higher number of parameters. We train the model using the Pytorch Image Models framework (TIMM) [154], with a learning rate of 0.016 for 75 epochs.

To put the results of the model in relation to the state of the art, we compare the performance of the model on the validation set of the Places365 dataset against MAE [155], iSQRT-COV-Net [130] and RS-VGG16 [131]. However, all these methods have a significantly higher number of parameters as EfficientNet-B3. Still, its performance is slightly better than that of RS-VGG16. The results are summarized in Table 20. Throughout the paper, we use accuracy at rank 1 (acc@1) as the main metric.

The results for bustle and shot type classification are provided in Table 21. We compare the results of the computationally quite demanding annotation toolchain with the classifier trained on the datasets. The models are trained for 25 epochs (50 for shot type) with a learning rate of 0.016. For bustle classification, we observe that the results obtained from the classifier are significantly worse than that obtained with the





Method	no. params	acc@1	acc@5
MAE (ViT-H) [155]	632M	60.3	-
iSQRT-COV-Net [130]	>26M	56.320	86.270
RS-VGG16 [131]	19M	51.680	82.040
EfficientNet-B3	12M	51.874	82.825

Table 20. Comparison on Places365 validation (365 classes).

Method	bustle	bustle0	bustle1	shot type	
	acc@1	acc@1	acc@1	acc@1	acc±1@1
Toolchain	81.020	95.892	95.538	56.726	70.604
E2E	66.337	84.158	81.683	50.715	67.437

Table 21. Performance for bustle and shot type. Toolchain refers to the toolchain in Section 3.11.2, E2E refers to an end-to-end trained classifier.

Pretraining	ToD acc@1	ToD+ acc@1	Season acc@1
none	63.918	20.000	28.310
ImageNet	52.577	66.182	84.225
Places365	54.639	69.818	86.197

Table 22. Top-1 accuracy for time of day and season classification using EfficientNetB3. The pretraining column specifies the base model being used, ToD+ refers to the time of day annotations after manual revision.

detectors in the annotation process. To investigate this further, we introduce two binary variants of the problem: *bustle0* classifies class *unpopulated* against all others, and *bustle1* classifies *{unpopulated, few people, few vehicles}* against all others. It turns out that in these cases the performance of the classifier is closer to that of the detector toolchain. Our interpretation is that the network can well discriminate the presence of people or vehicles, but responds similarly for images with different count or size of objects, which makes it more difficult to discriminate the intermediate classes. This means that if a binary bustle classification is needed, this can be done efficiently with the classifier, while for the multi-class problem, the (computationally more expensive) detector-based approach used for dataset annotation provides better results.

For shot type classification, we observe that the results come closer to that of the annotation toolchain, but still stay below. We observe that many of the wrongly classified shots are those in nearby classes (e.g., medium shot vs. medium full shot). We thus add an evaluation metric for measuring classification into the correct or an adjacent class, which we call *acc±1@1*. We can observe that the performance of the annotation toolchain is in this case significantly higher, and additionally the gap between the performance of the classifier and the toolchain is reduced. For practical cases in editing, shots with similar types (which may be border cases) might already be a useful result.

The results for bustle and shot type classification are provided in Table 22. The models are trained for 450 epochs for season and 1,000 epochs for time of day (stopping early if a performance ceiling is reached) with a learning rate of 0.016. We compare the performance when training EfficientNet-B3 from scratch and from models pretrained on ImageNet and Places365. We provide two results for time of day: ToD refers to the automatically generated annotations, and ToD+ to the annotations after manual corrections. Overall, the performance starting from a pretrained model is better than starting from scratch, and pretraining on Places365 provides slightly better results than pretraining on ImageNet. We assume this is due to the fact that Skyfinder images are more similar to images in many categories in Places365 than to those in ImageNet. The results of 86% accuracy for season and almost 70% accuracy for time of day show that the resulting classifiers are practically usable.





3.11.5. Conclusion

We have proposed two datasets to address relevant classification tasks in visual media production and archiving: one addressed bustle and shot type classification, the other season and time of day classification. We provide toolchains for generating the additional annotations, as well as the datasets, which include manually verified and corrected subsets. The datasets are useful for classifying these properties in images, and the toolchains enable adding these annotations to other similar datasets with limited manual effort. As a baseline, we provide experimental results using EfficientNet-B3 for the four tasks on the two datasets.

3.11.6. Relevance to AI4Media use cases and media industry applications

The two datasets and toolchain are useful for all media-related use cases where it is important to classify the scene (shot type), how populated it is, and to which time of day or season it belongs.

3.11.7. Relevant Publications

- W. Bailer, H. Fassold, "People@Places and ToDY: Two Datasets for Scene Classification in Media Production and Archiving", MultiMedia Modeling: 29th International Conference (MMM 2023), Bergen.
Zenodo record: <https://zenodo.org/records/7318045>

3.11.8. Relevant software/datasets/other outcomes

- The two datasets are available on Zenodo at <https://zenodo.org/records/8398916>.
- The code of the toolchain used to create the datasets are provided at <https://github.com/wbailer/PeopleAtPlaces> and <https://github.com/wbailer/ToDY>.

3.12. Deep Learning to detect objectification in films and visual media

Contributing partner: UCA

3.12.1. Introduction

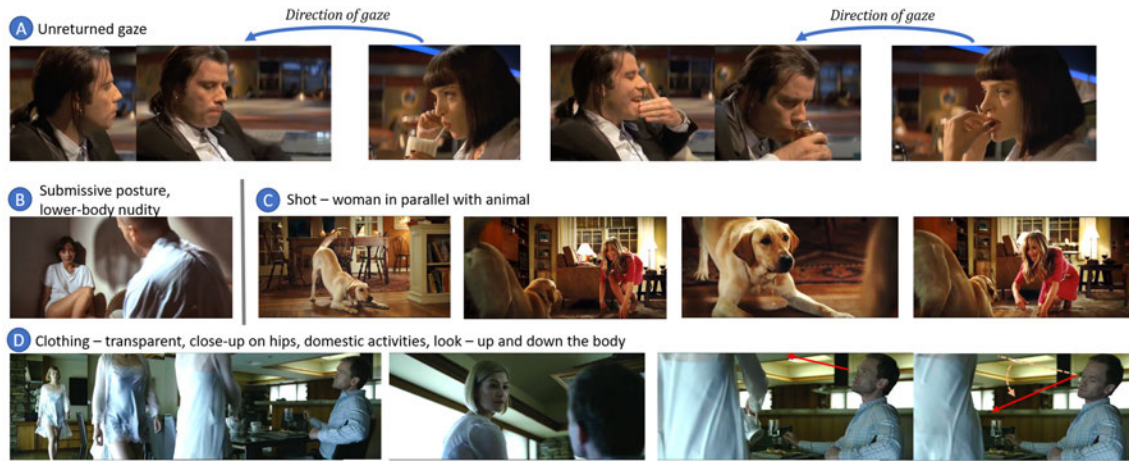
In film gender studies, the concept of “male gaze” refers to the way the characters are portrayed on-screen as objects of desire rather than subjects. In this article, we introduce a novel video-interpretation task, to detect character objectification in films. The purpose is to reveal and quantify the usage of complex temporal patterns operated in cinema to produce the cognitive perception of objectification. We introduce the ObyGaze12 dataset, made of 1,914 movie clips densely annotated by experts for objectification concepts identified in film studies and psychology. We evaluate recent vision models, show the feasibility of the task and where the challenges remain with concept bottleneck models. Our new dataset and code are made available to the community.

3.12.2. Methodology

In this section, we introduce the process that led to the creation of the ObyGaze12 dataset.

3.12.2.1. Definition of the objectification The first task was to define the objectification construct. To do so, a structured thesaurus based on the literature on psychology and films studies has been created. From this 5 sub-construct of objectification were identified resulting in 11 concepts spanning 3 modalities (vision, text and sound). The thesaurus is described in details in [156]. Figure 12 shows how objectification is manifested in various ways.





full body → approach → close-up on hips

Figure 12. In modern film media, the unequal characterization of gender on screen frequently evokes concepts of objectification, such as (A) unequal gaze (*Pulp Fiction*, 1994), (B) Nudity and submissive postures (*Pulp Fiction*, 1994), (C) animalisation or infantilisation (*Marley and Me*, 2008), and (D) transparent clothing, camera framing, domestic gender roles, and voyeurism (*Gone Girl*, 2014).

3.12.2.2. Annotation The annotation process consists in, at least, 2 experts densely annotating 12 movies: they manually delimit all the segments they find relevant for objectification, and label each with a level of objectification. To allow for fine-grained data and model analysis, they also annotate which objectification concepts are present.

Every selected movie is annotated by two experts for objectification level and concepts over the movie scenes. Specifically, the annotators were asked to repeat a three-step process for every scene they deemed interesting from an objectification perspective: (1) watch the movie entirely and when they identify a scene worth annotating, (2) delimit the clip, and (3) assign an objectification level and annotate the concept(s) involved in the objectification rating. We define four levels of objectification:

- **Easy Negative (EN)**: no objectifying concept is present.
- **Hard Negative (HN)**: one or some concepts are present, are annotated, but are deemed insufficient to produce a perception of objectification.
- **Not Sure (NS)**: objectification is perceived and concepts are annotated but the annotator considers they do not sufficiently explain the perception of objectification.
- **Sure (S)**: objectification is perceived and explained by the annotated concepts from the thesaurus.

3.12.2.3. Analysis of the data Here we comment on some interesting statistics of the resulting annotations and concepts of the 1,914 clips originally delimited in the MovieGraphs dataset.

First, we verify data consistency by computing the inter-annotator agreement (IAA). Given the task of annotating timespans, we choose the γ agreement measure introduced in [157]. It attributes a score between 1 (complete agreement) and $-\infty$. A value of $\gamma \leq 0$ indicates no agreement. Considering all four categories EN, HN, NS, S, we obtain an average $\gamma = 0.42$. Not considering the clips annotated Not Sure (NS), which is the uncertain and “noisy” class in human annotations, the IAA increases to $\gamma = 0.69$. This shows the consistency of the obtained annotations despite the interpretive nature of the task. Second, we analyze the obtained annotations in Fig 13. The Sure category is the least represented with 16%, the Easy Negative being, as expected, the most represented class with 52% of clips. It is interesting to note that every concept is approximately annotated with the same rate throughout the Hard Negative, Not Sure and Sure levels of objectification. Finally, it is very interesting to observe that the average number



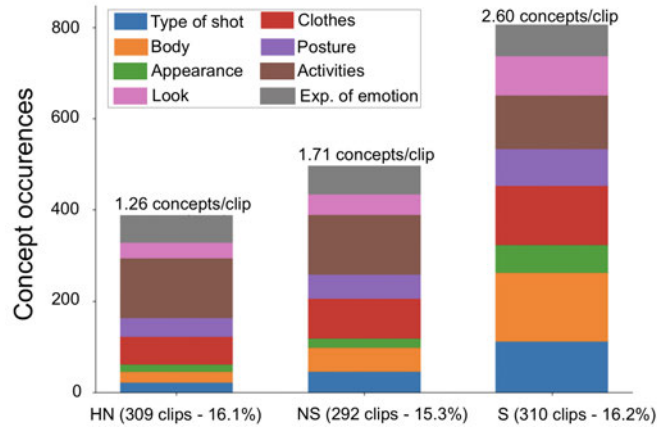


Figure 13. Distribution of visual factors annotated for each level of objectification (HN = Hard negative, NS = Not sure, S = Sure). The percentage of the dataset for each level of objectification as well as the average number of concepts per clip are also shown. (Best viewed in colors)

Table 23. F1-score on the binary task of objectification detection for models trained with easy or with hard negatives and tested on easy or all negative samples, with standard deviations.

Test	EN vs. S		(EN U HN) vs. S	
	EN vs. S	HN vs. S	EN vs. S	HN vs. S
ViViT-B/16	0.53 (0.18)	0.62 (0.13)	0.54 (0.24)	0.73 (0.1)
X-CLIP	0.79 (0.05)	0.71 (0.05)	0.66 (0.05)	0.82 (0.03)
Random	0.54		0.55	
All positive	0.08		0.06	
PCBM-DT	0.68	0.44	0.58	0.38
PCBM-LR	0.64	0.43	0.50	0.37

of concepts annotated per clip increases with the level of objectification: 1.26 concepts on average per Hard Negative clip, 1.71 for Not Sure, up to 2.6 for Sure. It gives an important insight into our video interpretation data: that objectification is a compositional process.

3.12.3. Experimental Results

The experiments have two objectives: to verify that the new classification task is feasible, and to identify the challenges of designing efficient models. To tackle these objectives, we consider pre-trained vision models and specifically address the following research questions:

- **Task accuracy** – What are the baseline performances by pre-trained vision models on the objectification detection task? How does the performance vary with hard negative examples?
- **Concept representation** – Can we implement interpretable models of objectification using concepts? What is the quality of representation of every concept, and what are the objectification concepts poorly captured by current models?

Task accuracy The setting of this experiment is described in detail in [156]. We report the F1-scores in Table 23. We observe that **the inclusion of Hard Negative examples improves the classification results**, showing the importance of a fine-grained annotation for highly-interpretive tasks. The best results based on existing models are of moderate quality, which calls for more investigation into where the difficulties lie.



Concept Accuracy To infer on-screen objectification, it is key for the model to detect the means of its production. The means of producing objectification through the eight concepts can be subtle to detect, making it difficult to provide the final interpretation. To investigate this difficulty, we implement Post-hoc Concept Bottleneck Models (PCBMs) [158], which allow us to approach a classification task with pre-trained models in an interpretable way when concept-annotated data is available. In our case, from the X-CLIP embedding space where our video clips are represented, we identify a Concept Activation Vector (CAV) [159] for every concept. We then project the X-CLIP embedding of every clip onto the subspace defined by the eight CAVs. The representation of the clip that is the output of this bottleneck is a low-dimensional vector with number-of-concepts components. This vector can then be fed to an interpretable classifier for the objectification detection task. A comprehensive description is provided in our first paper [156]. However the conclusion is that the X-CLIP embedding related to concepts *Type of shot*, *Posture*, *Look* and *Appearance* are harder to separate linearly.

3.12.4. Conclusion

This work has an explicit societal motivation in its purpose to tackle, with the help of AI, the analysis of complex temporal patterns operated in cinema that produce the perception of certain characters as objects. This is a challenging but valuable task that aims to uncover and quantify differences in how various identities may be portrayed on screen. A distinctive element of our work is the subjective judgement involved in annotating granular video elements for objectification. Video annotation is tedious, and approaching data annotation for such an interpretive task in a rigorous way is even more so, and difficult to scale. We therefore believe that pursuing high-quality, dense annotations with well-defined concepts goes a long way to tackle this new video interpretation task, which represents a valuable new challenge for the computer vision community.

3.12.5. Relevance to AI4Media use cases and media industry applications

This dataset and preliminary models can be used in use cases where one is concerned with investigating the possible representational biases present in videos included in training data or used for any other objective, such as illustrating a certain story.

3.12.6. Relevant Publications

- J.Tores, L.Sassatelli, H.Wu, C.Bergman, L.Andolfi, V.Ecrement, F.Precioso, T.Devars, M.Guaresi, V.Julliard, S.Lecossais, "Visual Objectification in Films: Towards a New AI Task for Video Interpretation", 2024 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle.

3.12.7. Relevant software/datasets/other outcomes

The dataset and code are made available at <https://github.com/husky-helen/ObyGaze12>.





4. Media content production

4.1. Overview

Task 5.2 (T5.2) “Media content production” of AI4Media investigated multiple aspects of automatic or semi-automatic media content production, focusing on the creation, adaptation, and enhancement of media content. The task examines both the pure synthesis of media content exploiting computational methods such as Deep Generative Models as well as methodologies that help in the acquisition and streaming of such content to end-user devices. Research activities in T5.2 cover a wide range of topics relevant to content production, including cinematography planning, with emphasis on UAV media production, procedural content generation and sound synthesis of musical instruments based on synthetic music sounds.

4.2. Photoconsistent and Trajectory Guided Novel-View Synthesis Tool for UAV Cinematography Based on Autoregressive Transformers

Contributing partner: AUTH

4.2.1. Introduction

Novel view synthesis is the task of generating new images that render an object or scene from a different viewpoint than the one given. It aims to create new views of a specific subject starting from a number of pictures taken from known points of view. The fields of computer science research, vision research, and artificial intelligence are involved in defining suitable approaches to the problem. The novel view synthesis problem can be approached in two different ways: as a problem of image interpolation between two known images or image extrapolation from one image or a subset of images. This work addresses the problem of image extrapolation. The goal is to synthesize a target image with an arbitrary target camera pose from given source images and their camera poses. These synthetic images can be very useful when they come from a UAV, taking advantage of the fact that it is possible to pre-calculate the trajectories that the camera has to execute, from a series of known UAV cinematography shot-types. Based on that and on Autoregressive Transformers, an end-to-end tool is presented that achieves novel-view synthesis from previously unvisited points of view for aerial cinematography robots.

The main contributions of this work are as follows:

- Trajectory guided novel-view synthesis from known, but not previously visited, viewpoints addressing the image extrapolation problem.
- An Open-Source end-to-end tool implemented in ROS for image generation from a known trajectory and an initial image.

The whole framework has been implemented end-to-end using the ROS (Robot Operative System) framework, [160], in such a way that it can be executed with real data.

This research work was conducted as part of a Junior Fellows Exchange programme between AUTH and Centro Avanzado de Tecnologías Aeroespaciales CATEC. The exchange began on 18 May 2023 and concluded on 14 July 2023

4.2.2. Methodology

The cinematographic aerial vehicle and its environment were simulated using the following tools; On the one hand, the environment was simulated using the 3D computer graphics game engine Unreal Engine 4.27. This allowed us to reduce the disparity with reality through its exceptional photorealistic capabilities. On the other hand, to simulate the aerial robot and the mounted camera, from which images of the environment were taken, AirSim [161], a cross-platform simulator designed for autonomous systems research, was used.





In order to generate the synthesized data and subsequently evaluate the tool, a trajectory generator has been developed based on the UAV shot types ORBIT, FLYBY, and FLYOVER from [162]. To produce the trajectories, the following simplifying assumptions have been made: the target is located at a known position in world coordinates and remains stationary, meaning that its velocity is zero.

4.2.2.1. GeoGPT Transformer The main module of the developed software is the Autoregressive transformer used to solve the image extrapolation task. For this purpose, an implementation based on GeoGPT [163]. Taking as input the initial camera image of the AirSim UAV and the following N positions and orientations of the trajectory being executed, GeoGPT will generate as output N novel views of the target inside the simulated environment.

In this autoregressive transformer architecture, a modified self-attention block, has been added in which the camera position is taken into account to generate the novel view. The architecture has been trained with default parameters. However, the number of transformer layers has been decreased to 16 to achieve training speed-ups and memory consumption decrease. It has been experimentally verified that this parameter change does not affect the tool performance in this specific task.

4.2.3. Experimental Results

In order to evaluate the performance of the tool, the lost visual consistency while generating novel views had to be measured. Therefore, the spacing between viewpoints was used as the variable to be modified. One image per second was taken at each position; therefore, the gap between positions was directly related to the speed of the aircraft. In other words, the aim was to evaluate at what speed the simulated aerial robot could take (or generate) images without losing visual consistency. In the experiments, the number of novel views to be generated was set to 5.

A total dataset of around 4.000 images with all their associated poses in the World Coordinate System (WCS) has been created at minimum speed, i.e. one image per second. The images have been generated from pre-calculated trajectories around the central target of the environment. The model has been trained on a workstation with 4 RTX1080 12 GB and 128 GB RAM for approximately 12 hours.

Regarding the metrics used to assess the consistency of the novel views, the same metrics will be used as in the original GeoGPT work. Two metrics will be used: Learned Perceptual Image Patch Similarity (LPIPS), [164] and PSNR. LPIPS measures the perceptual similarity in deep feature space, and PSNR measures pixel-wise differences between two images. Therefore, a series of images with different gap sizes between them has been generated to evaluate the tool's performance.

Table 24. Metrics comparison based on viewpoints spacing

	Gap 5	Gap 15	Gap 25	Gap 35
PSNR (\uparrow)	23.05	21.84	18.51	17.32
LPIPS (\downarrow)	1.8726	2.2412	2.7132	2.9870

Fig. 14 shows qualitatively the results obtained with each evaluated configuration. It is apparent that the consistency of the images is good when the gap between camera shots is small, which is analogous to the UAV moving at a very low speed. However, when the gap is around 25 positions between viewpoints, it is observed how, from the second image, there begins to be a notable offset between the predictions and the ground truth. Finally, in the most extreme case, which implies that the aircraft moves at high speed, only the first generated image is consistent, with the rest having a considerable visual difference from the ground truth. A metric-based comparison of the generated novel views compared to the AirSim UAV camera shots in the simulated environment is provided in Table 24



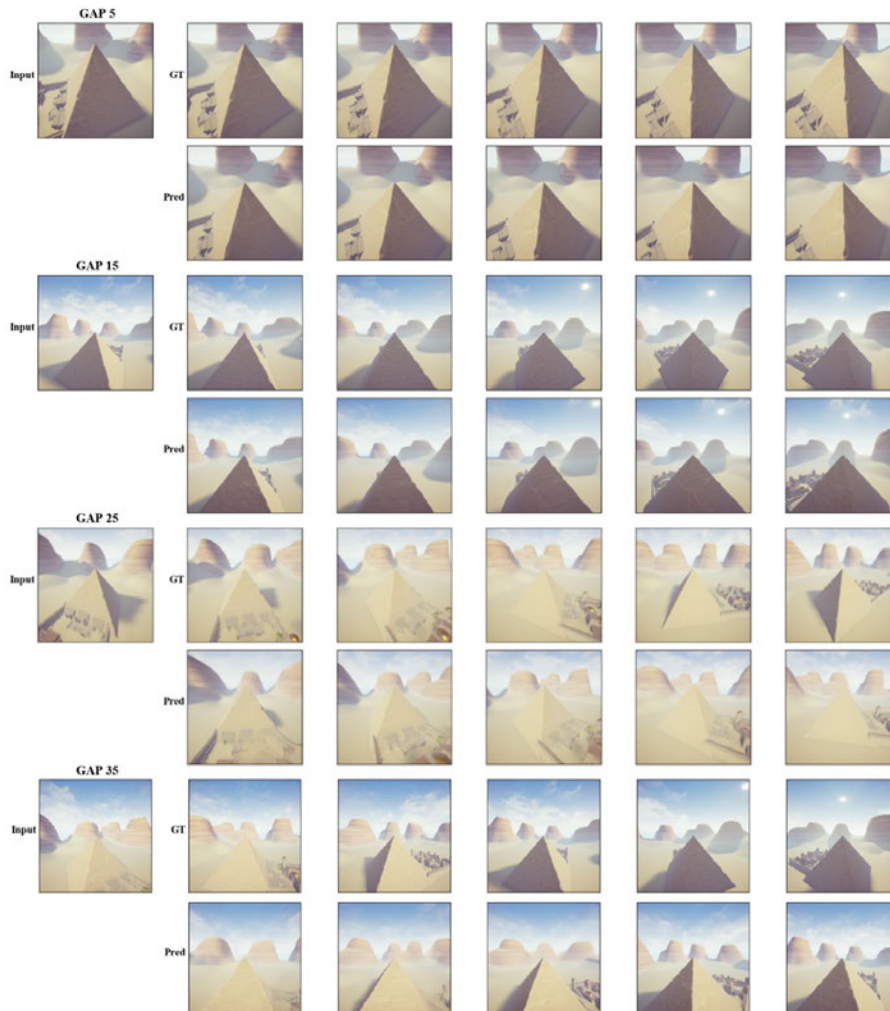


Figure 14. Comparison between prediction and ground truth with different gap length.

4.2.4. Conclusion

In this paper, we have presented an end-to-end tool for generating novel views from unvisited points of view. Future improvements for the tool could include replacing GeoGPT with a diffusion model, which would generate images even more consistently, especially in more photorealistic environments. Additionally, a great improvement of this method could be to modify the utilized DNN architecture training to include camera trajectory estimation. This could lead to an inference stage that would need only an initial image and the desired shot type as input, and a synthetic video shot as its output.

4.2.5. Relevance to AI4Media use cases and media industry applications

This method is useful for UC3 (AI in Vision - High quality Video Production and Content Automation) since it yields a methodology useful for automatic video prediction based on a desired UAV shot-type. This could be immensely helpful for media organizations, since, a UAV pilot could record multiple low FPS videos on the field to keep memory consumption minimal. Then, in the post-processing stage, using





novel-view synthesis methods like the one presented, intermediate, novel frames can be generated to achieve synthetic but natural-looking, higher-FPS, videos.

4.2.6. Relevant Publications

- Marco A. Montes-Grova, Vasileios Mygdalis, Francisco J. Pérez-Grau, Antidio Viguria and Ioannis Pitas, "Photoconsistent and Trajectory Guided Novel-View Synthesis Tool for UAV Cinematography Based on Autoregressive Transformer", 2024 IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2024)
Zenodo record: <https://zenodo.org/records/8276584>

4.2.7. Relevant software/datasets/other outcomes

The open-source implementation of the developed tool, implemented end-to-end using the ROS framework, can be found in the following github repository: https://github.com/catec/nvs_trajectory_guided_ros

4.3. Real-time object geopositioning from monocular target detection/tracking for aerial cinematography

Contributing partner: AUTH

4.3.1. Introduction

Currently, aerial cinematography plays an important role in media production. In recent years, the field of automated aerial cinematography has seen a significant increase in demand for real-time 3D target geopositioning for motion and shot planning. Targets to be filmed can be e.g., cars or persons. To this end, many of the existing cinematography plans require the use of complex sensors that need to be equipped on the subject or rely on external motion systems. This work addresses this problem by combining monocular visual target detection and tracking with a simple ground intersection model.

Inspired by [165], we have developed a complete Robot Operating System (ROS)-based software that consists of 3 modules: a *target detector*, a *tracker*, and an auxiliary *management* module. Operationally, the software works as follows: The management module handles input/output and triggers the target detection and tracking modules. Given that there have been no previous target detections, the management module provides images to the target detection module. This operation is repeated until there are output target detections. If a target has been detected, the management module uses the detected ROI to instantiate the tracking module. Unless a certain quality threshold is not achieved or a specific threshold of time has been exceeded, the outputs of the software module are given by the tracker module. The tracker module always provides two outputs, one bounding box prediction, and one tracking quality score [166]. The acceptable quality threshold and the tracking time windows are the system's hyperparameters, which can be set before a filming mission. The management module takes into account the time and quality variables and decides whether to re-employ the tracking module or ask for new detections from the detection module.

This research work was conducted as part of the Junior Fellows Exchange programme between AUTH and Public University of Navarre (UPNA). The exchange began on 1 November 2022 and concluded on 22 November 2022.

4.3.2. Methodology

4.3.2.1. 2D target detection Object detection is the task of identifying and localizing object instances within an image, formulated as a combined classification and regression problem. Inspired





by the whitened self-attention operation [167], we developed a transformer-based object (target) detector that replaces the attention operation with a linear multiplication, by introducing auxiliary (pre-computed) matrices that perform a transformation that highlights known or computed data properties, modeled in graph structures. An example of a target detection using this software is shown in Figure 15.



Figure 15. Sample output of the proposed 2D detection and tracking software.

4.3.2.2. 2D target tracking In aerial cinematography, a big challenge is that the target also disappears from the field of view quite often. In fact, according to Visual Object Tracking Challenge reports [168], occlusions are the most common causes of tracking failure, hence they should be taken into account.

To address the above-mentioned challenge, we have opted for a two-fold approach. Since we only focus on finding the correct analogy of the bounding box, we have selected the SiamFC siamese tracker [169], which is very fast, and very good at maintaining and finding the correct scale of the bounding box. Furthermore, inspired by [166], we have developed a framework that accounts for target occlusions and estimates the tracking quality for each output bounding box.

4.3.2.3. 3D Object geopositioning Once the boundary box of the target has been identified relative to the center of the camera, the distance is obtained from the height of the drone above ground level (AGL) h , the angle of pitch of the camera θ , and the vertical angle between the rays that project to the camera focal center and to the middle bottom of the boundary box.

4.3.3. Experimental Results

To test the model, a flight experiment was carried out with a heavy-lifting aerial cinematography hexacopter in a safe open-field environment. The key hardware components of the hexacopter system were:

1. *Flight controller*: responsible for maintaining stable flight across the flight plan and sending real-time telemetry data including GPS and attitude via ROS. The flight controller software is based on Ardupilot software.
2. *Positioning camera*: responsible for real-time image capturing and sending them to the onboard computer via ROS.
3. *Onboard computer*: Jetson AGX Xavier, responsible for receiving location, pose, and detection images, synchronizing times, and performing detection and tracking. The detection location is shared through ROS to coordinate aerial shots.





The selected objects of interest are electrical towers for their fixed and equally spaced positions. They also present features to guarantee the repeatability of the experiment. Nevertheless, this architecture is suitable for detecting all categories included in the COCO 2017 [170] pre-trained model without further training. The experiment conditions are listed below:

- Altitude: 30 m, constant Above-Ground-Level (AGL). Flight speed: 6 m/s.
- Parallel side-flight to the towers at a distance of 30 m.
- Number of towers flown for each pass: 4 towers.
- Camera mount angle: 15° of pitch to the bottom.
- Diagonal Field-Of-View (FOV): 92° . Sensor type: 3:2. C_w : 640 px. C_h : 480 px.
- Magnetic declination at location: $+0.76666^\circ$.

The estimated angular error and distance error for each target are summarized in Figures 16 and 17, respectively.

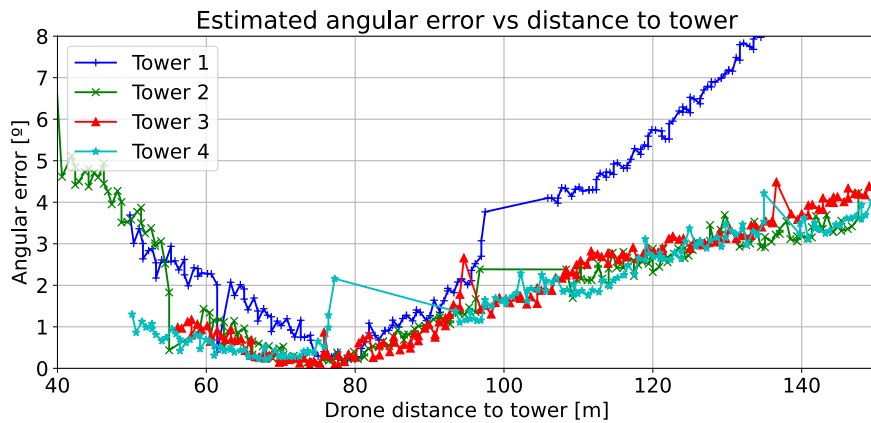


Figure 16. Angular error in estimation over hexacopter-to-target distance.

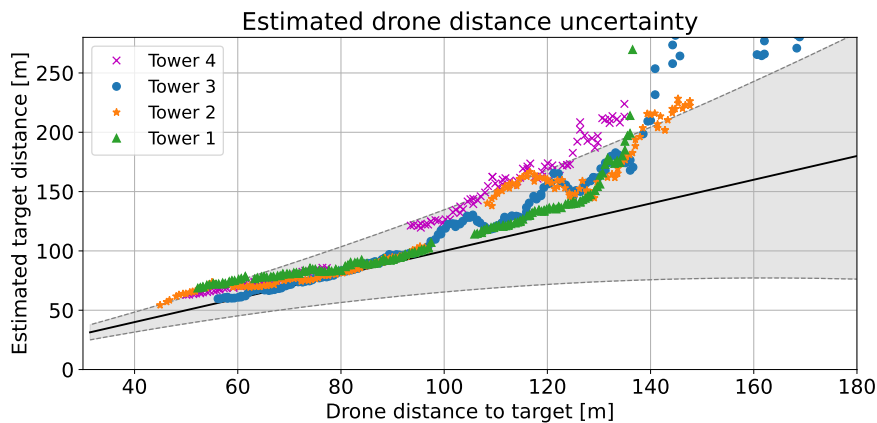


Figure 17. Hexacopter-to-target distance error estimation over actual distance to subjects.

The results demonstrate that distance and angular errors increase with distance to the target. This trend is more pronounced for distances greater than 100 meters. While the estimated distance values are well defined by the uncertainty model, errors are more significant than expected for long distances. This





discrepancy is attributed to the implicit limitations of the flat terrain assumption used in the algorithm, as the experimental location featured slight uphill inclinations causing an overestimation of the distance.

In some cases, it is also observed how the error tends to increase at distances below 60 meters. Since the experiments were carried out parallel to the power line when the drone is about to pass a tower (and thus the distance is minimum), the bottom of the tower falls out of the detection camera's field of view, but the algorithm is still able to detect it and track it. This produces inaccurate geopositioning of the tower. In future works, we will consider identifying whether the detection of the tower may be incomplete to improve this aspect.

Finally, Figure 18 presents the experimental position estimations for each target on the map. The position estimation for targets when they are further than 120 meters from the hexacopter, displays a higher error rate (red and orange dots). On the contrary, when the target is closer than the 120 meter-distance threshold, the approximation is reasonably good (yellow and green dots). This implies that the algorithm is suitable for most types of shots discussed in the bibliography [162].



Figure 18. Object of interest actual coordinates and algorithm-generated coordinates comparison. Pins represent the real-world positions of 4 aerial cinematography targets. Red and orange dots are erroneous positional estimations that happen when the target is >120 meters away from the camera. Yellow and green dots are acceptable or accurate position estimations that are produced with targets 80 to 120 meters away from the camera.

4.3.4. Conclusion

In this study, we have presented a real-time 3D position estimation algorithm for aerial cinematography based on image detection and tracking. The uncertainty of the algorithm was evaluated based on different variables, and the results were validated with experimental flight data. The results demonstrate a reasonable level of accuracy, with the vast majority of measurements below 100 meters of distance featuring an absolute error lower than 5 meters and 3 degrees of yaw. The algorithm's main limitation is using a flat-earth assumption for the ground model, which may be improved using an alternative ground model that does not require iterative solving. Future work includes implementing this algorithm in an automatic planning system for aerial cinematography shots.





4.3.5. Relevance to AI4Media use cases and media industry applications

This method is useful for UC3 (AI in Vision - High quality Video Production and Content Automation) since it yields a fast and automatic way for high-quality and accurate video production. By utilizing UAVs paired with modern versions of well-researched computer vision algorithms (i.e., target detection and tracking), fast and automatic high-accuracy cinematography can be achieved.

4.3.6. Relevant Publications

- D .Aláez, V. Mygdalis, J. Villadangos, and I. Pitas, "Real-Time Object Geopositioning from Monocular Target Detection/Tracking for Aerial Cinematography", 2023 IEEE 25th International Workshop on Multimedia Signal Processing (MMSP 2023)
Zenodo record: <https://zenodo.org/records/8276584>

4.3.7. Relevant software/datasets/other outcomes

Additional video comparison with a YOLOv5 [171] detector and flight experiment videos are provided as supplementary material at <https://youtu.be/Nwak08FnA5s> and <https://youtu.be/CJpHkhE2kSM>. This includes two videos featuring the 3D flight visualization with the detection overlay and a live bounding box comparison of a standard YOLO detector and the DETR detector without tracking.

4.4. Forecasting in Multimedia

Contributing partner: UNIFI

4.4.1. Introduction

In this subsection, we discuss UNIFI's contribution regarding forecasting quantities in media streams. Predicting future events is a fundamental prerequisite to implement automated production pipelines. Most of the work regarded the models for trajectory forecasts [172, 173, 174, 175]. While these models are in general preferred since they are able to reach longer timeframes and higher accuracy, they are also reliant on world reconstructions and accurate detection and tracking. UNIFI also worked in a more challenging setting where only first-person view of scenes is available [176, 177]. Finally, considering the central role of humans in media, UNIFI also studied *progress* in action recognition [178].

4.4.2. Methodology

4.4.2.1. SMEMO SMEMO is a new model with an end-to-end external working memory, which we refer to as Social MEMory MOdule, capable of modeling agent interactions for trajectory prediction. In SMEMO, the motion of each agent is processed into two streams, which we refer to as Egocentric and Social. The former is dedicated to modeling relative displacements of an agent from one timestep to another. This allows to understand how individual agents move, regardless of their actual position in space. The latter instead, processes the absolute agent positions to obtain knowledge of where an agent is with respect to the environment. This information is then stored into an external memory, shared across agents. Our model therefore can learn to perform social reasoning by manipulating memory entries to predict future positions for all agents in the scene.

In the Egocentric Stream, at each timestep t , past displacements Δx_t^i are observed for each agent trajectory $\mathbf{x}^i \in \mathcal{S}$. Each displacement is first processed by an encoder E_Δ to obtain a projection δ_t^i into a higher dimensional space. The temporal sequence of δ_t^i is then fed to a recurrent motion encoder E_T , which generates a condensed feature representation τ_t^i .





In the Social Stream, past absolute positions x_t^i are considered for each agent trajectory $\mathbf{x}^i \in \mathcal{S}$. A projection π_t^i is obtained with an encoder E_{Π} . This yields a sequence of temporized descriptors, which is directly fed to the Social Memory Module. This module acts as a recurrent neural network and processes a sequence of input features in parallel for each agent. It generates a compact social descriptor σ_t^i , summarizing social behaviors between all agents in the social context \mathcal{S} up to the current timestep t . The i superscript denotes a separate social descriptor for each agent, beyond the fact that all participate in a common social context. This is necessary since agents interact differently with the others depending on their position and movement.

The egocentric and social representations, τ_t^i and σ_t^i , are finally concatenated and fed to a recurrent motion decoder D_T and the model autoregressively predicts future displacements Δx_{t+1}^i , for each agent, with a decoder D_{Δ} . Each autoregressive step works as follows. E_{Δ} and E_{Π} respectively process each $\Delta \mathbf{x}_{0:P}^i$ and $\mathbf{x}_{0:P}^i$ independently, generating at each timestep the latent representations δ_t^i and π_t^i , until the present is reached.

For each timestep in the future, instead, δ_t^i and π_t^i are replaced with a vector of zeros to allow the autoregressive trajectory generation. The recurrent encoder E_T and the Social Memory Module therefore keep updating their internal state and new τ_t^i and σ_t^i are generated for each instant in the future.

4.4.2.2. FLODCAST We design **FLODCAST**, a novel optical **FLO**w and **Depth foreCASTing** network that anticipates both modalities at each future time step by observing the past ones.

FLODCAST takes a sequence $X = \{X_1, X_2, \dots, X_T\}$ of T past observations composed of dense optical flows and depth maps. In detail, each X_t encodes the input features for the image I_t in the past, that are obtained by concatenating the optical flow OF_t with the depth map D_t . In other words, $X_t = (OF_t \oplus D_t)$. We use a shared UNet to compute an intermediate representation Φ_t for each X_t . $X_{T-K}, X_{T-K+1}, \dots, X_T$ are then forwarded into our ConvLSTM module to extract our future prediction feature Ω that is used as an input for the two final branches. The model generates as output a sequence $\hat{X} = \{\hat{X}_{T+1}, \hat{X}_{T+2}, \dots, \hat{X}_{T+K}\}$, that is a sequence of K future optical flows and K depth maps. We set $T=3$ and $K=3$ in all our experiments.

Since optical flows and depth maps encode very different information about the scene, we add two separate heads after extracting features from the input in order to handle multimodal predictions. Therefore, we feed in input a sequence of concatenated optical flows and depths $\{X_1, X_2, \dots, X_T\}$ to a single recurrent ConvLSTM network, in which a UNet backbone is used to extract features at 64 channels for each input X_t , $t=1, \dots, T$, so to output a tensor of size $(H \times W \times 64)$, where $(H \times W)$ is the input resolution. Our feature extractor is the same UNet architecture as in [176], i.e. a fully convolutional encoder-decoder network with skip connections, consisting of 5 layers with filters $\{64, 128, 256, 512, 1024\}$ respectively. These 64-channel features capture meaningful spatio-temporal contexts of the input representation. The features are then passed to the two convolutional heads, which are end-to-end trained to simultaneously generate the sequence of future optical flows and depth maps. Each head is a fully convolutional network made of sequences of Conv2D+ReLUs with $\{32, 16, 8\}$ filters.

Finally, we append at the end of the optical flow head a convolution operation with $2 \times K$ channels and we use a \tanh activation function, so to produce the (u, v) flow field values normalized in $(-1, 1)$. Instead, after the depth head, we attach a convolution operation with a K channels and a sigmoid activation in order to get depth maps normalized in $(0, 1)$. Instead of outputting one prediction at a time as in prior work [176], we directly generate K flows and depth maps simultaneously, to make the model faster compared to autoregressive models which would require looping over future steps.

To train FLODCAST we compute a linear transformation of the original input values, by rescaling depth map values in $[0, 1]$ and optical flows in $[-1, 1]$ through a min-max normalization, with minimum and maximum values computed over the training set.

We use the reverse Huber loss [179], called *BerHu* for two main reasons: (i) it has a good balance





between the two L1 and L2 norms since it puts high weight towards values with a high residual, while being sensitive for small errors; (ii) it is also proved to be more appropriate in case of heavy-tailed distributions, that perfectly suits our depth distribution. BerHu minimizes the prediction error, through either the L2 or L1 loss according to a specific threshold c calculated for each batch during the training stage. Let $x = \hat{y} - y$ be the difference between the prediction and the corresponding ground truth. This loss $\mathcal{B}(x)$ is formally defined as:

$$\mathcal{B}(x) = \begin{cases} |x|, & |x| \leq |c| \\ \frac{x^2 + c^2}{2c}, & \text{otherwise} \end{cases} \quad (16)$$

Thus, we formulate our compound loss, using a linear combination of the optical flow loss $\mathcal{L}_{\text{flow}}$ and the depth loss $\mathcal{L}_{\text{depth}}$ (Eq. 17):

$$\mathcal{L} = \alpha \mathcal{L}_{\text{flow}} + \beta \mathcal{L}_{\text{depth}} \quad (17)$$

Specifically, we apply the reverse Huber loss to minimize both the optical flow and depth predictions, using the same loss formulation, since the threshold c is computed for each modality, and that value depends on the current batch data. Therefore, $\mathcal{L}_{\text{flow}}$ is the loss function for the optical flow computed as:

$$\mathcal{L}_{\text{flow}} = \frac{1}{M} \sum_{j=1}^M \mathcal{B}(|OF_j - \widehat{OF}_j|) \quad (18)$$

where $M = B \times R \times 2$, since the flow field has (u, v) components over R image pixels and B is the batch size, whereas OF_j and \widehat{OF}_j are the optical flows, respectively of the ground truth and the prediction at the pixel j . Likewise, we do the same for the depth loss $\mathcal{L}_{\text{depth}}$:

$$\mathcal{L}_{\text{depth}} = \frac{1}{P} \sum_{j=1}^P \mathcal{B}(|D_j - \widehat{D}_j|) \quad (19)$$

where $P = B \times R$, D_j and \widehat{D}_j are the depth maps, respectively of the ground truth and the prediction at the pixel j . We set $c = \frac{1}{5} \max_j (|y_j - \hat{y}_j|)$, i.e. the 20% of the maximum absolute error between predictions and ground truth in the current batch over all pixels.

4.4.3. Experimental Results

4.4.3.1. SMEMO In Table 25 we report results for SMEMO on Stanford Drone SDD for 20 futures in terms of Final Displacement Error (FDE) and Average Displacement Error (ADE). On the SDD dataset, SMEMO obtains state-of-the-art results, except for (FDE) at $K = 20$, where it reports competitive results with the top three performing methods.

4.4.3.2. FLODCAST We report depth forecasting results in Table 26. We exceed all the previous methods at short-term and mid-term predictions. Specifically, we beat all the existing approaches at short-term by a large margin for all the metrics, also reporting the highest inlier percentage. At mid-term term we exceed all the state-of-the-art approaches, in terms of AbsRel and SqRel, including the recent DeFNet (-42% and -8%), which employs both RGB frames and optical flows, even considering the camera pose during the training.

4.4.4. Conclusion

We presented several methods to perform forecasting in multimedia content. The algorithmic nature of SMEMO is able to learn the set of social rules yielding behaviors of pedestrians during their interaction.





K=20					
Method	ADE	FDE	Method	ADE	FDE
Trajectron++ [180]*	19.30	32.70	MID [181]	9.73	15.32
SoPhie [182]	16.27	29.38	MANTRA [183]	8.96	17.76
EvolveGraph [184]	13.90	22.90	LB-EBM [185]	8.87	15.61
CF-VAE [186]	12.60	22.30	PCCSNet [187]	8.62	16.16
P2TIRL [188]	12.58	22.07	MemoNet [189]	8.56	12.66
Goal-GAN [190]	12.20	22.10	LeapFrog [191]	8.48	11.66
Expert-Goals [192]	10.49	13.21	Y-Net [193]	8.25	12.10
SimAug [194]	10.27	19.71	SMEMO	8.11	13.06
PECNet [195]	9.96	15.88			

Table 25. Results on SDD. K is the number of predictions generated by the models.

We report state-of-the art results for SMEMO on ETH/UCY and SDD datasets. As a byproduct, we show that SMEMO can provide explainable predictions by design, simply looking at attention weights of its memory reading controllers.

Regarding FLODCAST, we shown the superiority of exploiting both optical flow and depth as input data against single-modality models, showing that leveraging both modalities in input can improve the forecasting capabilities for both flow and depth maps, especially at farther time horizons. Moreover, FLODCAST can be applied on the downstream task of segmentation forecasting, relying on a mask-warping architecture.

4.4.5. Relevance to AI4Media use cases and media industry applications

This methods are useful for UC3 (AI in Vision - High quality Video Production and Content Automation) since it yields a method to forecast trajectories of moving objects that can help UAV and other agents in planning their own trajectory for automatic cinematography applications.

4.4.6. Relevant Publications

- Marchetti, Francesco, Federico Becattini, Lorenzo Seidenari, and Alberto Del Bimbo. "Smemo: social memory for trajectory forecasting." IEEE Transactions on Pattern Analysis and Machine Intelligence (2024).
- Marchetti, Francesco, Federico Becattini, Lorenzo Seidenari, and Alberto Del Bimbo. "Explainable sparse attention for memory-based trajectory predictors." In European Conference on Computer Vision, pp. 543-560. Cham: Springer Nature Switzerland, 2022.
- Ciamarra, Andrea, Federico Becattini, Lorenzo Seidenari, and Alberto Del Bimbo. "FLODCAST: Flow and depth forecasting via multimodal recurrent architectures." Pattern Recognition(2024).

4.4.7. Relevant software/datasets/other outcomes

SMEMO AI4EU asset page: <https://www.ai4europe.eu/research/ai-catalog/smemo-social-memory-trajectory-for>

4.5. 3D, 4D and other Modalities

Contributing partner: UNIFI





Table 26. Quantitative results for depth forecasting after $t+k$ on Cityscapes test set, both at short-term and mid-term predictions, i.e. at $k=5$ and $k=10$ respectively.

Short term $k=5$							
	Lower is better ↓				Higher is better ↑		
Method	AbsRel	SqRel	RMSE	RMSE-Log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Copy last	0.257	4.238	7.273	0.448	0.765	0.893	0.940
Qi et al. [196]	0.208	1.768	6.865	0.283	0.678	0.885	0.957
Hu et al. [197]	0.182	1.481	6.501	0.267	0.725	0.906	0.963
Sun et al. [198]	0.227	3.800	6.910	0.414	0.801	0.913	0.950
Goddard et al. [199]	0.193	1.438	5.887	0.234	0.836	0.930	0.958
DeFNet [200]	0.174	1.296	5.857	0.233	0.793	0.931	0.973
FLODCAST w/o flow	<u>0.084</u>	<u>1.081</u>	<u>5.536</u>	<u>0.196</u>	<u>0.920</u>	<u>0.963</u>	<u>0.980</u>
FLODCAST	0.074	0.843	4.965	0.169	0.936	0.971	0.984
Mid term $k=10$							
	Lower is better ↓				Higher is better ↑		
Method	AbsRel	SqRel	RMSE	RMSE-Log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Copy last	0.304	5.006	8.319	0.517	0.511	0.781	0.802
Qi et al. [196]	0.224	3.015	7.661	0.394	0.718	0.857	0.881
Hu et al. [197]	0.195	<u>1.712</u>	6.375	0.299	0.735	0.896	0.928
Sun et al. [198]	0.259	4.115	7.842	0.428	0.695	0.817	0.842
Goddard et al. [199]	0.211	2.478	7.266	0.357	0.724	0.853	0.882
DeFNet [200]	0.192	1.719	<u>6.388</u>	0.298	0.742	0.900	0.927
FLODCAST w/o flow	<u>0.130</u>	2.103	7.525	0.320	<u>0.863</u>	<u>0.931</u>	<u>0.959</u>
FLODCAST	0.112	1.593	6.638	0.231	0.891	0.947	0.969

4.5.1. Introduction

In this subsection, we discuss UNIFI’s contribution regarding learning representation for underused modalities such as 3D static and dynamic streams and event data. In [201], Graph Neural Networks have been used to upsample point cloud data. Two contribution regarded the study of emotions from facial imagery with underused modalities. A novel 4D dataset was proposed in [202] and a neuromorphic sensor was used to understand emotions from event data in [203].

4.5.1.1. Graph Neural Networks for PointClouds Our proposed method makes use of message passing Graph Networks, different neighbourhood sampling techniques and Generative Adversarial training.

We employed a Graph Neural Network for this task. More in detail, our architecture has been developed starting from [204]. The employed architecture works on unordered lists of x, y, z, t points, representing the last n frames fused together, using two Graph Convolutional Neural Networks (GCNs from here on) in an adversarial setting. The discriminator is based on [204], while the generator improves on the architecture proposed in [204]. In particular, we used different neighbors sampling techniques that were developed with the intent of collecting, for each point, features contemporaneously of its immediate neighborhood and also from furthest vertices of the whole point cloud without making the computation too expensive.



The fully convolutional nature of our generator network allows us to potentially train and test at different input and output resolutions.

The basic module composing our generator network is made of the combination of Edge Convolution [205] and Graph Attention Networks (GAT) [206]. The Edge Convolution allows us to perform message passing over a dynamic graph in which the edges are updated as the point cloud changes. The GAT side is used to perform an attentional aggregation over the features collected from the dynamic local neighbourhood, this in contrast with much more common choices for aggregation such as *max* or *average*. We refer to this combination module as *Edge Convolution with Attention*.

The core of the generator side of the architecture is the Parallel Double Sampling (PDS) module that performs two different graph convolutions using two different sets of sampled points. A simplified illustration of this module is presented in Figure 19. For each point, two sets of operations are performed in a parallel fashion. The first set, is a pipeline composed of:

- **Radius filtering:** For each vertex, a filtering step leaves as neighbors, with the capability of passing messages, only those vertices that belong to a sphere of radius r , centered on the vertex.
- **Furthest Point Subsampling:** We use the Furthest Point Subsampling (FPS) algorithm in [204] in order to sample temporarily, a fraction s of the original points that are the farthest away, inside the radius, from a starting point.
- **Convolution:** Graph convolution is applied over the remaining vertices, **independently of their number**, and their features are aggregated.

The second set of operations, performed in parallel to the first one, is composed of:

- **K-NN:** A fixed number of k closest vertices is selected as neighbors.
- **Convolution:** Graph convolution is applied over the vertices, and their aggregated features.

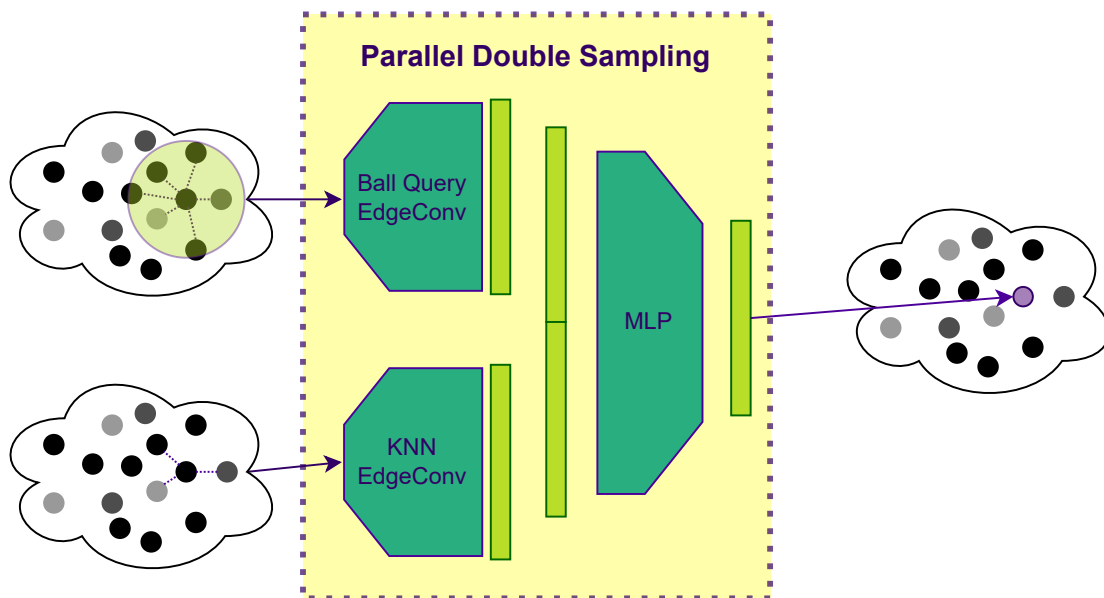


Figure 19. Schematic representation of the proposed Parallel Double Sampling (PDS) module.

Finally, the two sets of features are concatenated and fed to a linear layer that maps $2 \times Channels_{in} \rightarrow Channels_{out}$.

The developed architecture is composed of two Graph Convolutional Networks (GCNs) working in an adversarial setting (GAN) [207]. It is illustrated in the bottom of Figure 20. Basically, the point cloud given as input is processed as a graph using message passing based convolution.





The discriminator is inspired by the PointNet++ architecture [204], since it also targets a classification task. We use the same structure that progressively reduces the number of points using *max-pooling* operations and finally a sequence of linear layers before the output as shown in the bottom part of Figure 20.

The generator side of the model is instead built as an initial sequence of Edge Convolution with Attention modules followed by our Parallel Double Sampling (PDS) module. It is also inspired by the PointNet++ architecture [204]. In the top part of Figure 20 a simplified visualization of the PDS generator is presented. The generator is composed of multiple Graph Convolutions with Attention followed by a single PDS. The intuition behind this choice is to collect various features for each node, using different neighborhood sampling techniques. Once the original node has been enriched with the local features, the PDS will use them to generate multiple new vertices according to the scale factor. Finally this new vertices position is summed with the closest one that originated it, in a sort of residual fashion.

The generator loss L_g is composed of an adversarial component L_{adv} coming from the Discriminator, a full reference reconstruction loss L_{rec} computed as the Chamfer Distance between the restored point cloud and the original one and an additional Density Loss L_D . We used the LSGAN from [208] loss for our training, which assumes the form:

$$L_{adv} = \min_G L(G) = \frac{1}{2} \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})} \left[(D(G(\mathbf{z})) - c)^2 \right], \quad (20)$$

for the generator, and:

$$\min_D L(D) = \frac{1}{2} \mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})} \left[(D(\mathbf{x}) - b)^2 \right] + \quad (21)$$

$$+ \frac{1}{2} \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})} \left[(D(G(\mathbf{z})) - a)^2 \right], \quad (22)$$

for the discriminator.

The model is trained end-to-end using multiple losses. Beside the adversarial component L_{adv} , we also compute the point-to-set distance (Chamfer distance) L_{rec} between the reconstructed point cloud and the target one and, similarly to [209], we take into account the *neighbourhood* of each point. That is, for each reconstructed point $p_r \in P_r$, we find the closest point $p_t \in P_t$ in the target point cloud, and compute both the distance between them and the difference in terms of local neighbors:

$$L_{CD}(P_r, P_t) = \sum_{r \in P_r} \min_{t \in P_t} \|r - t\|_2^2 + \sum_{t \in P_t} \min_{r \in P_r} \|r - t\|_2^2. \quad (23)$$

We define a vertex p *neighbourhood* density $D(p)$ as the normalized sum of its neighbours in a given radius:

$$D(p \in P) = \frac{1}{N_{max}} \sum_{n \in Ball_p} 1, \quad (24)$$

$$L_D(P_r, P_t) = \sum_{r \in P_r} \min_{t \in P_t} \|D(r) - D(t)\|_2^2 + \sum_{t \in P_t} \min_{r \in P_r} \|D(r) - D(t)\|_2^2. \quad (25)$$

The generator final loss is therefore given by:

$$L_{rec} = \lambda_1 L_{CD} + \lambda_2 L_D + \lambda_3 L_{Adv}, \quad (26)$$

where values for λ_i have been empirically determined ($\lambda_1 = 1.0, \lambda_2 = 0.5, \lambda_3 = 0.1$).



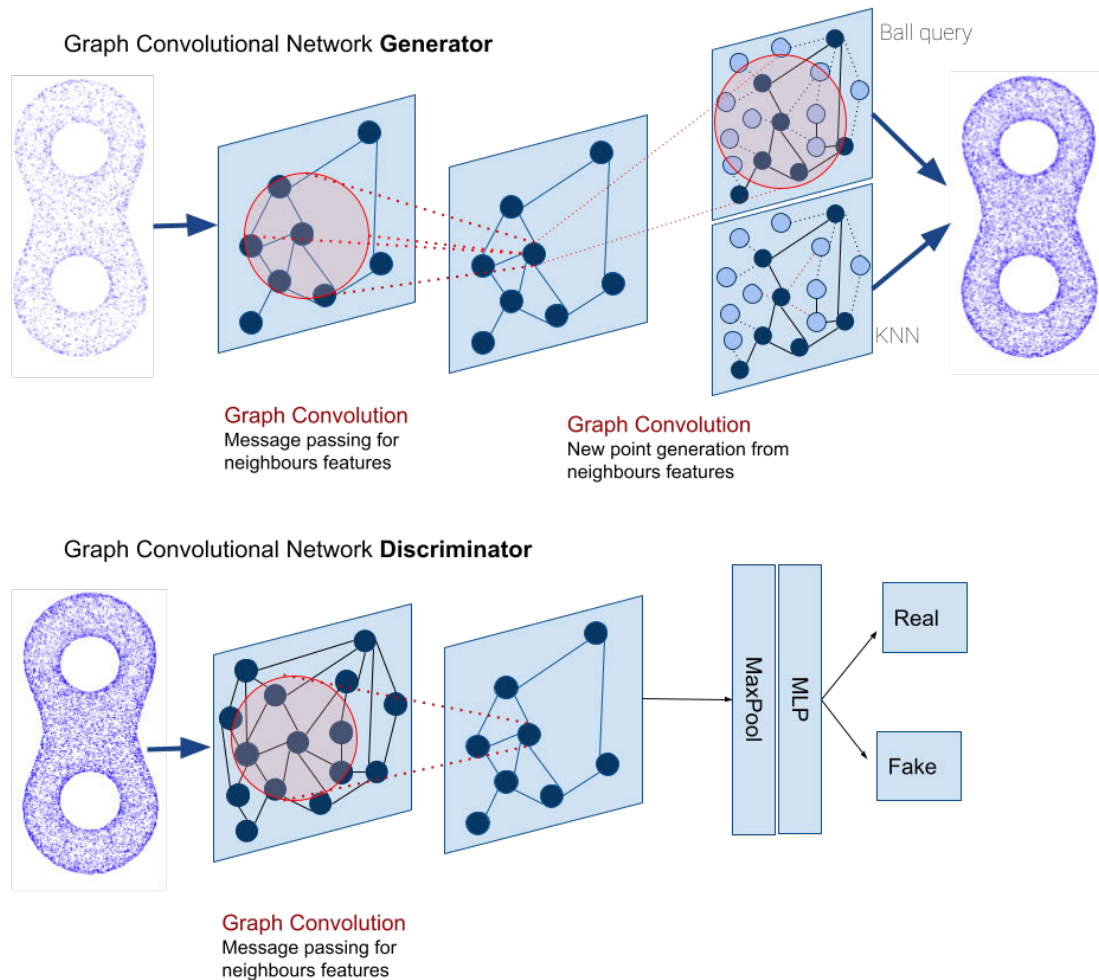


Figure 20. Schematic representation of the proposed GCN architecture. Top: Generator architecture; Bottom: Discriminator architecture.

4.5.1.2. Florence 4D Faces We identified a key missing aspect in the current literature of 4D face analysis, that is the ability of modeling complex, non-standard expressions and transitions between them. Indeed, current models and datasets are limited to the case, where a facial expression is performed assuming a neutral-apex-neutral transition. This does not hold in the real world, where people continuously switch between one facial expression to another. These observations motivated us to generate the proposed Florence 4D dataset, which is described in the following sections.

Florence 4D includes real and synthetic identities from different sources: (a) CoMA identities; (b) high-resolution 3D face scans of real identities; (c) synthetic identities.

The CoMA dataset [210] is largely used for the analysis of dynamic facial expressions. An important characteristic of this dataset that contributed to its large use is the fixed topology, according to which all the scans have 5,023 vertices that are connected in a fixed way to form meshes with 9,976 triangular facets. The dataset includes 12 real identities (5 females and 7 males).

On the Web, a large number of 3D models of synthetic facial characters, either females or males, can be purchased or downloaded for free. Using these online resources, we were able to add 63 synthetic

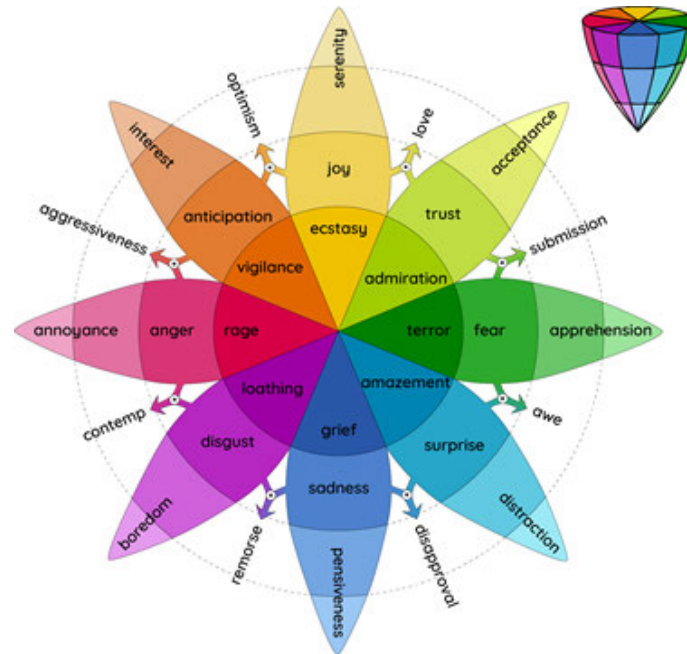


Figure 21. Plutchik's wheel of emotions [9], illustrating expression relations.

identities (33 females and 30 males) to the data, selecting those that allow editing and redistribution for non-commercial purposes. Subjects are split in three ethnic groups, Afro (16%), Asian (13%), and Caucasian (71%). Because such identities are synthetic, the resulting meshes are defect free, and perfectly symmetric, which is different from real faces. To make models more realistic, morphing solutions were applied to include face asymmetries.

We acquired 3D scans of 20 subjects (5 females and 15 males) with a 3DmD HR scanner. Subjects are mainly students and university personnel, 30 years old on average. Meshes have approximately 30k vertices. Written consents were collected for these subjects for using their 3D face scans.

Combining together the identities from the three sources indicated above, we obtained an overall number of 95 identities, 43 females and 52 males. Identities corresponding to synthetic 3D models and 3D scans of real subjects have different topology when compared with CoMA, and a variable number of facets and vertices. Instead, one objective of our dataset was that of providing identities with the same topology as the CoMA dataset (i.e., 5,023 vertices and 9,976 triangular facets). To this end, we used a workflow that involved the joint use of the DAZ Studio [211] and R3DS Wrap 3 [212] software to homogenize the correspondence of the identity meshes. All identities were converted into morphs of the DAZ Studio's Genesis 8 Female (G8-F) base mesh using the Wrap 3 software that allows one mesh to be wrapped over another by selecting corresponding points of the two meshes. The wrapped meshes were then associated with the G8-F mesh as morphs. At the end of the process, we got a G8-F mesh with 95 morphs of different identities. After animating the facial expressions and before exporting the sequence of meshes, we restored the animated G8-F to the original topology of the CoMA dataset.

With the basic Genesis 8 mesh, we also got a set of facial expressions, in the form of morphs that we used for our dataset. The number of presets was expanded by downloading free and paid packages from the DAZ Studio online shop and from other sites. The base set included 40 different expressions. A paid package of 30 more expressions was added, obtaining a total of 70 different expressions. These expressions were classified according to the Plutchik's wheel of emotions [9], which is illustrated in Figure 21. Following this organization of expressions, we generated a set of secondary expressions from



the eight primary ones (for each primary expression, the number of expression per class is indicated): *anger*, AR (6), *fear*, FR (6), *sadness*, SS (13), *disgust*, DT (9), *surprise*, SE (11), *anticipation*, AN (4), *trust*, TT (6), *joy*, JY (15). Details are given in Table 27.

Table 27

Primary expression	Expressions
<i>Anger</i> , AR (6)	Angry1, Angry2, Fierce, Glare, Rage, Snarl
<i>Fear</i> , FR (6)	Afraid, Ashamed, Fear, Scream, Terrified, Worried
<i>Sadness</i> , SS (13)	Agony, Bereft, Ill, Mourning, Pain, Pouting, Pouty, Sad1, Sad2, Serious, Tired1, Tired2, Upset
<i>Disgust</i> , DT (9)	Arrogant, Bored, Contempt, Disgust, Displeased, Ignore, Irritated1, Irritated2, Unimpressed
<i>Surprise</i> , SE (11)	Awe, Confused, Ditzzy, Drunk1, Frown, Hurt, Incredulous, Moody, Shock, Surprised, Suspicious
<i>Anticipation</i> , AN (4)	Cheeky, Concentrate, Confident, Cool
<i>Trust</i> , TT (6)	Desire, Drunk2, Flirting, Hot, Kissy, Wink
<i>Joy</i> , JY (15)	Amused, Dreamy, Excitement, Happy, Innocent, Laughing, Pleased, Sarcastic, Silly, Smile1, Smile2, Smile3, Smile4, Triumph, Zen

In the dataset, we named the expressions with pairs of names representing the abbreviation of the primary emotion and the facial expression represented, e.g., JY-smile or SE-incredulous. The Genesis 8 mesh also has 70 morphs of facial expressions available, in addition to 95 identity morphs.

Using the above expression classification, we generated the expression sequences of each identity by iterating through the activation of the expression morphs for each identity morph. The dataset includes two types of sequences for each identity: *single expression* and *multiple expressions*.

For each identity, the animation of each morph expression is generated as follows:

- Frame 0 - neutral expression (morph with weight 0);
- Random frame between 10 and 50⁷ - expression climax (morph with weight 1);
- Frame 60 - neutral expression (morph with weight 0).

The meshes in a sequence are named with the name of the expression and the number of the corresponding frame as a suffix (e.g., *Smile_01*). An example is shown in the top row of Figure 22.

For each identity, we created mesh sequences of transitions from a neutral expression to a first expression (expr. 1), then from this expression to a second one (expr. 2), then back from the latter to the neutral expression. Also in this case, the climax frames of the two expressions were randomized to obtain greater variability (i.e., the apex frame for each expression can occur at different times of the sequence). Summarizing, these sequences were created following this criterion:

- Frame 0 - neutral expression (morph expr. 1 weight 0);
- Random frame between 15 and 40 - morph expr. 1 with weight 1, and morph expr. 2 with weight 0;
- Random frame between 50 and frame 75 - morph expr. 1 with weight 0, and morph expr. 2 with weight 1;
- Frame 90 - neutral expression (morph expr. 2 with weight 0).

Meshes in a sequence are named with the initials of the primary emotions to which the two expressions involved in the animation belong to, followed by the name of the first and second expression plus a numeric suffix for the frame (e.g., *AN-AR_Confident_Glare_01*). An example is shown in the bottom row of Figure 22.

⁷With the randomization of the climax frame, we generated a greater variability in the speed of the transition from the neutral to the climax expression and back to the neutral expression for each identity.



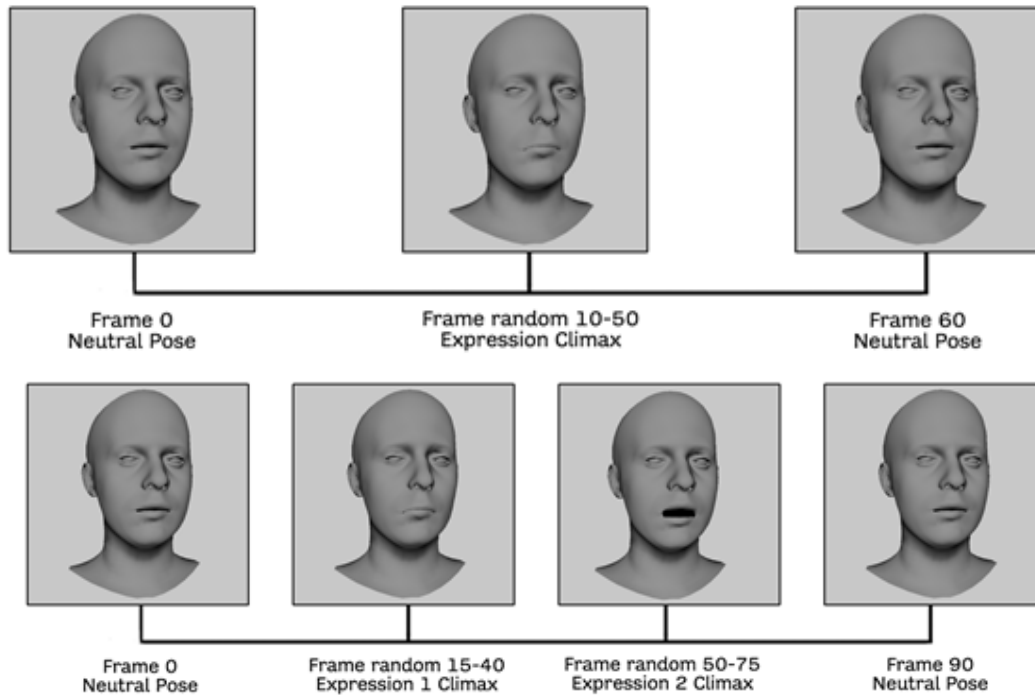


Figure 22. Sample frames from a generated sequence: (top) the expression passes from neutral to apex and to neutral again; (bottom) the expression passes from neutral to apex for expr. 1, then to apex for expr. 2, and finally to neutral again.

Table 28 reports a quick summary of the main characteristics of the Florence 4D released data. In particular, we reported the number of identities (male and female), the number of vertices per mesh (same topology for all models), the number of different expressions per identity, the number of sequences that show a neutral-apex expression-neutral transition (6,650 in total); the number of sequences with neutral-expr. 1-expr. 2-neutral transition. Note that, in this latter case, all the possible expression combinations have been generated for a total of 198,550 sequences.

We also note the neutral-expr-neutral sequences include 60 frames each, with the apex intensity for the expression occurring around frame 30; 90 frames are instead generated for the sequences with an expression-to-expression transition, with the expr. 1 apex and the expr. 2 apex occurring around frame 30 and 60, respectively.

Table 28. Florence 4D expression dataset: summary of released data

#IDs (m/f)	#vert	#exprs.	# n-exp-n/# f	# n-exp1-exp2-n/# f
95 (52/43)	5,023	70	70*95 / 60	2090*95 / 90

Some examples of the generated sequences are illustrated in Figure 23. In the top row, the apex frames of nine expression sequences (i.e., smile, wink, disgust, sad, angry, arrogant, fear, happy irritated) of a male synthetic subject are illustrated. The second and third row compare frames of an *angry* expression for a male and a female subject. The two bottom rows, instead, show the transitions *happy-pain*, and *confident-frown* for a given subject.

4.5.1.3. NEFER: Neuromorphic Event-based Facial Expression Recognition The purpose of NEFER is to capture genuine micro-expressions associated to specific emotions with both an event camera



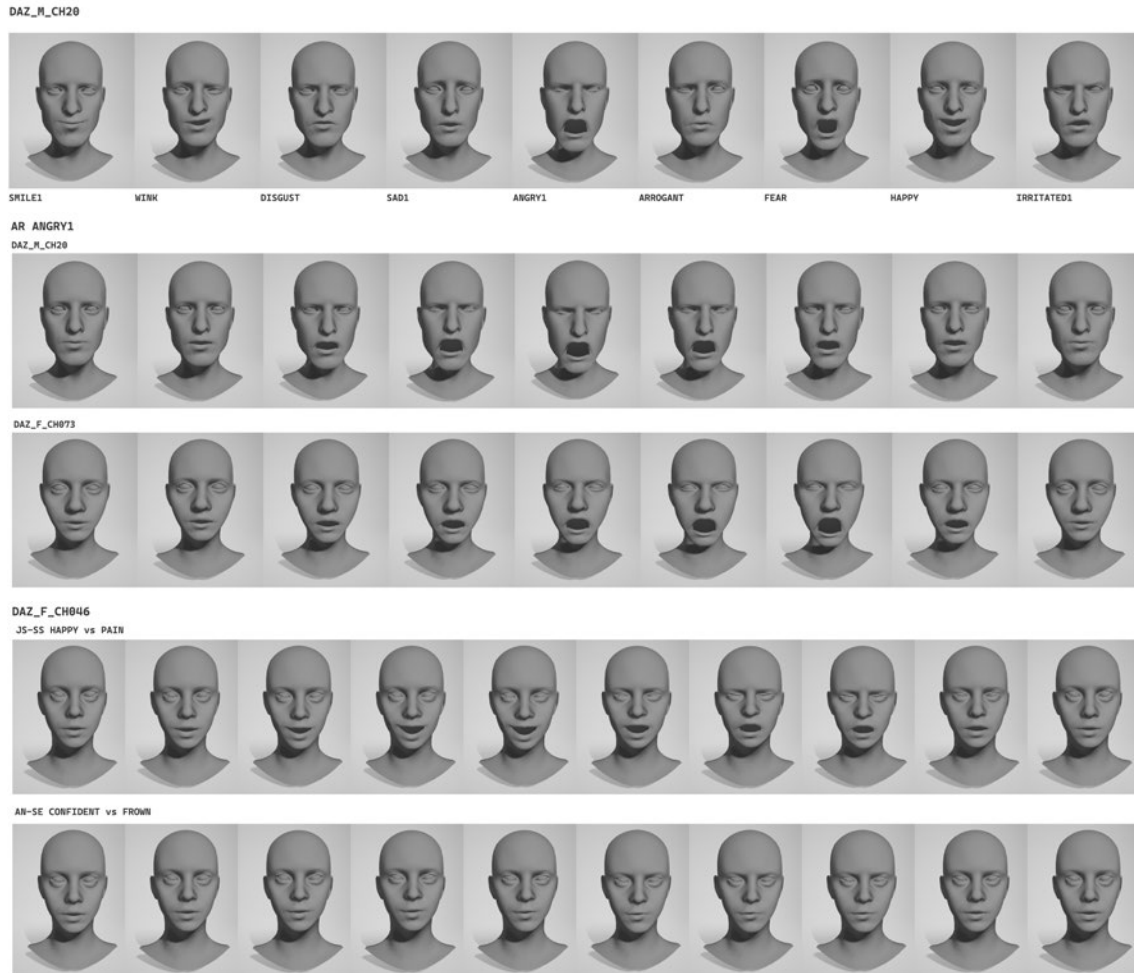


Figure 23. Examples frames from generated sequences: (top) apex frames of nine expression sequences for subject *DAZ_M_CH20*; (middle) angry expression for a male (*DAZ_M_CH20*) and a female (*DAZ_F_CH073*) subject; (bottom) For subject *DAZ_F_CH046* the transitions happy-pain, and confident-frown are shown.

and a standard RGB camera. We considered the 7 primary emotions defined by Ekman [213], namely *Disgust*, *Contempt*, *Happiness*, *Fear*, *Anger*, *Surprise* and *Sadness*, since these have been identified as independent from culture, history and personality and are performed in a similar way by everyone.

In order to obtain realistic and non-simulated expressions, we asked a set of volunteers to maintain a neutral facial expression while watching a selection of videos. A reward has been offered to the participants to encourage a proper behavior during the test (high-stakes situation). The volunteers that took part in the creation of NEFER are both males and females of age ranging between 24 and 52 years, for a total of 29 users.

We showed to each user 21 different videos, 3 for each of Ekman's basic emotions. The videos have been selected from online streaming platforms (e.g. YouTube). Each video was trimmed to the same length of 7s to keep the recording sessions as short as possible so not to induce unwanted expressions due to, for instance, boredom. This choice also simplifies training schemes with deep learning frameworks which process data in mini-batches of the same size. The overall procedure for the data acquisition and



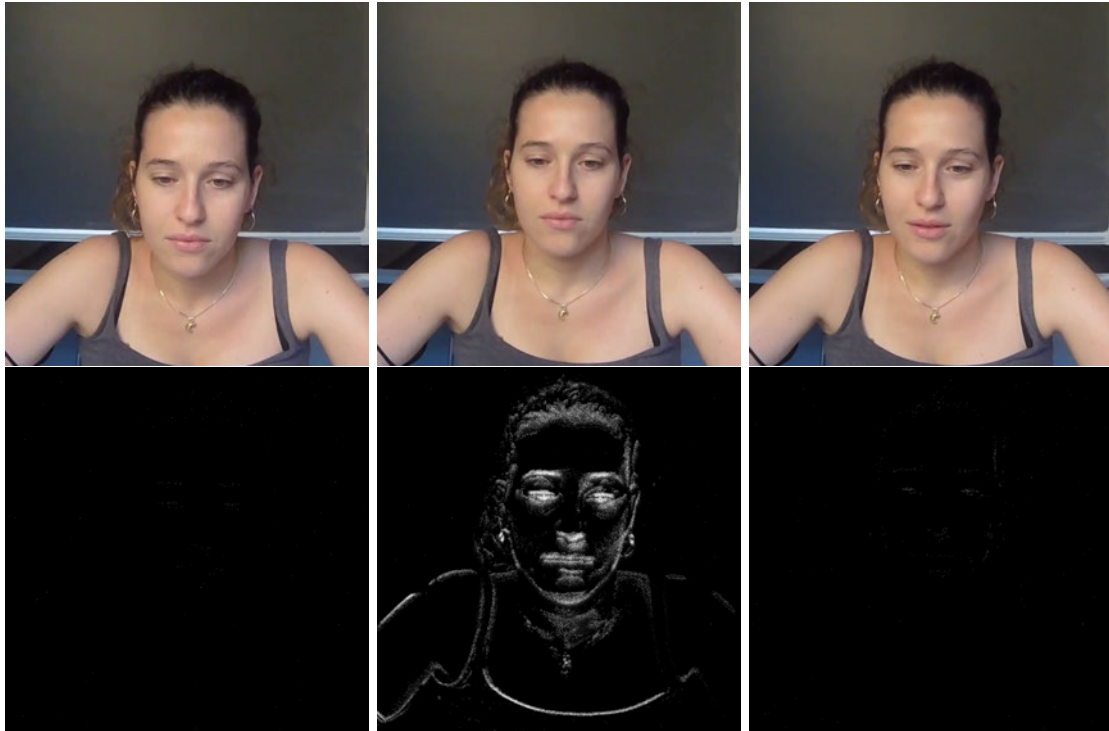


Figure 24. Four samples from the NEFER dataset. First row: happiness; Second row: fear; third row: disgust; fourth row: surprise. Subtle movements are almost invisible with RGB but are emphasized in event frames.

video selection was inspired by previously collected dataset from the state of the art [214, 215].

For the recording we used two capturing devices: a GOPRO Hero+ action camera, recording videos at 60FPS and 1920×1080 px resolution, and a Prophesee Evaluation Kit HD, recording event videos at a resolution of 1280×720 px. The cameras have been mounted on a fixed recording rig in a room lit with natural light. We specifically avoided any presence of artificial light to avoid background noise that could alter the event-based recordings. Users are also isolated from other people which could generate distractions.

Users have been asked to sit in front of the screen at approximately 60cm from the cameras. The RGB and event streams have been programmatically synchronized in order to capture two videos of the same duration and content. After viewing each video, we asked the volunteers to provide a personal evaluation of the observed footage. In particular, we asked two questions: (i) select among the 7 basic emotions, plus a "None" option, the most suitable one to describe the emotions stemmed from viewing the video; (ii) the intensity, on a 1 to 5 scale, of such emotion. We used the collected answers to create two alternative versions of the annotations, one considering the labeling of the user and one following our a-priori video-emotion assignment. The two versions mostly differ in the fact that following user labelings we have the additional neutral emotion and a slight unbalance in the sample distribution. Overall, recording sessions lasted 18 minutes on average. Figure 24 shows a few samples from the dataset.

The wide range of off-the-shelf functionalities for RGB-based computation is not available for event-based data. This includes modules that nowadays are common building blocks in computer vision pipelines such as face detectors and landmark estimators. In addition, it is necessary to preprocess the raw data of the neuromorphic sensor in order to use it with frame-based computational tools. Bridging this gap is not trivial, since due to the asynchronous nature of the domain, the usual annotation process for many different





Figure 25. Examples of detected faces and estimated landmarks on real event videos of NEFER. Better viewed in color on a PC screen. Bounding boxes are shown in green, landmarks are shown in yellow.

tasks becomes cumbersome and expensive. Even generating relatively simple annotations such as facial bounding boxes, which are reliably obtainable with RGB data, would require lots of manual annotation.

To provide additional annotations for event-based data we exploit RGB data and an event camera simulator, ESIM [216]. Through the use of the ESIM simulator we convert the RGB videos into physically accurate simulated event streams. We then run a face detector and facial landmark estimator on the RGB frames, which is easily done with tools such as FaceAlignment [217]. We train a face detector (Yolov2)[218] and a landmark estimator [217] on simulated data and test it on real event streams. This approach provides satisfactory results on most frames, decimating the annotation time. The final annotations are manually refined and validated using CVAT [219].

ESIM [216] is an event-based camera simulator that can generate a synthetic event-based stream from its RGB video counterpart in a physically realistic way. The images are rendered by the simulator at a high frame rate, interpolating pixel brightness along the camera trajectory using an adaptive sampling technique, which adapts the frame rate based on a prediction of the previous signals. We feed to the simulator all the RGB frames to generate a synthetic event-based version of each stream. In this way, we are able to associate the bounding boxes provided by face alignment on RGB frames with event data. The simulator-generated outputs are encoded using an exponential time surface [220]. Note the synthetic event-based videos obtained from the RGB data are used only as a means for training models to quickly collect annotations. These are not pixel-wise aligned with the real event streams and we do not treat them as part of the final dataset, which only comprises real event data.

Using the synthetic data from the simulator, we generated an annotated dataset in the event spectrum to train a face detector. First, we generated face annotation for RGB frames using FaceAlignment [217], an open-source tool for face analysis⁸. We then bound the face labels with the corresponding synthetic event frames obtained with ESIM. This allowed us to train a YOLOv2 [218] on the synthetic version of NEFER. We found the detector to have good generalization capabilities from synthetic to real event data, which yielded high-quality annotations at a slight cost of manual validation using CVAT [219].

The facial landmark detection is performed by an Xception [221] architecture trained on the synthetic data from ESIM to regress the position of 68 landmarks of the face. Similarly to face detection, we obtained the ground truth labels from the RGB videos by using FaceAlignment [217]. The Xception architecture is composed of three stages, all of them employing depthwise separable convolutions along skip connections, resulting in a faster convergence training [221]. The final linear layer outputs the 136 normalized numbers representing the coordinates of the standard 68 facial landmarks. The model is optimized using Adam with a learning rate of 8×10^{-4} for 10 epochs over 30K frame samples with the use of standard

⁸<https://github.com/1adrianb/face-alignment>





augmentation techniques (random changes in brightness, contrast, rotation, translation, and crop).

We provide a simple baseline for the dataset. This baseline architecture is based on a 3D convolutional network C3D [222]. It has been chosen as it has been a long-standing, simple, standard approach for video-based action and activity recognition tasks [223, 224, 225, 222]. The C3D model is implemented using 5 3D convolutional blocks, all with kernel size 3 and padding 1, followed by a 3D max-pooling of size 2 and stride 2. This chain of sequential blocks reduces the input stacked sequence of images down to a 72 channels feature map, which is then flattened and fed to two fully connected layers of size 512 and 64 before a final classification layer. ReLU activations are present between all layers.

We train the same model separately with RGB-frame-based data and with event data obtained by converting events into frame-wise representations using Temporal Binary Representation (TBR) [226]. We detect the face using our pre-trained detector, and resize the bounding box to a 200×200 px patch before feeding it as input to the model.

Temporal Binary Representation [226] (TBR) is an aggregation strategy to map the asynchronous events into a stream of synchronous frames that can be then processed by a standard computer vision pipeline. Given a fixed Δt we can build the binary representation b^i of a pixel at (x,y) by checking for an event in such a time interval, $b_{x,y}^i = \mathbb{1}(x,y)$.

We can then collect N consecutive representations and stack them together as $B \in \mathbb{R}^{H \times W \times N}$ forming for each pixel a binary string $[b_{x,y}^0, b_{x,y}^1, \dots, b_{x,y}^N]$.

This approach manages to create a frame processable by traditional Computer Vision algorithms with a minimal memory footprint and by retaining temporal information within the value of each pixel.

For our experiments, we used this representation setting $\Delta t = 15$ milliseconds and $N = 8$.

4.5.2. Experimental Results

4.5.2.1. Graph Neural Networks for PointClouds

To measure the reconstruction quality we applied the standard Chamfer Distance (CD), a point-to-set metric since following the same protocol as reported in [227] that uses the CD as reconstruction metric for measuring the dissimilarity between a point and a point set.

We compared our approach with respect to six state-of-the-art solutions in the literature for 4D reconstruction from point cloud sequences, namely, PSGN 4D, ONet 4D, OFlow, LPDC, 4DCR, and RFNet-4D. The PSGN 4D extends the PSGN approach [228] to predict a 4D point cloud, *i.e.*, the point cloud trajectory instead of a single point set. The ONet 4D network is an extension of ONet [229] to define the occupancy field in the spatio-temporal domain by predicting occupancy values for points sample in space and time. The OFlow network [230] assigns each 4D point an occupancy value and a motion velocity vector and relies on the differential equation to calculate the trajectory. The LPDC [231] learned a temporal evolution of the 3D human shape through spatially continuous transformation functions among cross-frame occupancy fields. The 4DCR solution [232] used a compositional representation that disentangles shape, initial state, and motion for a 3D object that deforms over a temporal interval. Finally, RFNet-4D [227] jointly reconstructs objects and their motion flows from 4D point clouds.

Tables 29 and 30 report results for our solution and for the other methods as given in [227]. For our method (last line in the tables) we used 3 frames for upscaling at 60fps with a scale factor of $\times 4$ starting from low-resolution point clouds composed of 1024 vertices. For the *unseen individual and seen motion* protocol in Table 29, our approach achieves the second best score. From Table 30, it can be observed that our method reached a reconstruction error of similar magnitude with respect to the two best performing methods, *i.e.*, RFNet-4D and LPDC. It is worth noting that RFNet-4D obtained the reported error using a larger number of input frames (*i.e.*, 17 against 3 to 8 as used in our tests). It was not possible to test the RFNet-4D with our setting because the code was not publicly available.

In Table 31, we report the inference time, in seconds, for various different configurations of our model. All the measurements correspond to experiments executed on an Nvidia 2080Ti GPU. The





Method	Chamfer Distance x 10^{-3} ↓
PSGN-4D [228]	0.6877
ONet-4D [229]	0.7007
OFlow [230]	0.2741
4DCR [232]	0.2220
LPDC [231]	0.2188
RFNet-4D [227]	0.1594
Ours	<u>0.1758</u>

Table 29. Reconstruction accuracy for the unseen individuals and seen motions protocol. We report the Chamfer distance (lower is better). Results for the best and second best performing methods are given in bold and underlined, respectively. Our approach scored the second best accuracy.

Method	Chamfer Distance x 10^{-3} ↓
PSGN-4D [228]	0.6189
ONet-4D [229]	0.5921
OFlow [230]	0.1773
4DCR [232]	0.1667
LPDC [231]	<u>0.1526</u>
RFNet-4D [227]	0.1504
Ours	0.1638

Table 30. Reconstruction accuracy for the seen individuals and unseen motions protocol. We report the Chamfer distance (lower is better). Results for the best and second best performing methods are given in bold and underlined, respectively. Our approach results in the third best performance.

values reported in the table evidence that our approach can open the way to real-time upscaling. As reported in [227], their method used 17 input frames to reconstruct an output frame, while our range of frames is between 3 (for models using larger input point clouds) and 8 (for smaller inputs) due to memory constraint at training time.

Method	Input size	Upscale ×	Inference time (s) ↓
Ours	1024	3	0.103
Ours	1024	2	0.089
Ours	512	4	0.046
Ours	512	2	0.039
Ours	256	8	0.034
Ours	256	4	0.030
Oflow[229]	-	-	0.95
LDPC[231]	-	-	0.44
RfNet4d[227]	-	-	0.24

Table 31. Inference time for different configurations of our model using a three-frames buffer. Every test was performed on an Nvidia2080Ti. For the other models it must be noted that they used a 17 frame input sequence to output a frame.





Table 32. Reconstruction error (mm) on expression-independent (left) and identity-independent (right) splits

Method	Expression Split			Identity Split		
	CoMA	D3DFACS	Florence 4D	CoMA	D3DFACS	Florence 4D
PCA	0.76±0.73	0.42±0.44	0.70±0.81	0.80±0.73	0.56±0.56	0.16±0.17
DL3DMM [235]	0.86±0.80	0.73±1.15	0.83±1.03	0.89±0.79	1.15±1.50	0.17±0.18
Neural3DMM [233]	0.75±0.85	0.59±0.86	1.45±1.43	3.74±2.34	2.09±1.37	1.41±1.09
S2D-Dec	0.52±0.59	0.28±0.31	0.57±1.24	0.55±0.62	0.27±0.30	0.10±0.08

4.5.2.2. Florence 4D faces In the following, we report a baseline evaluation for the proposed dataset. We are interested in assessing to what extent our dataset, composed of re-parameterized real scans and totally synthetic sequences, compares to a reference dataset of real scans. We do this by evaluating the task of landmark-based 3D model fitting. As reference datasets to compare with, we chose CoMA and D3DFACS as they share the same mesh topology as Florence 4D, and are composed of 4D expression sequences. They are also common benchmarks employed in other recent studies [233, 210]. For a consistent comparison and fulfill our goal, given the way larger amount and variability of sequences included in Florence 4D, we selected 1,222 sequences from it, corresponding to the 7 standard expressions, to make it comparable in size and content to CoMA and D3DFACS. Following similar previous works [233, 234], we performed experiments by splitting the data into train and test. To make sure they do not overlap, in one case, we divided the data based on the identities (Identity Split), in the other, based on expressions (Expression Split). In both the cases, we performed a 4-fold cross validation.

Since the main focus of Florence 4D is on expressions, we decided to exclude the problem of identity reconstruction, to avoid ambiguities in the results. The goal is to fit a neutral (not average) 3D face of a subject $\mathbf{S}^n \in \mathbb{R}^{N \times 3}$ to a target expressive face \mathbf{S}^e guided by a set of 3D landmarks $Z^e \in \mathbb{R}^{68 \times 3}$. For evaluation, we set up a baseline by first comparing against standard 3DMM-based fitting methods. Similar to previous works [235, 236], we fit \mathbf{S}^n to the set of target landmarks Z^e using the 3DMM components. Since the deformation is guided by the landmarks, we first retrieved the landmark coordinates in the neutral face by indexing into the mesh, *i.e.*, $Z^n = \mathbf{S}^n(\mathbf{I}_z)$, where $\mathbf{I}_z \in \mathbb{N}^{68}$ are the indices of the vertices that correspond to the landmarks. We then found the optimal deformation coefficients that minimize the Euclidean error between the target landmarks Z^e and the neutral ones Z^n , and use the coefficients to deform \mathbf{S}^n . We experimented the standard PCA-based 3DMM and the DL-3DMM [235]. We also evaluated against recent deep models, including the Neural3DMM [233] and the very recent S2D-Dec [234]. In order to use Neural3DMM as a fitting method, we used the modified architecture as defined in [234], where the model was trained to generate an expressive mesh given its neutral counterpart and the target landmarks Z^e as input. The mean per-vertex Euclidean error between the reconstructed meshes and their ground truth was used as measure, as in the majority of works [233, 237, 238, 210].

Table 32 reports the results. It can be noted that for the expression split, results are similar for all the compared datasets. We argue this represents a piece of evidence that the synthetic expressions are as difficult to reconstruct as the real ones, making them valid to be used in practice. Results for the identity split are instead much lower for the proposed Florence 4D. Likely, the variability of synthetic identities is lower than that of real ones, being obtained as a result of a generative software process.

4.5.2.3. NEFER: Neuromorphic Event-based Facial Expression Recognition We implemented our C3D model using PyTorch and trained it using the Adam optimizer initialized at the default learning rate value of 1×10^{-4} which is then reduced following the scheduling technique presented in [239] with the annealing strategy. As a loss function, we adopt the Binary Cross-Entropy Loss, regularized with weight decay.

We compare the performances of our model by training it separately first on the RGB videos and





Data	A-Priori Labels	%	Reported Labels	%
RGB	14.60	-	14.37	-
TBR Event	22.95	+57.2%	30.95	+115.4%

Table 33. Absolute accuracy and relative performances of our baseline model over the different data domains and using both labelling versions of NEFER.

then on the event streams, using both the self-reported user annotations and the a-priori expected one as labels for the target emotion. We define a validation split by selecting 20% of the users at random (thus keeping each user either in the training set or in the validation set to avoid unwanted biases), for a total of 126 videos.

We found that the RGB model results in poor accuracy, obtaining an average of 14.37% using the user labels and 14.60% using the expected ones. The event-based model instead showed much better performances, reaching an accuracy of 22.95% with the user labels and 30.95% using the expected ones. We report these experimental results in Table 33. This confirms that neuromorphic cameras are well suited for analyzing faces and that event footage carries valuable information for identifying subtle micro-expressions that are not easily detectable with RGB data.

Interestingly, we observed that our baseline model, just as the human a-priori assumptions, tends to confuse classes that share similar expressions, such as *fear* with *surprise* or *anger* with *contempt* even when trained on the self-reported emotions.

4.5.3. Conclusion

4.5.3.1. Graph Neural Networks for PointClouds We presented a fully convolutional graph-based approach for time-varying point clouds upscaling using a novel and different approach with respect to most of the state-of-the-art models. Our proposed method is comparable with state-of-the-art solutions in terms of upsampling performance but it has a much lighter architecture that allows the implementation on edge devices with limited computational capabilities.

As a future development this type of application could be implemented as an update for older LiDAR devices or to allow faster 3D point cloud streaming by only transmitting/sampling a subset of the original points.

While our method tackles the problem in a different way bringing some advantages it still has some limitations and drawbacks:

- Training time and memory footprint. Not relying on an encoder-decoder model implies having the whole point cloud at every stage of the network in memory, this slows down training and poses some limitations in the number of input frames;
- Results for the reconstruction accuracy are comparable with those reported in the state-of-the-art, though a bit lower.

4.5.3.2. Florence 4D Faces Florence 4D's design and generation was guided by the goal of advancing the research in 4D facial analysis, with a particular focus on dynamic expressions. Compared to current datasets, its unique characteristic is that of including sequences of complex, non-standard expressions. Differently from the existing ones, Florence 4D also includes dynamic transitions across expressions, extending the standard neutral-peak-neutral setting. All the sequences were generated with randomized velocity for improved realism. The dataset is a combination of real and synthetic identities, while the expressions are fully synthetic. An experimental validation highlights the little domain gap with respect to real expressive scans, making it a valuable resource for real applications.





4.5.3.3. NEFER: Neuromorphic Event-based Facial Expression Recognition We presented a first release of NEFER, a dataset for expression recognition based on event camera data. This dataset is composed of paired visual spectrum images and event camera streams. For every sequence of frames, both the expected emotion and the self reported one by the user are given. Every frame has multiple annotations, namely the user face bounding box and the respective facial landmarks that we collected by leveraging models trained on synthetic data obtained using a simulator. Finally, we presented and discussed a 3D convolutional baseline, trained on both version of our dataset, which achieved improved results on event camera data with respect to the RGB frame based data.

4.5.4. Relevance to AI4Media use cases and media industry applications

This methods and datasets can be leveraged not just to analyze data from new media (3D, 4D, events) but also to create new content and are therefore useful for 3C2-8 (Synthetic Video Generation from Single Semantic Label Map).

4.5.5. Relevant Publications

- Berlincioni, Lorenzo, Luca Cultrera, Chiara Albisani, Lisa Cresti, Andrea Leonardo, Sara Picchioni, Federico Becattini, and Alberto Del Bimbo. "Neuromorphic event-based facial expression recognition." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (2023).
- Principi, Filippo, Stefano Berretti, Claudio Ferrari, Naima Otberdout, Mohamed Daoudi, and Alberto Del Bimbo. "The florence 4D facial expression dataset." In 2023 IEEE 17th International Conference on Automatic Face and Gesture Recognition (FG), pp. 1-6. IEEE, 2023.
- Berlincioni, Lorenzo, Stefano Berretti, Marco Bertini, and Alberto Del Bimbo. "4DSR-GCN: 4D Video Point Cloud Upsampling using Graph Convolutional Networks." In Proceedings of the 1st International Workshop on Multimedia Content Generation and Evaluation: New Methods and Practice, 2023.

4.5.6. Relevant software/datasets/other outcomes

- NEFER dataset: <https://github.com/miccunifi/NEFER>
- Florence 4D Facial Expression dataset: <https://www.micc.unifi.it/resources/datasets/florence-4d-facial-expression/>

4.6. Image and Video Quality Enhancement

Contributing partner: UNIFI

4.6.1. Introduction

In this subsection, we discuss UNIFI's contribution regarding evaluation and improvement of multimedia. In [240] a multimodal approach was developed to turn captioning algorithms into full-reference and no-reference image quality assessors. A GAN for Video enhancement based on keyframes was proposed in [241].

4.6.2. Methodology

4.6.2.1. KeyFrame GAN Our architecture is based on U-Net [242] and it is composed of an encoder, that processes the input so that it is smaller in terms of spatial dimensions but deeper in terms





of the number of channels, and by a decoder, that inverts the process. Multi-scale reference features are combined with the features of the degraded image in a progressive manner. This approach can make the network learn coarse-to-fine details and is beneficial to the restoration process.

Our model takes 3 inputs:

- a degraded (i.e. highly compressed) image;
- a high-quality reference image (i.e. a video *I-frame*);
- a binary image that is white only in correspondence with the facial landmarks of the compressed image.

The model produces a restored image from the compressed one.

We use a pre-trained VGG-19 [243] to extract multi-scale features from the degraded, reference and landmarks binary images. The reference (guidance) image is previously warped to the degraded one based on the facial landmarks using Moving Least Squares. We extract features at 4 different scales from the layers `relu_2_2`, `relu_3_4`, `relu_4_4` and `conv_5_4` of the VGG-19.

To align the warped reference and degraded features we adopt AdaIN [244]. This helps reduce the difference in style and illumination between the two images and thus improves the restoration. We denote by F^d and F^g the degraded and guidance features. The AdaIN can be written as

$$F^{g,a} = \sigma(F^d) \left(\frac{F^g - \mu(F^g)}{\sigma(F^g)} \right) + \mu(F^d) \quad (27)$$

where $\sigma(\cdot)$ and $\mu(\cdot)$ represent the mean and the standard deviation.

After going through multiple dilated residual blocks, the degraded features are progressively upsampled by enlarging the spatial resolution and reducing the number of channels. At the same time, they are combined with the reference features by means of Adaptive Spatial Feature Fusion and Spatial Feature Transform (SFT) [245] blocks.

The SFT block generates affine transformation parameters for spatial-wise feature modulation incorporating some prior condition. The scale α and the shift β parameters are learned from the features outputted by the corresponding ASFF block. The output of the SFT block is formulated as

$$SFT = \alpha \odot F^r + \beta \quad (28)$$

where \odot is the element-wise product and F^r are the restored features, that is the features originated from the degraded ones and restored in the decoding part of the architecture.

Following [246], we train the network to learn the residual image, so there is a skip connection between the degraded image and the restored output. This choice reduces the overall training time and improves its stability.

The fusion of the features of the reference and degraded images is a fundamental part of exemplar-based approaches, as it allows to fully exploit the information supplied by the guidance image. Adopting a concatenation-based approach, as in [247, 248], does not take full advantage of the reference features.

Thus, in our multi-scale architecture, we rely on multiple Adaptive Spatial Feature Fusion (ASFF) blocks [249]. While the reference image generally contains more high-quality details, the degraded image should have more weight in the reconstruction of the overall face components. For example, if the mouth of the reference image is closed while that of the compressed image is open, the reconstruction of the teeth should be mainly based on the restored features from the degraded image. For this reason, ASFF blocks generate an attention mask based on the degraded image facial landmarks to guide the fusion of the guidance and restored features.

For most guided face restoration methods, the performance is diminished by the pose and expression difference between reference and degraded images because it introduces artifacts in the reconstruction result. Thus, we spatially aligned the reference and compressed images with an image deformation method based on Moving Least Squares (MLS) [250].





Let p and q be respectively the sets of facial landmarks of the reference and degraded image, with $|p|=|q|=N$. In our case, $N=68$. We aim to find a deformation function f to apply to all the points of the reference image. Given a point v in the image, we solve for the best affine transformation $l_v(x)$ that minimizes

$$\sum_{i=1}^N w_i |l_v(p_i) - q_i|^2 \quad \text{where } w_i = \frac{1}{|p_i - v|^2} \quad (29)$$

Because the weights w_i are dependent on the point of evaluation v we obtain a different transformation $l_v(x)$ for each v . We define the deformation function f to be $f(v) = l_v(v)$.

Since $l_v(x)$ is an affine transformation we can rewrite it in terms of a linear transformation matrix M

$$l_v(x) = (x - p_*)M + q_* \quad (30)$$

where p_* and q_* are weighted centroids

$$p_* = \frac{\sum_{i=1}^N w_i p_i}{\sum_{i=1}^N w_i} \quad q_* = \frac{\sum_{i=1}^N w_i q_i}{\sum_{i=1}^N w_i}$$

Based on this insight, the least squares problem of eq. (29) can be rewritten as

$$\sum_{i=1}^N w_i |\hat{p}_i M - \hat{q}_i|^2 \quad (31)$$

where $\hat{p}_i = p_i - p_*$ and $\hat{q}_i = q_i - q_*$. The affine deformation that minimizes eq. (31) is

$$M = \left(\sum_{i=1}^N \hat{p}_i^T w_i \hat{p}_i \right)^{-1} \sum_{j=1}^N w_j \hat{p}_j^T \hat{q}_j$$

With this closed-form solution for M , we can write a simple expression for the deformation function f

$$f(v) = (v - p_*) \left(\sum_{i=1}^N \hat{p}_i^T w_i \hat{p}_i \right)^{-1} \sum_{j=1}^N w_j \hat{p}_j^T \hat{q}_j + q_* \quad (32)$$

Applying this deformation function to each point of the reference image lets to warp it according to the facial landmarks of the degraded image.

4.6.2.2. LANBIQUE Classic Full-Reference image quality evaluation methods rely on the similarity between an image which has been processed by some algorithm D and a reference undistorted image. Considering the use case of image enhancement of an image that was compressed, GANs are a good solution since they are great at filling in high frequency realistic details in image enhancement tasks; in this case the resulting enhanced image is compared to the reference. Unfortunately, when using classical MSE based Full-Reference metrics such as SSIM and PSNR GAN restored images yield lower performance, although they appear as "natural" and pleasant to human evaluators. For this reason, in [251, 3] semantic tasks are used to evaluate the quality of restored images. Measuring the performance of a semantic task such as detection on restored images gives us an understanding of the "correctness" of output images. Given some semantic task (e.g. object detection), a corresponding evaluation metric (e.g. mAP) and a dataset, the evaluation protocol consists in measuring the variation of such metric on different versions of the original image. Interestingly, this evaluation methodology gives hints on what details are better recovered by GANs.





In certain cases, detection is a task describing scene semantics in a very approximate fashion; usually detectors do not degrade for object classes that are clearly identifiable by their shape since even high distortions in the image are not able to hide such features. The gain in image quality provided by GANs, according to object detection based evaluation, resides in producing high quality textures for deformable objects (e.g. cats, dogs, etc).

In this paper we advocate the use of a language generation task for evaluating image enhancement. The idea is that captioning maps the semantics of images into a much finer and rich label space represented by short sentences. To be able to obtain a correct caption from an image many details must be identifiable.

We devise the following evaluation protocol for image enhancement using reference captions. We pick an image captioning algorithm \mathcal{A} . Image captioning is the task of generating a sequence of words, possibly grammatically and semantically correct, describing the image in detail. Given a set of reference captions S and the caption generated from an input image $\mathcal{A}(I)$, we want to measure their similarity with a language metric \mathcal{D} :

$$\text{LANBIQUE}(\mathcal{D}, \mathcal{A}; I, S) = \mathcal{D}(\mathcal{A}(I), S) \quad (33)$$

We look at the performance of a captioning algorithm \mathcal{A} on different versions of a dataset (e.g. COCO): compressed, original and restored. The pipeline of this evaluation approach is depicted in Figure 62.

In particular, we analyze results from two highly performing captioning methods [252, 4] which combine a bottom-up model of visual entities and their attributes in the scene with a language decoding pipeline. Both methods are trained over several steps incorporating semantic knowledge at different levels of granularity. In particular, the bottom-up region generator is based on Faster R-CNN [253] which is based on a feature extractor pre-trained on ImageNet [86] and then fine-tuned to predict object entities and their attributes using the Visual Genome dataset [254]. In [252], further knowledge is incorporated into the model by training the caption generation model using a first LSTM as a top-down visual attention model and a second level LSTM as a language model. Meshed memory transformers [4] share the exact same visual backbone as [252] but exploit a stack of memory-augmented visual encoding layers and a stack of decoding layers to generate caption tokens.

No matter how captioning models are optimized, our results show that the behavior of the captioning model for image quality assessment is consistent over several metrics as shown in Table 35.

Captioning is evaluated with several specialized metrics measuring the word-by-word overlap between a generated sentence and the ground truth [255], in certain cases including the ordering of words [256], considering n-grams and not just words [257, 258] and the semantic propositional content (SPICE [259]). These metrics evaluate the similarity with respect to a set of reference captions S , that is usually composed of five references.

Unfortunately, in most of the cases reference captions are not available as they often must be collected with great expense of effort and resources; in fact, standard datasets used for image quality evaluation do not include captions. However, it is possible to evaluate any kind of test image with our language based approach by modifying the pipeline. The idea is that the reference image is enough high quality to provide a valid caption for the evaluation of LANBIQUE. We caption the reference image I_{HQ} using the same captioner \mathcal{A} we use for the test image I , then we maintain the same procedure we previously described:

$$\text{LANBIQUE-NC}(\mathcal{D}, \mathcal{A}; I, I_{HQ}) = \mathcal{D}(\mathcal{A}(I), \mathcal{A}(I_{HQ})) \quad (34)$$

Since we change the evaluation pipeline with respect to the previous case, we argue that there may be a drawback with respect to the original version of the approach. As a matter of fact, modern captioners provide just one description per image and this means that the computation of \mathcal{D} metric is done just between two sentences. However, this does not affect the performance of our approach significantly, provided that the \mathcal{A} generates high quality captions.

In the following, we show how our approach can be extended to work in a No-Reference setting. In many occasions we may not have a high quality image available to be compared with the one to be





Table 34. Quantitative comparison between the proposed approach and other state-of-the-art methods for Constant Rate Factor (CRF) 42 on DFD dataset. Best and second best results are in bold and underlined, respectively. \uparrow = higher values are better, \downarrow = lower values are better.

Method	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	BRISQUE \downarrow	CONTRIQUE \downarrow	CONTRIQUE-FR \downarrow	VMAF \uparrow	VMAF-NEG \uparrow
GWAINet	22.25	0.608	0.129	24.18	50.16	20.79	44.65	36.60
HiFaceGAN	<u>29.38</u>	0.828	0.075	28.41	48.75	18.67	47.77	45.11
PSFR-GAN	29.68	<u>0.833</u>	<u>0.057</u>	29.07	46.87	16.46	48.55	46.22
GFP-GAN	27.51	0.822	0.081	34.17	50.84	23.01	57.55	48.51
GPEN	27.61	0.813	0.075	28.67	49.42	21.36	55.86	<u>49.26</u>
DFDNet	27.03	0.827	0.065	32.38	46.84	<u>16.04</u>	55.15	48.95
ASFFNet	28.29	0.834	0.062	29.67	<u>46.27</u>	17.48	51.74	46.84
Ours	26.19	0.779	0.037	<u>27.41</u>	44.95	13.16	<u>56.87</u>	54.20

tested. For this reason, we modify our language based pipeline by adding an additional blind restoration module \mathcal{R} . We assume that the images to be tested are corrupted by one or a combination of unknown distortions that are responsible of a global reduction of the visual quality. In this extended model, our aim is to restore corrupted input image I in order to use the enhanced version as the reference image. After this operation is completed, we are able to feed both the corrupted image and the restored one to the same captioning module, hence we generate their text descriptions and finally we calculate the ultimate score based on some language metric \mathcal{D} :

$$\text{LANBIQUE-NR}(\mathcal{D}, \mathcal{A}, \mathcal{R}; I) = \mathcal{D}(\mathcal{A}(I), \mathcal{A}(\mathcal{R}(I))) \quad (35)$$

Typically, image distortions are not known a priori so it may be a difficult task to train many networks capable of handling all the possible combinations of corruption processes and then select the best one for a specific restoration. For this reason, we choose to train a single network following a degradation model, so that it can restore most types of distorted images and recover their original quality as best as possible. In order to ensure a good output quality, we employed Real-ESRGAN [260] as the restoration module. We have modified the original model by adding JPEG2000 in the training procedure, then we have fine-tuned a pre-trained version of such network with the new introduced distortion.

In most of the cases, recovered images represent a solid reference for our evaluation model, as they are very close to real images from the point of view of human perception. In this setup, our LANBIQUE-NR assigns high scores to slightly distorted images, as their reconstruction is likely very perceptually close, and the captions generated are pretty close. On the other hand, highly distorted images are transformed into better quality data that differ significantly from input. In this case, the captions between the two versions may differ much more, thus leading to lower scores of language metrics.

4.6.3. Experimental Results

4.6.3.1. Keyframe GAN

4.6.3.2. LANBIQUE In Table 35 we report results of LANBIQUE using various captioning metrics \mathcal{D} . Interestingly, all metrics show that captions over reconstructed images (REC rows) are better with respect to caption computed over compressed images (JPEG rows). This shows that image details that are compromised by the strong compression induce errors in the captioning algorithm. On the other hand, the GAN approach is able to recover an image which is not only pleasant to the human eye but recovers details which are also relevant to a semantic algorithm.

In Table 36 we report results on COCO for Full-Reference and No-Reference indexes. In this setup, we compress the original images at different QFs and then we restore them with a QF specific artifact





Table 35. Evaluation of image restoration over compression artifacts with GAN using LANBIQUE with different captioning metrics (best results highlighted in bold). For each metric we denote higher(\uparrow) or lower(\downarrow) is better. JPEG q indicates a JPEG compressed image with $QF=q$ (e.g. 10), while (REC q) indicates the corresponding reconstruction using [3]. Captions created from reconstructed images obtain a better score for every metric.

QUALITY	BLEU_1 \uparrow	METEOR \uparrow	ROUGE \uparrow	CIDEr \uparrow	SPICE \uparrow
JPEG 10	0.589	0.173	0.427	0.496	0.103
REC 10	0.730	0.253	0.527	1.032	0.189
JPEG 20	0.709	0.241	0.513	0.937	0.174
REC 20	0.751	0.266	0.543	1.105	0.201
JPEG 30	0.740	0.258	0.535	1.054	0.194
REC 30	0.757	0.269	0.549	1.133	0.205
JPEG 40	0.748	0.263	0.542	1.087	0.200
REC 40	0.758	0.270	0.549	1.132	0.206
JPEG 60	0.755	0.267	0.546	1.117	0.204
REC 60	0.760	0.270	0.550	1.137	0.207
ORIGINAL	0.766	0.274	0.556	1.166	0.211

Table 36. Evaluation using No-Reference and Full-Reference metrics on MS-COCO. For each metric we denote higher(\uparrow) or lower(\downarrow) is better. JPEG q indicates a JPEG compressed image with $QF=q$ (e.g. 10), while (REC q) indicates the corresponding reconstruction using [3]. NIQE and BRISQUE rate better GAN images than the ORIGINAL. SSIM always rate restored images worse than compressed. PSNR shows negligible improvement. [4] and CIDEr have been used by LANBIQUE-NC respectively as language model and language metric.

QUALITY	NIQE \downarrow	BRISQUE \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	LANBIQUE-NC \uparrow
JPEG 10	6.689	52.67	25.45	0.721	0.305	0.542
REC 10	3.488	17.93	25.70	0.718	0.144	1.118
JPEG 20	5.183	43.99	27.46	0.796	0.187	0.956
REC 20	3.884	17.85	27.60	0.784	0.085	1.289
JPEG 30	4.474	37.72	28.61	0.831	0.134	1.165
REC 30	3.601	18.32	28.81	0.819	0.060	1.370
JPEG 40	4.011	33.61	29.41	0.852	0.105	1.260
REC 40	3.680	18.68	29.44	0.836	0.048	1.424
JPEG 60	3.588	28.15	30.71	0.880	0.067	1.366
REC 60	3.885	19.45	30.61	0.862	0.032	1.482
ORIGINAL	3.656	21.79	-	-	-	-

removal GAN. We use the uncompressed image generated caption as ground truth. The results show that, for restored images, PSNR accounts for a slight improvement while SSIM indexes lower than the compressed counterparts. This is an expected outcome, as in [3] it is shown that state of the art results on PSNR can be obtained only when MSE is optimized and on SSIM if the metric is optimized directly. Nonetheless, GAN enhanced images are more pleasant to the human eye, therefore we should not rely just on PSNR and SSIM for GAN restored images. LANBIQUE, using [4], is in line with LPIPS [261].

4.6.4. Conclusion

4.6.4.1. Keyframe GAN We have proposed a novel GAN-based method and a keyframe selection system that improves the visual quality of videoconference videos enhancing the appearance of faces. A key element of the system is the policy that updates a set of previous I-frames and exploits them to improve the visual quality improvement process. The proposed approach improves over competing state-of-the-art methods in terms of perceptual metrics and is rated much better in terms of fidelity by human evaluators.





4.6.4.2. LANBIQUE We have proposed LANBIQUE, a new approach to evaluate image quality using language models. Existing metrics based on the comparison of the restored image with an undistorted version may give counter-intuitive results. On the other hand, the use of naturalness based scores may in certain cases rank restored images higher than original ones.

We show that instead of using signal based metrics, semantic computer vision tasks can be used to evaluate results of image enhancement methods. Our claim is that a fine grained semantic computer vision task can be a great proxy for human level image judgement. Indeed we find out that employing algorithms mapping input images to a finer output label space, such as captioning, leads to more discriminative metrics.

4.6.5. Relevance to AI4Media use cases and media industry applications

These methods are useful for 3C2-9 - Management of contribution under bandwidth constraints.

Keyframe GAN can be useful for 3B2-1 (Video super resolution). LANBIQUE can be used to assess the quality of restored content and to score existing content quality. Both methods provide a high degree of automation for several multimedia production process in m

4.6.6. Relevant Publications

- Agnolucci, Lorenzo, Leonardo Galteri, Marco Bertini, and Alberto Del Bimbo. "Perceptual quality improvement in videoconferencing using keyframes-based gan." *IEEE Transactions on Multimedia* (2023)
- Galteri, Leonardo, Lorenzo Seidenari, Pietro Bongini, Marco Bertini, and Alberto Del Bimbo. "Lanbique: Language-based blind image quality evaluation." *ACM Transactions on Multimedia* (2022)

4.6.7. Relevant software/datasets/other outcomes

Source code: <https://github.com/LorenzoAgnolucci/Keyframes-GAN>

4.7. Expressive Piano Performance Rendering from symbolic data

Contributing partner: IRCAM

4.7.1. Introduction

The research presented in this section and in section 4.8 aims to develop innovative algorithms to generate synthetic yet realistic musical sound mixes starting from musical scores present in a digital format. First, this approach can be employed to generate music content in media or video games, and it can also have artistic applications for music composers, by rendering previews of their compositions before hiring musicians. Second, on top of the aforementioned artistic purposes, this approach is interesting for producing large datasets of realistic musical mixes from symbolic annotations [262]. Such automatically generated datasets of realistic mixes will be used to further train models for various Music Information Retrieval (MIR) tasks, such as automatic transcription, instrument identification, tempo and down-beat estimation, or key and mode recognition.

To this end, we propose in the current section a neural model for rendering expressive performances of inputted piano music compositions. As a summary: the network applies changes in the input digital music scores, in the symbolic domain, in order to get expressive performances as humans would play, still in the symbolic domain. These changes are about: time, articulations and velocity of the notes. In section 4.8, a differentiable piano synthesizer is presented, which generates realistic piano sounds from symbolic performances.





Performance rendering: Performance rendering is the task of imbuing a music score with expressive features as if a musician performed the score in a way to bring out emotional qualities. To get an expressive rendition of the music, performers have the liberty to shape sound parameters that are not explicitly described by the written score [263]: for piano pieces, musicians make an interpretation of the score by mainly reshaping the timing, articulation and nuance of the notes.

Previous works for the task used data-driven methods to predict performance features that enhance the score note indications [264, 265, 263, 266]. More recently, Variational Auto-Encoders (VAE) conditioned on score features have proven to be successful at modeling the diversity in performance expressivity, as several renditions of the same piece are conceivable [267, 268, 269, 270]. The performance features are defined as the difference in timing, articulation, and velocity of the played notes compared to the exact rendition of the score [271]. However, obtaining such features requires the collection of MIDI (Musical Instrument Digital Interface) performances with their associated digital scores and to align them at note-level [272, 273]. These required matching and alignment steps limit the amount of data available for training [274] and the application of the models to piano music, where performance MIDI data can be collected more easily [275]. Also, most of these works are highly-informed as they take different markings in the digital scores into account for guiding the expressive rendering, such as rests, beat information, hand part, position in the measure, key and time signatures, articulation and ornament markings, slurs or beams. This reliance on markings specific to the sheet music format hinders the usage of these models in modern music production frameworks (Digital Audio Workstation, sequencers) where MIDI data are directly manipulated without using such markings.

Concurrently, GANs have been successfully applied for various tasks transferring data from one domain to another without aligned pairs, such as image-to-image translation [276], audio timbre matching [277] or music genre transfer [278]. In the light of such results, this work attempts to address expressive performance rendering as a domain transfer task, by transforming MIDI scores into human-like performances without supervision on the performance features and reliance on score markings. To this end, an adversarial approach is employed to map the outputs of a low-informed performance rendering model to the distribution of human performances, without providing matching pairs of scores and performances. Trained on publicly available datasets, the proposed method and its experiments are presented here, including an early subjective evaluation.

The experiments show promising results for the method as it can infer expressive qualities into scores, although not with the same amount of naturalness as in performances rendered by real pianists and by a highly-informed supervised baseline.

4.7.2. Methodology

The proposed approach, illustrated in Figure 26, is composed of a performance rendering model G that takes a score X as input and produces an expressive interpretation \tilde{X} . The rendered performances are fed into a discriminator D , among performances Y from a dataset of recorded human performances. The performance rendering model and the discriminator have opposed objectives, as the discriminator D aims to differentiate the real performances from the ones rendered by the model G , while the latter tries to produce performances indistinguishable from the real ones.

Data formatting: Both the scores X and real performances Y are encoded as sequences of N notes with the minimal amount of features needed for describing them:

$$\mathbf{X} = \{x_n\}_{n \leq N} = \{p_n, o_n, d_n, v_n\}_{n \leq N}. \quad (36)$$

The notes are ordered by their absolute onset time: for the n -th note, p_n is its normalized MIDI pitch, o_n its delta-time with the previous note onset, or relative Inter-Onset-Interval (IOI), capped at 4 seconds, d_n its duration in absolute time and v_n its normalized MIDI velocity.



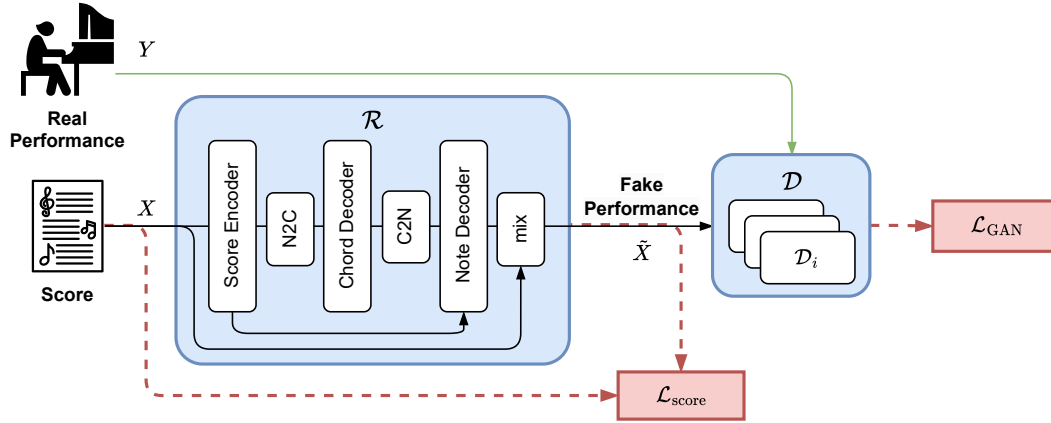


Figure 26. Training pipeline for the symbolic performance rendering model \mathcal{R} : its final mix function modifies the score X with modifying features output by the Note Decoder, in order to deceive the discriminators \mathcal{D}_i . During training, the unaligned score X and performance Y are drawn at random from their respective sets.

Rendering model: The performance rendering model G predicts modifying features $\Delta X = G(X)$ from the score note features in order to modify them into performance-like note features \tilde{X} through the mix function:

$$\begin{aligned} \tilde{X} &= \text{mix}(X, G(X)) \\ &= \{p_n, o_n + \delta o_n, d_n \times \delta d_n, v_n \times \delta v_n\}_{n \leq N}, \end{aligned} \quad (37)$$

with δo_n the micro-onset timing, δd_n the articulation and δv_n the expressive velocity of the n -th score note.

These modifying features are obtained by first processing the note-wise score features with a convolutional Score Encoder. Then, the same hierarchical modeling from [267] is applied: the note-wise features are merged into chord-wise features, which enables a more coherent modeling of the full sequence. This note-to-chord operation, or $N2C$, is performed by average pooling the features of simultaneous notes into a common chord-wise feature. The inverse operation $C2N$ can later convert chord-wise features into note-wise features by duplicating the chord feature for each of its notes. On the contrary of hierarchical strategies employed in other works [269, 270], the note-to-chord alignment matrix required for $N2C$ and $C2N$ can be directly extracted from our low-informed MIDI data representation, using the sequence of relative IOI $\{o_n\}_{n \leq N}$. Further implementation details on the $N2C$ and $C2N$ operations can be found in [267].

Before returning to the note-granularity, the chord-wise features are further processed by a Chord Decoder, which is a CRNN with a bidirectional GRU layer. Finally, fine-grained adjustments at note-level are made with the Note Decoder and a skip connection from the note-wise score encoding. The final micro-onset timing δo_n is obtained through a linear activation function, while the articulation δd_n and the expressive velocity δv_n are mapped to $[0.25, 4]$ with a scaled sigmoid function.

Discriminator: Taking inspiration from speech processing using discriminators with a multi-scale architecture [279], we use $k = 3$ discriminators D_k with identical architectures, mirrored from the performance rendering model, with the exceptions of the $N2C$ and $C2N$ operations, as chords in real performances are not as easily defined as in scores. Each discriminator is fed with a downsampled sequence of (real or rendered) performance notes by average pooling with sizes $\{1, 3, 9\}$. Discriminators with longer pool sizes look at features at higher levels in the performances and thus, can help transferring such knowledge and long-term coherence to the performance rendering model G . To stabilize the GAN training, Gaussian noise is added to the inputs of the discriminators, as in [278].



Loss functions: The least-square variant of the GAN objective (LSGAN) is used to train the discriminators and the performance rendering model. Their respective loss functions L_{D_k} and $L_{G, gan}$ are defined as:

$$L_{D_k} = \mathbb{E}_{Y \sim p_{perf}} [\|D_k(Y) - 1\|_2] + \mathbb{E}_{X \sim p_{score}} [\|D_k(G(X))\|_2],$$

$$L_{G, gan} = \mathbb{E}_{X \sim p_{score}} \left[\sum_{k=1,2,3} \|D_k(G(X)) - 1\|_2 \right]. \quad (38)$$

We have observed that the instability of the vanilla adversarial training may lead the performance rendering model to displace the notes in extreme values, causing the original piece to be unrecognizable. To ensure that the performances remain fairly close to their scores, an additional regularization term L_{score} is added:

$$L_{score}(X) = \lambda_{score} \left\| \frac{G(X) - X}{X} \right\|_2, \quad (39)$$

with λ_{score} a fixed vector weighting how much each performance component (timing, articulation, velocity) can deviate from the score indication. Here, $\lambda_{score} = \{1, 1, 0.1\}$.

The total loss for the performance rendering model G is:

$$L_G(X) = \lambda_{gan} L_{G, gan}(X) + L_{score}(X), \quad (40)$$

with λ_{gan} the balance between the GAN objective and the score regularization loss. This balance is decisive for the final behavior of G since the two loss components have opposite influences on its training: L_{score} refrains G from modifying the scores while $L_{G, gan}$ encourages exploring different interpretations in order to deceive the discriminator. In our experiments, $\lambda_{gan} = 2$.

4.7.3. Experimental Results

Datasets: The proposed approach was trained and evaluated using the scores from the ASAP dataset [274] and all performances from the MAESTRO dataset (v3.0.0) [275], which are both publicly available. The human performances from MAESTRO were recorded in MIDI format using Yamaha Disklaviers. The ASAP dataset has notably matched a set of these performances with their original scores at note-level, and has thus been used to some extent in previous performance rendering works [267]. However, since the proposed method does not require aligned scores and performance, the entirety of both datasets can be used, which amounts for 962 training performances, 137 validation performances, 107 training scores, 15 validation scores and 35 test scores (following the train-validation-test split of [280]).

The velocity indications were kept from the ASAP scores in MIDI format, which can either be constant throughout the piece or mapped from the score nuances and markings using simple rules. The scores and performances are split into segments of 128 consecutive notes, with random pitch shifting during training by ± 7 semi-tones, as in [268]. Validation data is used to monitor and avoid potential over-fitting of the performance rendering model by reproducing the training performances from their corresponding scores.

Subjective evaluation: A short listening test has been conducted to evaluate the interpretation quality of the performances rendered by the model. 7 scores from the ASAP test subset were selected, covering 5 different composers. 4 MIDI performances were generated by different methods for each score:

- **Human** is a corresponding human performance from the ASAP dataset.
- **Deadpan** is the direct export of the MIDI score.
- a rendition by our **Proposed** approach.



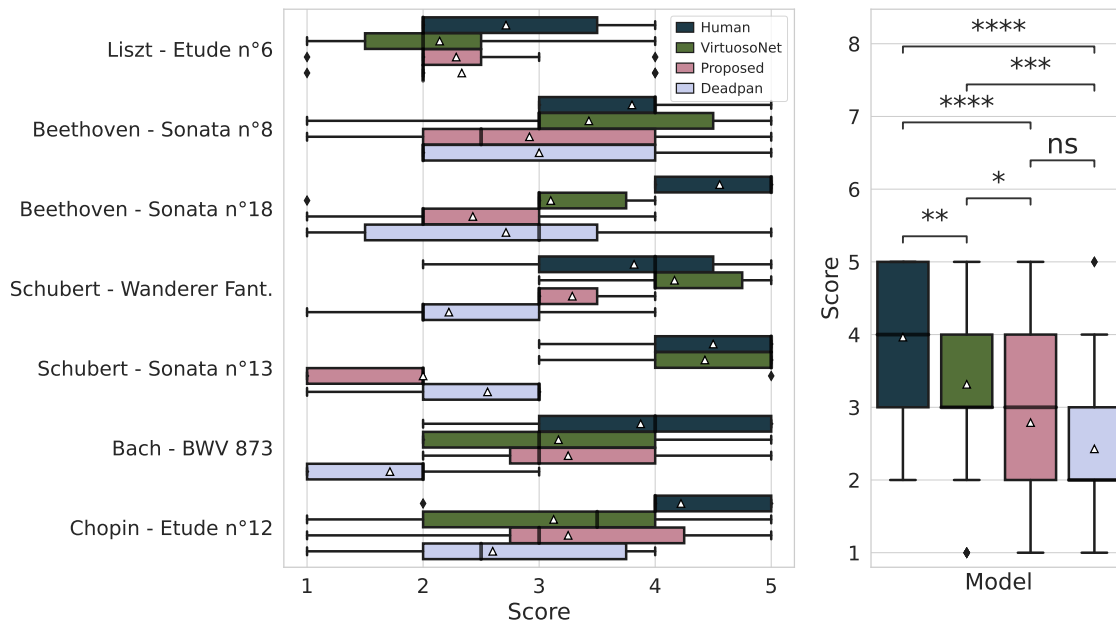


Figure 27. Box-plot of the MOS (Opinion Mean Score) of the different performance rendering methods: piece-wise at the left and overall at the right, with Holm-Bonferroni corrected two-sided Mann-Whitney U tests. The thickened bars indicate the median values while the white triangles indicate the mean values. p-value annotation legend: ns for $p > 0.05$; * for $p \leq 0.05$; ** for $p \leq 0.01$; *** for $p \leq 10^{-3}$ and **** for $p \leq 10^{-4}$.

- a rendition from the graph-based variant of **VirtuosoNet** [270], a highly-informed model using score markings in MusicXML format and is trained with a private dataset of 226 scores matched and aligned with MAESTRO performances, which is larger than ASAP.

The first 20s of each performance were synthesized using the Arturia Piano V3 software⁹, a physical-based piano synthesizer. 19 professional audio and piano players were asked to rate the naturalness of the presented performances, using a 5-point Likert scale (from 1 - Bad, to 5 - Excellent). Each trial randomly presented 3 different performances from each method. Results are reported in Figure 27.

The Holm-Bonferroni corrected two-sided Mann-Whitney U tests indicate a statistical difference at $\alpha = 0.05$ between the Human rendition and each of the other methods, and between VirtuosoNet and Deadpan. The overall results show that the proposed approach does enhance the scores with expressive features in comparison to the raw rendition of the piece, but still not with the same amount of naturalness as actual pianists and the highly-informed VirtuosoNet. This was to be expected as our proposed unsupervised training task without score markings is harder than the training objectives of VirtuosoNet, for about the same quantity of training data.

By examining the ratings piece-wise, one can notice the poorer renditions of the proposed method for slower tracks (Schubert’s 13th Sonata and Beethoven’s 18th Sonata). This may suggest that the model lacks an understanding of the global musical content of the scores and applies similar modifying features for every track, which renders inappropriate performances for slower-paced compositions. However, we have observed during preliminary experiments that some other configurations of the model (with different loss weightings for instance) do not exhibit such an issue, but they render less realistic performances overall than the presented configuration. Such sensibility to training hyper-parameters is typical of GAN and we hope strengthening the score understanding of the model would reduce this instability.

⁹<https://www.arturia.com/products/software-instruments/piano-v/overview>



4.7.4. Conclusion

In this section, we presented a neural model for rendering expressive performances of inputted piano music compositions. The training is based on two MIDI file datasets for: raw scores (inexpressive) and interpretations (expressive). Contrarily to other works about the same task, we use here a GAN approach, which makes it possible to train the model without aligned pairs. In section 4.8, a differentiable piano synthesizer is presented, which generates realistic piano sounds from symbolic performances.

4.7.5. Relevance to AI4Media use cases and media industry applications

- UC5-B (AI for Games: Music for games):
This method matches with the music sub-use case of UC5, for the music production of video games. The showcase demonstrator of UC5, which gathered the demonstrators of the two sub-use cases, also used the synthesized Piano sound from the DDSP-Piano module. Because of the lack of time, we had not the opportunity to improve the expression rendition of the chosen music MIDI files using the *Expressive Performance Rendering*, but this integration is feasible, and quite easy.
- UC6 (AI for Human Co-Creation):
This method also matches with UC6 which deals with music co-creation. For the integration of the DDSP-Piano synthesizer, the UC6 demonstrator accepts MIDI files as inputs, and so it is possible to insert this new module in the framework. For the same reason, this integration had not been realised.

4.7.6. Relevant Publications

- L. Renault, R. Mignot, and A. Roebel. "Expressive Piano Performance Rendering from Unpaired Data." International Conference on Digital Audio Effects (DAFx23), Copenhagen, Denmark, Sept. 2023. <https://doi.org/10.5281/zenodo.8386761>.

4.7.7. Relevant software/datasets/other outcomes

Demonstration page: http://renault.gitlab-pages.ircam.fr/thesis-support/chap_5-2

4.8. Differentiable Piano Synthesizer

Contributing partner: IRCAM

4.8.1. Introduction

From expressive piano performances, stored in MIDI format, see section 4.7, the production of realistic music recordings of piano needs a sound synthesizer. In deliverable D5.2 (section 3.8), the first version of the Differentiable Piano synthesizer (DDSP-Piano-v1) was presented. In this section we present first a summary of the previous model, then the last refinements of the architecture is detailed, and finally we present a subjective evaluation of the produced sounds.

DDSP-Piano-v1: Our proposed approach tackles the task of piano sound synthesis from a symbolic representation, by enhancing and adapting the DDSP framework (see [281]) to handle polyphonic MIDI input and to reproduce particular properties of the piano sound, such as partials inharmonicity, partials beating, and noise of the hammers, the keys and the pedals.

The full synthesis architecture is illustrated in Figure 28. It takes as input all the parameters that a pianist has over the instrument, being the played notes (pitches and velocities), the pedal actions, and the piano and room context. The synthesis is controlled by a neural network, and the audio signal is



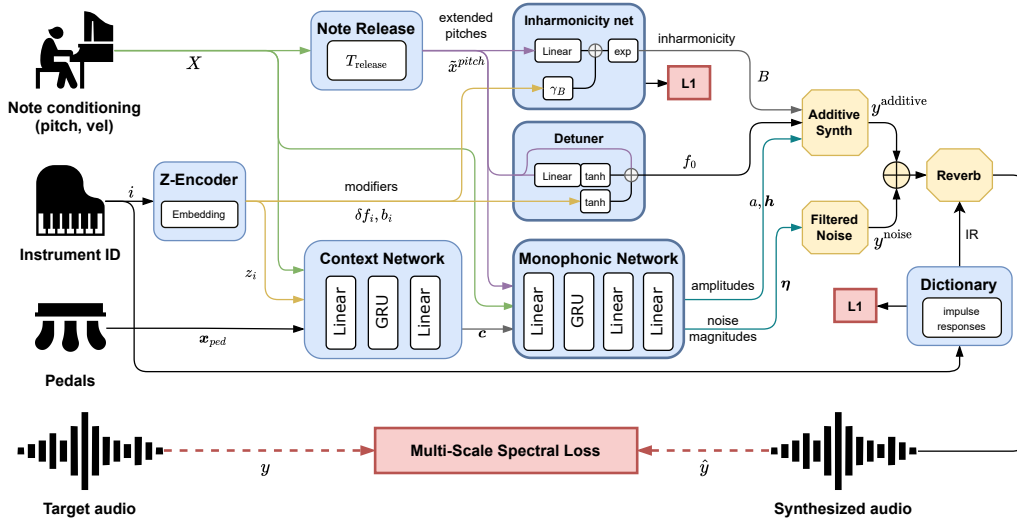


Figure 28. Synthesizer Architecture. The blue boxes represent the trained modules for the control of the synthesis. The synthesis modules from DDSF are represented by yellow boxes (Additive, Filtered Noise, and Reverberation). Finally, the Multi-Resolution Spectral Loss (MSS) compares the input target signal (bottom left) and the output synthesized sound (bottom right).

computed by summing the outputs of a multiple monophonic additive synthesizer (for sine parts) and a subtractive differentiable synthesizer (for stochastic components). Finally, the room reverberation is produced by a learned impulse response. The role of each trainable sub-module of our architecture is:

- **Z-Encoder:** encodes specific information related to the piano model and environment,
- **Note Release:** extends the note duration to take into account of the natural note damping after the key release,
- **Inharmonicity Network:** explicitly sets the inharmonic distribution of partials,
- **Detuner:** encodes data to reproduce natural partial beatings,
- **Context Network:** prepares context conditioning in order to take into account of the interaction between notes (sympathetic resonances e.g.),
- **Monophonic Network:** computes the synthesizer controls for individual notes,
- **Reverberation Dictionary:** stores learned impulse responses.

The differentiable synthesizer layers (yellow boxes in Figure 28) convert the controls into audio signals, in the spectral modeling paradigm [282]: the *additive synthesizer* generates the quasi-harmonic components of a piano note by summing multiple sinusoidal signals, and the *subtractive synthesizer* generates the noisy elements (hammer, key and pedal noises) by filtering white noise with filters computed from the noise magnitudes as in [281]. Finally, the reverberation sub-module convolves the summed signals with the learned impulse response. The final audio output is compared to the ground-truth audio with a Multi-Resolution Spectral loss (MSS), as in [283, 281].

4.8.2. Methodology

Section 4.8.1 summarized the first proposition of a DDSF-based piano audio synthesizer from MIDI. It combines expressive neural network layers with explicit modules that embed modeling knowledge of the instrument: this modular approach allows for tackling specificities of the piano sound in a targeted manner. However, while the overall synthesis quality appears to be quite decent and surpasses a pure neural benchmark, some individual modules did not converge as expected. This section will go over a few proposed modifications to the model, addressing some of these concerns, along with early evaluation results.

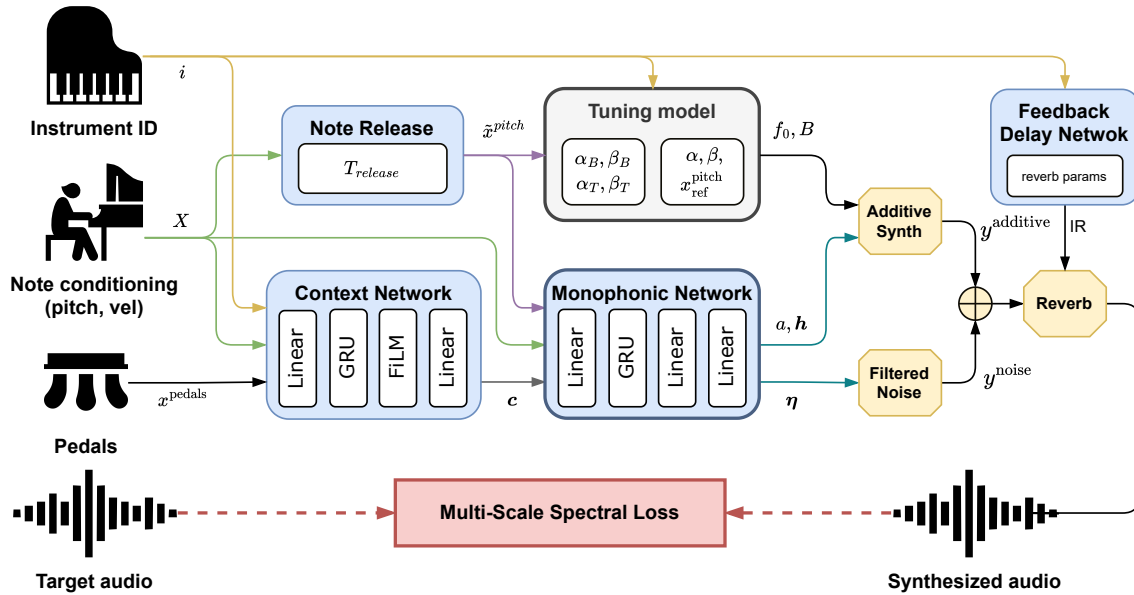


Figure 29. Full architecture of DDSP-Piano v2. The blue rounded boxes represent the trained modules for the control of the synthesis. The weights of the grey **Tuning Model** are optimized beforehand and are frozen during neural optimization. Modules with a thickened border are applied along each monophonic voice. Differentiable signal processing and synthesis layers are represented by yellow hexagons (Additive, Filtered Noise, and Reverberation). Finally, the MSS loss compares the input target signal (bottom left) and the output synthesized sound (bottom right).

The full architecture of the updated DDSP-Piano is illustrated in Figure 29, and the main changes are summarized here.

Tuning Model: The most apparent issue with the first iteration of DDSP-Piano is its inability to fine-tune the frequencies tuning parameters to the target pianos. The detuner is replaced by the **parametric tuning model** of [284] that takes the explicit **inharmonic** model into account for modeling the tuning deviations from the equal temperament (known as the *Railsback curve*). Added parameters are the per-piano reference notes $\{x_{ref,i}^{pitch}\}_{i \leq I}$, bass asymptotes $\{\beta_i\}_{i \leq I}$, and decrease slopes $\{\alpha_i\}_{i \leq I}$ of the parametric octave type model. Similarly for the **inharmonic** model, the instrument-specific modifiers $\{\delta_i, b_i\}_{i \leq I}$ are removed in favor of instrument-specific bass and treble linear asymptotes $\{\alpha_{B,i}\}_{i \leq I}$, $\{\beta_{B,i}\}_{i \leq I}$, $\{\alpha_{T,i}\}_{i \leq I}$ and $\{\beta_{T,i}\}_{i \leq I}$.

Back to signal-based analysis: From the attempts at strengthening the neural estimation of the frequency-related layers, we have concluded that the safest approach would be to first estimate the frequencies of individual piano notes, then fit the tuning and inharmonicity models on those estimations, rather than relying on matching through the audio modality. Thanks to the aligned MIDI data, we first extract all MAESTRO audio segments where only a single note is being played. Then, the method of [285] is used for the joint estimation of the notes f_0 and inharmonicity coefficient. The method has proven to be more efficient for such estimations than other contemporary approaches.

Sub-module refinements: Since the instrument-specific modifiers are removed, the **Z-encoder** is simply integrated into the **Context Network**. Its instrument embedding output is applied through a FiLM (Feature-wise Linear Modulation) layer [286], which has notably found usages for global conditioning [287].



Reverberation modeling: Another shortcoming of the first DDSP-Piano is its reverb module that has learned unrealistic features for usual room reverberations. Therefore, the explicitly learned FIR (Finite Impulse Response) are replaced by a **differentiable FDN-based reverb** (Feedback-Delay Network) module, with implementation and default parameters taken from [288]. In the same manner as spectral modeling for instrument sound synthesis, the FDN structure is motivated by modeling knowledge of natural reverberations, achieving realistic reverb FIRs with fewer parameters. One can refer to the works of [288, 289] for an in-depth explanation of the layer: notably, the early reflections are still modeled by a short FIR filter while the late reverberation is modeled by the FDN structure. The structural constraints inherent to the module should prevent it from learning unrealistic features and help to achieve better behavior disentanglement between the DDSP components.

Revised Training Procedure: Compared to the initial training strategy, we no longer alternate between two phases. Instead, the estimation of frequency-related parameters (from the parametric *tuning model*) is supposed to be completely done in a first stage, then the neural layers parameters are optimized afterward in a second stage. As for the neural optimization phase, since the reverb module was changed, the loss function is simply reduced to the MSS loss between the target and synthesized signals. Other optimization parameters remain unchanged (Adam learning rate, frame rate, segment duration, validation-based early stopping), with the exception of the increased output length due to the audio sampling rate upgrade.

4.8.3. Experimental results

This section presents a perceptual evaluation of the first version of DDSP-Piano, summarized in section 4.8.1. This evaluation was not presented in D5.2.

An objective evaluation of the refined DDSP-Piano-v2 architecture was performed and revealed improvements over the initial version. However, it will require a proper perceptual evaluation like the one performed for DDSP-Piano-v1, and we do not present this objective evaluation of DDSP-Piano-v2 here.

Baselines: The proposed DDSP-Piano-v1 model is evaluated against other piano sound synthesis methods. All samples synthesized with the following systems are also downsampled to 16kHz and converted to mono. The commercial software **Pianoteq 7**¹⁰ with the default preset **NY Steinway D Classical** is used as the physical-modeling-based baseline. Results from the physical modeling of the instrument are synthesized in real-time using modal synthesis [290]. For the wavetable synthesizer benchmark, performances are obtained by stitching isolated note recordings from the **YDP Grand Piano**¹¹ soundfont, using the open-source software **Fluidsynth**¹². Finally, **Piano-TTS v1**, the TTS-inspired model from [283] is elected as the pure neural audio synthesis benchmark. Also trained on MAESTRO dataset, this model is a modified Tacotron-2 acoustic model followed by a simplified Neural Source-Filter (NSF) vocoder model. MIDI-filter-bank-based spectra are used as the intermediate representation between the two sub-models, which has the advantage of being aligned with the input piano rolls in the frequency/pitch axis.

Default and Ablated models: The **Default** configuration of DDSP-Piano is also compared to 4 different variants of the proposed method:

- **Deep Inharmonicity:** this variant replaces the explicit inharmonicity model with a DNN. Hence, instead of using the inharmonicity equation known from a physics study, the DNN has the goal to learn it from the training data, without structural bias.

¹⁰<https://www.modartt.com/pianoteq>

¹¹<https://freepats.zenvoid.org/Piano/acoustic-grand-piano.html>

¹²<https://www.fluidsynth.org/>



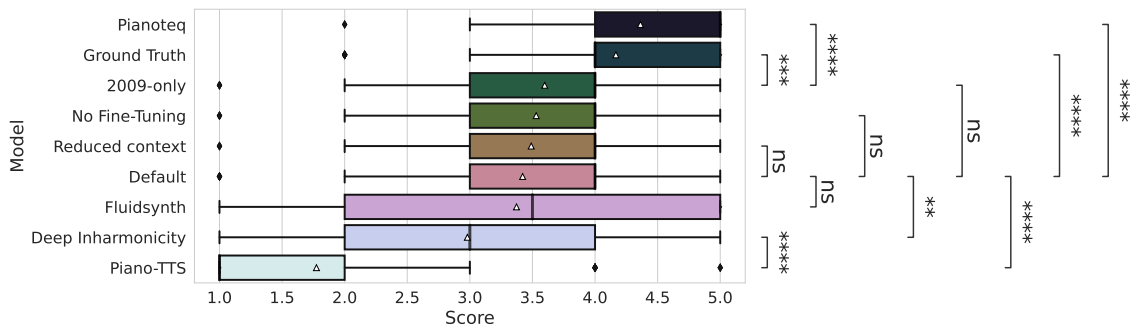


Figure 30. Box plots of MOS for each system. The thickened bars indicate the median values while the white triangles indicate the mean values. Two-sided Mann-Whitney U tests with Holm-Bonferroni correction were conducted on relevant systems pairs at $\alpha=0.05$. p -value annotation legend: *ns* for $p > 0.05$; * for $p \leq 0.05$; ** for $p \leq 0.01$; *** for $p \leq 10^{-3}$ and **** for $p \leq 10^{-4}$.

- **Reduced Context:** in this variant, the conditioning input X is removed from the context control computation. Hence, the network has not the ability to learn interactions between notes.
- **No fine-tuning:** in default configuration, two different training phases are done alternatively: the first used fixed values for the inharmonicity and the detuning, and the second refines these values. In this variant, only the first phase is computed.
- **2009-only:** the default configuration uses a joint training for the different pianos and environments of the MAESTRO dataset (10 configurations for 199 hours). This variant is trained only with recordings made in 2009 (20 hours of training data).

Evaluation Results: A listening test was conducted for gathering MOS (Mean Opinion Score) on all systems under evaluation. Eleven performances were hand-picked from the test data, covering all recording environments and with a diversity of composers, registers, and note densities. The first 9 seconds of the performances were synthesized with all systems. Listeners were asked to rate their overall quality on a 5-point Likert scale, from “very annoying” to “real recording”. In each trial, 2 samples from each of the 8 systems and 2 real recordings were randomly presented to the listener for rating. We gathered 52 participants that are musicians or audio professionals: 14 among them have notions of piano playing and 29 have been playing the instrument for several years. Box-plot and mean values of the MOS ratings are reported in Figure 30, with statistical tests following the evaluation procedure of [283].

Comparing the ratings of the model against its ablations:

- The quality difference between the **Deep Inharmonicity** variant and the models including the explicit inharmonicity model is confirmed perceptually. Only the **Default**-against-**Deep Inharmonicity** hypothesis is not statistically significant, but the median and quartile values still suggest a slight advantage in favor of the Default configuration.
- Ratings also confirm that the second training phase does not improve the perceived quality, suggesting that the natural beating between simultaneous notes in harmony may be sufficient for achieving realistic-sounding partial beatings during polyphonic performances.
- Reducing the context also does not significantly hinder the perceived quality of the DDSP-Piano model. It can be deduced that other components of the approach can be improved before the lack of note interaction limits the perceived quality.
- Single piano modeling is still perceived as good sounding as variants trained on several pianos simultaneously, which raises the question of the minimum amount of training data required for achieving such quality. Note that previous neural-based synthesis works did not report the model quality when trained on a single environment of MAESTRO.



As for comparisons with the other piano synthesis methods:

- All variants of DDSP-Piano have a significant difference over the neural-based Piano-TTS benchmark. Although this baseline is more versatile since it was developed for speech synthesis at first, our approach is better suited for piano sound synthesis, achieving better sound quality with significantly fewer training parameters.
- Only the physical-modeling-based method achieves sound quality comparable to the real recordings (even slightly better, although not significantly, as also found by [283]). Various unwanted noises and the recording quality of the real samples may have been perceived as slightly annoying compared to the clean sounds synthesized by the *Pianoteq* software. The quality of the training data represents the upper bound limit of neural-based synthesizers, thus our model can benefit from cleaner audio recordings.
- Nonetheless, there is still a significant gap in the perceived quality between the synthesis offered by the DDSP-Piano model and the real recordings.
- As it stands, all variants of our approach are not significantly different from the sampling-based synthesizer in terms of overall quality ratings, although with less variability. Among all evaluated systems, the ratings given to the synthesis from *Fluidsynth* are the most scattered: this may suggest that some listeners are more sensitive than others to an unrealistic feature in this synthesis algorithm.

4.8.4. Conclusion

In this work, the used framework follows the traditional music production workflow, where the two developed modules (Expressive Performance Rendering, sec. 4.7 and DDSP-Piano, sec. 4.8) operate the two transformations between the three entities: *composition*, *interpretation* and *sound*. From a given symbolic composition (made by humans or possibly generated by AI), the first model learns how to interpret it (in order to convey emotions as a musician does), and the second learns how to produce realistic sounds.

The neural synthesizer is informed by high-level physics knowledge (e.g. inharmonicity), and has structural constraints (e.g. sines+noise, FDN reverberation), making it lightweight and strongly interpretable. One of the motivations of this approach is to get explainable, reliable, trustworthy, and sustainable models.

Moreover, contrarily to other approaches of generative AI which generate full music mixes from textual prompts (see e.g. [291], or other commercial services), without other controls; in this work, by splitting the process into different modules which mimic the traditional music production, each step of the music creation is more controllable, which is a key point for AI assistants in music creation.

Finally, let us remark that the developed modules are differentiable, hence this work makes a differentiable bridge between the symbolic composition and the produced sounds, and allows the implementation of more complex neural architectures. For example, by using datasets made of raw composition scores and final mixes recordings, possibly not paired. The target is the pursuit of research works for realistic, lightweight, explainable and controllable generative models, and also for Music Information Retrieval (MIR) tasks, such as automatic transcription, instrument identification, tempo and down-beat estimation.

4.8.5. Relevance to AI4Media use cases and media industry applications

- UC5-B (AI for Games: Music for games):
This module has been integrated into the showcase demonstrator of UC5, which gathered the demonstrators of the two sub-use cases. After selecting relevant MIDI music piece, with suitable mood and ambiances, the music score has been adapted for piano solely. Then, DDSP-Piano synthesized the audio from modified MIDI files.
- UC6 (AI for Human Co-Creation):
This module has been integrated into the prototype of UC6. UC6 uses generative models to help musicians to create new sounds. It has been adapted to optionally get MIDI files as input, instead



of audio files. Finally, DDS-Piano has been integrated to this use case, as a generative model, in order to produce realistic piano sounds.

4.8.6. Relevant Publications

- L. Renault, R. Mignot, and A. Roebel. "Differentiable Piano Model for MIDI-to-Audio Performance Synthesis." 25th International Conference on Digital Audio Effects (DAFx20in22), Vienna, Austria, sept., 2022. <https://doi.org/10.5281/zenodo.7092602>.
- L. Renault, R. Mignot, and A. Roebel. "DDSP-Piano: A Neural Sound Synthesizer Informed by Instrument Knowledge." Journal of the Audio Engineering Society, 71(9), 552-565, 2023. <https://doi.org/10.17743/jaes.2022.0102>, <https://zenodo.org/records/8386706>.

4.8.7. Relevant software/datasets/other outcomes

Source codes and models:

- <https://github.com/lrenault/ddsp-piano>
- DDS-Piano repository on the *AIonDemand* platform

Demonstration pages:

- http://renault.gitlab-pages.ircam.fr/thesis-support/chap_4-1
- http://renault.gitlab-pages.ircam.fr/thesis-support/chap_4-2



5. Learning from scarce data

5.1. Overview

Despite their high accuracy, DNNs typically require a lot of high-quality data to be properly trained, making their deployment difficult in cases where large domain-specific datasets are not readily available. Of course, fully supervised learning is the hardest scenario, since all training examples have to be correctly annotated. **Task 5.3 (T5.3)** “Learning with scarce data” aimed to advance the state-of-the-art in methods attempting to facilitate DNN learning from multimedia content in the face of data scarcity. Unsupervised domain adaptation, semi-supervised learning, few-shot learning, data augmentation and unsupervised representation learning are approaches falling under this category, sharing a common theme of reducing the need for massive, domain-specific, fully and manually annotated training datasets. Methods of this type can increase the applicability of DNNs in real-world scenarios, with T5.3 also partially relating to WP3; notably to transfer learning and learning to count tasks.

T5.3 encompasses a wide range of activities, resulting in a substantial number of research outcomes. These are organized in the following subsections as follows: first, works that address data scarcity through various learning approaches, followed by those that focus on utilizing representation learning for accurate content retrieval.

5.2. Few-shot Object Detection as a Semi-supervised Learning Problem

Contributing partner: JR

5.2.1. Introduction

Most of the literature on few-shot learning focuses on n -way k -shot problems (i.e., problems with n classes and k samples per shot) on predefined splits (i.e., for base and novel classes) of a dataset. However, in many practical few-shot settings the concept of a dataset is fluid, and the available data will evolve over time, with different classes annotated on different datasets. Gupta et al. [292] call this setting of having a set of datasets, for which each has exhaustive annotations for only one or a small set of classes, a *federated dataset*. In such a setting, it is likely that unannotated samples of a class exist in all but one subsets of the dataset. Because of this fact, we argue that few-shot learning is essentially a semi-supervised learning problem. However, it appears that this view on few-shot learning is underrepresented in literature.

We, therefore, analyse whether approaches from semi-supervised learning could be applied in few-shot learning to address this issue of partial annotations on the dataset. While positive samples can be appropriately selected, the risk is that unannotated instances are considered negative samples, and thus penalize detection that may be correct. We select approaches for adjusting the loss and for using predicted samples in the training data. We perform experiments under a range of different settings, using a fine-tuning based few-shot learning framework.

Few-shot learning. Given the need to handle different numbers of samples and add set of classes incrementally, metric or contrastive learning seems more appropriate than meta-learning type of approaches in this setting. Methods of interest are thus [293], which uses a feature pyramid network (FPN) to create an object detection pipeline using metric learning. Classification is done differently for pretrained classes, while few-shot learning is done with FPN (in the DCN variant) instead. [294] propose to train a generic object detector on ImageNet, sampling positive and negative candidate regions. This approach is suitable for generic object detection, beyond the originally trained classes. An approach based on meta-features and learning re-weighting of those features is proposed in [295]. It has been proposed to apply fine-tuning only to region proposal and classification layers on a data set consisting of many base class and few new class samples while fixing the feature extraction part of the network,



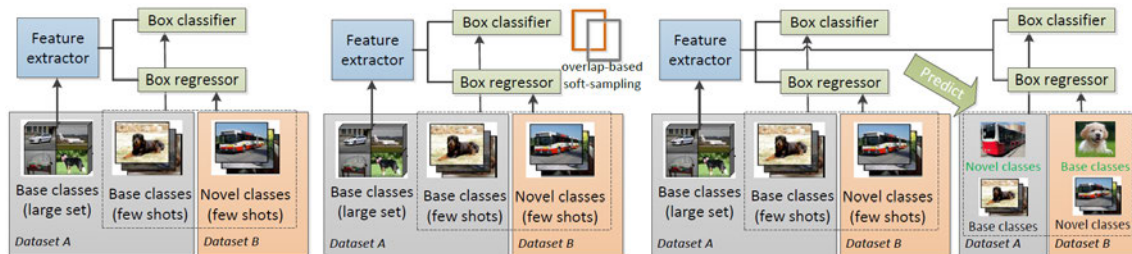


Figure 31. We address the issue of partial annotations in few-shot object detection in a two-stage fine-tuning (TFA) framework. Base setup of the framework (left), extended with soft-sampling to reduce the impact of negative samples caused by missing annotations (middle) and predicting additional annotations (right).

using Faster R-CNN as a backbone [296]. The two-stage fine-tuning approach proposed in [297] has been shown to outperform meta-learning approaches. A recent survey [298] confirms that fine-tuning approaches are a strong baseline for few-shot object detection tasks.

Semi-supervised learning. There are basically two types of approaches from semi-supervised learning that are relevant for our work. One type deals with reducing the impact of potentially missing annotations to be considered as negative samples, and thus influencing the gradient, when objects are detected in this region. The authors of [299] propose an extension of focal loss [300]. Focal loss contains a scaling factor that depends on the prediction confidence. A confidence threshold is introduced, and the loss below this threshold is defined as the mirrored positive branch of focal loss. Two variants of a soft-sampling loss are proposed in [301], using a Gompertz function [302]. One is based on overlap between the annotation and detection, rapidly downweighting the gradient when the overlap gets small. The other is applied when the detector is used to generate missing annotations, and weights the gradient according to the detection confidence.

The other type of approaches uses the partly trained detector to create annotations on the part of the data lacking ground truth for the particular class(es). A Siamese network for sparsely annotated object detection is proposed in [303]. The network consists of two detectors with shared weights, which are fed with an input image and an augmented version of the input image, respectively. From the detector outputs, a set of pseudolabels is generated, and each detector uses the union of the ground truth labels and the pseudo labels of the other detector for supervision. A student-teacher approach is proposed in [304], where the student learns both from annotated data and from pseudo labels generated by the teacher. The teacher model is updated from the student model using exponential moving average, and box proposals are generated using FixMatch [305]. The approach uses separate loss terms for labeled and unlabeled images, which are combined using a weighting factor.

5.2.2. Methodology

Given the good performance of fine-tuning approaches, and the potential to plug in other types of detectors, we use [297] as the basis of our work. This work proposes a two-stage fine-tuning (TFA) approach. A backbone model such as Faster R-CNN is trained on the base classes using a standard training approach. Then the last layer of the model is extended to include the novel classes, and the new weights are randomly initialized. Fine-tuning of the model is performed by training with a dataset formed from k samples from each of the base classes, and the samples of the novel classes. Both the classification and bounding box regression branches are trained using this balanced dataset, but the feature extraction part of the model is not updated. In addition, the fine-tuning step uses a cosine similarity based classifier, which results in improved accuracy for the novel classes and lower decrease for the base classes compared to an FC-based classifier. As an alternative to randomly initializing the new weights, a separate training



step for the last layer can be performed with the new classes, and the results can be used to initialize the weights of the novel classes in the combined model. We have done this training step for the novel classes in all our experiments. We integrated two approaches into our training pipeline: a soft-sampling function to handle missing annotations, and the use of a predictor to generate additional annotations for the missing classes. The experimental implementations of the approaches tested in this work are available at GitHub¹³.

Soft sampling. We modify the gradient calculation to account for cases where there is no or small overlap with a ground truth bounding box. The ground truth bounding boxes provide hard positive and negative samples, while other image regions cannot be considered negative samples in our setting, as they may contain samples of classes not annotated on the particular share of the dataset.

The overlap based soft-sampling function from [301] is defined as

$$\mathcal{G}(o) = a + (1-a)e^{-be^{-co}}, \quad (41)$$

where o is the overlap (i.e., IoU of annotation and detection regions), a is the minimum weight (for $o=0$), b determines the location of the decay along the overlap range, and c controls the growth rate. This function provides a weighting factor for the gradient of the specific head (classification or regression of the network).

The implementation of the two-stage fine-tuning approach we use¹⁴ uses cross-entropy loss for classification and smooth L1 loss for box regression. We implemented the weighting of the respective gradient based on box overlap in both branches. In order to inject the weighting factor into the gradient without modifying the actual output of the loss function, we use Pytorch backward tensor hooks. The vector of weighting factors for a batch is prepared when the loss is determined, and registered with the tensor hook. When the gradient calculation is performed on the tensor, the function registered with the hook is called and the weighting factors can be applied.

Using predictions. Inspired by the approach in [304], we use a similar approach of using a previous version of the detector as teacher, and train a student with a combination of ground truth annotations and predictions. The teacher model is always based on a model trained on the base classes. One option is to use the model trained for the novel classes only as a predictor for the novel classes. The second option is to use the model obtained from fine-tuning with the annotated few-shot set. With the first option, we actually use two teacher models, as we use the base model for the base classes (annotating images containing only annotations for novel classes), while we use the initial model trained for the novel set on images containing only annotations for the base classes. With the second option we have only one teacher model, but we use only the respective subset of classes. We generally use a single confidence threshold for the model outputs, which is rather low, as the model trained (or trained and fine-tuned) on few samples of the novel classes reports rather low confidence scores for these classes. However, it would be possible to apply different confidence scores for the predictions of base and novel classes. Note that currently the modified loss from [304] is not included in our approach. The complexity of supporting this loss comes from the fact that the original of each data sample needs to be traced, so that the loss function can consider the specific pair of class and data origin to determine the loss.

5.2.3. Experimental Results

We use a 10-shot training problem in our experiments. We use the MS COCO dataset [306], in particular seed 0 and a split of the classes into 60 base and 20 novel classes, both as defined in [297]. We use the same parameters for novel model training and fine tuning as in their work, i.e., when we fine-tune with 10 samples, we use the parameters from the 10-shot configuration, and when we fine-tune with 30 samples, we use those from the 30-shot configuration.

¹³<https://github.com/wbailer/few-shot-object-detection/tree/semi-supervised>

¹⁴<https://github.com/ucbdrive/few-shot-object-detection>





Data share	θ	samples 10-shot	samples additional	images searched
predicted 20	0.09	1406	194	2410
predicted 20	0.11	1296	304	10010
predicted 30	0.09	0	2400	5500

Table 37. Number of samples from different shares of the data, and additional images needed.

We create four baselines to compare our results against. All baselines start from the same model trained on 10 samples of the 20 novel classes, but use different approaches for fine-tuning. All baselines use ground truth data only.

Lower baseline. This is the default 10-shot pipeline with two-stage fine-tuning as described in [297].

Lower baseline (any 10). As the diversity of samples seems to have influence, especially when the number of samples is small, this baseline uses the same setting as the lower baseline, but selects 10 new samples for the fine-tuning stage for each class, instead of using the same that were used to train the novel classifier. In practice, this means that such a setup will require 20 annotated samples for the novel classes instead of 10.

Upper baseline. This baseline uses the subset for the 30-shot training task in [297] for fine-tuning. This means, 30 new samples are used for fine-tuning, and it total this would require 40 annotated samples.

Upper baseline (fixed 10). Similar to the upper baseline, we use 30 samples for fine-tuning, but 10 of them are those used in training the novel classifier.

For soft-sampling, the training pipeline is only slightly modified from the original setting. We only use the overlap-based soft-sampling loss in the fine-tuning stage. As proposed in [301], we set the parameters of the overlap-based soft-sampling function as $a=0.25, b=50, c=20$.

When using additional predictions, we would ideally want to use the same share of the data as in one of the baselines. However, we found in earlier experiments that a balanced number of samples is very important in a few-shot setting, due to the small sample sizes involved. Thus it will often not be possible to find a sufficient number of samples for a particular class in the share of the data used for baseline experiments. We use the following strategy to gather the required samples: We start from the set of images of the 10-shot fine-tuning set, i.e., including the annotated samples of all classes, both from the base and novel sets. We run the detector for novel classes on the images with annotations for the base classes, and vice versa. If less than the target number of samples have been obtained, we randomly sample additional images and run the detector. This process is repeated until the target number of samples has been obtained.

Table 37 lists the statistics of the data shares that have been created. The number in the data share (20, 30) specifies the target number of samples. If the number is 20, this means that the dataset is intended to be compared to the *upper baseline (fixed 10)*, and used in combination with the 10 samples used for training the classifier, while 30 means that it is intended to be used alone, to be compared with the *upper baseline*. The threshold θ is the confidence threshold, and predictions with a confidence $\geq \theta$ are used.

We test two detectors to generate the predictions for use in fine-tuning. The first is to use the detector obtained from training the novel classes, while the second performs initial fine-tuning using the 10-shot dataset with the samples as for training the novel classifier, and uses the resulting detector to generate additional annotations.

The results of our experiments are summarised in Table 38. In the table, the *soft-sampling* column indicates whether soft-sampling has been applied. The column *data share* indicates the dataset used for the experiments, where GT means ground truth data has been used, and predicted refers to one of the predicted datasets from Table 37 (for these experiments also the confidence threshold θ used for creating the dataset is reported). For the ground truth datasets, the *k-shot* datasets refer to the subsets created according to [297], while 10+20 refers to using the 10-shot subset, and sampling 20 additional





	soft-sampling	data share	fine-t. param.	all			novel		
				AP	AP50	AP75	AP	AP50	AP75
lower baseline	no	GT 10-shot	10	28.8003	44.3573	31.6091	6.7166	12.4130	6.4037
upper baseline	no	GT 30-shot	30	28.7469	44.5792	31.6566	12.2760	21.4859	12.4071
lower baseline (any 10)	no	GT any10	10	27.1242	41.2260	29.9908	6.8099	12.7262	6.3758
upper baseline (fixed 10)	no	GT 10+20	30	28.1997	43.9137	31.0884	10.8204	19.5422	10.7415
soft	both	GT 10shot	10	29.3976	45.8916	31.8329	7.6669	15.0144	6.7683
soft cl.	class.	GT 10shot	10	29.3745	45.8880	31.9559	7.8241	15.0659	7.1854
soft (any 10)	both	GT any10	10	28.7596	44.6806	31.2693	8.0540	15.5329	7.3533
pred. novel, $\theta=0.09$	no	predicted 20	30	22.7838	35.6272	25.0973	5.9141	10.5430	6.0409
pred. fine-tuned, $\theta=0.09$	no	predicted 20	30	22.5899	36.6769	24.7648	6.9385	13.9793	6.5406

Table 38. Results for baselines, soft-sampling and prediction experiments. Average precision (AP) for IoU 50% and 75% as well as average AP are provided for all and novel classes.

samples from the ground truth set, and any10 refers to 10 randomly sampled ground truth items. The *fine-tuned parameters* column specifies which hyperparameters are used during fine-tuning. We use the parameters for the 10 and 30 shot settings as proposed in [297].

We can make the following observations from the results. From the baseline experiments, we observe that as expected using more samples in fine-tuning improves results for the novel classes, while the overall results are very similar. Using different samples for fine-tuning than for the initial training provides almost identical results for 10 samples, while using 30 new samples provides a larger improvement for novel classes than adding only 20 samples to the 10 used before.

Using soft-sampling provides small but consistent improvements over the baseline, both for novel classes and overall. The results with 10 new samples are slightly better than those using the same samples for the novel classes. We also compared using the soft-sampling function in both heads and in the classification head only. Using it in classification only provides a small improvement over using it in both heads.

When using the predictions, the results using the novel classifier do not outperform the baseline. However, using the predictions after initial fine-tuning results in a small improvement for the novel classes, but at the cost of reducing the overall performance.

5.2.4. Relevance to AI4Media use cases and media industry applications

Few-shot object detection is useful in order to extend object detection capabilities in sourcing (e.g., annotation of feeds of raw material) or archiving with specific object classes of interest for a particular organization or production context. If the object class of interest is not covered by a publicly available dataset (or license conditions do not permit the use of such a dataset), the labeling of a large amount of training samples is typically not feasible. Few-shot object detection enables training with an amount of samples that can be labeled by a single user with acceptable effort. While the resulting classifier is likely to achieve lower performance than one trained on a thousands of samples, it may still provide detection of otherwise uncovered classes. In addition, detection results (possibly in combination with object tracking) can be used for retraining a classifier on a larger set.

5.2.5. Relevant Publications

- W. Bailer, H. Fassold, "Few-shot Object Detection as a Semi-supervised Learning Problem", Proceedings of the 19th International Conference on Content-based Multimedia Indexing (CBMI), 2022. Zenodo record: <https://zenodo.org/records/7037584>
- W. Bailer, M. Dogariu, B. Ionescu, H. Fassold, "Few-Shot Object Detection as a Service: Facilitating Training and Deployment for Domain Experts", Proceedings of the 19th International



Conference on Multimedia Modeling (MM), 2024.
Zenodo record: <https://zenodo.org/records/10636415>

5.2.6. Relevant software/datasets/other outcomes

The code for the framework is available at <https://github.com/wbailer/few-shot-object-detection>

5.3. Bioinspired learning approaches to data scarcity

Contributing partner: CNR

5.3.1. Introduction

Today's neural networks are generally trained using Stochastic Gradient Descent (SGD) with the error backpropagation algorithm (backprop), which reaches high accuracy when a large number of labeled samples are available for training. However, gathering labeled samples is expensive, requires a significant amount of human work, and, in many applications, a large amount of training data is simply not available.

Researchers started to investigate strategies for sample efficient learning [307, 308, 309, 310, 311, 312, 313]. In this setting, only a small number of labeled samples is assumed to be available. On the other hand, gathering unlabeled samples is relatively simple; therefore, these approaches exploit unlabeled samples to perform unsupervised training in addition to the supervised training process, leading to the so called *semi-supervised* learning technique. It is well known that unsupervised pre-training helps initializing the network weights in the neighborhood of a good local optimum [307, 308], thus easing convergence in a successive supervised fine-tuning phase. Current semi-supervised approaches leverage autoencoder architectures for the unsupervised part of the task [310, 311, 312], although they are still based on backprop. Another approach, SimCLR [313], exploits data augmentation and an unsupervised contrastive criterion.

We addressed the sample efficiency issue by developing a semi-supervised learning approach, where an initial unsupervised learning step, using all available data – unlabeled and labeled (but without using label information) –, is followed by a supervised learning step using a small amount of labeled data. To perform the unsupervised learning step we explore the use of the Hebbian learning paradigm, which mimics more closely the synaptic adaptation mechanisms found in biological brains, according to neuroscientists. Hebbian learning is a local learning rule [314, 315], i.e. it does not require error backpropagation, and it does not require supervision. Moreover, the capabilities of biological brains to learn and generalize only from a limited number of labeled samples make this approach appealing for the sample efficient learning setting. Note also that backprop-based approaches are considered to be biologically implausible [316].

5.3.2. Methodology

Despite their promising results, current Hebbian learning solutions could hardly be used to address large-scale problems, due to their demanding computational cost. In this perspective, we developed a new Hebbian learning solution, named *FastHebb*, which is designed to better take advantage of GPU acceleration. This is done in two steps. First, we notice that Hebbian learning with mini-batch processing evolves in two stages, one is the weight update computation for each sample in the mini-batch, and the other is the aggregation of updates over all the mini-batch elements. These two phases can be merged together with a significant speedup. Second, the resulting Hebbian equations of synaptic updates can be translated in terms of matrix multiplications, which can be executed very efficiently on GPU.

Our main contributions related to this activity are the following:

- We developed a semi-supervised learning approach that combines Hebbian learning with SGD on object recognition tasks with Deep Convolutional Neural Networks (DCNNs).

- All available training samples, unlabeled and labeled, are used for an unsupervised Hebbian pre-training phase (without using label information), where a nonlinear Hebbian Principal Component Analysis (HPCA) learning rule is used to train internal layers (both convolutional and fully connected);
- Then, labeled training samples and SGD are used to train a classifier, obtained as a final fully connected layer, on the features extracted from previous layers;
- We compared the results from a sample efficiency perspective with those obtained by a baseline network trained end-to-end with backpropagation, on the labeled samples, and with semi-supervised learning based on Variational Auto-Encoder (VAE) [317] unsupervised pre-training, the latter using all the available samples (VAE-based semi-supervised learning was also the approach considered in [310]);
- Different datasets and different regimes of sample efficiency were explored, and it was shown that the proposed semi-supervised approach (Hebbian + SGD) outperforms the other approaches in almost all the cases where a limited number of labeled samples is available;
- We developed a scalable solution for Hebbian synaptic updates (FastHebb) and performed exhaustive experimentation on large-scale datasets (ImageNet) and architectures (VGG) which (to the best of our knowledge) have been out of reach for Hebbian algorithms so far.

5.3.3. Experimental results

Experiments on Tiny ImageNet allowed us to validate the scalability of our methods to large datasets. Tiny ImageNet has 200 classes and the training set consists of 100,000 samples (90,000 of which are used for training and 10,000 for validation). Results are reported in Table 39, where the top-5 accuracy measures are shown, along with their 95% confidence interval.

In regimes where a limited number of labeled samples is available (between 1% and 5%), the Hebbian approach outperforms other counterparts, in almost all the cases. On the other hand, when the number of available labeled samples becomes larger, BP and VAE approaches (which exploit end-to-end fine tuning in the supervised phase) are able to take advantage of supervision and improve over HPCA, and our end-to-end fine tuning in HPCA+FT helps to further boost accuracy.

Specifically, HPCA outperforms BP in all layers up to 4% sample efficiency regime. In addition, we can observe that HPCA generally outperforms backprop by roughly 1-2 percent points, reaching a peak of almost 3 percent points on layer 3, in the 4% sample efficiency regime. With higher efficiency regimes, backprop begins to outperform HPCA, starting from the higher layers. At 100% sample efficiency regime, backprop outperforms HPCA on all layers. This is probably due to the fact that 90,000 labeled training samples are sufficient for BP to correctly train all network layers, exploiting the supervised information.

We observe that HPCA always performs better than the VAE method when low sample efficiency regimes are considered (between 1% and 5%), especially for higher network layers. VAE pre-training seems to be more effective in regimes where more labeled samples are available (beyond 10%).

The HPCA+FT strategy is still preferable in low sample efficiency regimes (between 1% and 5%), where it helps to further increase accuracy w.r.t. plain HPCA. In particular, in these regimes, we can observe a further increase in accuracy up to 3% points on layer 5 (in the 5% regime). Fine tuning also helps increasing accuracy in successive sample efficiency regimes, especially on higher layers.

5.3.4. Relevance to AI4Media use cases and media industry applications

This activity is related to UC3 (AI in Vision - High quality Video Production and Content Automation), where it can be used to train a neural network to extract visual features in an unsupervised fashion to allow effective retrieval of relevant visual material.



Table 39. Tiny ImageNet accuracy (top-5) and 95% confidence intervals obtained with a linear classifier on top of various layers, for the various sample efficiency regimes. Results obtained with supervised backprop (BP), VAE-based semi-supervised approach (VAE), Hebbian PCA (HPCA), and HPCA plus Fine Tuning (HPCA+FT) are compared. It is possible to observe that, in regimes where the number of available samples is low (roughly between 1% and 5% of the total available samples), HPCA performs better than BP and VAE approaches in almost all the cases, leading to an improvement up to almost 3% (on layer 3, in the 4% regime) w.r.t. non-Hebbian approaches. HPCA+FT helps to further boost accuracy.

Regimes	Method	L1	L2	L3	L4	L5
1%	BP	9.89 \pm 0.15	10.10 \pm 0.26	9.99 \pm 0.17	9.15 \pm 0.23	9.53 \pm 0.21
	VAE	9.63 \pm 0.26	9.49 \pm 0.39	7.58 \pm 0.28	5.99 \pm 0.19	5.55 \pm 0.23
	HPCA	10.83 \pm 0.28	10.87 \pm 0.26	11.85 \pm 0.19	10.84 \pm 0.26	10.86 \pm 0.23
	HPCA+FT	10.81 \pm 0.27	10.99 \pm 0.36	12.15 \pm 0.46	11.05 \pm 0.27	11.38 \pm 0.41
2%	BP	12.76 \pm 0.27	12.84 \pm 0.14	13.95 \pm 0.34	13.04 \pm 0.15	13.48 \pm 0.39
	VAE	12.94 \pm 0.37	13.06 \pm 0.23	10.86 \pm 0.28	7.40 \pm 0.27	6.74 \pm 0.20
	HPCA	13.84 \pm 0.17	14.35 \pm 0.15	16.18 \pm 0.15	14.52 \pm 0.32	14.03 \pm 0.15
	HPCA+FT	14.12 \pm 0.23	14.32 \pm 0.31	16.89 \pm 0.61	15.28 \pm 0.28	15.71 \pm 0.47
3%	BP	14.12 \pm 0.20	14.65 \pm 0.57	16.50 \pm 0.32	15.76 \pm 0.27	15.99 \pm 0.38
	VAE	14.31 \pm 0.18	15.17 \pm 0.20	13.67 \pm 0.36	8.35 \pm 0.29	7.74 \pm 0.19
	HPCA	16.13 \pm 0.14	16.32 \pm 0.33	18.87 \pm 0.29	17.04 \pm 0.26	16.38 \pm 0.25
	HPCA+FT	16.25 \pm 0.21	16.54 \pm 0.28	19.78 \pm 0.47	18.31 \pm 0.24	18.23 \pm 0.33
4%	BP	15.44 \pm 0.42	16.72 \pm 0.31	18.36 \pm 0.22	17.85 \pm 0.16	17.84 \pm 0.19
	VAE	16.09 \pm 0.20	17.05 \pm 0.20	16.83 \pm 0.51	8.86 \pm 0.11	8.45 \pm 0.21
	HPCA	17.64 \pm 0.49	18.27 \pm 0.34	21.07 \pm 0.17	19.16 \pm 0.33	18.13 \pm 0.39
	HPCA+FT	17.70 \pm 0.44	18.33 \pm 0.24	21.95 \pm 0.57	20.86 \pm 0.32	20.55 \pm 0.28
5%	BP	16.75 \pm 0.25	17.94 \pm 0.25	20.26 \pm 0.21	20.15 \pm 0.35	19.84 \pm 0.36
	VAE	17.44 \pm 0.26	18.62 \pm 0.32	19.16 \pm 0.52	9.92 \pm 0.24	9.29 \pm 0.17
	HPCA	18.93 \pm 0.14	19.67 \pm 0.36	22.65 \pm 0.35	21.01 \pm 0.38	19.57 \pm 0.15
	HPCA+FT	19.26 \pm 0.41	19.93 \pm 0.41	23.97 \pm 0.52	22.95 \pm 0.26	22.46 \pm 0.17
10%	BP	20.26 \pm 0.18	23.12 \pm 0.14	27.05 \pm 0.20	27.30 \pm 0.20	27.21 \pm 0.29
	VAE	21.62 \pm 0.25	23.83 \pm 0.19	27.42 \pm 0.18	16.69 \pm 0.18	13.51 \pm 0.34
	HPCA	22.15 \pm 0.43	23.69 \pm 0.24	27.02 \pm 0.24	25.73 \pm 0.34	23.08 \pm 0.17
	HPCA+FT	22.82 \pm 0.33	24.34 \pm 0.29	28.69 \pm 0.36	28.79 \pm 0.26	28.13 \pm 0.38
25%	BP	28.97 \pm 0.26	32.63 \pm 0.36	37.38 \pm 0.13	38.81 \pm 0.20	38.80 \pm 0.39
	VAE	29.40 \pm 0.31	32.42 \pm 0.29	39.93 \pm 0.31	37.97 \pm 0.62	37.89 \pm 0.54
	HPCA	27.05 \pm 0.47	28.39 \pm 0.34	32.08 \pm 0.19	31.30 \pm 0.26	29.51 \pm 0.23
	HPCA+FT	28.01 \pm 0.75	30.63 \pm 0.16	35.87 \pm 0.53	36.98 \pm 0.26	37.10 \pm 0.23
100%	BP	42.89 \pm 0.13	49.94 \pm 0.13	54.54 \pm 0.27	57.00 \pm 0.16	57.50 \pm 0.16
	VAE	42.32 \pm 0.16	48.54 \pm 0.53	58.31 \pm 0.12	59.60 \pm 0.23	60.23 \pm 0.65
	HPCA	35.74 \pm 0.15	38.29 \pm 0.19	38.78 \pm 0.07	38.61 \pm 0.21	36.99 \pm 0.36
	HPCA+FT	40.34 \pm 0.31	45.00 \pm 0.40	53.12 \pm 0.26	52.95 \pm 0.28	53.96 \pm 0.43





5.3.5. Relevant Publications

- G. Lagani et al., "Assessing Pattern Recognition Performance of Neuronal Cultures through Accurate Simulation," 2021 10th International IEEE/EMBS Conference on Neural Engineering (NER), 2021, pp. 726-729, [doi:10.1109/NER49283.2021.9441166](https://doi.org/10.1109/NER49283.2021.9441166).
- Gabriele Lagani, Fabrizio Falchi, Claudio Gennaro, Giuseppe Amato, "Hebbian semi-supervised learning in a sample efficiency setting", Neural Networks, Volume 143, 2021, Pages 719-731, ISSN 0893-6080
- Gabriele Lagani, Davide Bacciu, Claudio Gallicchio, Fabrizio Falchi, Claudio Gennaro, and Giuseppe Amato. 2022. Deep Features for CBIR with Scarce Data using Hebbian Learning. In Proceedings of the 19th International Conference on Content-based Multimedia Indexing (CBMI '22). Association for Computing Machinery, New York, NY, USA, 136-141. <https://doi.org/10.1145/3549555.3549587>
- Lagani, G., Gennaro, C., Fassold, H., Amato, G. (2022). FastHebb: Scaling Hebbian Training of Deep Neural Networks to ImageNet Level. In: Skopal, T., Falchi, F., Lokoč, J., Sapino, M.L., Bartolini, I., Patella, M. (eds) Similarity Search and Applications. SISAP 2022. Lecture Notes in Computer Science, vol 13590. Springer, Cham. https://doi.org/10.1007/978-3-031-17849-8_20
- Lagani, G., Falchi, F., Gennaro, C. et al. "Comparing the performance of Hebbian against backpropagation learning using convolutional neural networks". Neural Comput & Applic (2022). <https://doi.org/10.1007/s00521-021-06701-4>
- Lagani G., Falchi F., Gennaro C., Amato G. (2022) "Training Convolutional Neural Networks with Competitive Hebbian Learning Approaches". In: Nicosia G. et al. (eds) Machine Learning, Optimization, and Data Science. LOD 2021. Lecture Notes in Computer Science, vol 13163. Springer, Cham. https://doi.org/10.1007/978-3-030-95467-3_2
- Lagani G., Falchi F., Gennaro C., Amato G. (2022) "Evaluating Hebbian Learning in a Semi-supervised Setting. In: Nicosia G. et al. (eds) Machine Learning, Optimization", and Data Science. LOD 2021. Lecture Notes in Computer Science, vol 13164. Springer, Cham. https://doi.org/10.1007/978-3-030-95470-3_28
- Gabriele Lagani, Fabrizio Falchi, Claudio Gennaro, Hannes Fassold, Giuseppe Amato, Scalable bio-inspired training of Deep Neural Networks with FastHebb, Neurocomputing, Volume 595, 2024, 127867, ISSN 0925-2312, <https://doi.org/10.1016/j.neucom.2024.127867>

5.3.6. Relevant software/datasets/other outcomes

- GitHub repository of the Hebbian Learning CNN project: <https://github.com/aimh-lab/hebbian-learning-cnn>

5.4. Domain Adaptation and Counting techniques

Contributing partner: CNR

5.4.1. Introduction

Most CNN-based methods require a large amount of labeled data and make a common assumption: the training and testing data are drawn from the same distribution. The direct transfer of the learned features between different domains does not work very well because the distributions are different. Thus, a model trained on one domain, named *source*, usually experiences a drastic drop in performance when applied on another domain, named *target*. This problem is commonly referred as *Domain Shift* [318].

This problem is relevant, for instance, when counting techniques developed for one application need to be adapted to new applications.





One possible solution to tackle this issue is represented by *Unsupervised Domain Adaptation - (UDA)*. Specifically, it aims at mitigating domain shifts between different domains, relying on labeled data in the source domain and *unlabeled* data in the target domain. In other words, UDA techniques exploit annotated data from the source domain as well as *non-annotated* data coming from the target domain that is easy to gather since it does not require human effort for labeling. The challenge here is to automatically infer some knowledge from this latter data flow to reduce the gap between the two domains and, specifically, to learn feature representations that should be (i) discriminative for the main learning task on the source domain and (ii) indiscriminative concerning the shift between the domains.

5.4.2. Methodology

At the beginning of the project, as reported in D5.1, we applied UDA techniques to density estimations and, more specifically, to vehicle counting. We have also applied variations to these techniques to new applications:

DL-based pipeline for whitefly pest abundance estimation on chromotropic sticky traps: We developed an automated counting pipeline based on data-driven Artificial Intelligence, specifically Deep Learning (DL), for estimating the number of pests in images of sticky chromotropic traps. Our approach follows a modular paradigm and is model-agnostic: differently from most existing works that employ specific object detectors, the module responsible for counting can be implemented with recent SOTA methodologies, not only detection-based but also relying on regression. Its output is then fed into downstream modules that produce unified outputs expressing localization and confidence scores of the counted insects. The required data was collected by taking digital camera pictures of the traps placed in insect hot spot locations at the University of Pisa (Italy). Subsequently, images were annotated by putting dots over the centroids of the trapped insects of interest; dotting emulates the natural human technique for counting objects (at least when the number of objects is greater than the subitizing range), and it represents the golden standard concerning the labels needed for the supervised training of deep learning models for the counting task [319]. We named this collection of images PST - Pest Sticky Traps and publicly released it [320]. In this setting, we experiment with several approaches: our best-performing solution achieves an average counting error of approximately 9% compared to human capabilities while requiring mere seconds for computation, in contrast to the hours or days needed for manual human inspections.

Learning to count biological structures with raters' uncertainty: We developed a deep learning-based counting system for biological structures that takes as input a microscopy image and produces as output the localization of the objects to be counted; furthermore, it also produces associated scores indicating the reliability of the detections that practitioners can use to exclude or include from the total count. More in detail, we developed a two-stage architecture, each having its own separate training phase. In the first stage, a deep-learning network that takes as input an image and produces as output a set of coordinates localizing the biological structures to be considered will be developed, following several different architectures and strategies based on segmentation, detection, and density estimation. This model is trained with a large labeled dataset annotated by a single expert; due to the intricate patterns characterizing the distributions of the biological structures, this dataset likely encompasses errors, and consequently, the network output will present weak localization. In the second stage, the previously localized objects are considered, and they are assigned an “objectness” score that indicates the reliability of the detections. To do so, a scorer module that inputs a small cropped patch containing the previously localized objects and outputs a scalar score is employed.

Detection of violence in videos: We used UDA techniques in the the specific task of violence detection in *trimmed* videos, i.e., capturing an exact action (either violent or non-violent). This task is a subset of human action recognition. Specifically, the goal is to binary classify clips to predict if they contain (or not) any behaviors considered to be violent, differing from violent detection in *untrimmed*





videos, a subset of action localization where the purpose is also to seek the action in the temporal dimension. Despite its importance in many practical, real-world scenarios, it is relatively unexplored compared to other action recognition tasks. Although some annotated datasets for video violence detection in general contexts already exist, they are limited in size and in the considered different scenarios. Therefore, existing Deep Learning-based solutions trained using these data systematically experience performance degradation when applied to new specific contexts, such as violence detection in public transport environments [321].

To mitigate this problem, we proposed an end-to-end DL-based UDA solution to detect violent situations in videos in specific target scenarios where annotated data is scarce or lacking. Our proposal relies on *single* image classification randomly sampled from the frames making up the video, a simple technique already addressed by [322]. Starting from this, some UDA techniques for image classification are employed during the training pipeline, automatically gathering some knowledge from the unlabeled data belonging to the target domain. To the best of our knowledge, it is the first attempt at using a UDA schema for video violence detection. We conducted experiments by exploiting, as the source domain, several annotated datasets present in the literature dealing with video violence detection in general contexts and, as the target domain, the recently introduced *Bus Violence* benchmark [321], a collection of clips specific for detection of violent behaviors inside a moving bus. Experimental results show that by using our UDA pipeline, we can improve the performance of the considered models by a significant margin, thus suggesting that they generalize better over this new scenario without the need to use new labels.

5.4.3. Experimental Results

Here we report some results when using UDA solutions to adapt to violence detection.

The ResNet50 architecture with the UDA strategy, gains 7.4%, 0.37%, and 12.9% of accuracy compared with the ResNet50 network without UDA, overcoming all the other considered methods tested.

Considering *False Alarms* and *Missing Alarms*, ResNet50 architecture with UDA mitigates this issue, achieving better performance compared with the single ResNet50 model and often overtaking all the other techniques. This behavior is linked with a lower number of detected False Negatives and consequently affects and improves the *Recall* and *F1-score*. In Figure 32, we report some samples of True Positive, True Negative, False Positive, and False Negative coming out from the ResNet50 architecture with attached the UDA module.

5.4.4. Relevance to AI4Media use cases and media industry applications

This activity is related to UC3 (AI in Vision - High quality Video Production and Content Automation), where it can be used as a solution to adapt AI models to continuously evolving scenarios (eg. newly occurring events, facts, or trends) when dealing with large and highly dynamic audio-visual archives.

5.4.5. Relevant Publications

- Luca Ciampi, Nicola Messina, Gaetano Emanuele Valenti, Giuseppe Amato, Fabrizio Falchi, Claudio Gennaro (2023). MC-GTA: A Synthetic Benchmark for Multi-Camera Vehicle Tracking. ICIAP 2023: 22nd International Conference on Image Analysis and Processing. September 11-15, 2023, Udine, Italy. https://doi.org/10.1007/978-3-031-43148-7_27
- Ciampi L. and Santiago C. and Costeira J. P. and Falchi F. Gennaro C. and Amato G., Un-supervised domain adaptation for video violence detection in the wild, IMPROVE 2023 - 3rd International Conference on Image Processing and Vision Engineering, pp. 37–46, Prague, Czech Republic, 21-23/04/2023, <https://doi.org/10.5220/0011965300003497>
- Ciampi, Luca, Paweł Foszner, Nicola Messina, Michał Staniszewski, Claudio Gennaro, Fabrizio Falchi, Gianluca Serao, Michał Cogieł, Dominik Golba, Agnieszka Szczęśna, and Giuseppe Amato.



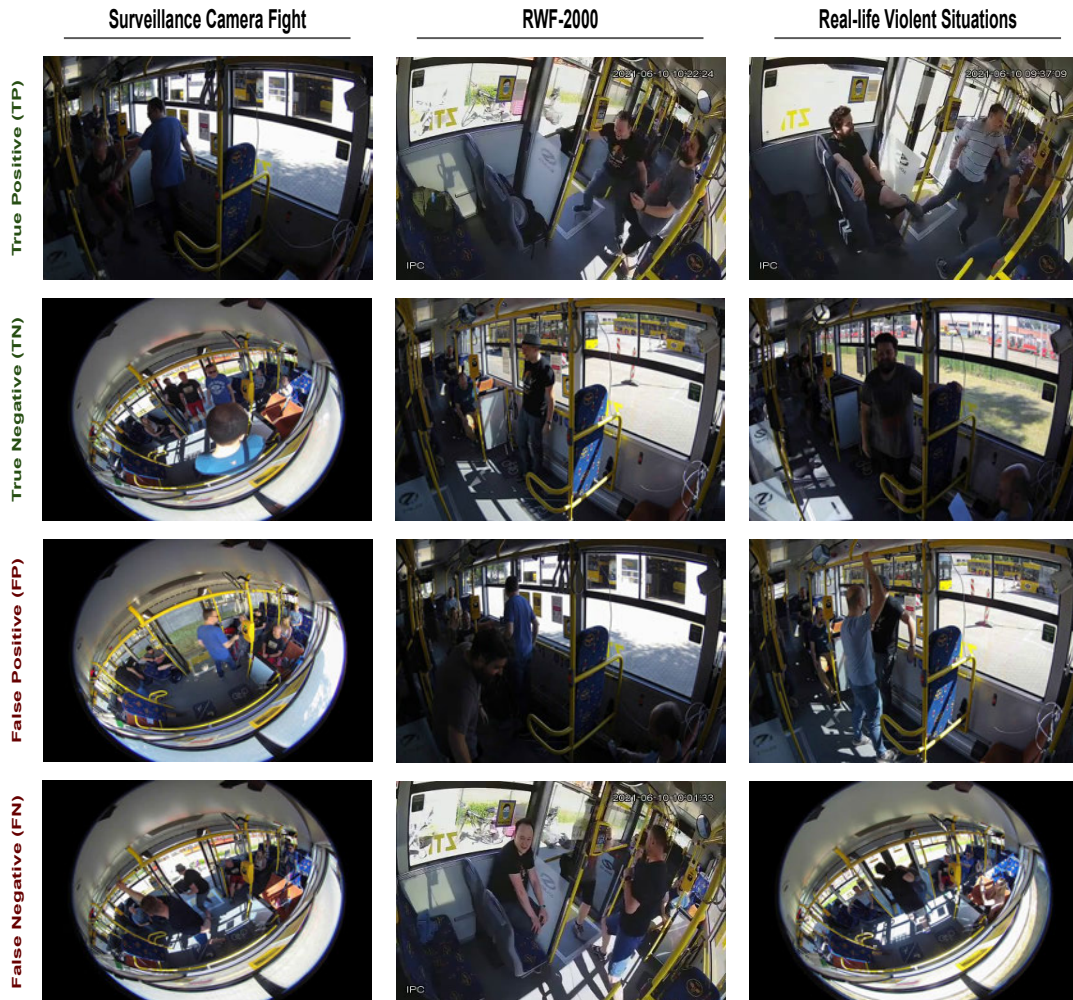


Figure 32. Some samples of predictions over the target domain. In the four rows, we report some samples of True Positives, True Negatives, False Positives, and False Negatives concerning the best model, i.e., ResNet50 + UDA, for each of the considered source domains (one for each column).

2022. "Bus Violence: An Open Benchmark for Video Violence Detection on Public Transport" Sensors 22, no. 21: 8345, <https://zenodo.org/records/7044203>

- Giuseppe Amato, Fabio Carrara, Luca Ciampi, Marco Di Benedetto, Claudio Gennaro, Fabrizio Falchi, Nicola Messina, Claudio Vairo, "AI and Computer Vision for Smart Cities", 8th Italian Conference on ICT for Smart Cities And Communities, 14-16 September, 2022 | University of Camerino - Ascoli Piceno, Italy
- Ciampi L., Carrara F., Amato G., Gennaro C. "Counting or Localizing? Evaluating Cell Counting and Detection in Microscopy Images", In Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2022) - Volume 4: VISAPP, pages 887-897, ISBN: 978-989-758-555-5; ISSN: 2184-4321, <https://zenodo.org/records/6367420>
- Marco Di Benedetto, Fabio Carrara, Luca Ciampi, Fabrizio Falchi, Claudio Gennaro, Giuseppe



Amato, "An embedded toolset for human activity monitoring in critical environments", Expert Systems with Applications, Volume 199, 2022, 117125, ISSN 0957-4174, <https://doi.org/10.1016/j.eswa.2022.117125>

- Luca Ciampi, Claudio Gennaro, Fabio Carrara, Fabrizio Falchi, Claudio Vairo, Giuseppe Amato, Multi-camera vehicle counting using edge-AI, Expert Systems with Applications, Volume 207, 2022, 117929, ISSN 0957-4174, <https://doi.org/10.1016/j.eswa.2022.117929>.
- Luca Ciampi, Fabio Carrara, Valentino Totaro, Raffaele Mazziotti, Leonardo Lupori, Carlos Santiago, Giuseppe Amato, Tommaso Pizzorusso, Claudio Gennaro, Learning to count biological structures with raters' uncertainty, Medical Image Analysis, Volume 80, 2022, 102500, ISSN 1361-8415, <https://doi.org/10.1016/j.media.2022.102500>
- M. Avvenuti, M. Bongiovanni, L. Ciampi, F. Falchi, C. Gennaro and N. Messina, "A Spatio-Temporal Attentive Network for Video-Based Crowd Counting," 2022 IEEE Symposium on Computers and Communications (ISCC), Rhodes, Greece, 2022, pp. 1-6, [doi:10.1109/ISCC55528.2022.9913019](https://doi.org/10.1109/ISCC55528.2022.9913019)
- Ciampi, L., Santiago, C., Costeira, J.P., Gennaro, C., Amato, G., "Domain adaptation for traffic density estimation", VISIGRAPP 2021 - Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, Volume 5, Pages 185-195, 2021
- Luca Ciampi, Valeria Zeni, Luca Incrocci, Angelo Canale, Giovanni Benelli, Fabrizio Falchi, Giuseppe Amato, Stefano Chessa, A deep learning-based pipeline for whitefly pest abundance estimation on chromotropic sticky traps, Ecological Informatics, Volume 78, 2023, 102384, ISSN 1574-9541, <https://doi.org/10.1016/j.ecoinf.2023.102384>

5.4.6. Relevant software/datasets/other outcomes

- Vehicle counting on the AI4Europe catalog: <https://www.ai4europe.eu/research/ai-catalog/ai-visual-vehicles-counting>

5.5. Augmentation for Self-supervised and semi-supervised learning

Contributing partner: UNIFI

5.5.1. Introduction

In this subsection, we discuss UNIFI's contribution on methods to perform learning in settings with limited access to annotations. Effective color space augmentation was studied in self-supervised learning in [323]. A pipeline for data augmentation based on synthetic object generation was presented in [324]. This approach is specifically addressing small-object detection when few small-objects are annotated.

5.5.2. Methodology

5.5.2.1. Planckian Jitter for Color Augmentation We call our color data augmentation procedure *Planckian Jitter* because it exploits the physical description of a black-body radiator to re-illuminate training images within a realistic illuminant distribution [325, 326]. The resulting augmentations are more realistic than those of the default color jitter. The resulting learned, self-supervised feature representation is thus expected to be robust to illumination changes commonly observed in real-world images, while simultaneously maintaining the ability to discriminate the image content based on color information.

Given an input RGB training image I , our Planckian Jitter procedure applies a chromatic adaptation transform that simulates realistic variations in the illumination conditions. The data augmentation procedure is as follows:



1. we sample a new illuminant spectrum $\sigma_T(\lambda)$ from the distribution of a black-body radiator;
2. we transform the sampled spectrum $\sigma_T(\lambda)$ into its sRGB representation $\rho_T \in \mathbb{R}^3$;
3. we create a jittered image I' by reilluminating I with the sampled illuminant ρ_T ;
4. we introduce brightness and contrast variation, producing a Planckian-jittered image I'' .

A radiating black body at temperature T can be synthesized using Planck's Law [327]:

$$\sigma_T(\lambda) = \frac{2\pi hc^2}{\lambda^5 (e^{\frac{hc}{kT\lambda}} - 1)} \text{ W/m}^3, \quad (42)$$

where $c = 2.99792458 \times 10^8$ m/s is the speed of light, $h = 6.626176 \times 10^{-34}$ Js is Planck's constant, and $k = 1.380662 \times 10^{-23}$ J/K is Boltzmann's constant. We sampled T in the interval between 3,000K and 15,000K which is known to result in a set of illuminants that can be encountered in real life [326]. Then, we discretized wavelength λ in 10nm steps ($\Delta\lambda$) in the interval between 400nm and 700nm.

The conversion from spectrum into sRGB is obtained according to [328]:

1. we first map the spectrum into the corresponding XYZ stimuli, using the 1931 CIE standard observer color matching functions $c^{\{X,Y,Z\}}(\lambda)$, in order to bring the illuminant into a standard color space that represents a person with average eyesight;
2. we normalize this tristimulus by its Y component, convert it into the CIE 1976 L*a*b color space, and fix its L component to 50 in a 0-to-100 scale, allowing us to constrain the intensity of the represented illuminant in a controlled manner as a separate task; and
3. we then convert the resulting values to sRGB, applying a gamma correction and obtaining $\rho_T = \{R, G, B\}$; the resulting distribution of illuminants is visualized with the Angle-Retaining Chromaticity diagram.

All color space conversions assume a D65 reference white, which means that a neutral surface illuminated by average daylight conditions would appear achromatic. Once the new illuminant has been converted in sRGB, it is applied to the input image I by resorting to a Von-Kries-like transform [329] given by the following channel-wise scalar multiplication:

$$I'^{\{R,G,B\}} = I^{\{R,G,B\}} \cdot \{R,G,B\} / \{1,1,1\}, \quad (43)$$

where we assume the original scene illuminant to be white (1,1,1). Finally, brightness and contrast perturbations are introduced to simulate variations in the intensity of the scene illumination:

$$I'' = c_B \cdot c_C \cdot I' + (1 - c_C) \cdot \mu(c_B \cdot I'), \quad (44)$$

where $c_B = 0.8$ and $c_C = 0.8$ represent, respectively, brightness and contrast coefficients, and μ is a spatial average function.

5.5.2.2. Down Sampling GAN We have designed a Downsampling GAN (DS-GAN) to overcome the poor performance from well-known methods like bilinear interpolation or nearest neighbor to obtain SLR objects. DS-GAN is a generative adversarial network that learns to correctly degrade HR objects into SLR objects to increase the training set for object detection.

In this downsampling problem the aim is to estimate an SLR object from an input HR object with a downsampling factor r . The problem to solve is an unpaired problem where HR objects do not have a corresponding LR pair, but the network would have to learn the distribution of the features of the whole



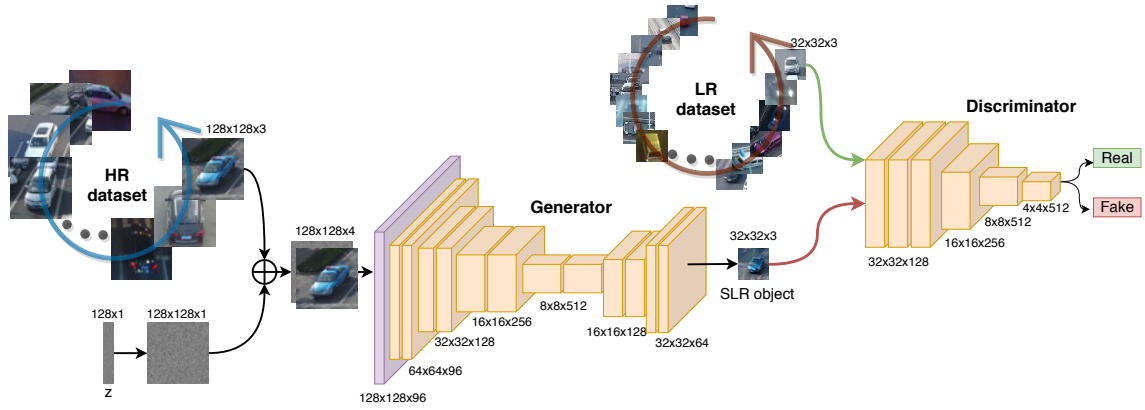


Figure 33. Downsampling Generative Adversarial Network (DS-GAN) architecture. The generator is trained with HR objects to synthesize small objects. A discriminator between real and fake small objects forces the generator to produce synthetic objects that are increasingly similar to real-world small objects.

LR subset while keeping similar visual appearance of the original HR object. For an image with C color channels, HR has size $W \times H \times C$ while both LR and SLR are described by $\frac{W}{r} \times \frac{H}{r} \times C$. So, for training the proposed GAN, two different image sets are required: (i) the *HR subset* composed of real large objects (HR objects) and (ii) the *LR subset* composed of real small objects (LR objects). Both the LR and HR subsets can be taken from the same dataset or from any additional one if more samples are needed.

Our DS-GAN architecture is shown in Fig. 33. The generator network (G) takes as input an HR image concatenated with a noise vector (z) and produces an SLR image $4 \times$ smaller than the input ($r=4$). For example, a 128×128 object will lead to a 32×32 object. The noise vector is randomly sampled from a normal distribution and it is attached to the input image. This allows to produce numerous SLR objects from a single HR object, thus modeling the fact that the HR image will be affected by multiple types of LR noise. Following the methodology of [330] we further define a discriminator network (D) which we optimize in an alternating manner along with the generator (G).

The generator is an encoder-decoder network —see Fig. 33— composed of six groups of residual blocks [331]. Each group has two same-dimension residual blocks with pre-activation and batch normalization as defined in [332]. To achieve a $4 \times$ downscaling, four $2 \times$ down-sample steps performed by pooling layers are placed at the end of each of the first four groups and two $2 \times$ up-sample steps performed by deconvolution layers at the end of each of the last two groups.

The discriminator —see Fig. 33— follows the same residual block structure (without batch normalization) followed by a fully connected layer and a sigmoid function. The discriminator comprises six residual blocks with two $2 \times$ down-sample steps. The details of the composition of both architectures are better shown in Fig. 33.

With this architecture, our goal is to train G to generate an SLR sample conditioned on an HR sample. To achieve this, the objective function chosen for the adversarial loss is the hinge loss [333]:

$$l_{adv}^D = \mathbb{E}_{s \sim \mathbb{P}_{LR}} [\min(0, 1 - D(s))] + \mathbb{E}_{\hat{s} \sim \mathbb{P}_G} [\min(0, 1 + D(\hat{s}))] \quad (45)$$

where \mathbb{P}_{LR} is the LR subset distribution and \mathbb{P}_G is the generator distribution to be learned through the alternative optimization. \mathbb{P}_G is defined by $\hat{s} = G(b, z) | b \in \mathbb{P}_{HR}$, where \mathbb{P}_{HR} is the HR subset. The general idea behind this formulation is that it allows to train G with the goal of fooling D , that is trained to distinguish SLR from LR images. With this approach our generator can learn to create SLR samples that are highly similar to real LR images, and thus difficult to classify by D .

Correspondingly, we train G by optimizing a loss function \mathcal{L} , defined as:

$$\mathcal{L} = l_{pixel} + \lambda l_{adv}^G, \quad (46)$$



Table 40. Evaluation on downstream tasks. Self-supervised training was performed on IMAGENET at (224×224) and testing performed on the downstream datasets resized to (224×224) .

AUGMENTATION	CUB-200	VEGFru	T1K+	USED	FLOWERS-102
Default Color Jitter (CJ)	54.52%	67.63%	71.44%	59.90%	93.16%
Planckian Jitter (PJ)	56.28%	65.84%	77.42%	60.03%	90.29%
LSC [CJ,PJ]	60.70%	74.73%	80.49%	64.07%	93.99%
LSC [CJ,CJ]	56.16%	70.59%	73.47%	61.07%	93.13%
LSC [CJ,CJ-]	53.14%	70.54%	78.32%	63.87%	93.47%

where l_{adv}^G is the adversarial loss, l_{pixel} is the L_2 pixel loss, and λ is a parameter that balances the weight of both components.

The adversarial loss l_{adv}^G is defined based on the probabilities of the discriminator as:

$$l_{adv}^G = -\mathbb{E}_{b \sim \mathbb{P}_{HR}} [D(G(b, z))], \quad (47)$$

where \mathbb{P}_{HR} is the HR subset and z is the noise vector. The adversarial loss is computed in an unpaired way, using the LR subset to make the SLR objects to be contaminated with real-world artefacts.

The l_{pixel} minimizes the L_2 distance between the input HR and the output SLR:

$$l_{pixel} = \frac{r^2}{WH} \sum_{i=1}^r \sum_{j=1}^r (AvgP(b)_{i,j} - G(b, z)_{i,j}) | b \in \mathbb{P}_{HR}, \quad (48)$$

where W and H denote the input HR size, r is the downsampling factor and $AvgP$ is an average pooling function that maps the HR input to the output $G(b, z)$ resolution. The l_{pixel} is computed in a paired way between the SLR object and the HR object downsampled to the output SLR resolution using an average pooling layer. This component aims to keep the appearance of the synthetic objects similar to the original HR objects.

In addition, to solve the stabilization of the discriminator training we normalize its weights by the spectral normalization technique [333].

5.5.3. Experimental results

5.5.3.1. Planckian Jitter for Color Augmentation Given the ablation study results, we performed the analysis of the proposed configurations on other downstream tasks using the backbone trained on higher resolution images (224×224) pixels. We report in Table 40 the results for: *Default Color Jitter*, *Planckian Jitter*, and latent space combinations.

Looking at the results, we see that the *Planckian Jitter* augmentation outperforms default color jitter on three datasets (CUB-200, T1K+, and USED). Comparing the results on FLOWERS-102 with those reported above at (32×32) pixels, we see that default color jitter actually obtains good results. We hypothesize that for high-resolution images the shape/texture information is very discriminative, and the additional color information yields little gain. Table 40 also contains results for latent space combination, which confirm that the two learned representations are complementary. Their combination yields gains of up to 9% on T1K+. As a sanity check we also include the latent space combination of two networks separately trained with Color Jitter. This provides a small gain on some datasets, but yields significantly inferior results than LSC.

5.5.3.2. DS-GAN For this experimentation, the SLR objects generated by the DS-GAN are compared with the LR objects —aiming for the greatest similarity— as well as with the resizing functions: linear interpolation, bicubic interpolation, nearest neighbours and Lanczos [334]. For this purpose, two metrics will be used to validate the quality of the synthetic objects generated by DS-GAN: the Fréchet Inception Distance (FID) [335] and object classification.





Figure 34. Real HR samples (left), and real LR samples (right).

FID is a popular metric for comparing the feature vectors calculated for real and generated images. The FID score summarizes how similar the two groups are in terms of statistics on computer vision features of the raw images calculated using a pre-trained image classification model. The lower the scores the greater the similarity of the two groups, meaning that they have more similar statistics, which is the purpose of our DS-GAN.

To support the above metrics, we also train an LR object classifier which differentiates between background (negative) and LR object (positive). We resort to this metric since it is closer to the objective of the full pipeline, i.e., the improvement of small object detection. On the one hand, the classifier is trained with the LR training set as positive examples and a background set as negative examples. On the other hand, the SLR set is used for positive examples and keeping the same backgrounds as negative examples. We have generated different SLR sets, one for each of the resizing functions, and another one for the DS-GAN. All the learned models are evaluated with the LR testing subset and different backgrounds. The higher the accuracy, the better the quality of the objects synthetically generated.

The DS-GAN generator architecture has a final stride $4\times$ smaller than the fixed size input image ($r=4$). Most of the popular datasets —MS COCO [336], UAVDT [337], VisDrone [338]— consider as small objects those smaller than 32×32 pixels. Therefore, we will train the DS-GAN to learn how to reduce HR objects to that range.

We validate our data augmentation for small object detection approach with the car category on the UAVDT dataset [337]. This dataset was selected because the whole set of objects are vehicles, which allows us to isolate the results for a specific category, and also provides a large number of small instances in the testing set. Quantitatively, UAVDT comprises 23,829 frames of training data and 16,580 frames of test data, belonging to 30 and 20 videos of $\approx 1,024 \times 540$ resolution, respectively. The videos are recorded with an UAV platform over different urban areas. UAVDT includes a total of 394,633 car instances for training, where 107,091 are considered within the *small* subset (52.38%), and a total of 361,055 car instances for testing, where 274,438 are considered within the *small* subset (76.01%).

Considering that the camera motion in UAVDT slightly modifies the appearance of consecutive frames, in this section, only 10% of the video frames are selected for training to avoid overfitting. The details on the datasets for evaluating DS-GAN are given below:

- **Real HR subset:** To obtain the HR objects we select those objects from 48×48 to 128×128 pixels, and we add context to have an area of 128×128 pixels in objects with a smaller area. These conditions result in a total number of 517 HR objects in the UAVDT dataset. To have a larger number, we also select the cars in the VisDrone dataset with the same restrictions. VisDrone is a dataset with a very similar nature to that of UAVDT, i.e., high-resolution videos recorded with UAVs. The total number of HR objects is 5,731 after joining both datasets. Some HR examples are shown in Fig. 34(left).



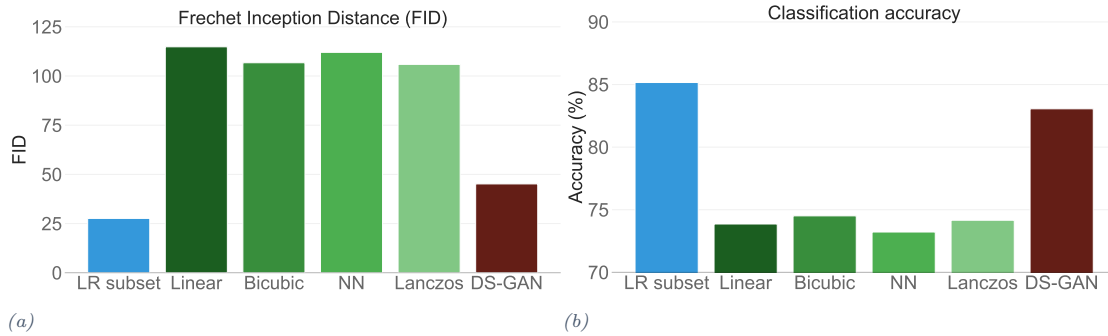


Figure 35. FID (a) and classification accuracy (b) for different subsampling methods on the LR testing subset of UAVDT.

- Real LR training subset:** To obtain the LR objects we select those objects under 32×32 with sufficient context to cover an area of 32×32 pixels. This results in a total of 18,901 objects coming from the UAVDT training set — these objects are a part of the UAVDT *small* subset, where redundant instances have been discarded. However, in order to simulate a small object scarcity scenario, the LR subset will only consist of approximately 25% of the videos of the UAVDT dataset. The selected videos include a total of 5,226 LR objects. Some LR examples are shown in Fig. 34(right).
- Real LR testing subset:** To evaluate the performance DS-GAN and the pipeline we use the 274,438 small objects coming from the UAVDT testing set with sufficient context to cover an area of 32×32 pixels.

For training the DS-GAN, we augment the training data by applying random image flipping to increase diversity. We provide a different noise vector (z) sampled from a normal distribution to each HR object in order to simulate a large variety of image degradation types. DS-GAN is trained during 1,000 epochs with an update ratio 1:1 between the discriminator and the generator, and it is optimized with Adam [339] with parameters $\beta_1 = 0$ and $\beta_2 = 0.9$. We set the base learning rate to $1e-4$, decreasing it twice during the training phase by a factor of 10. We use $\lambda = 0.01$ in Eq. 46 to balance the relevance of the two components in the image generation process — l_{adv}^G is two orders of magnitude higher than l_{pixel} . Thus, the adversarial loss helps to learn to contaminate the HR input with noise and artefacts coming from the LR subset, and the pixel loss helps to preserve the visual features from the original input.

Fig. 35a and Fig. 35b show the experimental results to evaluate the quality of the synthetic objects generated by DS-GAN over the LR testing subset of UAVDT. Our approach is compared to the main re-scaling functions: linear and bicubic interpolation, nearest neighbors and Lanczos [334]. The reference values are obtained by the models trained on the LR training subset (blue bars).

The FID value in Fig. 35a is measured using the final average pooling features in Inception-v3 [340]. The reference value of the LR training objects compared with the LR testing subset is 27.62. The graph of Fig. 35a shows how the small objects obtained by any re-scaling function lead to values above 100, which is a poor performance relative to the reference value. The FID value of the SLR objects generated by DS-GAN for the LR test objects is 45.15. This FID value shows how the objects generated by the DS-GAN have better quality than those obtained by a simple re-scaling function, i.e., are more similar to the real ones.

To complement the FID distance, we have trained a classification network (ResNet-50 pre-trained on ImageNet [341]) with each of the defined subsets and tested them with the LR testing subset. Fig. 35b shows, again, how the SLR object generated by DS-GAN provides a considerably higher accuracy (83.06%) than the re-scaling functions ($\approx 74\%$), and are very close to the reference accuracy obtained by the LR training subset (85.16%).

These results validate the conclusions reached in [342, 343], since re-scaling functions introduce artefacts that make the output object differ considerably from real-world objects. Even though these differences are

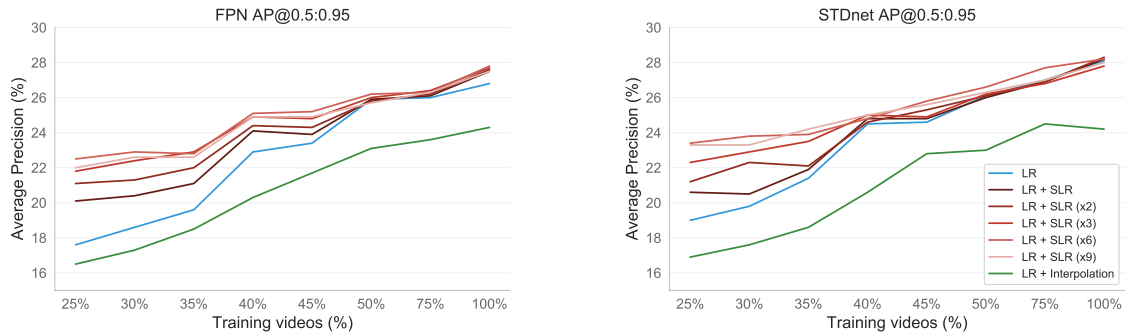


Figure 36. $AP_s^{@[.5,.95]}$ for small object detection in UAVDT for different percentage of training videos with the FPN and STDnet architectures.

Data augmentation	FPN		STDnet		CenterNet	
	$AP_s^{@.5}$	$AP_s^{@[.5,.95]}$	$AP_s^{@.5}$	$AP_s^{@[.5,.95]}$	$AP_s^{@.5}$	$AP_s^{@[.5,.95]}$
LR	39.0	17.6	41.2	19.0	51.9	22.6
LR + Interp.	38.1	16.5	38.8	16.9	46.9	18.4
LR + SLR	46.3	20.1	48.1	20.6	60.6	26.1
LR + SLR×6	50.9	22.5	51.5	23.4	63.5	26.8

Table 41. Comparison of several data augmentation approaches for small object detection with FPN, STDnet and CenterNet networks on the small object testing subset of UAVDT. The training phase was conducted by simulating a low instance small object scenario —25% of the UAVDT training videos.

not visually appreciable, they are identified by the layers within the CNNs (Inception-v3 and ResNet-50). DS-GAN significantly improves this issue by learning the different artefacts found in real-world objects.

In order to evaluate our pipeline for data augmentation for small object detection we use the UAVDT detection metrics that were originally defined by the MS COCO dataset, i.e., $AP^{@.5}$ and $AP^{@[.5,.95]}$. STDnet [344], FPN [345] and CenterNet [346] are adopted as the baseline detection networks.

The implementation details for DS-GAN are those defined in the previous section. The other component that requires training is DeepFill for image inpainting. In this case, the default parameters [347] are used to train the model on the UAVDT dataset. We have set $\tau=40$ as the frame search range for the position selector. The rest of the components of our pipeline are also configured with their default values.

We detail the results obtained by STDnet [344], FPN [345] and CenterNet [346] on the UAVDT testing set for small objects. The training phase for all the models was conducted from the same 25% of the videos as in the DS-GAN training, in order to simulate a scenario with a low number of LR objects, up to the whole UAVDT training set. Here, the *LR* label means that no data augmentation has been applied for training, so the images come directly from the standard UAVDT training set. The *LR + Interp.* and *LR + SLR* labels mean the same images with real objects as in *LR*, and also duplicating those images replacing the real LR objects with synthetic objects ones generated with the pipeline using bilinear interpolation and DS-GAN, respectively. So that, in *LR + Interp.* and *LR + SLR*, the number of synthetic objects is equal to the number of LR objects. Notice that *LR + Interp.* is a more elaborated solution than [348], as it is the proposed pipeline, but replacing DS-GAN by bilinear interpolation. Finally, the *LR + SLR×n* labels mean that the number of SLR objects is n times higher than the number of LR objects.



5.5.4. Relevance to AI4Media use cases and media industry applications

The proposed approaches can be used as augmentation procedures for any self-supervised framework and can be relevant in UC3 and UC7 for tasks such as visual indexing and search and visual concepts classification.

5.5.5. Relevant Publications

- Zini, Simone, Alex Gomez-Villa, Marco Buzzelli, Bartłomiej Twardowski, Andrew D. Bagdanov, and Joost van de Weijer. "Planckian Jitter: countering the color-crippling effects of color jitter on self-supervised training." International Conference on Learning Representations (2023). <https://dx.doi.org/10.48550/arXiv.2202.07993>
- Bosquet, Brais, Daniel Cores, Lorenzo Seidenari, Víctor M. Brea, Manuel Mucientes, and Alberto Del Bimbo. "A full data augmentation pipeline for small object detection based on generative adversarial networks." Pattern Recognition (2023) <https://doi.org/10.1016/j.patcog.2022.108998>

5.5.6. Relevant software/datasets/other outcomes

Source code: <https://github.com/TheZino/PlanckianJitter>

5.6. MaskCon: Masked Contrastive Learning for Coarse-Labeled Dataset

Contributing partner: QMUL

5.6.1. Introduction and methodology

Supervised learning with deep neural networks has achieved great success in various computer vision tasks such as image classification, action detection and object localization. However, the success of supervised learning relies on large-scale and high-quality human-annotated datasets, whose annotations are time-consuming and labour-intensive to produce. To avoid such reliance, various learning frameworks have been proposed and investigated.

In this work, we consider an under-explored problem setting aiming at reducing the annotation effort – learning fine-grained representations with a coarsely-labeled dataset. Specifically, we learn with a dataset that is fully labeled, albeit at a coarser granularity than we are interested in (i.e., that of the test set).

Differently than previous works, instead of using self-supervised contrastive learning as an auxiliary task, we propose a novel learning scheme, namely **Masked Contrastive Learning (MaskCon)**. Our method aims to learn by considering inter-sample relations of each sample with other samples in the dataset (Figure 37). Specifically, we always consider the relation to oneself as confidently positive. To estimate the relations to other samples, we derive soft labels by contrasting an augmented view of the sample in question with other samples, and further improve it by utilizing the *mask* generated based on the coarse labels.

5.6.1.1. Contrastive learning We first briefly introduce essential concepts about contrastive learning. Unlike the common supervised learning model, we can perform contrastive learning based on the inter-sample relations $\mathbf{Z} = \{z_i \in (0, 1)\}_{i=1}^N$, with each entry z_{ij} depicting the inter-sample relation between \mathbf{x}_i and \mathbf{x}_j . Intuitively, $z_{ij} = 1$ means that sample \mathbf{x}_i and \mathbf{x}_j generate a strong positive pair. Since each sample may form multiple positive sample pairs, for brevity, we abuse the notation here with \mathbf{Z} denoting also the sample-wise normalized inter-sample relations. To learn such inter-sample relations, instead of a parametric classifier, the f encoder is usually followed by a projector h , which is often implemented as an MLP and learned by regularizing the inter-sample relations \mathbf{Z} (eq. 50).



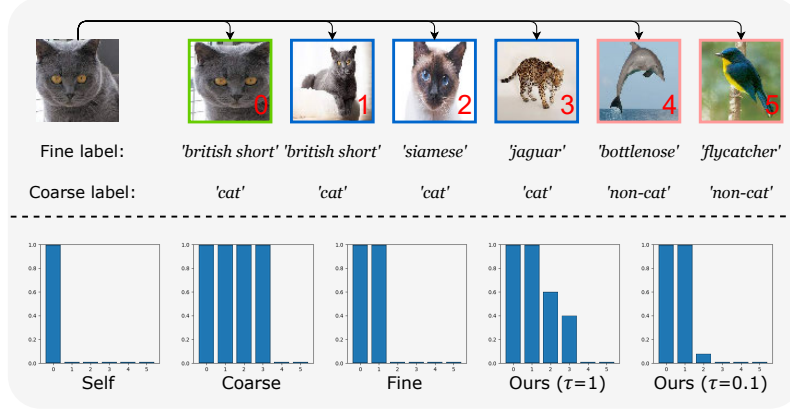


Figure 37. Contrastive learning sample relations using MaskCon (ours) and other learning paradigms when only coarse labels are available. MaskCon are closer to the fine ones.

More specifically, let us denote by $\mathbf{h}_i \triangleq h(f(\mathbf{x}_i))$ the *projection*. We first calculate the cosine similarity \mathbf{d}_i between a sample \mathbf{x}_i and the dataset $\mathbf{H} = \{\mathbf{h}_n\}_{n=1}^N$:

$$\mathbf{d}_i = [\cos(\mathbf{h}_i, \mathbf{h}_1), \cos(\mathbf{h}_i, \mathbf{h}_2), \dots, \cos(\mathbf{h}_i, \mathbf{h}_N)], \quad (49)$$

Let us further define $\mathbf{q}_i \triangleq \text{softmax}(\mathbf{d}_i / \tau_0)$, where τ_0 is the temperature hyperparameter. Then the following empirical risk will be optimized:

$$R(f, h) = \sum_{i=1}^N L_{con}(\mathbf{x}_i, \mathbf{z}_i; f, h), \quad (50)$$

where the contrastive loss L_{con} is defined as follows:

$$L_{con}(\mathbf{x}_i, \mathbf{z}_i; f, h) = -\sum_{n=1}^N z_i^n \log q_i^n. \quad (51)$$

5.6.1.2. Methodology To better learn with coarse labels, we introduce a novel contrastive learning method, namely **Masked Contrastive learning (MaskCon)**, within the framework of contrastive learning that utilizes inter-sample relations directly.

More specifically, for sample \mathbf{x}_i , we estimate its inter-sample relations \mathbf{z}'_i to other samples utilizing the key view projection \mathbf{h}_k and the whole dataset $\{\mathbf{h}_1, \dots, \mathbf{h}_N\}$ excluding itself (since it will always be considered as a trustworthy positive), as below:

$$z'_{ij} = \frac{\mathbb{1}(\mathbf{y}_j = \mathbf{y}_i) \cdot \exp(d'_{ij} / \tau)}{\sum_{n=1, n \neq i}^N \mathbb{1}(\mathbf{y}_n = \mathbf{y}_i) \cdot \exp(d'_{in} / \tau)}, i \neq j, \quad (52)$$

where the similarity \mathbf{d}'_i is given by

$$\mathbf{d}'_i = [\cos(\mathbf{h}_i^k, \mathbf{h}_1), \dots, \cos(\mathbf{h}_i^k, \mathbf{h}_{i-1}), \cos(\mathbf{h}_i^k, \mathbf{h}_{i+1}), \dots, \cos(\mathbf{h}_i^k, \mathbf{h}_N)]. \quad (53)$$

Please note the use of the mask ($\mathbb{1}(\mathbf{y}_j = \mathbf{y}_i)$) that excludes from the softmax that estimates inter-sample relationships, the samples j that have a different coarse label with the sample i (and sets their z'_{ij}



to 0). While it is risky to consider all samples from the same coarse class as positive, we can confidently identify those samples that do not have the same coarse class as negative. This reduces the noise in z'_i . Finally, we re-scale the z'_i with its maximum

$$z'_{ij} = z'_{ij} / \max(z'_i), \quad (54)$$

to make the closest neighbour as positive as the sample itself and arrive at:

$$z'_{ij}{}^{mask} = \begin{cases} 1, & \text{if } i=j \\ z'_{ij}, & \text{if } i \neq j \end{cases} \quad (55)$$

Compared to \mathbf{Z}^{supcon} , we thus reweight the samples of the same coarse label according to the similarities in the feature space.

We denote the masked contrastive loss as $L_{maskcon}$ and, similarly to Grafit [349] and CoIns [350], we also consider a weighted loss as the final objective:

$$L = wL_{maskcon} + (1-w)L_{selfcon} \quad (56)$$

5.6.2. Experimental results

We compare our method with two competing methods: **Grafit** and **CoIns**. For a fair comparison, we exhaust the weight w choices for both methods and report the best achievable results in all experiments. Note that when $w=0$, Grafit and CoIns degenerate to self-supervised contrastive learning denoted as **SelfCon**; Conversely, when $w=1$, Grafit degenerates to supervised contrastive learning [351] denoted as **SupCon**, while CoIns degenerates to conventional supervised cross-entropy learning denoted as **SupCE**. For reference, we also show the results when training with fine labels – this is denoted as **SupFINE**.

5.6.2.1. Evaluation protocol To evaluate the different methods on the test set with fine labels, we use the recall@K [352] metric widely used in the image retrieval task. Each test image first retrieves top-K nearest neighbours from the test set and receives 1 if there exists at least one image from the same fine class among the top-K nearest neighbours, otherwise 0. Recall@K averages this score over all the test images.

5.6.2.2. Experiments on CIFAR100 dataset The common CIFAR100 dataset has 20 classes of coarse labels in addition to the 100 classes of fine labels, with each coarse class containing five fine-grained classes (500 samples). The results in Table 42 show that our method achieves significant improvements over the SOTAs. In particular, it improves the top-1 retrieval precision from 47.25% to 65.52%, approaching the results by the model learned with fine labels (71.13%).

5.6.3. Conclusion

In this work, we propose a **Masked Contrastive learning framework (MaskCon)** for learning fine-grained information with coarse-labeled datasets. On the basis of two baseline methods, we utilize coarse labels and the instance discrimination task to better estimate inter-sample relations. We show theoretically that our method can reduce the optimization error bound. Extensive experiments with various hyperparameter settings on multiple benchmarks, including the CIFAR datasets and the more challenging fine-grained classification datasets show that our method achieves consistent and large improvement over the baselines.

5.6.4. Relevance to AI4Media use cases and media industry applications

MaskCon represents a novel method for learning with coarse labels. It can be relevant in UC3 for tasks such as visual indexing and search and visual concept classification.





Table 42. Results on CIFAR100 dataset.

Method	Recall@1	Recall@2	Recall@5	Recall@10
SelfCon	40.50	51.83	66.23	76.66
Grafit	60.57	71.13	82.32	89.21
SupCon	58.65	70.04	82.18	89.09
CoIns	60.10	70.89	83.14	89.52
SupCE	47.25	61.24	77.78	87.01
SupFINE	71.13	80.03	87.61	91.59
MaskCon (Ours)	65.52 (18.17↑)	74.46	83.64	89.25

5.6.5. Relevant publications

- Chen Feng, Ioannis Patras. "MaskCon: Masked Contrastive Learning for Coarse-Labeled Dataset." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023. Zenodo record: <https://zenodo.org/records/8014242>

5.6.6. Relevant software/datasets/other outcomes

- The Pytorch implementation can be found in https://github.com/MrChenFeng/MaskCon_CVPR2023

5.7. Self-Supervised Video Similarity Learning

Contributing partners: QMUL, CERTH

5.7.1. Introduction and methodology

Self-supervised learning is a popular approach, especially for learning representations that are amenable to transfer to different tasks [353, 354, 355, 356, 357]. SSL allows to scale-up the dataset size by not relying on manual labeling and is known to obtain representations with high transferability. The commonly studied setup is to consider SSL for pre-training on a proxy task and then perform supervised fine-tuning on different target tasks [353, 354, 355]. In this work, we rather perform SSL and directly use the model on video similarity-related tasks.

In this work, we adopt the ViSiL [358] architecture for video similarity, which needs labeled video datasets for its development in prior works [358, 359], but we train it in a self-supervised way and argue that instance-discrimination through augmentations is well suited for all the aforementioned tasks. To pronounce the synergy, we develop an appropriate composition of video augmentations and propose a model-tailored loss combined with a standard SSL loss. By eliminating the need for video annotations, we are able to train on large video datasets and achieve state-of-the-art results on all target retrieval and detection tasks. Evaluation is performed on three standard benchmarks, namely, VCDB [360], FIVR [361], and EVVE [362].

Our aim is to learn a video similarity function $s: \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$, where \mathcal{V} is the space of all videos. The goal is for two videos to have high similarity if they are relevant, and low otherwise. The definition of relevance is task-dependent. In our experiments, we consider several evaluation tasks, where relevance ranges from video copies to videos of the same physical event. Nevertheless, we perform training in a single universal way without video labels for supervision. We perform training with self-supervision in





the spirit of instance-discrimination, i.e., two augmented videos originating from the same original video are considered as positive to each other, or negative otherwise. In some parts, we follow the work of Pizzi *et al.* [363], who perform SSL for image copy detection.

5.7.1.1. Similarity network We adopt the ViSiL variant proposed in DnS [359], namely the fine-grained attention student, as our similarity network architecture. It consists of a representation network, a hand-crafted spatial matching function, a learnable temporal matching function, and a final hand-crafted matching function that estimates the final video-level similarity.

The representation network $f_{\theta,\phi}: \mathcal{V} \rightarrow \mathbb{R}^{T \times R \times D}$ maps an input video to a D -dimensional vector per region, for R regions per frame, for T frames, where R and T vary according to the frames' size and video length, respectively. This network consists of a pre-trained backbone network and has a parameter set θ that is fixed in this work, similar to the prior ones [364, 361, 365]. The learnable part corresponds to the parameter set ϕ , a dot-attention scheme [366] that is applied to weigh region vectors based on their saliency.

Given two input videos v and u and their corresponding representations, the hand-crafted spatial matching is performed by the function $g: \mathbb{R}^{T_v \times R_v \times D} \times \mathbb{R}^{T_u \times R_u \times D} \rightarrow \mathbb{R}^{T_v \times T_u}$, that takes as input two video representations and estimates the temporal similarity matrix. It computes the $R_v \times R_u$ spatial similarity matrix for all frame pairs and then applies Chamfer similarity on each of them to estimate the frame-to-frame similarity.

The temporal matching is performed by function $h_\psi: \mathbb{R}^{T_v \times T_u} \rightarrow \mathbb{R}^{T'_v \times T'_u}$. This is a four-layer CNN that learns to capture temporal patterns in the input similarity matrices. It outputs a filtered temporal similarity matrix. It holds that $T_v = 4T'_v$, and similarly for u , due to the CNN design that contains strided max pooling operations. The parameters of the CNN, denoted by ψ , are learnable.

Chamfer similarity is applied and denoted by the function $m: \mathbb{R}^{T'_v \times T'_u} \rightarrow \mathbb{R}$, taking as input the filtered temporal similarity matrix and estimating the final video-level similarity, i.e., the scalar similarity between the two videos.

To summarize, similarity $s(v,u)$, for the video pair consisting of videos v and u , is equivalent to $s(v,u) = m(h_\psi(g(f_{\theta,\phi}(v), f_{\theta,\phi}(u))))$, and the goal in this work is to learn ϕ and ψ with self-supervision on videos, while θ remains fixed and is obtained from supervised pre-training on ImageNet. The reader is referred to the original ViSiL work [358] for additional details.

5.7.1.2. Weak/strong video augmentations We apply two sets of augmentations to generate two corresponding versions of a training video, i.e., one weakly and one strongly augmented version. Formally, given an original video v , the output of an augmentation function A is a video tensor $\tilde{v} = A(v) \in \mathbb{R}^{T_B \times H_B \times W_B \times 3}$, where T_B , H_B , and W_B correspond to the number of frames, height, and width of the video in the batch, respectively.

Weak augmentations consist of conventional geometric transformations (i.e., resized crop and horizontal flip), applied globally on the entire video, and temporal cropping to select T_B consecutive frames.

Strong augmentations consist of the weak augmentations and several other transformations grouped into the following four categories:

Global transformations are frame transformations applied to all frames in a consistent way. We use RandAugment [367], an automatic augmentation strategy that includes different geometric and photometric image transformations and requires two hyperparameters, namely N_{RAug} and M_{RAug} . These correspond to the number of randomly-applied consecutive transformations and their magnitude value that determines their severity, respectively.

Frame transformations are applied independently per frame. We use overlay and blurring transformation¹⁵. Following advanced augmentations from prior work [363], we add random emojis and text,

¹⁵The RandAugment implementation we use does not contain blurring operations. Hence, global transformations do not blur videos.





each with probability $p_{overlay}$, and blur frames with probability p_{blur} . We opt for these operations to emulate common video copy transformations.

Temporal transformations act only on the temporal dimension and include five operations, with one applied per video. Following [358], we use fast forward, slow motion, reverse play, and frame pause, where a single frame is duplicated several times consecutively. In addition, we propose Temporal Shuffle-Dropout (TSD) to alter the global temporal structure but preserve the local one. The video is first split into short clips, each of them with length randomly chosen in $[4, \dots, T_B/2]$. In the shuffling phase, applied with probability p_{shuf} , the clip order is shuffled. In the dropout phase, a clip is dropped with probability p_{drop} , where it is either discarded or filled with empty frames or Gaussian noise with probability p_{cont} .

Video-in-video randomly mixes two strongly augmented videos, the *host* and the *donor*, in the same batch. The donor video is randomly spatially down-sampled with a factor λ_{viv} and is overlaid in a random location within the host video. Each strongly augmented video is chosen as donor with probability p_{viv} . Then, a host video is randomly chosen, while the mixed output replaces the donor video. This process requires properly adjusting the instance-discrimination labels since the generated video is the outcome of two others. Video-in-video transformation is very common in real-life video cases.

5.7.1.3. Loss on video similarity A random set of N videos, where each video is augmented once with the weak and once with the strong augmentations, forms a training batch of size $B = 2N$ denoted by $\mathcal{B} = [v_1, \dots, v_{2N}]$. We compute the similarity matrix $S \in [0, 1]^{B \times B}$, with elements $S_{i,j} = s(v_i, v_j)$, comprising all pairwise video similarities within the batch. Each row of S consists of the self-similarity on the diagonal, one positive-pair similarity, and $B-2$ negative-pair similarities¹⁶. Note that S is not symmetric and that the diagonal elements are not equal to 1 because of h_ψ . For the i -th row of the similarity matrix, let $p(i)$ be the set of column indices of the positive pairs. Additionally for the i -th row, let $n(i)$ be the set of column indices of the negative pairs.

The total loss is a combination of two losses that optimize different parts of S : (i) the widely used InfoNCE [368] loss estimated per row excluding the self-similarity value, and (ii) a loss that maximizes the self-similarity, i.e., main diagonal, and minimizes the similarity with the hardest negative, i.e., the negative with the highest similarity, for each video in the batch.

InfoNCE loss is estimated for each positive pair by

$$\mathcal{L}_{nce}(i,j) = -\log \frac{\exp(S_{i,j}/\tau)}{\exp(S_{i,j}/\tau) + \sum_{k \notin p(i) \cup i} \exp(S_{i,k}/\tau)}, \quad (57)$$

where τ is a temperature hyper-parameter and (i,j) is a positive pair. The final InfoNCE loss is given by the average over all positive pairs as

$$\mathcal{L}_{nce} = 1/P \sum_i \sum_{j \in p(i)} \mathcal{L}_{nce}(i,j), \quad (58)$$

where P is the total number of positive pairs in the batch.

Self-similarity – hardest negative loss: Since the self-similarity is not equal to 1 by design, we add a loss term that is trying to push it to high values. Together with that, an additional term pushes the hardest negative of each row to have small similarity. For the i -th row, this loss is given by

$$\mathcal{L}_{sshn}(i) = \underbrace{-\log(S_{i,i})}_{self-sim} - \log \underbrace{\max_{j \notin p(i) \cup i} (1 - S_{i,j})}_{hard-negative sim}, \quad (59)$$

¹⁶This is the case where video-in-video augmentation is not used; otherwise, there can be more (less) positives (negatives).



Approach	Lab.	Retrieval					Detection				
		VCDB ($\mathcal{C}+\mathcal{D}$)	FIVR-200K			EVVE	VCDB ($\mathcal{C}+\mathcal{D}$)	FIVR-200K			EVVE
			DSVR	CSVR	ISVR			DSVD	CSVD	ISVD	
DML [371]	✓	-	52.8	51.4	44.0	61.1	-	39.0	36.5	30.0	75.5
LAMV [364]	✗	78.6	61.9	58.7	47.9	62.0	62.0	55.4	50.0	38.8	<u>80.6</u>
TCA _f [372]	✓	-	87.7	83.0	70.3	-	-	-	-	-	-
VRL _f [373]	✗	-	90.0	85.8	70.9	-	-	-	-	-	-
ViSiL _f [358]	✗	82.0	89.0	84.8	72.1	62.7	40.9	66.9	59.5	45.9	74.6
ViSiL _v [358]	✓	-	89.9	85.4	72.3	65.8	-	75.8	69.0	53.0	79.1
DnS [359]	✓	87.9	92.1	87.5	<u>74.1</u>	65.1	74.0	79.7	69.5	54.2	74.3
S ² VS (Ours)	✗	-	92.7	87.9	74.6	67.2	-	<u>85.7</u>	<u>76.9</u>	<u>62.8</u>	80.7
S ² VS (Ours)	✗	87.9	<u>92.5</u>	<u>87.8</u>	73.9	<u>65.9</u>	<u>73.0</u>	89.3	80.2	64.9	78.9

Table 4.3. State-of-the-art comparison via retrieval mAP (%) and detection μ AP (%) on three evaluation datasets. **Bold** and underline indicate the best and second best approach, respectively. Missing values are either due to unavailability or unfair comparison due to leak of evaluation data during training.

and the total loss is given by the average over rows as $\mathcal{L}_{\text{sshn}} = 1/B \sum_i \mathcal{L}_{\text{sshn}}(i)$. Note that the hard-negative term resembles entropy maximization through the Kozachenko-Leononenko estimator and a consequent spreading of elements in the representation space [369]. Differently to them, we perform this directly on pairwise similarities and not on distances over a vector space.

To this end, we optimize a weighted sum of the losses presented above, as follows

$$\mathcal{L} = \mathcal{L}_{\text{ncc}} + \lambda \mathcal{L}_{\text{sshn}}, \quad (60)$$

where λ is a hyperparameter that tunes the impact of $\mathcal{L}_{\text{sshn}}$.

5.7.2. Experimental results

5.7.2.1. Datasets DnS-100K [359] consists of 115,792 unlabeled videos. It is used for knowledge distillation in the original work, but we use it as a training set.

VCSL [370] is originally created for video copy localization. It contains 9,207 videos with more than 281K copied segments split into training, validation, and test set. Due to the unavailability of several videos, we managed to collect only 8,384 videos. We use this dataset to train our model in a supervised way, only to provide an indicative comparison with the proposed SSL approach.

VCDB [360] is created for partial video copy detection. The core dataset (\mathcal{C}) contains 528 videos from 28 discrete sets with over 9,000 copied segments. It also contains a set \mathcal{D} of 100,000 distractor videos. We use this dataset for evaluation for detection and retrieval of video copies, considering as related the videos that share at least one copied segment. Moreover, we use the distractor set as an alternative unlabeled training set. We use VCDB, VCDB (\mathcal{D}), or VCDB ($\mathcal{C}+\mathcal{D}$) to indicate that only set \mathcal{C} , only set \mathcal{D} , or both sets are used, respectively.

FIVR-200K [361] is used as a benchmark for fine-grained incident video retrieval. It consists of 225,960 videos and 100 queries. FIVR-200K includes three different subtasks: a) Duplicate Scene Video Retrieval (DSVR), b) Complementary Scene Video Retrieval (CSVR), and c) Incident Scene Video Retrieval (ISVR). In this work, we use the same subsets to evaluate for the corresponding detection tasks, denoted by DSVD, CSVD, and ISVD. For quick comparisons, we also use FIVR-5K [358], a subset of FIVR-200K. We use it in our ablations, denoted by FIVR, where the average performance of the three subtasks is reported.

EVVE [362] is a dataset for video retrieval. It consists of 620 queries and 2,375 database videos. Due to the unavailability of several videos, we use only 504 queries and 1906 database videos [358], which is roughly $\approx 80\%$ of the initial dataset. All reported methods are evaluated on this subset.

In summary, we train on DnS-100K, or VCDB(\mathcal{D}), and evaluate on VCDB for video copies, on FIVR for video copies, and incidents, and on EVVE for video copies, incidents, and events.



5.7.2.2. Experiments We evaluate the performance of the proposed approach on different retrieval and detection tasks related to video similarity, compare its performance to the state-of-the-art methods.

We compare the proposed S²VS method in Table 43 with the following approaches. **DML** [371] extracts a video embedding based on a network trained with supervised deep learning. **LAMV** [364] trains a video representation using a generated dataset while relying on kernel-based temporal alignment. **TCA_f** [372] is a transformer-based architecture trained with supervised contrastive learning. **VRL** [373] is a CNN and transformer-based network trained end-to-end with no labeled data. **ViSiL_f** [358] is a baseline without any training on videos that corresponds to the frame-to-frame similarity part of ViSiL combined with Chamfer similarity. **ViSiL_v** is the full similarity model trained with supervision. **DnS** [359] is a ViSiL-based student network trained with distillation from a teacher trained with supervision; we compare with the best-performing fine-grained attention student **S_A^f**. For TCA and VRL, the reported results are taken from the original works. For the remaining approaches, we run the provided pretrained networks, and following **DnS** [359], we implement LAMV and DML with the same features provided in the official repository¹⁷.

5.7.3. Conclusion

In this work, we proposed a self-supervised learning approach for training video similarity networks. Eliminating the need for labels allows us to train on large-scale video corpora, which, together with a diverse set of video augmentations, form the key ingredient for achieving top performance. The obtained single model has been evaluated on several target retrieval and detection tasks. It manages to perform on par or outperform existing models that exploit labeled datasets, especially for detection due to better similarity calibration across queries.

5.7.4. Relevance to AI4Media use cases and media industry applications

S²VS is a self-supervised framework video similarity network training, and can be relevant in UC3 and UC7 for tasks such as visual indexing and search and visual concepts classification.

5.7.5. Relevant publications

- Kordopatis-Zilos, G., Tzilepis, C., Kompatsiaris, I., Patras, I., & Papadopoulos, S. (2023). Self-supervised video similarity learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 4756-4766). Zenodo record: <https://zenodo.org/record/8314217>

5.7.6. Relevant software/datasets/other outcomes

Code is available at <https://github.com/gkordo/s2vs>.

5.8. Efficient Data Utilization for enhanced DNN Inference Reliability

Contributing partner: AUTH

5.8.1. Introduction

In this work, we introduce an efficient data utilization strategy for enhanced DNN inference reliability. We propose an innovative approach that evaluates the performance of a DNN in handling large datasets without the need to generate inferences for the entire dataset. Our primary goal is to ascertain whether a

¹⁷<https://github.com/mever-team/distill-and-select>





dataset has been previously encountered by a DNN and, thus, if the DNN can produce reliable and accurate inferences. To achieve this, we aim to determine the optimal data quantity required by DNNs to ensure that their inferences are both accurate and reliable, without the necessity of utilizing the entire dataset. This approach significantly reduces the need for extensive testing across large datasets, thereby decreasing computational complexity typically associated with large-scale data processing. Additionally, it focuses on assessing how well a DNN has integrated knowledge from its training data and its capability to apply this knowledge effectively to new, unseen data. We advocate that this strategy not only aims to reduce the computational complexity of the DNNs but also ensures that the networks maintain high reliability and accuracy in their predictions, thus addressing some of the key challenges in Big Data analysis.

5.8.2. Methodology

In Big data environments, it is critical to establish the minimal amount of test data necessary for DNN classifiers to reliably predict outcomes across an entire dataset without the need to individually analyze each data point. We introduce a novel method for statistically analyzing DNN inferences, aiming at reducing computational complexity while ensuring the reliability of these inferences.

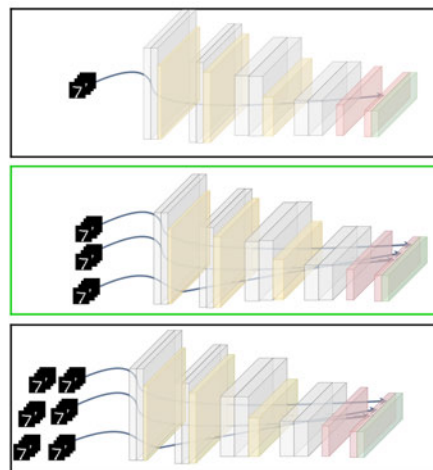


Figure 38. Through our experiments, we determine the minimum amount of data required to provide reliable inferences while maintaining high performance. Our method demonstrates that achieving optimal inference accuracy in Big data environments does not require processing the entire dataset; instead, it efficiently delivers reliable inferences using the fewest necessary data points.

To establish the necessary sample size for accurate DNN inferences in the classification problems we explore, individual tests of the DNN are carried out across a range of dataset cardinalities. These range from a single data sample to larger quantities (e.g., up to two thousand data samples, N , in our simulation experiments).

For each one of these cardinalities in the range $[1, \dots, 2,000]$, the networks are tested repeatedly multiple times so that the mean value and the variation of the evaluation metric can safely be estimated. The selection of data cardinalities was determined through extensive experimentation, revealing that further expansion of the dataset is unnecessary. In the scenarios examined, where the focus is on image classification, classification accuracy is utilized as the evaluation metric.

To verify the normal distribution of the evaluation metrics, essential for applying statistical methods accurately, the Shapiro-Wilk normality test [374] is employed on a random sample of 500 values. The normal distribution of our data adheres to the empirical rule showcasing the characteristics of a normal distribution





and its relevance to our statistical analyses. Using the 95% confidence interval, we determined the sufficient number of testing samples needed for reliable DNN inferences. A confidence interval is a statistical range derived from a sample of data that is utilized to estimate a population parameter. The interval is accompanied by a confidence level, representing the degree of confidence in the interval containing the true population parameter. The most commonly used confidence intervals include those with confidence levels of 90%, 95%, and 99% with critical values (Z-scores) of 1.645, 1.96, and 2.576 respectively. Specifically, a 95% confidence interval indicates that if we were to repeatedly sample from a population and calculate a 95% confidence interval for each sample, approximately 95 out of 100 intervals would contain the true mean value (μ).

When aiming to generate a 95% confidence interval estimate for an unknown population mean, it implies that there is a 95% probability that the confidence interval will encompass the true population mean. The 95% confidence interval for the population n mean can be expressed as:

$$95\% \text{ confidence interval} = \bar{X} \pm 1.96\sigma/\sqrt{n}. \quad (61)$$

According to our proposed method, the minimum sample size was established to ensure that, with this quantity of data, the evaluation metric falls within the 95% confidence interval of the overall sample mean. More specifically, to ascertain the sufficient number of testing samples needed, we first confirm that the evaluation metric values adhere to a Gaussian distribution. This verification allows for the calculation of the sample mean, sample standard deviation and confidence intervals for the population mean. Then, the minimum number of testing samples needed is determined based on the condition that the evaluation metric corresponding to that number equals the total sample size mean value, with a 95% confidence interval. In particular, when using the proposed method, the quantity of testing samples needed corresponds to the number of samples necessary for the network to attain an evaluation metric equivalent to:

$$\bar{e} + 1.96\sigma_e/\sqrt{N}. \quad (62)$$

This criterion ensures a sufficient level of confidence in the DNN's conclusions while maintaining the high performance of the model.

5.8.3. Experimental results

In this section, we present the experimental results of our method applied to various datasets. Our aim is to identify the minimum number of samples required for DNNs to provide reliable inferences. By determining the minimum sample size, we can assess whether the dataset was previously encountered during the model's training. This approach ensures optimal performance while reducing computational complexity and maintaining the production of reliable predictions in Big Data environments. To ascertain the optimal number of test samples necessary for trustworthy inferences, the DNN is subjected to testing using varying unknown testing sample sizes. In our experiments, the AlexNet [5] DNN architecture is used. To assess the performance of the DNN classifier, classification accuracy is employed as the evaluation metric. The primary outcomes of this study are presented in Table 44, which showcases the experimental results across multiple datasets.

Figure 39 depicts the plot illustrating the correlation between the number of samples and the accuracy scores. The plot demonstrates that with an increasing number of samples, the range of classification accuracy scores becomes narrower. It is observed that when testing with limited data, misclassified data significantly affect the model's evaluation. The classification accuracy scores stabilize after a certain number of samples, suggesting that additional increases in sample size do not significantly affect the performance. This stabilization indicates that the model can deliver accurate inferences even with few data points.

The minimum test data set cardinality needed to ensure reliable and accurate DNN inferences is determined based on the results presented in Table 44. The values indicating the minimum sample sizes for various training datasets are presented in Table 45. The outcomes reveal that the DNN model can





Table 44. Alex-Net Classification Accuracy of Multiple Datasets

Dataset	Number of Testing Samples	Mean Evaluation Metric	Standard Deviation
F-MNIST [10]	15	0.9000	0.0573
	500	0.9214	0.0113
	1000	0.9249	0.0051
	2000	0.9219	0.0057
Cifar-10 [375]	15	0.8667	0.0894
	500	0.8184	0.0160
	1000	0.8206	0.0119
	2000	0.8175	0.0090
Cifar-100 [375]	15	0.5438	0.1282
	500	0.6122	0.0366
	1000	0.6107	0.0175
	2000	0.6167	0.0117
MNIST [376]	15	1.0000	0.0000
	500	0.9952	0.0034
	1000	0.9956	0.0019
	2000	0.9948	0.0014

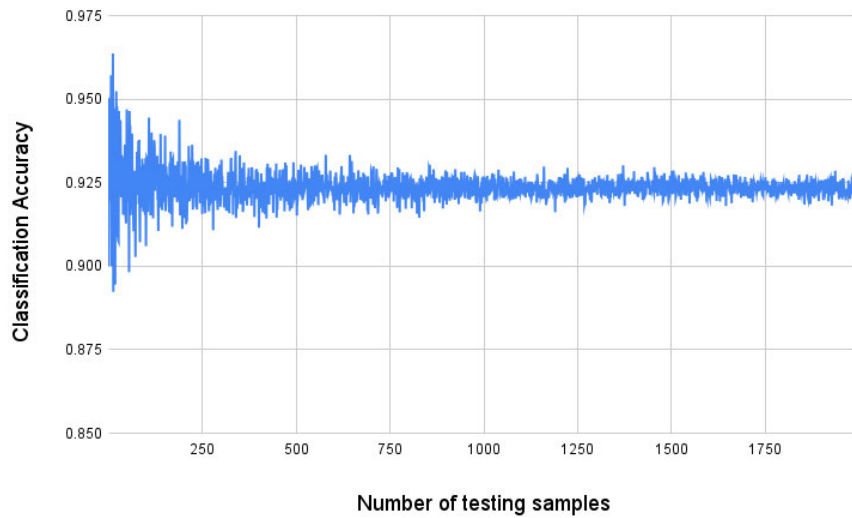


Figure 39. AlexNet classification accuracy scores and the number of samples plot on the F-MNIST dataset [10].

be deemed reliable when delivering inferences for around 1% of the data it is trained on. This finding indicates that the model can effectively handle large volumes of data while requiring significantly fewer data points to produce reliable inferences.





Table 45. Minimum number of data required to ensure reliable DNN (AlexNet [5]) inferences for each dataset and the percentage in relation to the training dataset size.

Dataset	Samples needed	Percentage of samples needed
F-MNIST [10]	335	0.55%
Cifar-10 [375]	305	0.61%
Cifar-100 [375]	362	0.72%
MNIST [376]	253	0.42%

5.8.4. Relevance to AI4Media use cases and media industry applications

This method contributes to UC7 "AI for Content Organization and Content Moderation" by proposing a novel method designed to enhance the efficiency of Deep Neural Networks (DNNs) in managing vast amounts of data. This method addresses the primary challenge of enabling DNNs to provide high-quality inferences using minimal data, a crucial aspect in big data analytics. Consequently, media companies can manage visual content efficiently and cost-effectively. For example, a media organization may implement the proposed methodology to compare and select the most effective AI tool for a specific task using only a small portion of incoming task-specific data.

5.8.5. Relevant Publications

- "Efficient data utilization in deep neural networks for inference reliability", I. Valsamara, C. Papaioannidis, and I. Pitas, "Big Visual Data Analytics Workshop (ICIP 2024)"
Zenodo record: <https://zenodo.org/records/13384355>

5.9. Representation learning for knowledge distillation: teaching representations in triplets

Contributing partner: AUTH

5.9.1. Introduction

Representation learning reveals complex dependencies in the dimensions of the feature vectors used as data representations [377]. Learning a similarity measure that captures small differences within the same class and significant between distinct classes is the main goal of deep metric learning [378]. A popular loss function capturing dependencies in the feature space is triplet loss, which has shown remarkable results in several computer vision tasks [379, 380, 381, 382]. Triplet loss increases the gap between the intra-class and inter-class distances in order to develop a discriminative feature embedding [383]. Basic distillation methods transfer knowledge but often fail to adequately compress the structural knowledge, such as these dependencies between output dimensions. To address this issue, we propose a method able to capture correlations and higher-order output dependencies.

This work investigates representation learning via knowledge distillation. We propose an approach that optimizes the feature structure of the student DNN. Utilizing the concept of triplets, our method seeks to capture data correlations and transfer structural knowledge. The objective is to compress the knowledge of representations and its structural data dependencies from larger to smaller DNN models while preserving performance accuracy. We propose a *Triplet-Based Knowledge Distillation* (TBKD)





method that guides the student model to extract optimal representations and enhances its learning process to effectively capture data similarities. The results demonstrate the efficiency of our method in consistently enhancing the student’s training process, showing improvements in performance accuracy across various DNN architectures and datasets.

5.9.2. Methodology

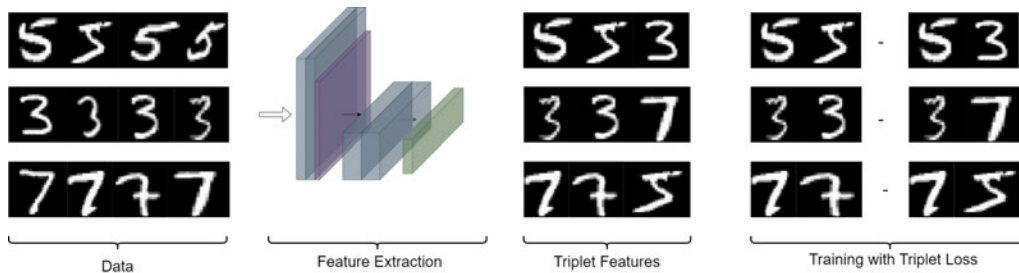


Figure 40. A visual explanation of a deep metric learning framework using triplet loss.

5.9.2.1. Triplet-based Knowledge Distillation (TBKD) loss The key idea of a typical deep metric learning pipeline using triplet loss involves learning a representation that brings “positive” samples closer to an anchor point in a given metric space while pushing “negative” samples further away from the same anchor point as presented in Figure 40. As a result, for each triplet $(\mathbf{x}_a, \mathbf{x}_p, \mathbf{x}_n)$ where \mathbf{x}_a is called the anchor point, \mathbf{x}_p is called the positive point having the same label with \mathbf{x}_a and \mathbf{x}_n is called the negative point having a different label, the intra-class distance $d(\mathbf{x}_a, \mathbf{x}_p)$ will be smaller than the inter-class distance $d(\mathbf{x}_a, \mathbf{x}_n)$ in the learned embedding space. All the data contained in a dataset \mathcal{D} are used to create triplets, which are constructed from each batch, resulting in a new dataset $\mathcal{D}_{triplet}$.

The triplet loss is designed to ensure that the anchor point \mathbf{x}_a is closer to the positive point \mathbf{x}_p than to the negative point \mathbf{x}_n by a margin m . When using a batch of triplets, the overall loss is the mean of the individual triplet losses:

$$L(\mathbf{x}_a, \mathbf{x}_p, \mathbf{x}_n)_{batch} = \frac{1}{N} \sum_{i=1}^N \max(0, d(\mathbf{x}_{a_i}, \mathbf{x}_{p_i}) - d(\mathbf{x}_{a_i}, \mathbf{x}_{n_i}) + m) \quad (63)$$

where N is the number of triplets in the batch, $d(\mathbf{x}_i, \mathbf{x}_j)$ represents the Euclidean distance between the embeddings of points \mathbf{x}_i and \mathbf{x}_j and $\mathbf{f}(\mathbf{x}; \boldsymbol{\theta})$ is the embedding function that maps input samples into a high-dimensional space.

Given two DNNs, a teacher DNN \mathbf{f}^T and a student DNN \mathbf{f}^S , their representations at the penultimate layer for an input image \mathbf{x} are denoted as $\mathbf{f}^T(\mathbf{x}; \boldsymbol{\theta}^T)$ and $\mathbf{f}^S(\mathbf{x}; \boldsymbol{\theta}^S)$ respectively. For a basic KD scheme, the representations $\mathbf{f}^T(\mathbf{x}; \boldsymbol{\theta})$ and $\mathbf{f}^S(\mathbf{x}; \boldsymbol{\theta})$ should be pushed closer. Hence the student is trained with the KL-divergence based KD loss \mathcal{L}_{KD} [384] where KL denotes the Kullback Leibler (KL) divergence.

To facilitate the transfer of structural knowledge from the teacher to the student and achieve optimal representations, our method organizes the data into triplets $(\mathbf{x}_a, \mathbf{x}_p, \mathbf{x}_n)$. This strategy leverages the advantages of triplet loss and representation learning, thereby enabling effective structural KD. The distillation loss for triplet-arranged data is defined as follows:

$$\mathcal{L}_{KDtriplet} = \alpha \sum_{i \in \{a, p, n\}} \text{KL} \left(\frac{\mathbf{f}^S(\mathbf{x}_i; \boldsymbol{\theta}^S)}{T}, \frac{\mathbf{f}^T(\mathbf{x}_i; \boldsymbol{\theta}^T)}{T} \right) \cdot (T^2), \quad (64)$$

where T is the temperature, and α is the distillation hyperparameter. For the combined triplet-based knowledge distillation loss, we incorporate both the distillation loss and the triplet loss. The final TBKD



loss is defined as:

$$\mathcal{L} = \mathcal{L}_{KDtriplet} + \lambda \mathcal{L}_{Triplet}, \quad (65)$$

where \mathcal{L}_{KD} is the distillation loss with temperature scaling defined in Equation 64, $\mathcal{L}_{Triplet}$ the triplet loss defined in Equation 63 and λ is a hyperparameter that controls the relative influence of the triplet loss in comparison to the distillation loss. In practice, the combined loss ensures that the student network learns from both the triplet relationships and the teacher’s guidance as illustrated in Figure 41.

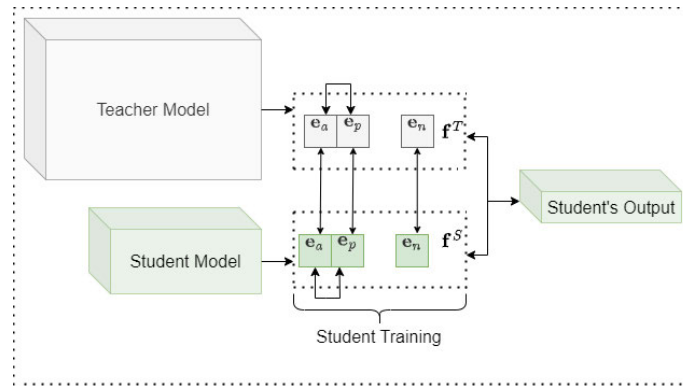


Figure 41. Our method achieves knowledge distillation by minimizing the discrepancy between the feature representations of the teacher and the student, while simultaneously learning a representation $(\mathbf{e}_a, \mathbf{e}_p, \mathbf{e}_n)$ that brings “positive” samples closer to an anchor point and pushes “negative” samples further away in the metric space. To facilitate the transfer of structural knowledge and obtain optimal representations, our method uses a triplet-based knowledge distillation loss (TBKD) that combines both the distillation and the triplet loss.

To evaluate the performance of the DNN teacher and student models, a metric referred to as *triplet accuracy*, which indicates the DNN capability to derive optimal representations is employed. The triplet accuracy measures how well the DNN can distinguish between positive and negative samples in a triplet. During training, the Euclidean distances in the embedding space produced by the DNN, $d(\mathbf{f}(\mathbf{x}_a; \boldsymbol{\theta}), \mathbf{f}(\mathbf{x}_p; \boldsymbol{\theta}))$ - the distance between the anchor and the positive sample and $d(\mathbf{f}(\mathbf{x}_a; \boldsymbol{\theta}), \mathbf{f}(\mathbf{x}_n; \boldsymbol{\theta}))$ - the distance between the anchor and the negative sample, are calculated for each triplet. The triplet accuracy is defined as the ratio of triplets for which the anchor-positive distance is less than the anchor-negative distance:

$$\text{Triplet Accuracy} = \frac{\text{Number of correct triplets}}{\text{Total number of triplets}} \quad (66)$$

where the triplet is considered correct if:

$$d(\mathbf{f}(\mathbf{x}_a; \boldsymbol{\theta}), \mathbf{f}(\mathbf{x}_p; \boldsymbol{\theta})) < d(\mathbf{f}(\mathbf{x}_a; \boldsymbol{\theta}), \mathbf{f}(\mathbf{x}_n; \boldsymbol{\theta})) \quad (67)$$

5.9.3. Experimental Results

We evaluate our method by training the DNN model in three different scenarios: (a) model training with triplet loss, (b) model training with KL-divergence based KD loss [384], and (c) model training with our proposed triplet-based knowledge distillation loss. In the first scenario, we do not employ a KD method; instead, we leverage the benefits of triplet loss to train the architecture of the model that will be used as the student in the distillation scenarios. In the second scenario, we employ the KL-divergence based KD loss. In the third scenario, we use our proposed TBKD objective to train the student model.

The overall aim of KD algorithms is to enable the student DNN to mimic the teacher DNN output, thereby achieving similar performance. To evaluate how well the student mimics the teacher, we measure the student’s ability to create triplets using triplet accuracy as the evaluation metric. The key findings of



our study are presented in Tables 46 and 47, which show the experimental results for multiple datasets and two different combinations of DNN architectures for the teacher and student DNN models: ResNet 101- ResNet 50 and ResNet 34- ResNet 18. Our experiments demonstrate that our approach consistently improves the training of the student model across all scenarios and datasets examined.

Table 46. Comparison between the teacher model (ResNet 101) and the student model (ResNet 50) trained under the examined scenarios for different datasets.

Dataset		Triplet accuracy
MNIST [376]	Teacher	96.22%
	Student	
	Triplet loss	95.80%
	KD loss	93.64%
	TBKD loss	96.96%
FMNIST [10]	Teacher	87.76%
	Student	
	Triplet loss	87.38%
	KD loss	86.77%
	TBKD loss	88.48%
Flowers102 [385]	Teacher	69.83%
	Student	
	Triplet loss	68.26%
	KD loss	65.50%
	TBKD loss	71.43%

Table 47. Comparison between the teacher model (ResNet 34) and the student model (ResNet 18) trained under the examined scenarios for different datasets.

Dataset		Triplet accuracy
MNIST [376]	Teacher	99.07%
	Student	
	Triplet loss	99.34%
	KD loss	94.11%
	TBKD loss	99.49%
FMNIST [10]	Teacher	92.97%
	Student	
	Triplet loss	91.99%
	KD loss	90.87%
	TBKD loss	94.01%
Flowers102 [385]	Teacher	67.91%
	Student	
	Triplet loss	66.67%
	KD loss	65.09%
	TBKD loss	69.94%





Table 48. Comparison of different datasets with their respective teacher (*T*) and student (*S*) models in the image retrieval task.

Dataset		mAP	
		1 neighbor	25 neighbors
MNIST [376]	<i>T</i> : ResNet 101	0.9472	0.9433
	<i>S</i> : ResNet 50		
	Triplet loss	0.9068	0.9069
	KD loss	0.8932	0.8997
	TBKD loss	0.9481	0.9468
FMNIST [10]	<i>T</i> : ResNet 101	0.7763	0.7369
	<i>S</i> : ResNet 50		
	Triplet loss	0.7683	0.7264
	KD loss	0.7476	0.7189
	TBKD loss	0.7898	0.7475
MNIST [376]	<i>T</i> : ResNet 34	0.9871	0.9846
	<i>S</i> : ResNet 18		
	Triplet loss	0.9793	0.9821
	KD loss	0.8959	0.8871
	TBKD loss	0.9882	0.9905
FMNIST [10]	<i>T</i> : ResNet 34	0.8459	0.8167
	<i>S</i> : ResNet 18		
	Triplet loss	0.8402	0.8131
	KD loss	0.7705	0.7740
	TBKD loss	0.8590	0.8287

To assess the representations learned by the DNNs, image retrieval experiments are performed using the teacher and student models trained under various scenarios. A Faiss index [386] is utilized for efficient similarity searches. The retrieval performance is measured using the mean Average Precision (mAP) metric, which evaluates the DNN ability to identify and retrieve relevant images (specifically 1 or 25 nearest neighbors) based on learned embeddings. This evaluation method provides a robust indication of the student model’s capability to extract meaningful and discriminative features from the images. The key findings are presented in Table 48, which displays the experimental results across different datasets and combinations of DNN architectures. Our experiments show that our approach consistently enhances the student DNN model’s retrieval performance across all examined scenarios and datasets.

5.9.4. Relevance to AI4Media use cases and media industry applications

This method matches with UC7 (AI for Content Organization and Content Moderation) as it proposes a Triplet-Based Knowledge Distillation (TBKD) method that can be incorporated into advanced deep learning techniques for content analysis. For instance, a media organization may implement the proposed methodology to enhance visual content retrieval efficiency by employing smaller, less computationally demanding models





5.9.5. Relevant Publications

- I. Valsamara, C. Papaioannidis, and I. Pitas, "Distilling Structural Knowledge: teaching representations for model compression", Under review



5.10. Self-Supervised Facial Representation Learning with Facial Region Awareness

Contributing partner: QMUL

5.10.1. Introduction

Human face understanding is an important and challenging topic in computer vision [387, 388]. Self-supervised pre-training has been proved to be effective in learning transferable representations that benefit various visual tasks. This leads to the question: can self-supervised pre-training learn general facial representations for various facial analysis tasks? Recent efforts toward this goal are limited to treating each face image as a whole, i.e., learning consistent facial representations at the image-level, which overlooks the “**consistency of local facial representations**” (i.e., facial regions like eyes, nose, etc). In this work, we make a **first attempt** to propose a novel self-supervised facial representation learning framework, **Facial Region Awareness (FRA)** that learns consistent global and local facial representations. Specifically, we use learnable positional embeddings as facial queries to look up the facial image for facial regions. The facial queries are learned by solving a pixel-level deep clustering problem.

5.10.2. Methodology

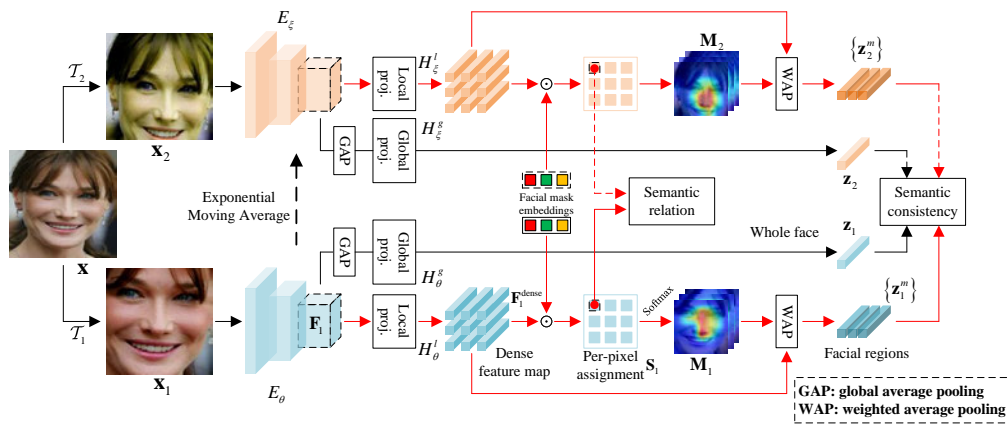


Figure 42. Overview of the proposed FRA framework. \odot denotes cosine similarity. For each input image \mathbf{x} , its augmented views \mathbf{x}_1 and \mathbf{x}_2 are passed into two network branches to produce the global embeddings \mathbf{z}_1 and \mathbf{z}_2 . In addition, we produce a set of heatmaps \mathbf{M}_1 and \mathbf{M}_2 indicating the local facial regions, via the correlation between the pixel features and “facial mask embeddings” computed from a set of learnable positional embeddings. Then we aggregate the feature map to obtain the local facial embeddings $\{\mathbf{z}_1^m\}$ and $\{\mathbf{z}_2^m\}$. The semantic consistency loss is applied to global embeddings and facial embeddings to maximize the similarity across augmented views. To learn such heatmaps, i.e., facial mask embeddings, we treat the facial mask embeddings as facial region clusters and propose a semantic relation loss to align the cluster assignments of each pixel feature over the facial region clusters between the online and momentum network.

The overview of the proposed FRA is shown in Figure 42. We propose two objectives: **pixel-level semantic relation** and **image/region-level semantic consistency**. Semantic relation aligns the per-pixel cluster assignments of each pixel feature over the facial mask embeddings between the online and momentum network to learn heatmaps for facial regions while semantic consistency directly matches the global and local facial representations across augmented views with the learned heatmaps.

Given an input image \mathbf{x} , two random augmentations are applied to generate two augmented views $\mathbf{x}_1 = \mathcal{T}_1(\mathbf{x})$ and $\mathbf{x}_2 = \mathcal{T}_2(\mathbf{x})$, following BYOL [389]. Each augmented view $\mathbf{x}_i \in \{\mathbf{x}_1, \mathbf{x}_2\}$ is fed into an encoder E to obtain a feature map $\mathbf{F}_i \in \mathbb{R}^{C \times H \times W}$ (before global average pooling), where C, H, W are the

number of channels, height and width of \mathbf{F}_i and a latent representation $\mathbf{h}_i \in \{\mathbf{h}_1, \mathbf{h}_2\}$ (after global average pooling), i.e., $\mathbf{h}_1 = E_\theta(\mathbf{x}_1)$ and $\mathbf{h}_2 = E_\xi(\mathbf{x}_2)$. Then each latent representation \mathbf{h}_i is transformed by a global projector H^g to produce a global embedding $\mathbf{z}_i \in \{\mathbf{z}_1, \mathbf{z}_2\}$ of dimension $\mathbf{z}_i \in \mathbb{R}^D$, i.e., $\mathbf{z}_1 = H_\theta^g(\mathbf{h}_1)$ and $\mathbf{z}_2 = H_\xi^g(\mathbf{h}_2)$. Then we obtain a set of heatmaps $\mathbf{M}_i \in \{\mathbf{M}_1, \mathbf{M}_2\}$ highlighting the facial regions from the feature map \mathbf{F}_i for each view. Take view \mathbf{x}_1 as an example, the projected feature map can be expressed as:

$$\mathbf{F}_1^{\text{dense}}[* , u, v] = H_\theta^l(\mathbf{F}_1[* , u, v]), \quad (68)$$

where $\mathbf{F}_1[* , u, v] \in \mathbb{R}^C$ is the pixel feature at the (u, v) -th grid of \mathbf{F}_1 . Then inspired by supervised segmentation [390], we use a Transformer decoder followed by a MLP, which takes as input the feature map \mathbf{F}_i and N learnable positional embeddings (i.e., facial queries for looking up the facial image globally for facial regions) to predict N “facial mask embeddings” $\mathbf{Q}_i \in \mathbb{R}^{N \times D}$ of dimension D , where each row associated with a facial region. Next, we compute the cosine similarity between facial mask embeddings \mathbf{Q}_i and dense feature map $\mathbf{F}_i^{\text{dense}}$ along the channel dimension, yielding **per-pixel cluster assignments** $\mathbf{S}_i \in \mathbb{R}^{N \times H \times W}$, where $\mathbf{S}_i[* , u, v]$ denotes the relation between the dense pixel feature $\mathbf{F}_1^{\text{dense}}[* , u, v]$ and facial mask embeddings \mathbf{Q}_i .

For both augmented views, we define the symmetrized semantic relation objective as:

$$\mathcal{L}_r = \frac{1}{HW} \sum_{u,v} (CE(\mathbf{s}_1^{u,v}, \widehat{\mathbf{s}}_1^{u,v}) + CE(\mathbf{s}_2^{u,v}, \widehat{\mathbf{s}}_2^{u,v})), \quad (69)$$

where $CE(\mathbf{s}_2^{u,v}, \widehat{\mathbf{s}}_2^{u,v})$ is the cross-entropy loss for view \mathbf{x}_2 .

For semantic consistency, we enforce the consistency of global embeddings and local facial embeddings. With the learned heatmaps \mathbf{M}_i , we generate the latent representations for the local facial regions through weighted average pooling. We then match the global embeddings and local facial embeddings across views using the negative cosine similarity in BYOL [389]:

$$\begin{aligned} \mathcal{L}_{\text{sim}}(\mathbf{z}_1, \mathbf{z}_2) = & -(\lambda_c \times f_s(G_\theta^g(\mathbf{z}_1), \mathbf{z}_2) + \\ & + (1 - \lambda_c) \times \frac{1}{N} \sum_{m=1}^N f_s(G_\theta^l(\mathbf{z}_1^m), \mathbf{z}_2^m)), \end{aligned} \quad (70)$$

where $f_s(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u}^\top \mathbf{v}}{\|\mathbf{u}\|_2 \|\mathbf{v}\|_2}$ denotes the cosine similarity between the vectors \mathbf{u} and \mathbf{v} , λ_c is the loss weight, G_θ^g and G_θ^l are the predictors on top of the projectors H_θ^g and H_θ^l , respectively. Following BYOL [389], we symmetrize the loss $\mathcal{L}_{\text{sim}}(\mathbf{z}_1, \mathbf{z}_2)$ defined in eq. 70 by passing \mathbf{x}_1 through the momentum network ξ and \mathbf{x}_2 through the online network θ to compute $\mathcal{L}_{\text{sim}}(\mathbf{z}_2, \mathbf{z}_1)$. The semantic consistency objective can be expressed as follows:

$$\mathcal{L}_c = \mathcal{L}_{\text{sim}}(\mathbf{z}_1, \mathbf{z}_2) + \mathcal{L}_{\text{sim}}(\mathbf{z}_2, \mathbf{z}_1). \quad (71)$$

We jointly optimize the semantic relation objective (eq. 69) and the semantic consistency objective (eq. 71), leading to the following overall objective:

$$\mathcal{L} = \mathcal{L}_c + \lambda_r \mathcal{L}_r, \quad (72)$$

where λ_r is the loss weight for balancing \mathcal{L}_c and \mathcal{L}_r .

5.10.3. Experimental results

Following the common practice in previous works [387, 388], we evaluate the transfer performance of the self-supervised pre-trained facial representations on several popular downstream facial analysis tasks: facial expression recognition (FER) [407, 408], and face alignment (FA) [409, 410, 411, 412].

We report the FER and FA results in Table 49 and 50, respectively. The results on classification (e.g., facial expression recognition) and regression tasks (e.g., face alignment) show that **our FRA achieves SOTA results using vanilla ResNet [400] as the unified backbone for various facial analysis tasks.**



Table 49. *Comparisons on facial expression recognition. We report the Top-1 accuracy on test set. Text denotes text supervision. †: our reproduction using the official codes.*

Method	Text	FERPlus	RAF-DB	AffectNet
Supervised				
KTN [391]	×	90.49	88.07	63.97
RUL [392]	×	88.75	88.98	61.43
EAC [393]	×	90.05	90.35	65.32
Weakly-Supervised				
FaRL [387]†	✓	88.62	88.31	64.85
CLEF [394]	✓	89.74	90.09	65.66
Self-supervised				
MCF [395]†	×	88.17	86.86	60.98
Bulat <i>et al.</i> [396, 397]	×	-	-	60.20
BYOL [389]	×	89.25	89.53	65.65
LEWEL [398]	×	85.61	81.85	61.20
PCL [388]	×	85.87	85.92	60.77
FRA (LP)	×	78.13	73.89	57.38
FRA (FT)	×	89.78	89.95	66.16
FRA (EAC)	×	90.62	90.76	65.85

5.10.4. Conclusion

In this work, we propose a novel self-supervised facial representation learning framework to learn consistent global and local facial representations, **Facial Region Awareness (FRA)**. We learn a set of heatmaps indicating facial regions from learnable positional embeddings, which leverages the attention mechanism to look up facial image globally for facial regions. We show that our FRA outperforms previous pre-trained models on several facial classification and regression tasks. More importantly, using ResNet as the unified backbone, our FRA achieves comparable or even better performance compared with SOTA methods in facial analysis tasks.

5.10.5. Relevance to AI4Media use cases and media industry applications

FRA represents a novel method for self-supervised pre-training with a focus on facial analysis. It can be relevant in tasks such as visual indexing and search and visual concepts classification.

5.10.6. Relevant publications

- Zheng Gao, Ioannis Patras. “Self-Supervised Facial Representation Learning with Facial Region Awareness”, In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024. Zenodo record: <https://zenodo.org/records/13592955>

5.10.7. Relevant software/datasets/other outcomes

- The Pytorch implementation can be found in https://github.com/zaczgao/Facial_Region_Awareness





Table 50. Comparisons on face alignment. †: our reproduction using the official codes.

Method	Arch.	WFLW			300W (NME ↓)		
		NME ↓	FR _{10%} ↓	AUC _{10%} ↑	Full	Comm.	Chal.
Supervised							
SLPT [399]	ResNet [400]	4.20	3.04	0.588	3.20	2.78	4.93
DTLD [401]	ResNet [400]	4.08	2.76	-	2.96	2.59	4.50
RePFormer [402]	ResNet [400]	4.11	-	-	3.01	-	-
ADNet [403]	Hourglass [404]	4.14	2.72	0.602	2.93	2.53	4.58
STAR [405]	Hourglass [404]	4.02	2.32	0.605	2.87	2.52	4.32
Self-supervised							
MCF [395]	ViT [406]	3.96	1.40	0.609	2.98	2.60	4.51
Bulat <i>et al.</i> [396, 397]	ResNet [400]	4.57	-	-	3.20	-	-
BYOL [389]	ResNet [400]	4.29	2.96	0.579	3.03	2.66	4.55
LEWEL [398]	ResNet [400]	4.52	4.50	0.563	3.09	2.70	4.71
PCL [388]†	ResNet [400]	4.84	6.18	0.535	3.35	2.77	5.12
FRA	ResNet [400]	4.11	2.53	0.591	2.91	2.60	4.46

5.11. Self-Supervised Representation Learning with Cross-Context Learning between Global and Hypercolumn Features

Contributing partner: QMUL

5.11.1. Introduction and methodology

Whilst contrastive learning yields powerful representations by matching different augmented views of the same instance, it lacks the ability to capture the similarities between different instances. One popular way to address this limitation is by learning global features (after the global pooling) to capture inter-instance relationships based on knowledge distillation, where the global features of the teacher are used to guide the learning of the global features of the student. Inspired by cross-modality learning, we extend this existing framework that only learns from global features by encouraging the global features and intermediate layer features to learn from each other. This leads to our novel self-supervised framework: **C**ross-**c**ontext learning between **G**lobal and **H**ypercolumn features (CGH), that enforces the consistency of instance relations between low- and high-level semantics. Specifically, we stack the intermediate feature maps to construct a “hypercolumn” representation so that we can measure instance relations using two contexts (hypercolumn and global feature) separately, and then use the relations of one context to guide the learning of the other. This cross-context learning allows the model to learn from the differences between the two contexts. The experimental results on linear classification and downstream tasks show that our method outperforms the state-of-the-art methods.

Given an image \mathbf{x} , we generate a weakly augmented view \mathbf{x}_2 through weak augmentation for the teacher and a heavily augmented view \mathbf{x}_1 through contrastive augmentation for the student. We then proceed to generate the contexts of global feature and hypercolumn for the teacher and the student separately, as shown in Figure 43. First, \mathbf{x}_2 is passed to the teacher encoder E_t to produce the “*global feature context*” (after the global average pooling) $\mathbf{h}_2^g = E_t(\mathbf{x}_2)$. Then \mathbf{h}_2^g is transformed by a global projector H_t to produce a low-dimensional global embedding by $\mathbf{z}_2^g = H_t(\mathbf{h}_2^g)$. As for the hypercolumn of the teacher, let $E_t^l(\mathbf{x}_2) \in \mathbb{R}^{c_l \times h_l \times w_l}$ be the intermediate feature maps of the l -th convolutional block, $l \in \{0, \dots, L\}$, where



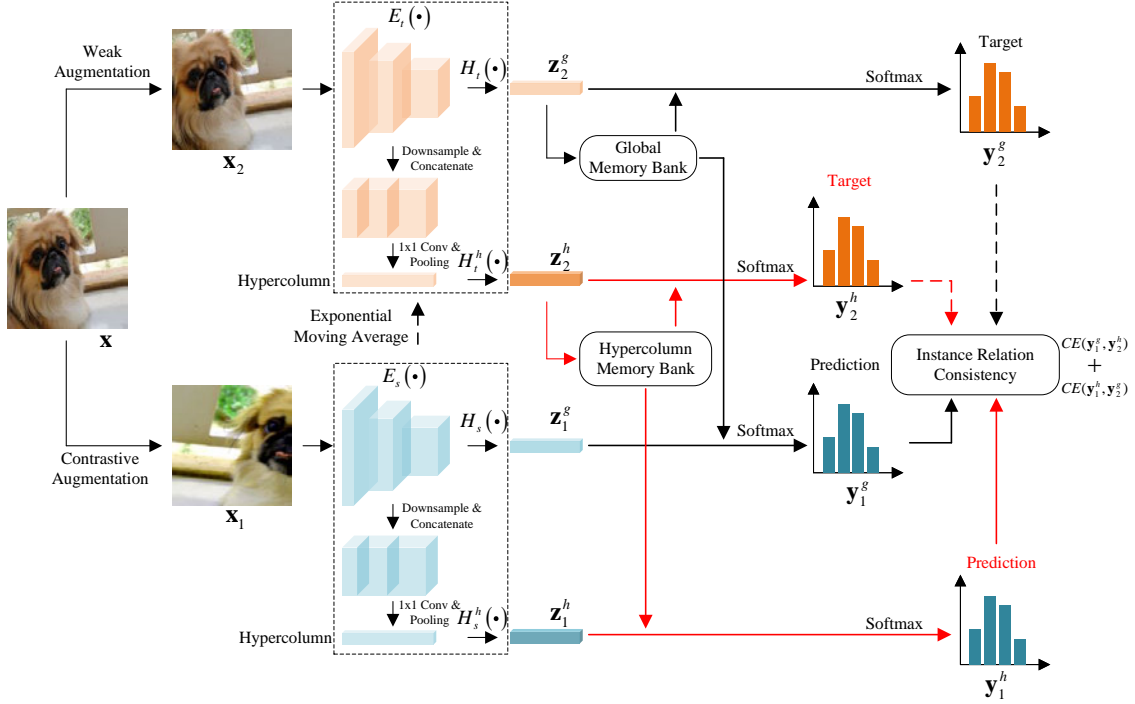


Figure 4.3. **Overview of the proposed CGH framework.** We adopt a knowledge distillation framework where the teacher is the exponential moving average of the student. A heavily corrupted view \mathbf{x}_1 is fed into the student E_s to obtain both a hypercolumn embedding \mathbf{z}_1^h and a global embedding \mathbf{z}_1^g while a weakly augmented view \mathbf{x}_2 is passed to the teacher E_t to obtain a hypercolumn embedding \mathbf{z}_2^h and a global embedding \mathbf{z}_2^g . The embeddings are used to measure the similarity relationships between the augmented views \mathbf{x}_1 , \mathbf{x}_2 and the samples in the memory bank – this leads to a similarity distribution. We enforce two instance relations alignments: “global-hypercolumn alignment” and “hypercolumn-global alignment”, which are detailed in the text.

c_l denotes the number of channels, h_l is the height and w_l is the width. The intermediate feature maps $\{E_t^l(\mathbf{x}_2)\}$, which are downsampled to the same spatial size as the output of the last convolutional block $E_t^l(\mathbf{x}_2)$ to reduce GPU memory consumption, are concatenated first and then mapped to a d -dimensional latent space through a 1×1 convolution followed by average pooling to obtain the “hypercolumn context” $\mathbf{h}_2^h \in \mathbb{R}^d$. \mathbf{h}_2^h is transformed by another projector H_t^h to obtain the hypercolumn embedding by $\mathbf{z}_2^h = H_t^h(\mathbf{h}_2^h)$. Thus the contexts of the global feature \mathbf{h}_2^g and hypercolumn \mathbf{h}_2^h are obtained for the teacher. Likewise, for the student, we produce the global feature context $\mathbf{h}_1^g = E_s(\mathbf{x}_1)$, hypercolumn context \mathbf{h}_1^h and the corresponding embeddings $\mathbf{z}_1^g = H_s(\mathbf{h}_1^g)$ and $\mathbf{z}_1^h = H_s(\mathbf{h}_1^h)$ for the heavily corrupted view \mathbf{x}_1 .

Next we measure the similarity relationships between the augmented views (\mathbf{x}_1 and \mathbf{x}_2) and the samples in the memory bank. To guide the learning of the global feature context \mathbf{h}_1^g for the student, we use the similarity relationships between \mathbf{h}_2^h and the embeddings $\hat{\mathbf{z}}_i^h$ in the hypercolumn memory bank \mathcal{Q}^h as the target. The relationships are measured using the cosine similarity between \mathbf{z}_2^h and $\hat{\mathbf{z}}_i^h$. We normalize the similarities with a softmax operation and produce a target probabilistic distribution \mathbf{y}_2^h for the teacher:

$$\mathbf{y}_2^h[i] = \frac{\exp(\text{sim}(\mathbf{z}_2^h, \hat{\mathbf{z}}_i^h) / \tau_h)}{\sum_{k=1}^M \exp(\text{sim}(\mathbf{z}_2^h, \hat{\mathbf{z}}_k^h) / \tau_h)}, \quad (73)$$

where $\mathbf{y}_2^h[i]$ is the i -th element of the target similarity distribution generated by hypercolumn context \mathbf{h}_2^h , $\hat{\mathbf{z}}_i^h$ is the i -th embedding in the hypercolumn memory bank \mathcal{Q}^h , τ_h is the temperature parameter for the hy-

percolumn context, M is the size of the memory bank and $\text{sim}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u}^\top \mathbf{v}}{\|\mathbf{u}\|_2 \|\mathbf{v}\|_2}$ denotes the cosine similarity between the vectors \mathbf{u} and \mathbf{v} . Similarly, the predicted distribution from the student is expressed as follows:

$$\mathbf{y}_1^g[i] = \frac{\exp(\text{sim}(\mathbf{z}_1^g, \hat{\mathbf{z}}_i / \tau_s))}{\sum_{k=1}^M \exp(\text{sim}(\mathbf{z}_1^g, \hat{\mathbf{z}}_k) / \tau_s)}, \quad (74)$$

where $\mathbf{y}_1^g[i]$ is the i -th element of the predicted similarity distribution generated by global feature context \mathbf{h}_1^g , $\hat{\mathbf{z}}_i$ is the i -th embedding in the memory bank \mathcal{Q} and τ_s is the temperature for global feature context of the student. The global-hypercolumn alignment predicts the hypercolumn based similarity distribution \mathbf{y}_2^h from the global feature based distribution \mathbf{y}_1^g by minimizing the cross-entropy loss:

$$\mathcal{L}_{\text{gh}} = CE(\mathbf{y}_1^g, \mathbf{y}_2^h), \quad (75)$$

where $CE(\mathbf{y}_1, \mathbf{y}_2) = -\sum_{k=1}^M \mathbf{y}_2[k] \log \mathbf{y}_1[k]$.

Similarly, the objective for hypercolumn-global alignment is expressed as:

$$\mathcal{L}_{\text{hg}} = CE(\mathbf{y}_1^h, \mathbf{y}_2^g). \quad (76)$$

Altogether, we enforce the cross-context learning between the global feature context and hypercolumn context with the following objective:

$$\mathcal{L} = \mathcal{L}_{\text{gh}} + \mathcal{L}_{\text{hg}} = CE(\mathbf{y}_1^g, \mathbf{y}_2^h) + CE(\mathbf{y}_1^h, \mathbf{y}_2^g). \quad (77)$$

Table 51. *Linear and KNN evaluation results on IN-1K with ResNet-50 backbone. All methods are evaluated with the single-crop setting. Top-1 and Top-5 validation accuracy are reported. †: our reproduction using the official codes. *: results cited from [6].*

Method	Backprop	Epochs	Batch Size	Linear Acc.	KNN Acc.
Supervised	1x	100	256	76.5	-
Asymmetric loss.					
MoCo-v2 [413]	1x	200	256	67.5	55.9
PCL-v2 [414]	1x	200	256	67.6	58.1
HCSC [415]	1x	200	256	69.2	60.7
OBoW [416]†	1x	200	256	69.5	57.2
ReSSL [417]†	1x	200	256	69.3	61.3
ReSSL-pred [418]	1x	200	1024	72.0	-
CGH	1x	200	256	70.5	62.9
CGH-pred	1x	200	256	72.3	65.8
Symmetric loss. 2× FLOPS					
SimCLR [353]*	2x	200	4096	68.3	-
SwAV [396]*	2x	200	4096	69.1	-
SimSiam [6]*	2x	200	256	70.0	-
BYOL [389]*	2x	200	4096	70.6	-
NNCLR [419]	2x	200	4096	70.7	-



5.11.2. Experimental results

We perform performance evaluation on ImageNet-1k classification in Table 51. The proposed method outperforms MoCo-v2/ReSSL by 3.0%/1.2% on linear classification and 7.0%/1.6% on KNN classification, respectively. The consistent improvement compared with the baselines shows the effectiveness of the proposed cross-context learning strategy.

5.11.3. Conclusion

In order to solve the class collision problem in contrastive learning, inspired by cross-modality learning [420, 421], we present a novel framework based on knowledge distillation, cross-context learning between global and hypercolumn features (CGH) that learns representations by capturing cross-context information from the context of global features and hypercolumns. The cross-context learning strategy allows the model to identify more similar samples (true positives) in the memory bank and keep low false positives. The extensive experiments on classification and downstream tasks demonstrate the effectiveness and generality of our method.

5.11.4. Relevance to AI4Media use cases and media industry applications

CGH represents a novel method for self-supervised pre-training of visual data. It can be relevant in tasks such as visual indexing and search and visual concepts classification.

5.11.5. Relevant publications

- Zheng Gao, Chen Feng and Ioannis Patras. “Self-Supervised Representation Learning with Cross-Context Learning between Global and Hypercolumn Features”, In IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2024. Zenodo record: <https://zenodo.org/records/8364210>

5.11.6. Relevant software/datasets/other outcomes

- The Pytorch implementation can be found in <https://github.com/zaczgao/CGH-Hypercolumn>

5.12. SSR: An Efficient and Robust Framework for Learning with Unknown Label Noise

Contributing partner: QMUL

5.12.1. Introduction

It is now commonly accepted that supervised learning with deep neural networks can provide excellent solutions for a wide range of problems, so long as there is sufficient availability of labeled training data and computational resources. However, these results have been mostly obtained using well-curated datasets in which the labels are of high quality. In the real world, it is often costly to obtain high-quality labels, especially for large-scale datasets. A common approach is to use semi-automatic methods to obtain the labels (e.g. “webly-labeled” images where the images and labels are obtained by web-crawling). While such methods can greatly reduce the time and cost of manual labelling, they also lead to low-quality noisy labels. In such settings, noise is one of the following two types: closed-set noise where the true labels belong to one of the given classes (Set B in Figure 44) and open-set noise where the true labels do not belong to the set of labels of the classification problem



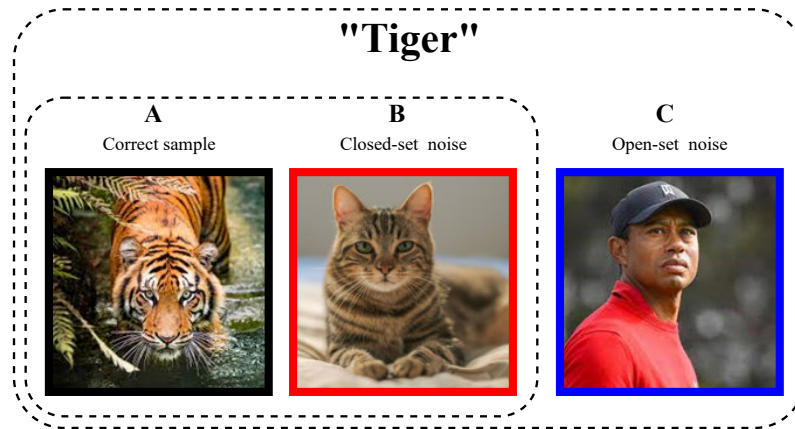


Figure 44. Different “tigers”.

(Set C in Figure 44). To deal with different types of noise, two main types of methods have been proposed, which we name here as probability-consistent methods and probability-approximate methods.

Probability-consistent methods usually model noise patterns directly and propose corresponding probabilistic adjustment techniques, e.g., robust loss functions [422, 423, 424] and noise corrections based on noise transition matrix [425]. However, accurate modelling of noise patterns is non-trivial, and often cannot even model open-set noise. Also, due to the necessary simplifications of probabilistic modelling, such methods often perform poorly with heavy and complex noise. More recently, probability-approximate methods, that is methods that do not model the noise patterns explicitly become perhaps the dominant paradigm, especially ones that are based on sample selection. Earlier methods often reduce the influence of noise samples by selecting a clean subset and training only with it [426, 427, 428, 429]. Recent methods tend to further employ semi-supervised learning methods, such as MixMatch [430], to fully explore the entire dataset by treating the selected clean subset as labeled samples and the non-selected subset as unlabeled samples [431, 432]. These methods, generally, do not consider the presence of open-set noise in the dataset, since most current semi-supervised learning methods can not deal with open-set noise appropriately. To address this, several methods [433, 434] extend the sample selection idea by further identifying the open-set noise and excluding it from the semi-supervised training.

In general, the above methods make assumptions about the pattern of the noise, such as the confidence penalty specifically for asymmetric noise in DivideMix [431]. However, these mechanisms are often detrimental when the noise pattern does not meet the assumptions – for example, explicitly filtering open-set noise in the absence of open-set noise may result in clean hard samples being removed. Furthermore, due to the complexity of combining multiple modules, the above methods usually need to adjust complex hyperparameters according to the type and degree of noise.

In this work, we consider a novel problem setting — *Learning with Unknown Label Noise (LULN)*, that is, learning when both the degree and the type of noise are unknown. Striving for simplicity and robustness, we propose a simple method for LULN, namely *Sample Selection and Relabelling (SSR)*, with two components that are clearly decoupled: a selection mechanism that identifies clean samples with correct labels, and a relabelling mechanism that aims to recover correct labels of wrongly labeled noisy samples. These two major components are based on the two simple and necessary assumptions for LULN, namely, that samples with highly-consistent annotations with their neighbours are often clean, and that very confident model predictions are often trustworthy. Once a well-labeled subset is constructed this way we use the most basic supervised training scheme with a cross-entropy loss.



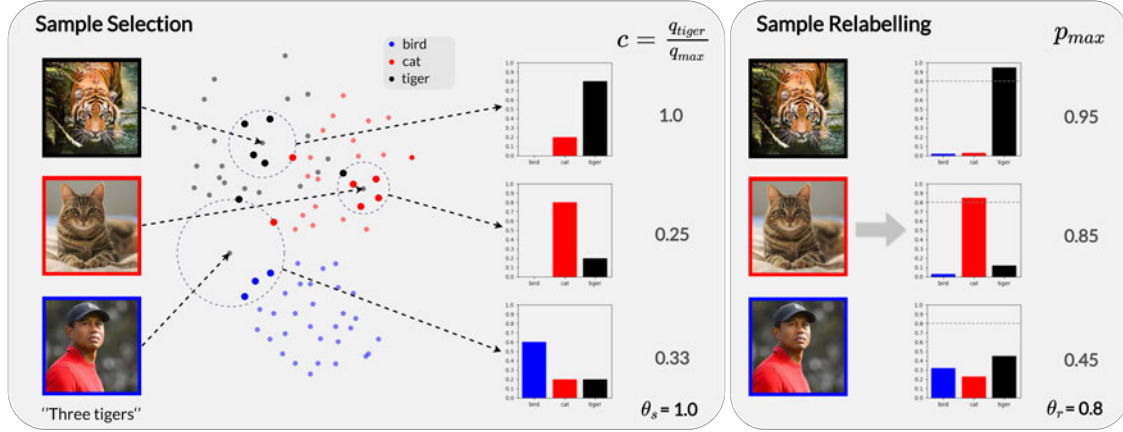


Figure 45. A toy example of SSR with a noisy animal dataset.

Optionally, a feature consistency loss can be used for all data so as to deal better with open-set noise.

5.12.2. Methodology

Let us denote with $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N, \mathbf{x}_i \in R^d$, a training set with the corresponding one-hot vector labels $\mathcal{Y} = \{\mathbf{y}_i\}_{i=1}^N, \mathbf{y}_i \in \{0,1\}^M$, where M is the number of classes and N is the number of samples. For convenience, let us also denote the label of each sample \mathbf{x}_i corresponding to the one-hot label vector \mathbf{y}_i as $l_i = \arg_j [y_i(j)=1] \in \{1, \dots, M\}$. Finally, let us denote the true labels with $\mathcal{Y}' = \{\mathbf{y}'_i\}_{i=1}^N$. Clearly, for an open-set noisy label it is the case that $\mathbf{y}'_i \neq \mathbf{y}_i, \mathbf{y}'_i \notin \{0,1\}^M$, while for closed-set noisy samples $\mathbf{y}'_i \neq \mathbf{y}_i, \mathbf{y}'_i \in \{0,1\}^M$.

We view the classification network as an encoder f that extracts a feature representation and a parametric model classifier (PMC) g_p that deals with the classification problem in question. We also define a non-parametric KNN classifier (NPK) g_q based on the feature representations from encoder f . For brevity, we define $\mathbf{f}_i \triangleq f(\mathbf{x}_i)$ as the feature representation of sample \mathbf{x}_i , and $\mathbf{p}_i \triangleq g_p(\mathbf{f}_i)$ and $\mathbf{q}_i \triangleq g_q(\mathbf{f}_i)$ as the prediction vectors from PMC g_p and NPK g_q , respectively. Following recent works [431, 434, 432, 426, 427], we adopt an iterative scheme for our method consisting of two stages: 1) sample selection and relabelling, and 2) model training.

5.12.2.1. Sample selection and relabelling For a better exposition, we first introduce our sample selection mechanism. Please note, that we actually relabel the samples before each sample selection.

Clean sample selection by balanced neighbouring voting Our sample selection is based on the consistency, as quantified by a measure c_i , between the label \mathbf{y}'_i ¹⁸ of sample \mathbf{x}_i and an (adjusted) distribution, \mathbf{q}_i , of the labels in its neighbourhood in the feature space. More specifically, let us denote the similarity between the representations \mathbf{f}_i and \mathbf{f}_j of any two samples \mathbf{x}_i and \mathbf{x}_j by $s_{ij}, i, j = 1, \dots, N$. By default, we used the cosine similarity, that is, $s_{ij} \triangleq \frac{\mathbf{f}_i^T \mathbf{f}_j}{\|\mathbf{f}_i\|_2 \|\mathbf{f}_j\|_2}$. Let us also denote by N_i the index set of the K nearest neighbours of sample \mathbf{x}_i in \mathcal{X} based on the calculated similarity. Then, for each sample \mathbf{x}_i , we can calculate the KNN-voted label distribution $\mathbf{q}'_i = \frac{1}{K} \sum_{n \in N_i} \mathbf{y}_n^r$ in its neighbourhood, and a balanced version, $\mathbf{q}_i \in R^M$, of it that takes into consideration/compensates for the distribution $\boldsymbol{\pi} = \sum_{i=1}^N \mathbf{y}_i^r$ of the labels in the dataset. More specifically,

$$\mathbf{q}_i = \boldsymbol{\pi}^{-1} \mathbf{q}'_i, \quad (78)$$

¹⁸Please note, we use the labels \mathcal{Y}^r (80) that a relabelling mechanism provides as mentioned above.



where we denote with π^{-1} the vector whose entries are the inverses of the entries of the vector π — in this way we alleviate the negative impact of possible class imbalances in sample selection.

The vector \mathbf{q}_i can be considered as the (soft) prediction of the NPK g_q classifier. We then, define a consistency measure c_i between the sample's label $l_i^r = \operatorname{argmax}_j \mathbf{y}_i^r(j)$ and the prediction \mathbf{q}_i of the NPK as

$$c_i = \frac{\mathbf{q}_i(l_i^r)}{\max_j \mathbf{q}_i(j)}, \quad (79)$$

that is the ratio of the value of the distribution \mathbf{q}_i at the label l_i^r (eq. 80) divided by the value of its highest peak $\max_j \mathbf{q}_i(j)$. Roughly speaking, a high consistency measure c_i at a sample \mathbf{x}_i means that its neighbours agree with its current label l_i^r — this indicates that l_i^r is likely to be correct. By setting a threshold θ_s to c_i , a clean subset $(\mathcal{X}_c, \mathcal{Y}_c^r)$ can be extracted. In our method, we set $\theta_s = 1$ by default, that is, we consider a sample \mathbf{x}_i to be clean only when its neighbours' voting \mathbf{q}_i is consistent with its current label \mathbf{y}_i^r .

Noisy sample relabelling by classifier thresholding Our sample relabelling scheme aims at adding well-labeled samples to the training pool and is based on the PMC classifier g_p . Specifically, we "relabel" all samples for which the classifier is confident, that is all samples i for which the prediction \mathbf{p}_i of the classifier PMC g_p exceeds a threshold θ_r . Formally,

$$l_i^r = \begin{cases} \operatorname{argmax}_l \mathbf{p}_i(l), & \max_l \mathbf{p}_i(l) > \theta_r \\ l_i, & \max_l \mathbf{p}_i(l) \leq \theta_r \end{cases} \quad (80)$$

Please note, we denote the one-hot label corresponding to l_i^r as \mathbf{y}_i^r — this will be used in eq. 78. By setting a high θ_r , a highly confident sample \mathbf{x}_i will be relabeled — this can in turn further enhance the quality of sample selection. Note, that this scheme typically avoids mis-relabelling open-set noise samples as those tend not to have highly confident predictions. In this way, our method can deal with open-set noise datasets effectively even though we do not explicitly propose a mechanism for them.

5.12.2.2. Model training In the training stage, we use the most basic form of supervised learning, i.e., using the cross-entropy loss on the clean subset selected in the first stage — this updates both the encoder f and the PMC g_p . With our sample relabelling mechanism, the size of the clean subset grows progressively by including more and more relabeled closed-set noise in training. Optionally, we use a feature consistency loss that enforces consistency between the feature representations of different augmentations of the same sample — this updates the encoder f and helps to learn a strong feature space on which the selection mechanism of the first stage can rely.

Supervised training using the clean subset For each sample $(\mathbf{x}, \mathbf{y}^r)$ in the selected subset $(\mathcal{X}_c, \mathcal{Y}_c^r)$, we train the encoder f and PMC g_p with common cross-entropy loss, that is, $L_{ce} = -\mathbf{y}^{rT} \log g_p(f(\mathbf{x}))$. Moreover, to deal with the possible class imbalance in the selected subset, we simply over-sample minority classes. In the ablations study, we report the effect of balancing — the over-sampling and also the balanced sample selection in eq. 78.

Optional: feature consistency regularization using all samples Although our relabeling method can progressively relabel and introduce closed-set noise samples into training, open-set samples can also improve generalization. Motivated by commonly used prediction consistency regularization methods, we propose a feature consistency loss L_{fc} [6]. Specifically, with a projector h_{proj} and predictor h_{pred} , we minimize the cosine distance between two different augmented views (\mathbf{x}_1 and \mathbf{x}_2) of the same sample \mathbf{x} . That is,

$$L_{fc} = -\frac{\mathbf{h}_1^\top \mathbf{h}_2}{\|\mathbf{h}_1\|_2 \|\mathbf{h}_2\|_2}, \quad (81)$$





where $\mathbf{h}_1 \triangleq h_{pred}(h_{proj}(f(\mathbf{x}_1)))$ and $\mathbf{h}_2 \triangleq h_{proj}(f(\mathbf{x}_2))$. In summary, the overall training objective is to minimize a weighted sum of L_{ce} and L_{fc} , that is

$$L = L_{ce} + \lambda L_{fc}. \quad (82)$$

We set $\lambda=1$. For brevity, we name our method as SSR when $\lambda=0$, and SSR+ when $\lambda \neq 0$.

5.12.3. Experimental results

We conduct extensive experiments on two standard benchmarks with synthetic label noise, CIFAR-10 and CIFAR-100, and three real-world datasets, Clothing1M [435], WebVision [436], and ANIMAL-10N [437]. For brevity, we define abbreviated names for the corresponding noise settings, such as "sym50" for 50% symmetric noise, "asym40" for 40% asymmetric noise and "all30_open50" for 30% total noise with 50% open-set noise.

Table 52 shows results on CIFAR10 and CIFAR100 — we note again for SSR/SSR+ this is without the use of model cotraining or pre-training. It is clear that our method far outperforms them (e.g. 66.6% accuracy on CIFAR100 with 90% symmetric noise), not only in the case of symmetric noise but also in the more realistic asymmetric synthetic noise settings.

Table 52. Results on CIFAR10/CIFAR100 datasets with synthetic noise.

Dataset	CIFAR10					CIFAR100			
	Symmetric		Assymmetric			Symmetric			
Noise type	20%	50%	80%	90%	40%	20%	50%	80%	90%
Cross-Entropy	86.8	79.4	62.9	42.7	85.0	62.0	46.7	19.9	10.1
Co-teaching+ [427]	89.5	85.7	67.4	47.9	-	65.6	51.8	27.9	13.7
F-correction [438]	86.8	79.8	63.3	42.9	87.2	61.5	46.6	19.9	10.2
PENCIL [439]	92.4	89.1	77.5	58.9	88.5	69.4	57.5	31.1	15.3
LossModelling [440]	94.0	92.0	86.8	69.1	87.4	73.9	66.1	48.2	24.3
DivideMix* [431]	96.1	94.6	93.2	76.0	93.4	77.3	74.6	60.2	31.5
ELR+* [441]	95.8	94.8	93.3	78.7	93.0	77.6	73.6	60.8	33.4
RRL [442]	95.8	94.3	92.4	75.0	91.9	79.1	74.8	57.7	29.3
NGC [434]	95.9	94.5	91.6	80.5	90.6	79.3	75.9	62.7	29.8
AugDesc* [443]	96.3	95.4	93.8	91.9	94.6	79.5	77.2	66.4	41.2
C2D* [444]	96.4	95.3	94.4	93.6	93.5	78.7	76.4	67.8	58.7
SSR(ours)	96.3	95.7	95.2	94.6	95.1	79.0	75.9	69.5	61.8
SSR+(ours)	96.7	96.1	95.6	95.2	95.5	79.7	77.2	71.9	66.6

5.12.4. Conclusion

In this work, we propose an efficient *Sample Selection and Relabelling* (SSR) framework for *Learning with Unknown Label Noise* (LULN). Unlike previous methods that try to integrate many different mechanisms and regularizations, we strive for a concise, simple and robust method. The proposed method does not utilize complicated mechanisms such as semi-supervised learning, model co-training and model pre-training, and is shown with extensive experiments and ablation studies to be robust to the values of its few hyper-parameters, and to consistently and by large surpass the state-of-the-art in various datasets.





5.12.5. Relevance to AI4Media use cases and media industry applications

SSR represents a novel method for learning with noisy labels. It can be relevant in tasks such as visual indexing and search and visual concepts classification.

5.12.6. Relevant publications

- Chen Feng, Georgios Tzimiropoulos and Ioannis Patras. "SSR: An Efficient and Robust Framework for Learning with Unknown Label Noise." In 33rd British Machine Vision Conference, 2022. Zenodo record: <https://zenodo.org/records/8364210>

5.12.7. Relevant software/datasets/other outcomes

- The Pytorch implementation can be found in https://github.com/MrChenFeng/SSR_BMVC2022

5.13. Adaptive Soft Contrastive Learning

Contributing partner: QMUL

5.13.1. Introduction

Self-supervised learning learns meaningful representation information through label-independent tasks, achieving performance that approaches or even exceeds that of supervised learning models in many tasks. Early self-supervised learning methods are often based on heuristic tasks, such as the prediction of image rotation angles, while the current mainstream methods are generally based on instance discrimination tasks, i.e., treating each individual instance as a separate semantic class. Methods in this category usually share the same framework, named as contrastive learning. For a specific view of a specific instance, they define as positives other views of it and negatives views from other instances, and minimize its distance to positives while maximizing its distance to negatives. Meanwhile, a large number of works have been done to improve this framework, such as using a momentum encoder and memory bank to increase the number of negatives [354].

In this work, we focus on an inherent deficiency of contrastive learning, namely "class collision" [445, 446]. The instance discrimination hypothesis violates the natural grouping in visual datasets and induces false negatives, e.g., the representation of two similar dogs should be close to each other rather than pushed away. To bridge the gap, we need to introduce meaningful inter-sample relations in contrastive learning.

Debiased contrastive learning [447] proposes a theoretical unbiased approximation of contrastive loss with the simplified hypothesis of the dataset distribution, however, does not address the issue of real false negatives. Some works [448, 449] apply a progressive mechanism to identify and remove false negatives in the training. NNCLR [419] tries to define extra positives for each specific view by ranking and extracting the top- K neighbors in the learned feature space. Considering soft inter-sample relations, Co2 [450] introduces a consistency regularization enforcing relative distribution consistency of different positive views to all negatives. Clustering-based approaches [451, 452] also provide additional positives, but assuming the entire cluster is positive early in the training is problematic and clustering has an additional computational cost. In addition, all these methods rely on a manually set threshold or a predefined number of neighbors, which is often unknown or hard to determine in advance.

In this work, we propose **ASCL**, an efficient and effective module for current contrastive learning frameworks. We reformulate the contrastive learning problem and introduce inter-sample relations in an adaptive style. To make the training more stable and the inter-sample relationships more accurate, we use weakly augmented views to compute the relative similarity distribution and obtain the sharpened soft label



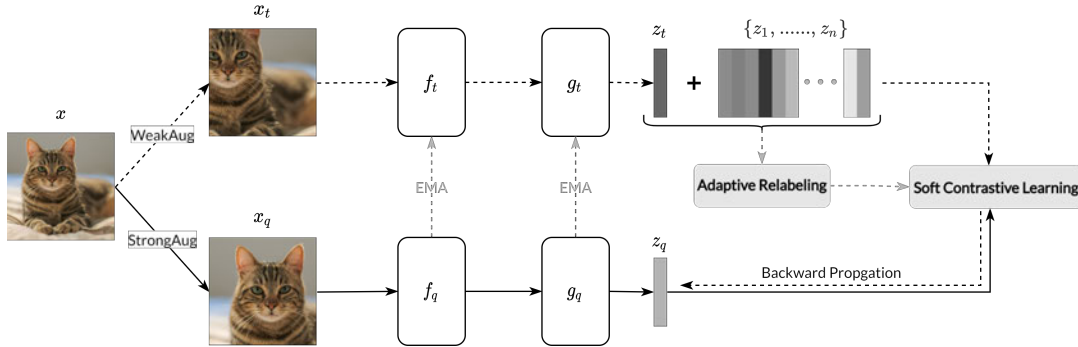


Figure 46. Structure of ASCL. When we remove the adaptive relabelling step (indicated in light grey), ASCL can be considered as a general contrastive learning framework such as MoCo.

information. Based on the uncertainty of the similarity distribution, we adaptively adjust the weights of the soft labels. In the early stages of training, due to the random initialization, the weights of the soft labels are low and the training of the model will be similar to the original contrastive learning. As the features mature and the soft labels become more concentrated, the model will learn stronger inter-sample relations.

5.13.2. Methodology

Current self-supervised learning methods focus on the instance discrimination task, more specifically, learning by considering each image instance as a separate semantic class. In this work, we follow the representative structure in MoCo [354]. More specifically, given a specific sample x , and two different transformed views of it, as query x_q and target x_t , we want to minimize the distance of the corresponding representation projection z_q and z_t while maximizing the distance of z_q and representations of other samples cached in a memory bank $\{z_1, \dots, z_n\}$. Here $z_- = g(f(x_-))$. The learned representation $f(x_-)$ will be fixed and utilized in subsequent tasks such as image classification with an extra linear classifier (Figure 46). With the encoders f_q, f_t and projectors g_q, g_t , we optimize the infoNCE loss:

$$L = -\log \frac{\exp(z_q^T z_t / \tau)}{\exp(z_q^T z_t / \tau) + \sum_{i=1}^n \exp(z_q^T z_i / \tau)} \quad (83)$$

Where τ is a temperature hyperparameter that controls the feature density.

Soft contrastive learning Combining z_t and memory bank $\{z_1, \dots, z_n\}$ together as $\{z'_1, z'_2, \dots, z'_{n+1}\} \triangleq \{z_t, z_1, \dots, z_n\}$ ¹⁹, we can easily rewrite eq. 83 below:

$$L = -\sum_{j=1}^{n+1} y_j \log p_j \quad (84)$$

where

$$p_j = \frac{\exp(z_q^T z'_j / \tau)}{\sum_{i=1}^{n+1} \exp(z_q^T z'_i / \tau)} \quad (85)$$

$$y_j = \begin{cases} 1, & j=1 \\ 0, & \text{otherwise} \end{cases} \quad (86)$$

¹⁹For the convenience, we may use these two notations interchangeably in the following.



Here $\mathbf{y}=[y_1,\dots,y_{n+1}]$ is the one-hot **pseudo label** while $\mathbf{p}=[p_1,\dots,p_{n+1}]$ is the corresponding prediction probability vector. Recalling normal supervised learning, prediction over-confidence has inspired research on label smoothing and knowledge distillation. Similarly in self-supervised learning, this problem is more pronounced due to the fact that the distance between individual samples is smaller compared to that between categories, especially when there are duplicate samples or extremely similar samples in the dataset, i.e., the false negatives described earlier. By modifying **pseudo label**, especially the part regarding with other samples, we can convert original contrastive learning problem as a soft contrastive learning problem, with the optimization goal in eq. 84.

Adaptive Relabelling As mentioned above, the **pseudo label** in infoNCE loss ignores the inter-sample relations which will result in false negatives. To address this problem, we propose to modify the **pseudo label** based on the neighboring relations in the feature space. We first calculate the cosine similarity d between self positive view z'_1 and other representations in memory bank $\{z'_2, z'_3, \dots, z'_{n+1}\}$:

$$d_j = \frac{z'_1{}^T z'_j}{\|z'_1\|_2 \|z'_j\|_2}, j=2, \dots, n+1 \quad (87)$$

- *Hard relabelling* According to $d_j, j=2, \dots, n+1$, we define the top- K nearest neighbors set \mathcal{N}_K in the memory bank of z'_1 as extra positives for z_q . The new **pseudo label** \mathbf{y}_{hard} will be defined as below:

$$y_j = \begin{cases} 1, & j=1 \text{ or } z_j \in \mathcal{N}_K \\ 0, & \text{otherwise} \end{cases} \quad (88)$$

Intuitively speaking, we consider not only z'_1 as positive for z_q but also the top- K nearest neighbors of z'_1 .

- *Adaptive hard relabelling* However, it is risky to recklessly assume that the top- K nearest neighbors are positive, and, especially early in the training, some hard samples may have fewer close neighbors compared to others. To alleviate these problems of \mathbf{y}_{hard} , we propose an adaptive mechanism that automatically modifies the confidence of the **pseudo label**. More specifically, with cosine similarity d we build the relative distribution \mathbf{q} between self positive view z'_1 and other representations in memory bank $\{z'_2, z'_3, \dots, z'_{n+1}\}$:

$$q_j = \frac{\exp(d_j/\tau')}{\sum_{l=2}^{n+1} \exp(d_l/\tau')}, j=2, \dots, n+1 \quad (89)$$

To quantify the uncertainty of relative distribution, i.e., how confident when we extract the neighbors, we define a confidence measure as the normalized entropy of the distribution \mathbf{q} :

$$c = 1 - \frac{H(\mathbf{q})}{\log(n)} \quad (90)$$

Here $H(\mathbf{q})$ is the Shannon entropy of \mathbf{q} . We further use $\log(n)$ to normalize c into $[0,1]$. We then get the adaptive hard label \mathbf{y}_{ahcl} by augmenting \mathbf{y}_{hard} with c :

$$y_j = \begin{cases} 1, & j=1 \\ c, & j \neq 1 \text{ and } z_j \in \mathcal{N}_K \\ 0, & j \neq 1 \text{ and } z_j \notin \mathcal{N}_K \end{cases} \quad (91)$$

- *Adaptive soft relabelling* Moreover, instead of using top- K neighbors for the extra positives, we also propose using the distribution \mathbf{q} itself as soft labels. Intuitively speaking, a more concentrated



distribution yields a higher degree of confidence, implying a more reliable neighboring relationship for the sample. We then define the adaptive soft label \mathbf{y}_{ascl} as:

$$y_j = \begin{cases} 1, & j=1 \\ \min(1, c \cdot K \cdot q_j), & j \neq 1 \end{cases} \quad (92)$$

Here, c is defined in eq. 91 to weight the soft labels, and K is the number of neighbors in \mathcal{N}_K . Please note, that we put an upper bound of one – that means that the most confident positive neighbor is not more confident than a view of the sample itself, i.e., than z'_1 .

Finally, \mathbf{y}_{ascl} , \mathbf{y}_{ahcl} and \mathbf{y}_{hard} are then normalized, that is:

$$y_j = \frac{y_j}{\sum_- y_-} \quad (93)$$

For simplicity, we use the same notation for the normalized **pseudo label** as the unnormalized ones. By default we use \mathbf{y}_{ascl} for training — this is the **ASCL** method. We call the training method that uses \mathbf{y}_{ahcl} as **AHCL**, and the one with \mathbf{y}_{hard} as **Hard**. When we set K as zero, the method degenerates to the original MoCo framework.

5.13.3. Experimental results

We evaluate **ASCL** on ImageNet-1k in Table 53. With all methods pretrained for 200 epochs, **ASCL** outperforms the current state-of-the-art methods. Also, please note that **ASCL** requires only one backpropagation pass, which reduces a significant amount of computational cost compared to methods such as BYOL, SimCLR, etc.

Table 53. Results on ImageNet-1K dataset.

Method	Architecture	BackProp	EMA	Batch Size	Epochs	Top-1 Acc
Supervised	ResNet50	1x	No	256	120	76.5
InstDisc [453]	ResNet50	1x	No	256	200	58.5
LocalAgg [454]	ResNet50	1x	NO	128	200	58.8
MoCo [354]	ResNet50	1x	Yes	256	200	67.5
CO2 [450]	ResNet50	1x	No	256	200	68.0
PCL [446]	ResNet50	1x	Yes	256	200	67.6
ReSSL [417]	ResNet50	1x	Yes	256	200	69.9
ASCL(Ours)	ResNet50	1x	Yes	256	200	71.5
SimCLR [353]	ResNet50	2x	No	4096	200	66.8
NNCLR [419]	ResNet50	2x	No	4096	200	70.7
CLSA [455]	ResNet50	2x	Yes	256	200	69.4
SwAV [396]	ResNet50	2x	No	4096	200	69.1
SimSiam [6]	ResNet50	2x	No	256	200	70.0
BYOL [389]	ResNet50	2x	Yes	4096	200	70.6

5.13.4. Conclusion

In this work, we propose **ASCL**, a reliable and efficient framework based on the current contrastive learning framework. We utilize a sharpened inter-sample distribution to introduce extra positives and





adaptively adjust its confidence based on the entropy of the distribution. Our method achieves the state of the art in various benchmarks, with a negligible extra computational cost. We also show the potential of our method with self-supervised learning methods requiring no memory bank and explicit negative pairs.

5.13.5. Relevance to AI4Media use cases and media industry applications

ASCL represents a novel method for self-supervised representation learning. It can be relevant in tasks such as 3visual indexing and search and visual concepts classification.

5.13.6. Relevant publications

- Chen Feng, Ioannis Patras. "Adaptive Soft Contrastive Learning." In 2022 26th International Conference on Pattern Recognition (ICPR), 2022. Zenodo record: <https://zenodo.org/records/8014131>

5.13.7. Relevant software/datasets/other outcomes

- The Pytorch implementation can be found in https://github.com/MrChenFeng/Adaptive-Soft-Contrastive-Learning_ICPR2022

5.14. DivClust: Controlling Diversity in Deep Clustering

Contributing partner: QMUL

5.14.1. Introduction

Clustering has been a major research subject in the field of machine learning, one to which deep learning has recently been applied with significant success. However, an aspect of clustering that is not addressed by existing deep clustering methods, is that of efficiently producing multiple, diverse partitionings for a given dataset. This is particularly important, as a diverse set of base clusterings are necessary for consensus clustering, which has been found to produce better and more robust results than relying on a single clustering. QMUL's work has focused on this area, and we developed a diversity enforcing clustering loss component that can be used to train models to produce multiple clusterings of controlled diversity with each other, and which explore different partitionings of a given dataset. We conduct experiments with multiple datasets and deep clustering frameworks and show that: a) our method effectively controls diversity across frameworks and datasets with very small additional computational cost, b) the sets of clusterings learned by DivClust include solutions that significantly outperform single-clustering baselines, and c) using an off-the-shelf consensus clustering algorithm, DivClust produces consensus clustering solutions that consistently outperform single-clustering baselines, effectively improving the performance of the base deep clustering framework.

5.14.2. Methodology

The architecture of our method can be seen in Figure 47. It consists of a backbone network f , followed by K projection heads h_1, \dots, h_K , each corresponding to a clustering C_k . Assuming a set X of N unlabeled samples, the backbone network maps those samples $x \in X$ to vector representations $f(x)$, and each projection head h_k maps the representations to C clusters, such that $p_k(x) = h_k(f(x)) \in \mathbb{R}^C$ represents the probability assignment vector mapping the sample $x \in X$ to C clusters in clustering $k = 1, \dots, K$. The column $p_k(i)$, that is the probability assignment vector for the i -th sample, shows to which clusters that





sample has been assigned. The row vector $q_k(j)$, that is the cluster membership vector for a cluster j , shows which samples are assigned to cluster j .

To quantify the similarity between clusterings A and B we define the inter-clustering similarity matrix $S_{AB} \in \mathbb{R}^{C \times C}$. We define each element $S_{AB}(i, j)$ as the cosine similarity between the cluster membership vector $q_A(i)$ of cluster $i \in A$ and the cluster membership vector $q_B(j)$ of cluster $j \in B$:

$$S_{AB}(i, j) = \frac{q_A(i) \cdot q_B(j)}{\|q_A(i)\|_2 \|q_B(j)\|_2} \quad (94)$$

This measure expresses the degree to which samples in the dataset are assigned similarly to clusters i and j . Specifically, $S_{AB}(i, j) = 0$ if $q_A(i) \perp q_B(j)$ and $S_{AB}(i, j) = 1$ if $q_A(i) = q_B(j)$. It is, therefore, a differentiable measure of the similarity of clusters i and j .

Based on the inter-clustering similarity matrix S_{AB} , we define DivClust's loss L_{div} to softly enforce that the aggregate similarity S_{AB}^{agg} between clusterings A and B does not exceed a similarity upper bound d . L_{div} regulates the diversity between clusterings A and B by forcing that $S_{AB}^{agg} < d$, for $d \in [0, 1]$.

$$S_{AB}^{agg} = \frac{1}{C} \sum_{i=1}^C \max_j (S_{AB}(i, j)) \quad , \quad L_{div}(A, B) = [S_{AB}^{agg} - d]_+ \quad (95)$$

The similarity upper bound d is dynamic and updated in regular intervals of $T = 10$ steps as:

$$d_{s+1} = \begin{cases} \max(d_s(1-m), 0), & \text{if } D^R > D^T \\ \min(d_s(1+m), 1), & \text{if } D^R \leq D^T \end{cases} \quad (96)$$

where D^T is a user-defined similarity target D^R is the inter-clustering similarity measured over a small memory bank. Both values measure inter-clustering similarity with the Normalized Mutual Information metric averaged over the clusterings. Following this update rule, we decrease d when the measured inter-clustering similarity D^R needs to decrease, and increase it otherwise.

Having defined the diversity loss L_{div} between two clusterings, we extend it to multiple clusterings K and combine it with the base deep clustering framework's objective for the joint loss $L_{joint}(k)$ for each clustering k , where $L_{main}(k)$ depends on cluster assignment matrix P_k , while $L_{div}(k, k')$ depends on P_k and $P_{k'}$. Accordingly, the model's training loss L_{total} is the average of L_{joint} over all clusterings.

$$L_{joint}(k) = L_{main}(k) + \frac{1}{K-1} \sum_{k'=1, k' \neq k}^K L_{div}(k, k') \quad , \quad L_{total} = \frac{1}{K} \sum_{k=1}^K L_{joint}(k) \quad (97)$$

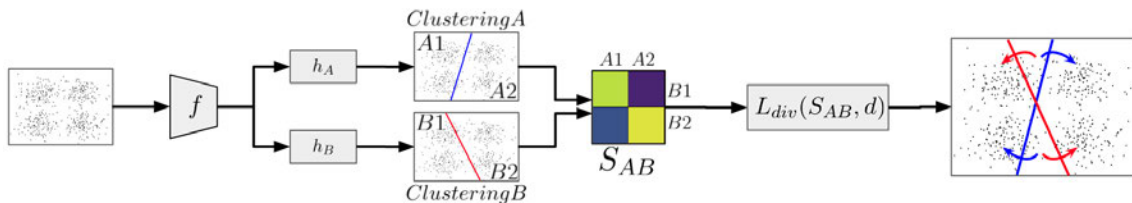


Figure 47. Overview of DivClust. Assuming clusterings A and B , the proposed diversity loss L_{div} calculates their similarity matrix S_{AB} and restricts the similarity between cluster pairs to be lower than a similarity upper bound d . In the figure, this is represented by the model adjusting the cluster boundaries to produce more diverse clusterings. Best seen in color.





Table 54. Results combining DivClust with CC for various diversity targets D^T . We underline DivClust results that outperform the single-clustering baseline CC, and note with **bold** the best results for each metric across all methods and diversity levels. We emphasize that the NMI in this table measures the similarity between the single clustering produced by each method and the ground truth classes.

Dataset	D^T	CIFAR10			CIFAR100			ImageNet-10			ImageNet-Dogs		
Metric	NMI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI
K-means	-	0.087	0.229	0.049	0.084	0.130	0.028	0.119	0.241	0.057	0.55	0.105	0.020
DEC	-	0.257	0.301	0.161	0.136	0.185	0.050	0.282	0.381	0.203	0.122	0.195	0.079
DAC	-	0.396	0.522	0.306	0.185	0.238	0.088	0.394	0.527	0.302	0.219	0.275	0.111
ADC	-	-	0.325	-	-	0.160	-	-	0.530	-	-	-	-
DCCM	-	0.496	0.623	0.408	0.285	0.327	0.173	0.608	0.710	0.555	0.321	0.383	0.182
IIC	-	-	0.617	-	-	0.257	-	-	-	-	-	-	-
PICA	-	0.591	0.696	0.512	0.310	0.337	0.171	0.802	0.870	0.761	0.352	0.352	0.201
CC	-	0.705	0.790	0.637	0.431	0.429	0.266	0.859	0.893	0.822	0.445	0.429	0.274
CC-Kmeans	-	0.654	0.698	0.523	0.429	0.405	0.235	0.792	0.841	0.669	0.457	0.444	0.284
DeepCluE	-	0.727	0.764	0.646	0.472	0.457	0.288	0.882	0.924	0.856	0.448	0.416	0.273
DivClust	1.	0.678	0.763	0.604	0.418	0.424	0.257	<u>0.86</u>	<u>0.895</u>	<u>0.825</u>	<u>0.459</u>	<u>0.451</u>	<u>0.298</u>
	0.95	0.677	0.76	0.602	0.431	<u>0.434</u>	<u>0.276</u>	0.891	0.936	0.878	<u>0.461</u>	<u>0.451</u>	<u>0.297</u>
	0.9	0.678	<u>0.789</u>	<u>0.641</u>	0.422	0.426	0.258	<u>0.879</u>	<u>0.92</u>	<u>0.859</u>	<u>0.48</u>	<u>0.487</u>	<u>0.332</u>
	0.8	<u>0.724</u>	0.819	0.681	0.422	0.414	0.26	<u>0.879</u>	<u>0.918</u>	<u>0.851</u>	<u>0.458</u>	<u>0.448</u>	<u>0.296</u>
	0.7	<u>0.71</u>	<u>0.815</u>	<u>0.675</u>	<u>0.44</u>	<u>0.437</u>	<u>0.283</u>	0.85	0.90	0.819	0.516	0.529	0.376

Table 55. Avg. inter-clustering similarity scores D^R for clustering sets produced by DivClust combined with CC for various diversity targets D^T . The objective of DivClust is that $D^R \leq D^T$.

D^T	D^R			
	CIFAR10	CIFAR100	ImageNet-10	ImageNet-Dogs
1.	0.976	0.939	0.987	0.941
0.95	0.946	0.926	0.948	0.945
0.9	0.9	0.848	0.897	0.87
0.8	0.814	0.806	0.807	0.795
0.7	0.699	0.705	0.696	0.702

5.14.3. Experimental results

We conduct experiments on four datasets (CIFAR10, CIFAR100, Imagenet-10, Imagenet-Dogs) using the deep clustering method CC as our baseline, aggregate the learned clusterings using the off-the-shelf consensus clustering method SCCBG [456], and report the Accuracy (ACC), Normalized Mutual Information (NMI) and Adjuster Rand Index (ARI) scores of the resulting clustering solution, relative to the ground truth labels in Table 54. Furthermore, we report the target and reported similarities D^T and D^R for DivClust in Table 55. Our results show that DivClust can effectively control inter-clustering diversity without reducing the quality of the clusterings. Furthermore, we demonstrate that, with the use of an off-the-shelf consensus clustering algorithm, the diverse base clusterings learned by DivClust produce consensus clustering solutions that outperform the base frameworks, effectively improving them with minimal computational cost.

5.14.4. Conclusion

We introduce DivClust, a method that can be incorporated into existing deep clustering frameworks to learn multiple clusterings while controlling inter-clustering diversity. To the best of our knowledge,





this is the first method that can explicitly control inter-clustering diversity based on user-defined targets, and that is compatible with deep clustering frameworks that learn features and clusters end-to-end. Experiments confirm the effectiveness of DivClust in controlling inter-clustering diversity and its adaptability. Furthermore, results demonstrate that DivClust learns high quality clusterings, which, in the context of consensus clustering, lead to improved performance compared to single clustering baselines and alternative ensemble clustering methods.

5.14.5. Relevance to AI4Media use cases and media industry applications

DivClust represents a novel method for generating diverse clusterings of visual data. It can be relevant in tasks such as visual indexing and search and visual concepts classification.

5.14.6. Relevant publications

- Maniadis Metaxas, Ioannis, Georgios Tzimiropoulos, and Ioannis Patras. "Divclust: Controlling diversity in deep clustering." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023. Zenodo record: <https://zenodo.org/records/8013831>

5.14.7. Relevant software/datasets/other outcomes

- The Pytorch implementation can be found in <https://github.com/ManiadisG/DivClust>

5.15. Efficient Unsupervised Visual Representation Learning with Explicit Cluster Balancing

Contributing partner: QMUL

5.15.1. Introduction

Self-supervised learning has recently emerged as the preeminent pretraining paradigm across and between modalities, with remarkable results. In the image domain specifically, group (or cluster) discrimination has been one of the most successful methods. However, such frameworks need to guard against heavily imbalanced cluster assignments to prevent collapse to trivial solutions. Existing works typically solve this by reweighing cluster assignments to promote balance, or with offline operations (e.g. regular re-clustering) that prevent collapse. However, the former typically requires large batch sizes, which leads to increased resource requirements, and the latter introduces scalability issues with regard to large datasets. To tackle this challenge, QMUL developed ExCB, a framework that uses a novel online cluster balancing method that is stable without requiring a large batch size. ExCB estimates the relative size of the clusters across batches and balances them by adjusting cluster assignments, proportionately to their relative size and in an online manner. Thereby, it overcomes previous methods' dependence on large batch sizes and is fully online, and therefore scalable to any dataset. We conduct extensive experiments to evaluate our approach and demonstrate that ExCB: **a)** achieves state-of-the-art results with significantly reduced resource requirements compared to previous works, **b)** is fully online, and therefore scalable to large datasets, and **c)** is stable and effective even with very small batch sizes.

5.15.2. Methodology

Following previous works, ExCB utilizes a teacher-student framework, where the student is trained to match the cluster assignments of the teacher, and the teacher's weights follow the student via momentum



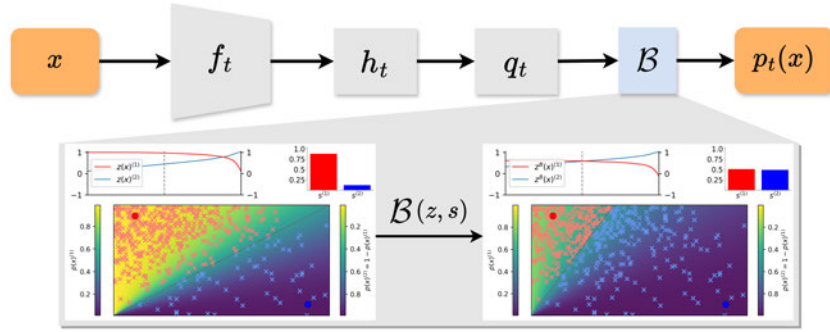


Figure 48. Illustration of ExCB’s balancing operator \mathcal{B} for two clusters c_1 (red) and c_2 (blue). $\mathcal{B}(z; s)$ adjusts sample-cluster cosine similarities z according to the relative cluster sizes, as measured in s . For smaller clusters the similarities are increased ($z^B > z$), whereas for larger clusters the similarities are decreased ($z^B < z$). The impact, as seen in the figure, is that the boundary between clusters shifts, undersized (oversized) clusters are assigned more (fewer) samples, and clusters become more balanced.

update. We define as z_s^h and z_s^g the sample-cluster cosine similarities produced by the student’s projector and predictor MLP heads respectively, and as z_t the sample-cluster cosine similarities produced by the teacher’s projector. In order to balance the target cluster assignments produced by the teacher, we apply a balancing operator \mathcal{B} to z_t , resulting in updated sample-cluster similarities z_t^B , as illustrated in Table 48. We then obtain the probability assignment vectors \mathbf{p}_s^h , \mathbf{p}_s^g and $\mathbf{p}_t \in \mathbb{R}^K$, mapping sample x to each cluster $k \in K$:

$$\mathbf{p}_s^h(x)^{(k)} = \frac{\exp(z_s^h(x)^{(k)}/\tau_s)}{\sum_{i=1}^K \exp(z_s^h(x)^{(i)}/\tau_s)}, \quad \mathbf{p}_s^g(x)^{(k)} = \frac{\exp(z_s^g(x)^{(k)}/\tau_s)}{\sum_{i=1}^K \exp(z_s^g(x)^{(i)}/\tau_s)}, \quad (98)$$

$$\mathbf{p}_t(x)^{(k)} = \frac{\exp((z_t^B(x)^{(k)}/\tau_t)}{\sum_{i=1}^K \exp((z_t^B(x)^{(i)}/\tau_t)}, \quad (99)$$

where τ_s , τ_t are temperature hyperparameters. The student is then trained to minimize the loss L :

$$L = \frac{1}{2} \sum_{x' \in G} \sum_{\substack{x'' \in G \cup L \\ x'' \neq x'}} H(\mathbf{p}_t(x'), \mathbf{p}_s^h(x'')) + \frac{1}{2} \sum_{x' \in G} \sum_{x'' \in G \cup L} H(\mathbf{p}_t(x'), \mathbf{p}_s^g(x'')), \quad (100)$$

where x' and x'' are different views of sample x and G , L represent global and local crops.

To define the balancing operator \mathbb{B} , we first define the relative cluster size vector $\mathbf{s} \in \mathbb{R}^K$. For each batch of N_B samples X we obtain the teacher’s cluster assignments $\mathbf{P}_t(X) = [\mathbf{p}_t(x_1), \dots, \mathbf{p}_t(x_{N_B})] \in \mathbb{R}^{N_B \times K}$, and calculate the in-batch relative cluster size vector $\mathbf{s}_B \in \mathbb{R}^K$ as the proportion of samples assigned to each cluster:

$$\mathbf{s}_B^{(k)} = \frac{1}{N_B} \sum_{n=1}^{N_B} \mathbf{1}_{\arg\max(\mathbf{p}_t(x_n))=k}. \quad (101)$$

The vector \mathbf{s} is then updated for each batch as:

$$\mathbf{s} = m_s \mathbf{s} + \mathbf{s}_B (1 - m_s), \quad (102)$$

where m_s is a momentum hyperparameter.



Method	Batch Size	Epochs	Linear	k-NN
Supervised	-	-	75.6	-
SimCLR	4096	800	71.7	-
BYOL	4096	1000	74.4	64.8
MoCo-v3	4096	1000	74.6	-
DeepCluster v2	4096	800	75.2	-
Barlow Twins	2048	1000	73.2	66.0
SwAV	4096	800	75.3	65.7
DINO	4096	800	75.3	67.5
NNCLR	4096	1000	75.4	-
TWIST*	2048	800+50	75.5	-
MIRA	4096	800	75.7	68.8
MAST	2048	1000	75.8	-
CoKe	1024	800	<u>76.4</u>	-
SMoG	4096	400	<u>76.4</u>	-
ExCB	1024	400	76.5	71.0

Table 56. *Linear & k-NN classification on ImageNet. We report linear and k-NN classification accuracy on ImageNet, along each method’s pretraining batch size and epochs. *TWIST follows standard pretraining with filtered self-labeled training.*

Essentially, $\mathbf{s} \in [0,1]$ measures the proportion of samples assigned to each cluster over multiple batches with an exponential moving average whose window length is determined by m_s . This approach yields an accurate estimate of cluster sizes across the dataset, without requiring a large batch size. If samples are distributed among clusters with absolute uniformity, then $\mathbf{s}^{(k)} \rightarrow \frac{1}{K} \forall k \in K$, whereas $\mathbf{s}^{(k)} < \frac{1}{K}$ for undersized clusters and $\mathbf{s}^{(k)} > \frac{1}{K}$ for oversized clusters.

We then define $\mathcal{B}(z;s)$ as follows:

$$z^B = \mathcal{B}(z;s) = \begin{cases} 1 - [1 - z]sK & , \text{ if } s < \frac{1}{K} \\ [1 + z]\frac{1}{sK} - 1 & , \text{ if } s > \frac{1}{K} \\ z & , \text{ otherwise} \end{cases} \quad (103)$$

For any cluster k , \mathcal{B} increases sample-cluster similarity if k is undersized ($z^B > z$ for $s < \frac{1}{K}$), and decreases it if k is oversized ($z^B < z$ for $s > \frac{1}{K}$). In this way, undersized (oversized) clusters are assigned more (fewer) samples, in an effort to approximate evenly sized clusters ($\mathbf{s}^{(k)} \rightarrow 1, \forall k \in K$). This simple approach softly balances cluster assignments without requiring a large batch size.

5.15.3. Experimental results

We conduct extensive experiments with both CNN and ViT backbones, and demonstrate that ExCB achieves state-of-the-art performance, while requiring fewer resources (batch size and/or pretraining epochs) compared to competitive methods. Specifically, we present results for the main classification downstream task in Table 56 and in Table 57 for ResNet50 and ViT-S/16 backbones, respectively. In both cases, ExCB outperforms previous works, and we note that, for ResNet50, it does so with fewer epochs and a smaller batch size, whereas for ViT-S/16 it outperforms DINO without any hyperparameter tuning (i.e. using DINO’s recommended hyperparameters), which highlights ExCB’s effectiveness and reliability.

5.15.4. Conclusion

We present ExCB, a novel clustering-based framework for self-supervised representation learning. ExCB relies on a novel cluster balancing method that explicitly measures their sizes across multiple batches, and





Method	Batch Size	Epochs		
		100	300	800
MoCo-v3	4096	-	72.5	-
DINO	1024	73.8	75.9	77.0
TWIST	1024	-	76.3	-
ExCB	1024	73.9	76.4	77.1

Table 57. *Linear classification with ViT. We report linear classification accuracy on ImageNet for various epochs.*

adjusts their assignments to promote evenly sized clusters. We conduct extensive experiments and find that ExCB achieves state-of-the-art results across benchmarks and backbone architectures. However, crucially, our experiments demonstrate that ExCB is also remarkably efficient, as it achieves the strong performance reported in this work with less training and a much smaller batch size than most other frameworks. Overall, we believe that the proposed framework is not only significant in terms of its performance, but also as a step toward decreasing the resources required for self-supervised pretraining with visual data.

5.15.5. Relevance to AI4Media use cases and media industry applications

ExCB represents a novel self-supervised representation learning method for visual data. It can be relevant in tasks such as visual indexing and search and visual concepts classification.

5.15.6. Relevant publications

- Maniadis Metaxas, Ioannis, Georgios Tzimiropoulos, and Ioannis Patras. "Efficient Unsupervised Visual Representation Learning with Explicit Cluster Balancing." European Conference on Computer Vision. 2024. Zenodo record: <https://zenodo.org/records/13510391>

5.15.7. Relevant software/datasets/other outcomes

- The Pytorch implementation can be found in <https://github.com/ManiadisG/ExCB>

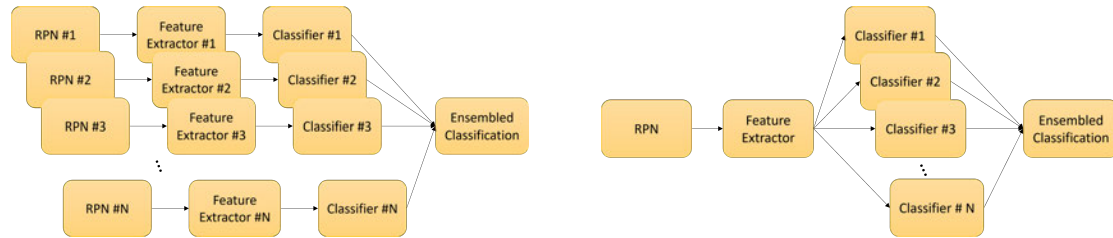
5.16. Few-shot Object Detection as a Semi-Supervised Learning Problem

Contributing partners: UPB, JR

5.16.1. Introduction

Few-shot object detection is by itself a relatively new topic, being addressed by a modest, albeit growing, number of works. Due to the difficult nature of the task, results in this field are not spectacular, leaving much room for improvement. One interesting approach to the problem is to benefit from ensembling strategies, which combine individual systems into a single setup whose performance exceeds every individual performance of its components. Ensembling efforts have been mostly made in the image classification field [457, 458, 459], or for object detection [460, 461, 462], in both cases approaching techniques such as cooperation, competition or voting schemes. Separately, different groups of researchers tackled the few-shot object detection problem [463, 464, 465, 297], with focus on meta-learning, weight sharing or fine-tuning approaches. However, there are no other works related to ensemble learning in the few-shot object detection scenario.





(a) Regular ensembling of FSOD systems.

(b) Proposed ensembling of FSOD systems.

Figure 49. Comparison between regular and proposed ensembling architectures for FSOD systems.

5.16.2. Methodology

We base our ensembling strategy on a mixture between the works of Wang et al. [297] and Dvornik et al. [457]. In the former, FSOD is introduced as a fine-tuning step on top of a pre-trained two-stage object detector. The authors argue that having a pre-trained detector, it is sufficient to freeze the entire network, except for the last two layers and perform fine-tuning solely on these layers in order to obtain improved performance. In the latter, the authors tackle the problem of image classification and argue that having several networks performing the same classification task together yields better results due to having as little as possible different random weight initialization. The authors study the impact brought by having several almost identical classifiers perform the same job, with the only difference between them being the random values used to initialize the weights in the training process.

Two-stage object detectors generally consist of two fundamental sections: the region (or object) proposal network (RPN) and the feature extractor, together with a classifier, working on top of it. Ensembling strategies usually deploy several networks and process their set of outcomes, as depicted in Figure 49a. However, this bears the cost of training N different networks with N usually being greater than 5. From the resource point of view, this type of processing is very costly, especially GPU-wise. Furthermore, the ensemble is usually distilled in order to reduce inference time, possibly at the cost of also reducing the system’s performance.

Our ensemble learning paradigm takes advantage of the fact that the framework presented in [297] freezes almost the entire two-stage object detection network. This leaves the object classifier part open for fine-tuning. Following [457], we apply a set of classifiers on top of the features extracted from the RPN’s proposed boxes and generate N classification decisions for each proposed box, as depicted in Figure 49b, thus approximately simulating an ensemble of N complete networks. Then, a regular non-maximum suppression (NMS) algorithm is applied for the resulting proposals. From this point on, the system approximates a single object detector, with enhanced detection capabilities.

5.16.3. Experimental Results

Performance-wise, our ensembling method adds a slight improvement to the detection performance of the original system [297]. To the best of our knowledge, this type of approach has not been tried before. Therefore, we compared our proposed system on the MS COCO dataset with the original work of Wang et al. [297], while keeping the evaluation protocol unchanged. We obtained an AP@0.5 of 10.1 and 13.6 on 10 and 30 shots, respectively, compared to the original results of 10.0 and 13.4, respectively. Thus, our method essentially adds a marginal improvement with virtually no additional cost incurred.

5.16.4. Conclusion

The main difference between our method and the usual ensembling strategy is that we reduce the use of resources N -fold. One could argue that our proposed system’s RPN does not behave in the same manner



as in the original case, having N times less proposed boxes to work on, but allowing the ensembled system's RPN to propose a large amount of possible objects ($>2,000$) reaches the same performance as a combination of several RPNs, since the vast majority of the proposed regions are, in fact, not objects, and are therefore redundant. Another significant advantage of our method is that it is almost free to scale. Adding another network to the ensemble is reduced to adding another classification head to the architecture, which has insignificant impact from a memory standpoint. This method adds flexibility in the sense that it can be applied to a large number of network architectures that follow this working environment. Both the classifiers and the ensembling algorithm can remain unchanged from the regular ensembling setup.

5.16.5. Relevance to AI4Media use cases and media industry applications

Few-shot object detection is particularly useful in tasks that do not possess sufficient data. This can be found especially in areas where the user would like to retrieve less common objects from a dataset, objects for which the detectors did not have enough data to train on. Leveraging the power of several DNNs, an ensemble of common object detectors improves the performance of the overall decision by intelligently selecting the best individual predictions so as to gain better results at a negligible computational price.

5.16.6. Relevant Publications

- W. Bailer, M. Dogariu, B. Ionescu, H. Fassold, "Few-shot Object Detection as a Semi-supervised Learning Problem", Proceedings of the 19th International Conference on Content-based Multimedia Indexing (CBMI), 2024.
Zenodo record: <https://zenodo.org/records/10636415>

5.16.7. Relevant software/datasets/other outcomes

The code for the framework is available at <https://github.com/wbailer/few-shot-object-detection>

5.17. Deep Learning for Image Retrieval: An Overview

Contributing partner: AUTH

5.17.1. Introduction

Over the past few years, Deep Neural Networks (DNNs) have significantly advanced and facilitated the task of searching in databases for similar data, particularly in the domain of Content-Based Image Retrieval (CBIR). DNNs have emerged as the most accurate and widely adopted approach for such tasks. While a few notable studies have explored deep image retrieval methods, these investigations have primarily concentrated on specific approaches. They often fall short of encapsulating the latest developments in the rapidly evolving field. Given the evolution of image retrieval, it is challenging yet crucial to conduct a comprehensive evaluation of the relevant studies. Therefore, the objective of this survey is to provide a concise summary of the Deep Image Retrieval concept and comprehensively gather, describe, and interpret various image retrieval methods. By doing so, we aim to enhance the understanding of the current state-of-the-art research and foster future ideas in this area.

5.17.2. Literature overview

Deep neural networks have emerged as a highly efficient and widely employed artificial intelligence technique, demonstrating exceptional performance across diverse tasks. One such significant task is information retrieval, which enables retrieving specific data items that satisfy predefined criteria from a





designated database [466]. Digital images constitute a substantial portion of multimedia. Their analysis is crucial for numerous computer vision applications in real-world scenarios. Industries and services spanning social media to autonomous systems, and remote sensing daily generate an enormous volume of digital images. Their exponential growth has stimulated profound scientific interest in the field of image retrieval. Its primary objective is to enable users to browse, search, and retrieve images from an image database that meet specific user requirements. Consequently, image retrieval has become a prominent research area within both Information Retrieval and Computer Vision domains. Due to the ever-increasing volume of image data worldwide, the significance of addressing the challenges and advancing the methodologies of image retrieval has been amplified in recent years.

Image retrieval applications are practically limitless, extending from medical image search to facial images and, social media image content search. Typically, digital $H \times W$ pixel images can have a huge dimensionality R^{HW} for large H, W . It is advantageous to reduce high image dimensionality, before conducting image search. This reduction is typically achieved through *image feature extraction* [467]. The image representation vector (also called image feature vector) $\mathbf{f}_i = f(\mathbf{x}_i, \boldsymbol{\theta})$ of database images $\mathbf{x}_i, i = 1, \dots, N$ is extracted and compared to the $\mathbf{f}_q = f(\mathbf{x}_q, \boldsymbol{\theta})$ of the query image \mathbf{x}_q . In the case of DNN features, $\mathbf{f} = f(\mathbf{x}, \boldsymbol{\theta})$ denotes a DNN having a learnable parameter vector $\boldsymbol{\theta}$. By extracting relevant features $\mathbf{f} \in R^d$ from images, such as color histograms, texture descriptors, or deep neural features the image dimensionality $\mathbf{x} \in R^m$ can be significantly reduced *i.e.*, $d \ll m$. The similarity between images $\mathbf{x}_i, \mathbf{x}_q$ can be accurately assessed based on the extracted features. By comparing the extracted image features of the query image to those of the database images, an image similarity measure $S(\mathbf{f}_q, \mathbf{f}_i)$ can be computed, enabling the retrieval of images that exhibit content-based similarity to the query image. This facilitates efficient image retrieval by reducing computational complexity and enhancing retrieval accuracy. A similarity metric is employed to evaluate appropriate image similarity (or dis-similarity) [468]. Numerous image similarity functions have been proposed *e.g.* the Euclidean distance, the cosine similarity, or the KL divergence.

Content Based Image Retrieval (CBIR) aims to identify and extract database images that exhibit visual content similarity to a given *query image* based on their content [469]. Given the feature representations of the images to be searched and the feature representation of the query image acquired through a feature extraction approach, the output of the retrieval process is a ranked set of images based on their similarity measure to the query representation, as illustrated in Figure 50. CBIR techniques can be categorized into two types, namely at category or instance level. *Category-level retrieval* aims at extracting images belonging to the same class as the query image. *Instance-level retrieval* focuses on finding images depicting a specific instance of an object or scene with the query image and not just the object class, even if the images are captured under different imaging conditions. A common framework for deep learning-based CBIR is described in Figure 51. Typically it operates through several key steps to effectively match and retrieve images based on their content. Firstly, a DNN model is trained on a large training image dataset. The model learns to extract discriminative image feature representations from input images. The features of the query image are also extracted. Subsequently, a similarity metric such as cosine similarity or Euclidean distance is employed to compute the similarity between the query image and the images in the database. This framework enables efficient and accurate retrieval of images with similar content, leveraging the power of deep learning to handle large-scale image databases and diverse visual content [470].

Since its inception, a primary objective of the image retrieval task has been to bridge the semantic gap, which refers to the disparity between low-level image representations and higher-level conceptual understanding [471]. Earlier image retrieval approaches have aimed to extract semantically rich and geometrically invariant image representations to describe the image based on its shape [472], texture [473] or color [474]. These include techniques such as Fisher Vector descriptors [475] or Scale-Invariant Feature Transform (SIFT) [476, 477], Histograms of Oriented Gradients (HOG) [478], Oriented FAST and rotated BRIEF (ORB) [479], Speeded-Up Robust Features (SURF) [480] or Local Binary Pattern (LBP) [481]. Due to the use of ImageNet [86], deep CNNs have emerged as foundational descriptors in various computer vision tasks, notably image retrieval.



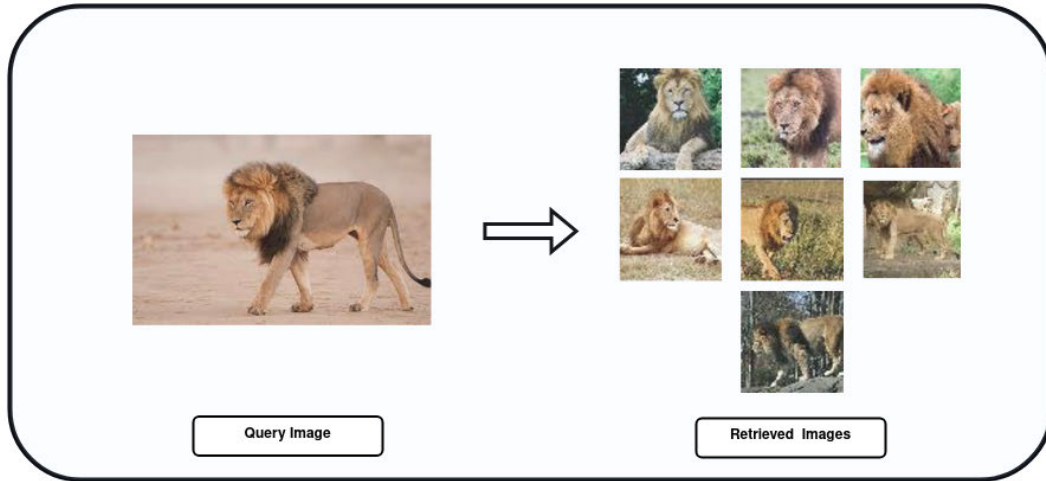


Figure 50. An example of image retrieval from an image database (Tiny ImageNet) [11]. Given the query image (left), the images on the right are retrieved.

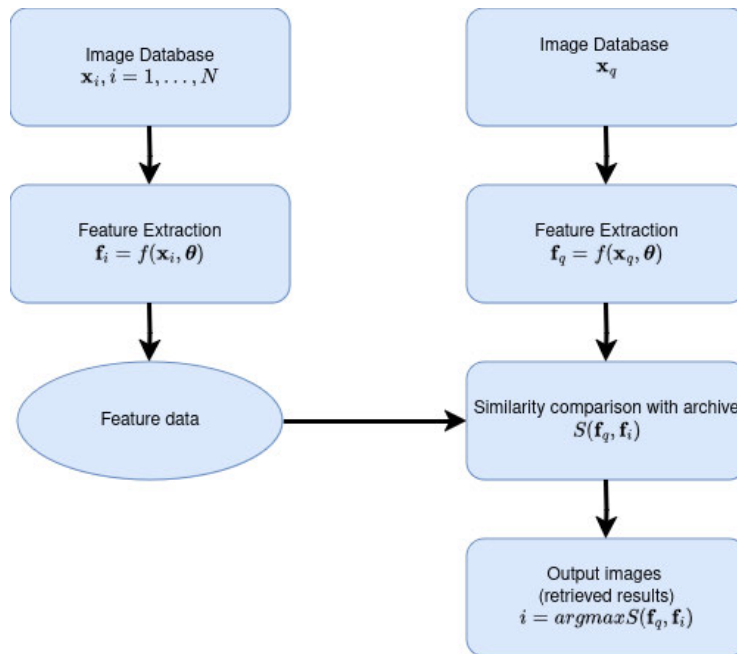


Figure 51. A deep CBIR framework.

Two prevalent types of image representations have been extensively explored: global features [482, 483, 484], which offer high-level semantic image signatures, and local features [485, 486, 487], which encapsulate discriminative geometry information about specific image regions. Global features are typically designed to be invariant to viewpoint and illumination, while local features excel in capturing local geometry and textures. Traditionally, image retrieval relied on image descriptor matching [488]. The advent of CNN-based descriptors revolutionized this approach and there has been a noticeable shift



in feature representation, transitioning from hand-engineered approaches to learning-based methods following the emergence of deep learning [489].

Utilizing neural features has been extensively explored as a promising means to bridge the semantic gap. These techniques leverage the power of machine learning algorithms to extract rich image descriptors (representations) from raw image data, enabling a more effective and accurate retrieval process. In image retrieval, representations encode image contents and measure their similarities. Image retrieval has witnessed the adoption of fully-connected layers after convolutional layers to generate global descriptors, followed by dimensionality reduction [483]. These deep CNNs global descriptors have superseded conventional hand-crafted features such as SIFT [477]. Many existing approaches leveraging deep architectures for image retrieval primarily utilize pre-trained networks as local feature extractors. Consequently, significant efforts have been directed toward employing image representations suitable for retrieval tasks atop these features. Different types of loss functions, including classification loss [485, 490] or contrastive loss [491, 492], have also been thoroughly investigated in the field. Various pooling techniques, including sum pooling [482], max pooling [484], and average pooling [493], have been employed. Some studies integrate attention mechanisms [494] to amplify the activations of crucial features on the feature map. Notably, recent research has focused on extracting representative and distinctive features for global and local representations using deep learning methods [495, 485, 496, 497, 498]. This poses challenges, as representations for retrieval must be compact while preserving intricate image details. Innovations have been made to enable deep architectures to accurately represent images of various sizes and aspect ratios [482, 484]. More recently, Vision Transformers are predominantly utilized to map images to their compact representations [15, 16, 499, 492, 500, 501]. Typically the similarity between features can be computed using distance metrics or assessed using re-ranking methodologies [502, 495]. There has been a growing research focus on approximate nearest neighbor search methods aiming at accelerating the search process [503]. For faster distance computation, hashing techniques have been employed in the past years [504, 505].

This work endeavors to present a thorough examination of the recent advancements in deep image retrieval methods. Throughout our survey, we categorize the most utilized and successful methods into six categories. Over the years, researchers have proposed several methods for similarity comparison, and thus image retrieval, using machine learning techniques, such as metric learning and representation learning. The objective of representation and metric learning is to build new spaces of representations to improve performance. Metric learning focuses on learning a distance to measure the similarity between different instances [506], while representation learning aims at learning meaningful representations from data that can be used later for comparison [507]. Representation learning aims at learning a projection function that can transform the data points in the original space to a discriminative space where points from the same class will be gathered together, and points from different classes will be pushed far apart. In feature aggregation, aggregated features are utilized for training the DNNs [508], [509]. These methods involve embedding convolutional features into a high-dimensional space to generate more compact and discriminative feature representations [510]. Attention mechanisms, enable the DNN to focus on the most relevant image parts during feature extraction by computing an attention map, thus enhancing the discriminative power of the extracted features. Hashing algorithms have been proposed as a solution for large-scale image search and retrieval due to their computational and storage efficiency. These algorithms map images to compact binary codes while preserving the underlying data structure. By employing hashing, computational requirements are reduced, leading to faster retrieval times in various information retrieval-based applications. In the context of image retrieval, DNN fine-tuning involves taking a pre-trained DNN that has been trained on a source dataset and retraining it on a new target dataset specifically for image retrieval tasks. This fine-tuning process helps adapt the DNNs to the characteristics and requirements of the new datasets, hence improving their performance in image retrieval.





5.17.3. Relevance to AI4Media use cases and media industry applications

This work is relevant to UC3 (AI in Vision - High quality Video Production and Content Automation) and UC7 (AI for Content Organization and Content Moderation) since it provides an overview of advanced deep learning image retrieval methods and thus, offers information regarding novel solutions to analyze visual content.

5.17.4. Relevant Publications

- I. Valsamara, and I. Pitas, "Deep Learning for Image Retrieval: An Overview", Under review

5.18. Solutions to large scale Video Browsing and Retrieval

Contributing partner: CNR

5.18.1. Introduction

With the pervasive use of digital cameras and social media platforms, we witness a massive daily production of multimedia content, especially videos and photos. This phenomenon poses several challenges for the management and retrieval of visual archives. On one hand, the use of content-based retrieval systems and automatic data analysis is crucial to deal with visual data that typically are poorly-annotated (think for instance of user-generated content). On the other hand, there is an increasing need for scalable systems and algorithms to handle ever-larger collections of data.

We developed a video search system, named VISIONE, which provides users with various functionalities to easily search for targeted videos. It relies on artificial intelligence techniques to automatically analyze and annotate visual content and employs an efficient and scalable search engine to index and search for videos.

5.18.2. Methodology

VISIONE integrates several search functionalities that allow a user to search for a target video segment by formulating textual and visual queries, which can be also combined with a temporal search. In particular it supports *free text search*, *spatial color and object search*, *visual similarity search*, and *semantic similarity search*. The system architecture is summarized in Figure 52, while a screenshot of the user interface is shown in Figure 53. We already reported about first versions of VISIONE in D5.1. Here we report the features of latest versions.

To support the free text search and the semantic similarity search, we employ two cross-modal feature extractors based on, respectively, CLIP2Video [511] and ALADIN [512] pre-trained models. For the object detection, we use three models: VfNet[513] (trained on COCO dataset), Mask R-CNN [514] (trained on LVIS dataset), and a Faster R-CNN+Inception ResNet V2²⁰ (trained on the Open Images V4). The color annotation process relies on two chip-based color naming techniques [515, 516]. Finally, the visual similarity search is based on comparing GEM [517] features. We employ two indexes: the first to store the CLIP2Video features (searched using the Facebook FAISS library²¹), and the second to store all the other descriptors (searched using Apache Lucene²²). Note that to index the extracted descriptors with Lucene, we designed special text encodings, based on the Surrogate Text Representations (STRs) approach [518, 519, 520].

²⁰http://tfhub.dev/google/faster_rcnn/openimages_v4/inception_resnet_v2/1

²¹<https://github.com/facebookresearch/faiss>

²²<https://lucene.apache.org/>



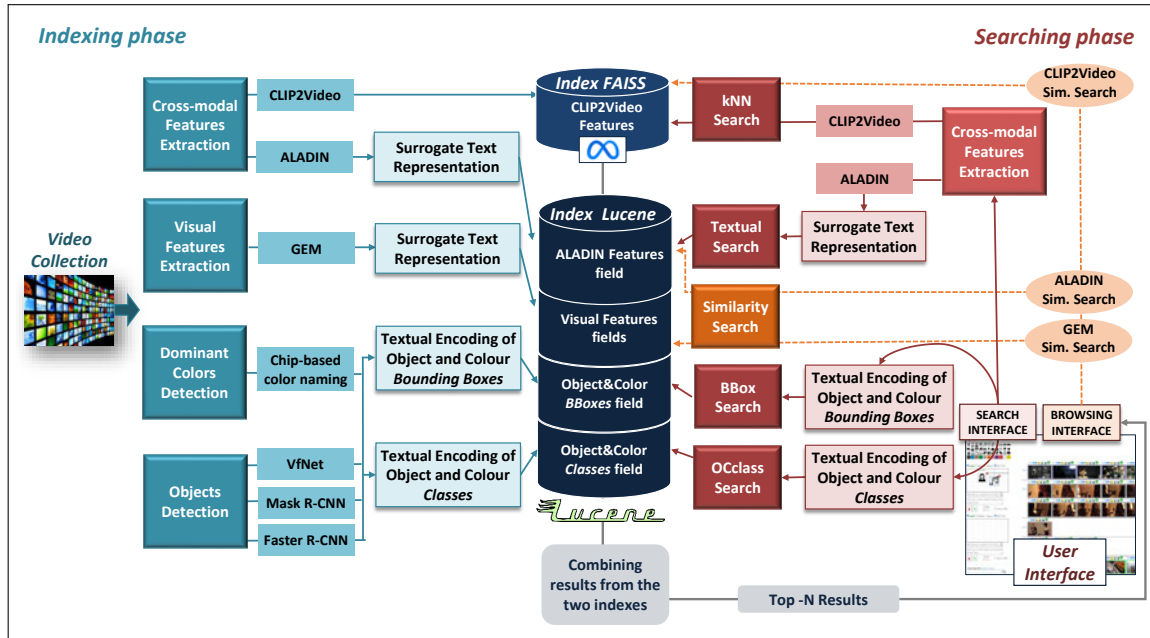


Figure 52. VISIONE System Architecture

VISIONE has improved over the years, during the project lifetime and now it includes, among the others, the following advanced features:

- ALADIN for text-to-image retrieval.** We developed a new cross-modal retrieval deep neural network, called ALADIN (ALign And DIstill Network) [512]. ALADIN first produces high-effective scores by aligning at fine-grained level images and texts. Then, it learns a shared embedding space – where an efficient kNN search can be performed – by distilling the relevance scores obtained from the fine-grained alignments. We empirically found that this network is able to compete with state-of-the-art vision-language Transformers while being almost 90 times faster at inference time.
- CLIP2Video for text-to-video retrieval.** In order to deeply understand videos, in particular temporal correlations and actions among multiple frames of a shot, we use CLIP2Video [511], which is one of the state-of-the-art networks for text-to-video retrieval. We re-engineered the code for easily extracting fixed-sized descriptors for texts and images that can be compared with cosine similarity. We found some problems in post-processing these features using our STR representation for textual-based indexing [519]. In particular, looking at Figure 54, we noticed that the distribution of the cosine similarities of the CLIP2Video features has a very low mean value in the text-to-video cross-modal setup. This may happen if element-wise products underlying the dot-product computation have a negative sign, which in turn implies that there could be a lot of mixed-sign factors. This is a bad scenario for the STR representation, given that the CReLU operation at the core of the STR method zeroes out the contribution from mixed-sign factors. Therefore, for the CLIP2Video features, the approximated cosine similarity computed in the STR representation badly approximates the original one. For these reasons, we indexed and searched these cross-modal features with FAISS, using an exact search and an 8-bit scalar quantization for reducing the index size in memory. The visual features extracted using CLIP2Video are also employed for a semantic reverse video search, where a video segment displayed in the results can be used as a query to search other video clips semantically similar to it.
- Browsing Interface.** To improve the user’s browsing experience, we included the possibility of

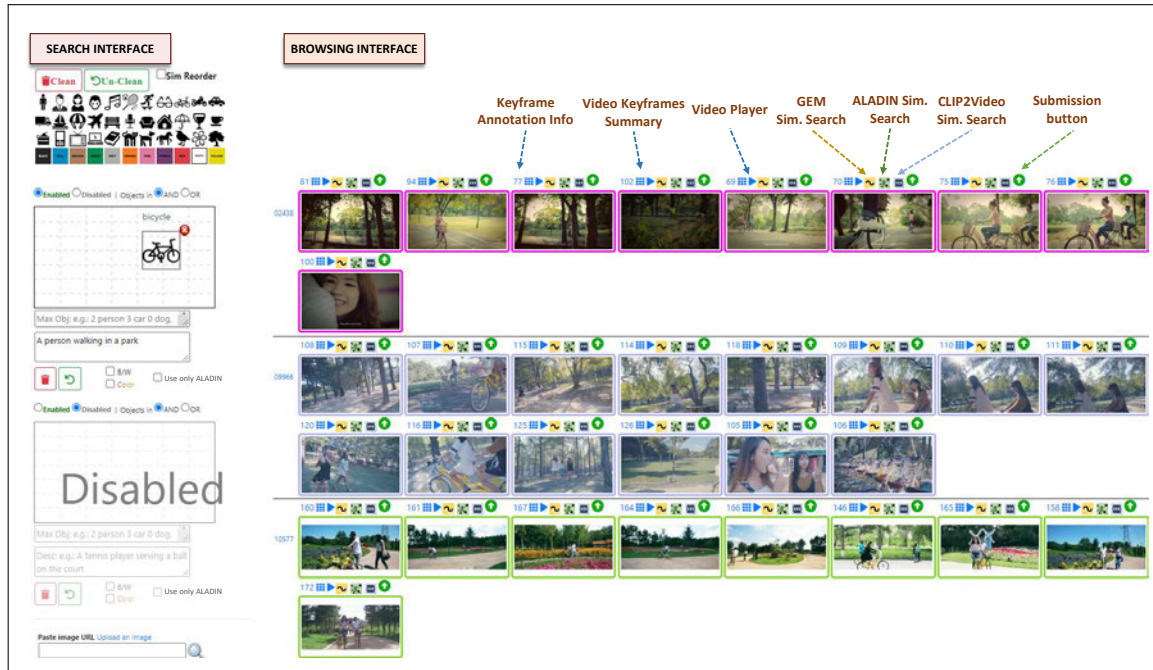


Figure 53. VISIONE User Interface

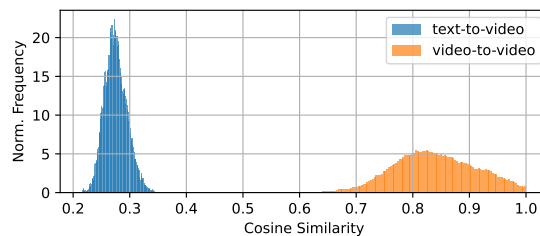


Figure 54. Distribution of cosine similarities between text-video and video-video CLIP2Video features.

displaying a short preview of a video clip by right-clicking on one of the results displayed in the user interface. We also introduced multiple selections of frames to submit during Ad-Hoc Video Search (AVS) tasks and the ability to submit a given instant of a video directly from the video player.

- **Object search.** We used three object detectors trained on three different datasets (COCO, LVIS, and Open Images V4), which have different classes. We built a mapping of these classes using a semi-automatic procedure in order to have a unique final list of 1,460 classes²³. We also generated a hierarchy for each class, using wordnet²⁴, which is used for query expansion both at index time and at runtime.

VISIONE was integrated by CNR and RAI within use case UC3 (3A3) to index a dataset of videos provided by RAI (<http://visione.isti.cnr.it/>). VISIONE, specifically its reverse image search functionality, was also integrated by CNR and ATC within use case UC1, to search for images that might lead to misinformation.

²³<https://doi.org/10.5281/zenodo.7194300>

²⁴<https://wordnet.princeton.edu/>



5.18.3. Experimental Results

CNR has participated in the Video Browser Showdown (VBS) 2024 with VISIONE version 5.0, and VISIONE was the Overall Winner of the competition (Best AVS/Experts, AVS/Novices, Visual KIS/Novices, QA/Experts) as can be seen in <https://videobrowsershowdown.org/hall-of-fame/>.

5.18.4. Relevance to AI4Media use cases and media industry applications

This activity is related to UC3 (AI in Vision - High quality Video Production and Content Automation), where it can be used to manage large audio-visual archives offering advanced retrieval functionalities. It is also related to UC1 (AI against Disinformation), where it can be used to build a reverse image retrieval system to verify if query images are contained in databases of images known to be used to spread disinformation. Also UC7 (AI for (re-)organization and content moderation) can benefit from this solution, to organize and manage content.

5.18.5. Relevant Publications

- Jakub Lokoč, Stelios Andreadis, Werner Bailer, Aaron Duane, Cathal Gurrin, Zhixin Ma, Nicola Messina, Thao-Nhu Nguyen, Ladislav Peška, Luca Rossetto, Loris Sauter, Konstantin Schall, Klaus Schoeffmann, Omar Shahbaz Khan, Florian Spiess, Lucia Vadicamo, Stefanos Vrochidis, "Interactive Video Retrieval in the Age of Effective Joint Embedding Deep Models: Lessons from the 11th VBS," Springer Multimedia Systems, 2023
- Giuseppe Amato, Paolo Bolettieri, Fabio Carrara, Fabrizio Falchi, Claudio Gennaro, Nicola Messina, Lucia Vadicamo, Claudio Vairo, VISIONE: A Large-Scale Video Retrieval System with Advanced Search Functionalities, ICMR '23: Proceedings of the 2023 ACM International Conference on Multimedia Retrieval, June 2023, Pages 649–653, <https://doi.org/10.1145/3591106.3592226>
- Lucia Vadicamo, Claudio Gennaro, & Giuseppe Amato. (2021). "On Generalizing Permutation-Based Representations for Approximate Search". International Conference on Similarity Search and Applications (SISAP), https://doi.org/10.1007/978-3-030-89657-7_6
- Silvan Heller, Viktor Gsteiger, Werner Bailer, Cathal Gurrin, Björn Þór Jónsson, Jakub Lokoc, Andreas Leibetseder, Frantisek Mejzlík, Ladislav Peska, Luca Rossetto, Konstantin Schall, Klaus Schoeffmann, Heiko Schuldt, Florian Spiess, Ly-Duyen Tran, Lucia Vadicamo, Patrik Veselý, Stefanos Vrochidis, Jiaxin Wu: "Interactive Video Retrieval Evaluation at a Distance: Comparing Sixteen Interactive Video Search Systems in a Remote Setting" at the 10th Video Browser Showdown, International Journal of Multimedia Information Retrieval, 2022
- Carrara, F., Vadicamo, L., Gennaro, C., Amato, G. (2022). Approximate Nearest Neighbor Search on Standard Search Engines. In: Skopal, T., Falchi, F., Lokoč, J., Sapino, M.L., Bartolini, I., Patella, M. (eds) Similarity Search and Applications. SISAP 2022. Lecture Notes in Computer Science, vol 13590. Springer, Cham.
- J. Lokoc, L. Rossetto, W. Bailer, K. Schoeffmann, S. Vrochidis, C. Gurrin, S. Heller, L. Vadicamo, K.U. Barthel, L. Peška, J. Wu, B.Þ. Jonsson. "A Task Category Space for User-Centric Comparative Multimedia Search Evaluations", MMM 2022
- Amato, G., Bolettieri, P., Falchi, F., ...Vadicamo, L., Vairo, C, "VISIONE at Video Browser Showdown 2021", 27th International Conference on MultiMedia Modeling, MMM 2021; Prague;Czech Republic; 22 June 2021 through 24 June 2021; Code 254419, Volume 12573 LNCS, 2021, Pages 473-478
- Giuseppe Amato, Paolo Bolettieri, Fabio Carrara, Fabrizio Falchi, Claudio Gennaro, Nicola Messina, Lucia Vadicamo, and Claudio Vairo. 2023. VISIONE for newbies: an easier-to-use video retrieval system. In Proceedings of the 20th International Conference on Content-based Multimedia Indexing (CBMI '23). Association for Computing Machinery, New York, NY, USA, 158–162. <https://doi.org/10.1145/3617233.3617261>



- Carrara, F., Gennaro, C., Vadicamo, L., Amato, G. (2023). Vec2Doc: Transforming Dense Vectors into Sparse Representations for Efficient Information Retrieval. In: Pedreira, O., Estivill-Castro, V. (eds) Similarity Search and Applications. SISAP 2023. Lecture Notes in Computer Science, vol 14289. Springer, Cham. https://doi.org/10.1007/978-3-031-46994-7_18
- Lucia Vadicamo, Giuseppe Amato, Claudio Gennaro, Induced permutations for approximate metric search, Information Systems, Volume 119, 2023, 102286, ISSN 0306-4379, <https://doi.org/10.1016/j.is.2023.102286>
- L. Vadicamo, R. Arnold, W. Bailer, F. Carrara, C. Gurrin, N. Hezel, X. Li, J. Lokoc, S. Lubos, Z. Ma, N. Messina, T.-N. Nguyen, L. Peska, L. Rossetto, L. Sauter, K. Schöffmann, F. Spiess, M.-T. Tran, S. Vrochidis, "Evaluating Performance and Trends in Interactive Video Retrieval: Insights from the 12th VBS Competition," IEEE Access, May 2024.
- Nicola Messina, Giuseppe Amato, Andrea Esuli, Fabrizio Falchi, Claudio Gennaro, Stéphane Marchand-Maillet, "Fine-grained visual textual alignment for cross-modal retrieval using transformer encoders", ACM Transactions on Multimedia Computing, Communications, and Applications, Volume 17, Issue 4, November 2021 Article No.: 128 pp 1–23, <https://doi.org/10.1145/3451390>
- Messina, N., Falchi, F., Esuli, A., Amato, G., "Transformer reasoning network for image-text matching and retrieval", 25th International Conference on Pattern Recognition (ICPR)
- Nicola Messina, Fabrizio Falchi, Claudio Gennaro, Giuseppe Amato, "AIMH at SemEval-2021 Task 6: Multimodal Classification Using an Ensemble of Transformer Models", Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)
- Nicola Messina, Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, Fabrizio Falchi, Giuseppe Amato, Rita Cucchiara, ALADIN: Distilling Fine-grained Alignment Scores for Efficient Image-Text Matching and Retrieval, Proceedings of the 19th International Conference on Content-based Multimedia Indexing, September 14–16, 2022, Graz, Austria
- W. Bailer, R. Arnold, V. Benz, D. A. Coccomini, A. Gkagkas, G. P. Guomundsson, S. Heller, B. P. Jonsson, J. Lokoc, N. Messina, N. Pantelidis, J. Wu, "Improving Query and Assessment Quality in Text-Based Video Retrieval Evaluation," ACM International Conference on Multimedia Retrieval (ICMR), Thessaloniki, GR, June 2023.
- Messina, N., Amato, G., Carrara, F., Gennaro, C., Falchi, F. (2022). Recurrent Vision Transformer for Solving Visual Reasoning Problems. In: Sclaroff, S., Distanto, C., Leo, M., Farinella, G.M., Tombari, F. (eds) Image Analysis and Processing – ICIAP 2022. ICIAP 2022. Lecture Notes in Computer Science, vol 13233. Springer, Cham. https://doi.org/10.1007/978-3-031-06433-3_5

5.18.6. Relevant software/datasets/other outcomes

- VISIONE demo: <http://visione.isti.cnr.it/>
- VISIONE integrated with RAI videos: <http://visione.isti.cnr.it/>
- GitHub repository of the VISIONE project: <https://github.com/aimh-lab/visione>

5.19. DnS: Distill-and-Select for Efficient and Accurate Video Indexing and Retrieval

Contributing partners: QMUL, CERTH

5.19.1. Introduction and methodology

Due to the popularity of Internet-based video sharing services, the volume of video content on the Web has reached unprecedented scales. For instance, YouTube reports that more than 500 hours of content





are uploaded every minute²⁵. This puts considerable challenges for all video analysis problems, such as video classification, action recognition, and video retrieval, that need to achieve high performance at low computational and storage requirements in order to deal with the large scale of the data. The problem is particularly hard in the case of content-based video retrieval, where, given a query video, one needs to calculate its similarity with all videos in a database to retrieve and rank the videos based on relevance. In such scenario, this requires efficient indexing, i.e., storage of the representations extracted from the videos in the dataset, and fast calculations of the similarity between pairs of them.

In this work, we propose to address the problem of high retrieval performance and computationally efficient content-based video retrieval in large-scale datasets. The proposed method builds on the framework of Knowledge Distillation, and starting from a well-performing, high-accuracy-high-complexity teacher, namely a fine-grained video similarity learning method (ViSiL), trains a) both fine-grained and coarse-grained student networks on a large-scale unlabeled dataset and b) a selection mechanism, i.e., a learnable re-ranking module, that decides whether the similarity estimated by the coarse-grained student is accurate enough, or whether the fine-grained student needs to be invoked. By contrast to other re-ranking methods that use a threshold on the similarity estimated by the fast network (the coarse-grained student in our case), our selection mechanism is a trainable, lightweight neural network. All networks are trained so as to extract representations that are stored/indexed, so that each video in the database is indexed by the fine-grained spatio-temporal representation (3D tensor), its global, vector-based representation (1D vector), and a scalar self-similarity measure that is extracted by the feature extractor of the selector network, and can be seen as a measure of the complexity of the videos in question. The latter is expected to be informative of how accurate the coarse-grained, video-level similarity is, and together with the similarity rapidly estimated by the coarse-grained representations, is used as input to the selector. We note that, by contrast to other Knowledge Distillation methods in videos that address classification problems and typically perform distillation at intermediate features, the students are trained on a similarity measure provided by the teacher – this allows training on large scale datasets as intermediate features of the networks do not need to be stored, or estimated multiple times. Due to the ability to train on large unlabeled datasets, more complex models, i.e., with more trainable parameters, can be employed leading to even better performance than the original teacher network.

Figure 55 depicts the DnS framework. It consists of three networks: (i) a coarse-grained student (\mathbf{S}^c) that provides very fast retrieval speed but with low retrieval performance, (ii) a fine-grained student (\mathbf{S}^f) that has high retrieval performance but with high computational cost, and (iii) a selector network (\mathbf{SN}) that routes the similarity calculation of the video pairs and provides a balance between performance and time efficiency.

Each video in the dataset is stored/indexed using three representations: (i) a spatio-temporal 3D tensor $f^{\mathbf{S}^f}$ that is extracted (and then used at retrieval time) by the fine-grained student \mathbf{S}^f , (ii) a 1D global vector $f^{\mathbf{S}^c}$ that is extracted (and then used at retrieval time) by the coarse-grained student \mathbf{S}^c , and (iii) a scalar $f^{\mathbf{SN}}$ that summarises the similarity between different frames of the video in question that is extracted (and then used at retrieval time) by the selector network \mathbf{SN} . The indexing process that includes the feature extraction is illustrated within the blue box in Figure 55 and is denoted as $f^{\mathbf{X}}(\cdot)$ for each network \mathbf{X} . At retrieval-time, given an input query-target video pair, the selector network sends to the coarse-grained student \mathbf{S}^c the global 1D vectors so that their similarity is rapidly estimated (i.e., as the dot product of the representations) $g^{\mathbf{S}^c}$. This coarse similarity and the self-similarity scalars for the videos in question are then given as input to the selector \mathbf{SN} , which takes a binary decision $g^{\mathbf{SN}}$ on whether the calculated coarse similarity needs to be refined by the fine-grained student. For the small percentage of videos that this is needed, the fine-grained network calculates the similarity $g^{\mathbf{S}^f}$ based on the spatio-temporal representations. The retrieval process that includes the similarity calculation is illustrated within the red box in Figure 55 and is denoted as $g^{\mathbf{X}}(\cdot, \cdot)$ for each network \mathbf{X} .

²⁵<https://www.youtube.com/yt/about/press/>, accessed June 2021



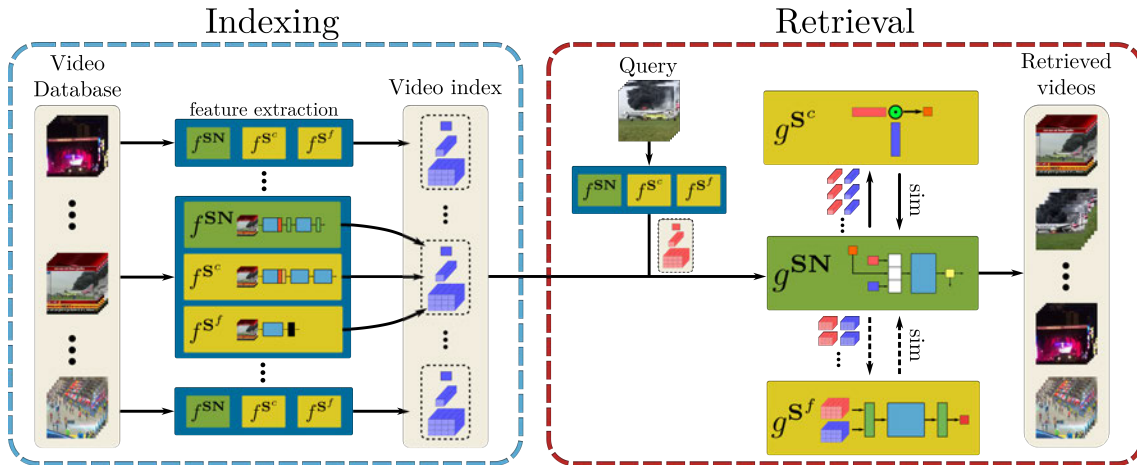


Figure 55. Overview of the proposed framework. It consists of three networks: a coarse-grained student S^c , a fine-grained student S^f , and a selector network SN . Processing is split into two phases, Indexing and Retrieval. During indexing (blue box), given a video database, three representations needed by our networks are extracted and stored in a video index, i.e., for each video, we extract a 3D tensor, a 1D vector, and a scalar that captures video self-similarity. During retrieval (red box), given a query video, we extract its features, which, along with the indexed ones, are processed by the SN . It first sends all the 1D vectors of query-target pairs to S^c for an initial similarity calculation. Then, based on the calculated similarity and the self-similarity of the videos, the selector network judges which query-target pairs have to be re-ranked with the S^f , using the 3D video tensors. Straight lines indicate continuous flow, i.e., all videos/video pairs are processed, whereas dashed lines indicate conditional flow, i.e., only a number of selected videos/video pairs are processed. Our students are trained with Knowledge Distillation based on a fine-grained teacher network, and the selector network is trained based on the similarity difference between the two students.

In practice, we apply the above process on every query-target video pair derived from a database, and a predefined percentage of videos with the largest confidence score calculated by the selector is sent to the fine-grained student for re-ranking. With this scheme, we achieve very fast retrieval with very competitive retrieval performance.

5.19.2. Experimental results

In Table 58, the mAP of the proposed method in comparison to the video retrieval methods from the literature is reported. The proposed students achieve very competitive performance achieving state-of-the-art results in several cases. First, the fine-grained attention student achieves the best results on the two large-scale datasets, i.e., FIVR-200K and SVD, outperforming ViSiL (our teacher network) by a large margin, i.e., 0.022 and 0.021 mAP, respectively. It reports almost the same performance as ViSiL on the CC_WEB_VIDEO dataset, and it is slightly outperformed on the EVVE dataset. Additionally, it is noteworthy that the fine-grained binarization student demonstrates very competitive performance on all datasets. It achieves similar performance with ViSiL and the fine-grained attention student on the CC_WEB_VIDEO, the second-best results on all three tasks of FIVR-200K, and the third-best on SVD with a small margin from the second-best. However, its performance is lower than the teacher’s on the EVVE dataset, highlighting that feature reduction and hashing have considerable impact on the student’s retrieval performance on this dataset. Also, another possible explanation for this performance difference could be that the training dataset does not cover the included events sufficiently.

Second, the coarse-grained student exhibits very competitive performance among coarse-grained approaches on all datasets. It achieves the best mAP on two out of four evaluation datasets, i.e., on SVD and EVVE, reporting performance close or even better than several fine-grained methods. On FIVR-200K and CC_WEB_VIDEO, it is outperformed by the BoW-based approaches, which are trained with



samples from the evaluation sets. However, when they are built with video corpora other than the evaluation (which simulates more realistic scenarios), their performance drops considerably [371, 361]. Also, their performance on the SVD and EVVE datasets is considerably lower.

Third, our DnS runs maintain competitive performance. It improves the performance of the coarse-grained student by more than 0.2 on FIVR-200K and 0.02 on SVD by re-ranking only 5% of the dataset with the fine-grained students. However, on the other two datasets, i.e., CC_WEB_VIDEO and EVVE, the re-ranking has negative effects on performance. A possible explanation for this might be that the performance of the coarse- and fine-grained students is very close, especially on the EVVE dataset. Also, this dataset consists of longer videos than the rest, which may impact the selection process. Nevertheless, the performance drop on these two datasets is mitigated when 30% of the dataset is sent to the fine-grained students for re-ranking; while on the FIVR-200K and SVD, the DnS method reaches the performance of the corresponding fine-grained students, or it even outperforms them, i.e., $\text{DnS}_{\mathcal{B}}^{30\%}$ outperforms $\text{S}_{\mathcal{B}}^f$ on SVD dataset.

Additionally, Table 59 displays the storage and time requirements and the reference performance of the proposed method on each dataset. In comparison, we include the video retrieval methods that are implemented with the same features and run on GPU. For FIVR-200K and CC_WEB_VIDEO datasets, we display the DSVR and cc_web_c^* runs, respectively. We have excluded the TN and DP methods, as they have been implemented on CPU and their transfer to GPU is non-trivial. Also, the requirements of the TCA runs from [372] are approximated based on features of the same dimensionality. All times are measured on a Linux machine with the Intel i9-7900X CPU and an Nvidia 2080Ti GPU.

	Approach	FIVR-200K			CC_WEB_VIDEO				SVD	EVVE
		DSVR	CSVR	ISVR	cc_web	cc_web^*	cc_web_c	cc_web_c^*		
Coarse-grained	ITQ [†] [521]	0.491	0.472	0.402	0.967	0.941	0.976	0.954	0.842	0.606
	MFH [†] [522]	0.525	0.507	0.424	0.972	0.950	0.981	0.967	0.849	0.604
	CSQ [523]	0.267	0.252	0.219	0.948	0.899	0.954	0.909	0.364	0.400
	BoW [†] [524]	0.699	0.674	0.581	0.975	0.958	0.985	0.977	0.568	0.539
	LBoW [525]	0.700	0.666	0.566	0.976	0.960	0.984	0.975	0.756	0.500
	DML [371]	0.411	0.392	0.321	0.971	0.941	0.979	0.959	0.785	0.531
	DML [†] [371]	0.503	0.487	0.410	0.971	0.951	0.979	0.965	0.850	0.611
	R-UTS-GV [526]	0.509	0.498	0.432	-	-	-	-	-	-
	TCA _f [†] [372]	0.570	0.553	0.473	0.973	0.947	0.983	0.965	-	0.598*
	S ^c (Ours)	0.574	0.558	0.476	0.972	0.952	0.980	0.967	0.868	0.636
Fine-grained	TMK [†] [527]	0.524	0.507	0.425	0.977	0.959	0.986	0.975	0.863	0.618
	LAMV [364]	0.496	0.466	0.371	0.975	0.956	0.986	0.975	0.781	0.531
	LAMV [†] [364]	0.619	0.587	0.479	0.978	0.964	0.988	0.982	0.880	0.620
	TN [†] [528]	0.844	0.804	0.660	0.982	0.970	0.993	0.989	0.894	0.471
	DP [†] [529]	0.827	0.783	0.642	0.980	0.966	0.991	0.987	0.880	0.580
	R-UTS-FRP [526]	0.769	0.724	0.611	-	-	-	-	-	-
	A-DML [530]	0.627	-	-	0.964	0.949	-	-	0.885*	-
	TCA _f [†] [372]	0.877	0.830	0.703	0.983	0.969	0.994	0.990	-	0.603*
	ViSiL [358]	0.899	0.854	0.723	0.985	0.973	0.995	0.992	0.881	0.658
	S _A ^f (Ours)	0.921	0.875	0.741	0.984	0.973	0.995	0.992	0.902	0.651
	S _B ^f (Ours)	0.909	0.863	0.729	0.984	0.974	0.995	0.991	0.891	0.640
	Re-ranking	PPT [529]	-	-	-	0.959	-	-	-	-
HM [531]		-	-	-	0.977	-	-	-	-	-
TMK [†] +QE [527]		0.580	0.564	0.480	0.977	0.960	0.986	0.976	0.774	0.648
LAMV [†] +QE [364]		0.659	0.629	0.520	0.979	0.964	0.990	0.984	0.786	0.653
DnS _A ^{5%} (Ours)		0.874	0.829	0.699	0.972	0.951	0.983	0.969	0.895	0.594
DnS _A ^{5%} (Ours)		0.862	0.817	0.687	0.970	0.948	0.981	0.967	0.884	0.584
DnS _B ^{30%} (Ours)		0.913	0.868	0.733	0.978	0.958	0.990	0.977	0.902	0.646
DnS _B ^{30%} (Ours)		0.900	0.854	0.720	0.977	0.954	0.988	0.974	0.894	0.634

Table 58. *mAP* comparison of our proposed students and re-ranking method against several video retrieval methods on four evaluation datasets. [†] indicates that the runs are implemented with the same features extracted with the same process as ours. * indicates that the corresponding results are on different dataset split.

5.19.3. Conclusion

In this work, we proposed a video retrieval framework based on Knowledge Distillation that addresses the problem of performance-efficiency trade-off focused on large-scale datasets. In contrast to typical





	Approach	FIVR-200K			CC WEB VIDEO			SVD			EVVE	
		mAP	KB	Sec	mAP	KB	Sec	mAP	KB	Sec	mAP	KB
Coarse-grained	ITQ [521]	0.491	0.1	0.733	0.954	0.1	0.045	0.842	0.1	1.793	0.606	0.1
	MFH [522]	0.525	0.1	0.733	0.967	0.1	0.045	0.849	0.1	1.793	0.604	0.1
	BoW [524]	0.699	0.3	1.540	0.977	0.3	0.053	0.568	0.1	3.308	0.539	0.2
	DML [371]	0.503	2	0.769	0.965	2	0.047	0.850	2	1.915	0.611	2
	TCA _c [372]	0.570	4	0.812	0.965	4	0.047	-	-	-	0.598*	4
	S ^c (Ours)	0.574	4	0.812	0.967	4	0.047	0.868	4	1.920	0.636	4
Fine-grained	TMK [527]	0.524	256	119.8	0.975	256	6.949	0.863	256	282.5	0.618	256
	LAMV [364]	0.619	256	167.2	0.975	256	9.703	0.880	256	394.5	0.620	256
	TCA _f [372]	0.877	438	36.15	0.990	596	2.097	-	-	-	0.603*	932
	ViSIL [358]	0.899	15124	451.9	0.992	20111	24.26	0.881	2308	319.8	0.658	31457
	S _A ^f (Ours)	0.921	2016	149.1	0.992	2682	8.260	0.902	308	271.8	0.651	4194
	S _B ^f (Ours)	0.909	63	146.9	0.991	84	8.129	0.891	10	266.5	0.640	131
Re-ranking	TMK+QE [527]	0.580	256	239.6	0.976	256	13.90	0.774	256	576.0	0.648	256
	LAMV+QE [364]	0.659	256	334.4	0.984	256	19.41	0.786	256	766.0	0.653	256
	DnS _B ^{5%} (Ours)	0.874	2020	8.267	0.969	2686	0.463	0.895	312	15.41	0.594	4198
	DnS _B ^{5%} (Ours)	0.862	67	8.154	0.967	88	0.456	0.884	14	15.14	0.584	135
	DnS _B ^{30%} (Ours)	0.913	2020	45.55	0.974	2686	2.528	0.902	312	83.36	0.646	4198
	DnS _B ^{30%} (Ours)	0.900	67	44.87	0.974	88	2.489	0.894	14	81.76	0.634	135

Table 59. Performance in mAP, storage in KiloBytes (KB) and time in Seconds (Sec) requirements of our proposed students and re-ranking method and several video retrieval implemented with the same features. * indicates that the corresponding results are on different dataset split.

video retrieval methods that rely on either a high-performance but resource demanding fine-grained approach or a computationally efficient but low-performance coarse-grained one, we introduced a Distill-and-Select approach. Several student networks were trained via a Teacher-Student setup at different performance-efficiency trade-offs. We experimented with two fine-grained students, one with a more elaborate attention mechanism that achieves better performance and one using a binarization layer offering very high performance with significantly lower storage requirements. Additionally, we trained a coarse-grained student that provides very fast retrieval with low storage requirements but at a high cost in performance. Once the students were trained, we combined them using a selector network that directs samples to the appropriate student in order to achieve high performance with high efficiency. It was trained based on the similarity difference between a coarse-grained and a fine-grained student so as to decide at query-time whether the similarity calculated by the coarse-grained one is reliable or the fine-grained one needs to be applied. The proposed method has been benchmarked to a number of content-based video retrieval datasets, where it improved the state-of-art in several cases and achieved very competitive performance with a remarkable reduction of the computational requirements.

5.19.4. Relevance to AI4Media use cases and media industry applications

We propose a knowledge distillation based video retrieval framework that can be relevant in tasks such as visual indexing and search and visual concepts classification.

5.19.5. Relevant publications

- Kordopatis-Zilos, G., Tzelepis, C., Papadopoulos, S., Kompatsiaris, I., & Patras, I. (2022). Dns: Distill-and-select for efficient and accurate video indexing and retrieval. International Journal of Computer Vision, 130(10), 2385-2407, <https://dl.acm.org/doi/10.1007/s11263-022-01651-3>.

5.19.6. Relevant software/datasets/other outcomes

Code is available at <https://github.com/mever-team/distill-and-select>.





6. Language analysis in Media

6.1. Overview

Pre-trained word embeddings (WE) have been the standard way of initializing Natural Language Processing (NLP) neural models. **Task 5.4 (T5.4)** “Language analysis in Media” focuses on automatic language analysis in the media sector and develops methods to improve Natural Language Processing performance and adapt language models to specialized domains that can be directly useful in media organizations and consumers. Some of the main challenges in this field are: (1) the ever-growing number of new topics and public personalities that emerge in the news and that need to be detected by the algorithms; (2) the fine-grained opinions expressed in those documents that need to be accounted for when performing document retrieval.

6.2. MAD-TSC: A Multilingual Aligned News Dataset for Target-dependent Sentiment Classification

Contributing partner: CEA

6.2.1. Introduction

Text analysis needs to address both *objective* aspects, such as topic extraction, and *subjective* aspects, such as sentiment and opinion classification. In spite of recent progress brought by the introduction of large language models [532, 533, 534], sentiment classification remains a challenging task. Expression of sentiments varies according to the data sources, languages, and domain of the texts. These challenges are particularly important in target-dependent sentiment classification (TSC), which focuses on determining the sentiment expressed toward a given entity in a given context. The bulk of TSC-related research efforts are focused on major languages. This focus is an effect of the availability of generic and task-specific resources in these languages [535, 536]. A majority of datasets are monolingual, and when they are multilingual [537, 538, 539, 540, 541], the examples are not aligned across languages. Equally, a majority of existing datasets and methods are devised for texts such as tweets, reviews, or comments [542, 540, 541] in which sentiment is most often expressed explicitly. Somewhat surprisingly, less attention is given to TSC in the news, despite the usefulness of automatic analysis of their content for the understanding of societally impactful processes such as disinformation or polarization [543].

6.2.2. Methodology

Our main contribution is the introduction of MAD-TSC, the first large multilingual aligned dataset for TSC in news articles. It includes 5,110 annotated entities mentioned in 4,714 unique sentences. Each sentence has professionally translated and aligned versions in eight languages (English, Spanish, German, Italian, French, Portuguese, Dutch, and Romanian). Sentences originate from 286 news sources published in over 30 countries. These characteristics differentiate the proposed dataset from existing resources, and in particular from NewsMTSC [543], a monolingual dataset focused on American politics, which is the closest to MAD-TSC. We discuss the main steps of the dataset creation methodology below.

Data Sources. Voxeurop²⁶ is a multilingual news website that aims to offer interesting and high-quality news to European audiences. The content is translated by professional translators, thus ensuring high-quality texts in all available languages. The content is published using a Creative Commons BY-NC-ND, an open license that facilitates its redistribution and reuse for non-commercial purposes, which will also be used to distribute MAD-TSC. We have collected 7,370 news articles with translations

²⁶<https://voxeurop.eu>



in all eight languages, amounting to 122,263 sentences in English and comparable numbers in other languages. Most of the entities mentioned in the articles refer to prominent political figures from different European countries at the time of publication (2009–2013).

Sample Selection. Named entity detection was performed using the Flair model for English [544], which led to an initial pool of 30,303 sentences with at least one mention of a person. We combine entity linking with Blink [545] and coreference resolution with neuralcoref²⁷ to obtain reliable entity counts in articles. Entities mentioned only once in an article are not kept for annotation because they are not considered in focus. This filtering led to 19,223 candidate sentences. The alignment of sentences for the eight languages is inspired by lingtrain²⁸, and uses a similarity threshold. Automatic alignment was manually checked for three languages (EN, FR, RO), with a sample of 1,000 examples. It was correct in 98.1% of cases. Following sentence alignment, entity mentions across languages must also be aligned, and we used a rule-based approach for this task. Normalization form compatibility decomposition is first applied to examples in all languages. Then we computed a normalized Levenshtein distance between the English mention of the entity and the words from the target sentence. A similarity threshold of 80% between the English and the target mention in any of the other languages was used. To add flexibility to the matching process, we also considered nearly contiguous sequences as valid matches. We have checked this matching and it is correct in over 99% of cases on a subset of 1,000 mentions.

The sentiment classes are not evenly distributed in the news, and we followed an initial selection procedure inspired by the one introduced in [543]. It involves an undersampling of potentially neutral mentions as predicted by a simple binary classifier. This led to a pool of 11,000 examples which were selected and proposed to annotators.

Sample Annotation and Aggregation. Annotations were crowdsourced using a custom web interface. The annotation guide made the annotators aware of the complexity of the task and were asked to annotate from the author’s point of view. They were presented with examples of sentences that include intricate and/or implicit sentiment expressions, as well as irony. We used a Likert scale with five labels: negative, weakly negative, neutral, weakly positive, and positive. Annotators also had the possibility to label examples as unknown whenever they could not determine the label of an example. Annotations were provided by a total of 21 volunteers. They were recruited via a call for participation, which was circulated via group and personal e-mails. Participants provided explicit consent to use their annotations and demographic data at the beginning of the experiment. Samples were presented randomly in order to avoid any ordering effect, and users were free to stop at any point. Each sample was labeled by three annotators in order to allow annotation consolidation.

Following [543], we reduced the five initial labels to three classes (negative, neutral, and positive) by aggregating the two possible labels for the negative and positive sentiments. Finally, we kept only samples for which there was a unanimous voting or majority agreement with a third vote in a neighboring class. The inter-rater reliability, measured using Fleiss’ kappa [546], reaches $\mathcal{K}_F=0.58$ and $\mathcal{K}_F=0.67$, before and after consolidation, respectively.

6.2.3. Experimental Results

We run experiments with MAD-TSC in monolingual and multilingual settings, and we also use NewsMTSC for English experiments. The training/validation/test subsets are sampled randomly and include 3,810/300/1,000 labeled mentions, respectively. Results for NewsMTSC are reported with the official splits from [543]. We use the usual macro F1 ($F1_m$) on all classes as primary metric [543, 542, 540].

Experiments with Individual Languages. Results for the eight MAD-TSC languages are presented in Table 60. They are reported using SPC, a commonly used TSC method [547, 543, 548, 549]. The best $F1_m$ scores are obtained for English and French, and the lowest scores are reported for Dutch

²⁷<https://github.com/huggingface/neuralcoref>

²⁸<https://github.com/averkij/lingtrain-aligner>

Pretrain	EN	ES	DE	IT	FR	PT	NL	RO
<i>TG</i>	72.3	63.9	66.1	65.8	70.8	68.2	62.1	66.9
<i>ML</i>	67.8	67.2	64.8	65.1	67.2	66.2	66.4	68.5

Table 60. $F1_m$ results for the eight languages included in MAD-TSC. SPC is applied on top of models pretrained specifically for each target language (*TG*) or with a multilingual corpus (*ML*) using SPC.

Train	Test	ES	DE	IT	FR	PT	NL	RO
<i>EN</i>	<i>EN_{M2M}</i>	73.3	70.8	71.4	70.6	71.9	71.1	73.0
<i>EN</i>	<i>EN_{DL}</i>	73.9	73.2	72.5	73.5	73.1	72.1	73.8
<i>TG</i>	<i>TG</i>	63.9	66.1	65.8	70.8	68.2	62.1	66.9
<i>TG_{M2M}</i>	<i>TG</i>	64.7	65.0	66.0	70.6	66.9	64.2	65.7
<i>TG_{DL}</i>	<i>TG</i>	63.7	65.2	65.8	71.3	68.3	62.0	67.5

Table 61. $F1_m$ results for machine translation languages included in MAD-TSC, compared to the results obtained when without machine translation for English-only (72.3) and monolingual models (fourth row copied from Table 60). Notations: *EN* - English, *TG* - target language. The original train/test sets were used if no subscripts are present. *DL* (DeepL) and *M2M* [7] subscripts give the machine translation model used. All results are reported with language-specific pretrained models. TSC models are trained with SPC.

and Spanish. When using monolingual pretraining (*TG*), the difference between the best and worst scores is over 10 points. In contrast, the results obtained with multilingual pretraining (*ML*) are much more similar across all languages. Performance variability is explained by the quality of pretrained models, and in particular by the size of the datasets and that of the subsets relevant for politics. The results from Table 60 indicate that strong monolingual pretraining is preferable in TSC, but it can be successfully replaced by multilingual pretraining when the dataset for a particular language is insufficient.

Experiments with Machine Translation. Machine translation (MT) has strongly progressed in recent years, notably due to the introduction of neural architectures [550]. A successful deployment of MT for sentiment classification would greatly facilitate the task in the multilingual setting because it would reduce, or even remove, the need for specific annotations in each language. Building on previous works that apply MT to TSC [537, 551], we report results with English as the pivot language. Test and/or train subsets of the other languages are translated to English. The translation is performed with two methods: (1) M2M100 [7], a recent massively multilingual translation model, by using the largest available model (12B parameters); (2) the API of DeepL²⁹, a well-known commercial machine translation service.

The $F1_m$ scores obtained with different MT configurations are reported in Table 61. The results are very interesting, particularly when translating the test set to English with DeepL (row with *EN* train and *EN_{DL}* test). $F1_m$ scores are globally better than 72.3, the performance of SPC obtained with manual translations for English. The maximum gain is 1.6 points for Spanish, and Dutch is the only language for which DeepL translations are slightly worse (-0.2 points). $F1_m$ is also interesting with M2M100, albeit lower than that of *EN_{DL}*.

Results are also interesting when the English training set is translated toward the target languages using DeepL and M2M100 (rows with *TG_{DL}* / *TG_{M2M}* train and *TG* test). The associated $F1_m$ scores are on par with those obtained with the manual translations. However, the translation of training sets is less effective than that of test sets. This happens because TSC training is done in languages other than English and is based on weaker pretrained language models.

6.2.4. Conclusion

We introduce MAD-TSC, a dataset for multilingual target-dependent sentiment classification. The proposed dataset is aligned across languages and includes examples of geographically diversified entities.

²⁹<https://www.deepl.com/en/docs-api>



Examples are longer and more complex because sentiment is often expressed in an implicit way. Given its aligned character, MAD-TSC dataset enables a comparison of sentiment classification between languages. Performance varies significantly, and this variation is to a large extent explained by the quality of pretrained models available for each language.

Importantly, the MT experiments show that human translations can be replaced by automatic ones. The automatic translation of test sets from target languages to English is particularly interesting since it brings target-dependent sentiment classification in different languages to the same quality level as that of English. This allows TSC to be scaled for the languages included in this study without the need to develop language-specific training sets.

6.2.5. Relevance to AI4Media use cases and media industry applications

MAD-TSC enables a fine-grained analysis of sentiments expressed in political texts in eight European languages. Initial experiments showed that automatic translation toward a language having strong pretrained models followed by sentiment classification is preferable to direct classification in the original languages. This result is usable by media organizations to optimize their opinion mining pipelines. The dataset was used in a secondment between CEA and VRT, as part of UC2, to analyze the political positioning of Belgian news sources. This work is described in detail in deliverable D6.4.

6.2.6. Relevant Publications

- Evan Dufraisse, Adrian Popescu, Julien Tourille, Armelle Brun, Jerome Deshayes. "MAD-TSC: A Multilingual Aligned News Dataset for Target-dependent Sentiment Classification." Proc. of the 61st Annual Meeting of the Association for Computational Linguistics. 2023.

6.2.7. Relevant software/datasets/other outcomes

- The data and associated Pytorch code are available at https://github.com/EvanDufraisse/MAD_TSC

6.3. Same or Different? Diff-Vectors for Authorship Analysis

Contributing partners: CNR

6.3.1. Introduction

Automated *authorship analysis* is concerned with inferring characteristics such as the gender [552], the age group [553], or the native language [554] of the author, among others; these subtasks usually go under the name of *author profiling* [555]. Alternatively, authorship analysis may be concerned with inferring the *identity* of the author; tasks in which this is the goal are collectively referred to as *authorship identification* tasks, and include *authorship verification* (AV – the task of predicting whether a given author is or not the author of a given anonymous text [556]), *authorship attribution* (AA – the task of predicting who among a given set of candidates is the most likely author of a given anonymous text [557, 558]), and *same-author verification* (SAV – the task of predicting whether two given documents are by the same, possibly unknown, author or not [559]). Authorship analysis has several applications, e.g., in supporting the work of philologists who try to identify the authors of texts of literary or historical value [560, 561], or in aiding linguistic forensics experts in crime prevention or criminal investigation [562, 563].

All of these tasks are usually approached as *text classification* tasks, whereby a supervised machine learning algorithm, using a set of labeled documents, is used to train a classifier to perform the required prediction task. As in many supervised learning endeavors, each training example is usually represented





as a vector of features, where the value of a feature in a vector usually corresponds to the relative frequency with which a certain linguistic phenomenon (say, an exclamation mark, or a POS-gram) occurs within the document.

[559] describes an alternative method for generating vectorial representations of texts for authorship identification. Specifically, while in the standard representation methodology a vector represents a document, in this alternative method a vector represents an unordered *pair* of different documents. While in the standard methodology the value of a feature is (an increasing function of) the relative frequency of occurrence of a given linguistic phenomenon in the document, in this alternative method it is the absolute value of the *difference* between the relative frequencies (or increasing functions thereof) of this phenomenon in the two documents. Since these vectors represent differences, we call these representations *Diff-Vectors* (DVs). While in the standard methodology, the class label is the author of the document, in this DV-based methodology, the class label is one of the two classes Same or Different (standing for “same author” or “different authors”, respectively).

However, the goal of [559] was actually to propose a different method (the “impostors” method for SAV), and not to propose the DV-based methodology, which they dismiss as a “simplistic baseline method” [559, p. 179]. Since then, the use of DVs has never been studied systematically; to carry out such a systematic study is the goal of the present work, documented in [564].

The contributions of this work are thus as follows.

First, we study the consequences of the fact that, given n labeled documents, while the standard methodology gives rise to n training vectors, the DV-based methodology gives rise to $O(n^2)$ training vectors, which seems, at first sight, advantageous. Is this advantage for real? Does this quadratic number of training vectors pose computational problems? The present study answers this question.

Second, we carry out extensive experiments on a number of publicly available datasets (including one that we here make available for the first time) representative of different textual genres, lengths, and styles, with the goal of determining whether using DVs in place of “standard” vectors brings about higher accuracy in authorship identification tasks. In these experiments we tackle different authorship identification tasks, including SAV (for which DVs are naturally geared), AA, and AV; for these two latter tasks we propose two new methods, *Lazy AA* and *Stacked AA* (two AA methods that can also be used for AV) that solve AA by using a DV-based SAV classifier as a building block. Our experiments show that the DV-based representation is advantageous, since it brings about substantially increased effectiveness at the price of a tolerable increase in computational cost. The experiments also show that DVs bring about substantial improvements especially in low-resource authorship analysis tasks, i.e., in tasks characterised by small quantities of training data (which is the case in many real-life authorship analysis scenarios, such as those dealing with ancient texts). Like the standard representation, the DV-based representation is learner-independent, i.e., it can be used in connection with any (supervised or unsupervised) learning method.

Third, we carry out an extensive comparative analysis of the efficiency of the two methodologies, both by studying the computational complexity of authorship analysis tasks and by clocking actual experiments. This study confirms that, as expected, the DV-based methodology is computationally more expensive; however, as we argue in detail, the additional computational cost is tolerable, especially in the light of the fact that, in authorship analysis, practical application scenarios often involve *a single* document of uncertain paternity, which means that classification efficiency is not a primary concern.

6.3.2. Methodology

In “standard” authorship identification, each document x_i is represented via a labeled vector \mathbf{x}_i of features, where each feature usually represents a linguistic phenomenon that may occur (possibly several times) in a document of \mathcal{D} , the label $y_i \in \mathcal{A}$ represents the true author of x_i , and the value \mathbf{x}_i^k of the k -th feature in vector \mathbf{x}_i represents a non-decreasing function (e.g., tfidf) of the relative frequency of the linguistic phenomenon in x_i . For instance, if the k -th feature stands for character 3-gram “car”, then the value





of \mathbf{x}_i^k may be the number of occurrences of character 3-gram “car” in x_i divided by the number of all character 3-grams that x_i contains.

We here study an alternative type of vectorial representation for authorship identification tasks. Here, a labeled vector \mathbf{x}_{ij} represents an *unordered pair* (x_i, x_j) of documents in \mathcal{D} such that $i \neq j$, each feature represents a linguistic phenomenon that may occur (possibly several times) in a document of \mathcal{D} , the label $y_{ij} \in \mathcal{P} = \{\text{Same, Different}\}$ indicates whether the true authors of x_i and x_j are the same person or not, and the value \mathbf{x}_{ij}^k of the k -th feature in vector \mathbf{x}_{ij} represents the absolute difference between non-decreasing functions of the relative frequencies of the linguistic phenomenon in x_i and x_j . (In this section we provisionally assume this function to be the identity function $f(x) = x$, while in the sections to come this function will be some well-established feature weighting function.) Since the *difference* between relative frequencies is central to the definition of these vectors, we call them *Diff-Vectors* (DVs).

Any set of labeled documents $\mathcal{L} = \{(x_1, y_1), \dots, (x_n, y_n)\}$ can be represented either in the standard way or via DVs. One of the main differences between the two representations is that the “standard” representation gives rise to n labeled vectors, while the alternative representation gives rise to $n(n-1)/2$ labeled vectors. The other main difference is that a classifier using the “standard” representation attempts to predict, given an unlabeled document, its true author, while a classifier using the DV-based representation attempts to predict, given two unlabeled documents, whether the two documents are or not by the same author. *In other words, the standard representation is geared towards AV or AA, while the DV-based representation is geared towards SAV.* However, AV and AA can (as discussed below) be recast in terms of SAV, and vice-versa; as a result, we will consider the two representations as general-purpose alternatives, and we will study them as such.

6.3.2.1. Solving SAV, AA, and AV, by means of Diff-Vectors One difference between the standard representation, in which class labels represent authors, and the representation based on DVs, in which class labels are in $\{\text{Same, Different}\}$, is that the tasks that can be solved “directly” are AV and AA for the former, and SAV for the latter. That is, by using the standard representation, AV and AA can be solved directly by setting up a classifier that, for a given document, returns a class label in \mathcal{A} (for AA) or in $\{A^*, \bar{A}^*\}$ (for AV); SAV is instead to be solved as a derivative, “downstream” task, e.g., by first determining the true authors of documents x_i and x_j by means of two calls to an AA engine, and then checking whether the two returned class labels are the same or not.³⁰ On the contrary, when using the DV-based representation, SAV is solved directly; AV and AA are instead to be solved as derivative tasks, using SAV as the building block of any algorithm for solving them. In [564] we first formally define our method for performing SAV, and then devise two alternative solutions for solving both AV and AA that build on top of the former.

6.3.3. Experimental results

In [564], in order to test whether a representation based on DVs is advantageous with respect to a representation based on standard vectors, we compare these two different design choices in experiments that we run on four publicly available datasets (among which one that we here make available for the first time) and for all three authorship analysis tasks (AA, AV, SAV). The code to reproduce our experiments is available online at <https://github.com/AlexMoreo/diff-vectors>.

We run experiments on four datasets (IMDB62, PAN2011, Victorian, arXiv) consisting of textual documents annotated by author; our datasets are representative of different textual genres, lengths, and styles, are publicly available, and all consist of English texts. We use logistic regression (LR) as the learning method.

As for the choice of features, we stick to ones well-known and broadly adopted in the field of authorship analysis, i.e., features of a frequentistic nature that can be extracted automatically and that are believed to

³⁰This is possible only for closed-set SAV, though, since open-set SAV cannot be recast in terms of AA.





	mean	std	ttest
DV-Bin	.756	0.017	
DV-2xAA	.803	0.025	
STD-CosDist	.629	0.022	
STD-2xAA	.646	0.014	

Table 62. Intrinsic evaluation of DVs: results on closed-set SAV, using vanilla accuracy as the evaluation measure on dataset *arXiv*. **Boldface** indicates the best method. The first two methods are DV-based, while the last 2 methods are based on standard representations. Symbols * and ** denote the method (if any) whose score is not statistically significantly different from the best one at $\alpha=0.05$ (*) or at $\alpha=0.001$ (**) according to a paired sample, two-tailed t-test. No symbols * and ** appear in this particular table since all differences are statistically significant.

convey stylistic information; see for example [565, 566, 558] for an overview, and [567, 568] for a discussion of the most frequently used features in recent shared tasks focused on authorship analysis. These features are considered a standard in the authorship analysis field because they represent linguistic traits that are believed to remain more or less constant in an author’s production and, conversely, to vary in noticeable fashion across different authors [566, p. 241]; as such, they tend to be identifiers of the idiosyncratic style of an author. Note that other sets of features could have been equally plausible; however, this is not an important concern for our work, since it is completely agnostic with respect to the specific features that should be used.

We run two types of experiments:

- An “intrinsic” evaluation of DVs, which consists of SAV experiments, since SAV is the task that a classifier using DVs can solve directly. We perform experiments in both closed-set SAV and open-set SAV settings.
- An “extrinsic” evaluation of DVs, which consists of closed-set AA experiments. We do not run experiments for AV since each of our AA experiments is also a set of m AV experiments, and can be evaluated as such.

For reasons of brevity (a) we do not report AA and AV results, and only concentrate on SAV results; (b) we only report the results for one dataset only (the arXiv dataset).

Across all four datasets, results clearly indicate that the DV-based variants perform well; of the two methods that achieve SAV by running AA on both documents (i.e., the DV-2xAA and STD-2xAA methods), the DV-based method is always better or much better than the standard vector-based method, and the same happens of the two non-AA-based methods. The top-performing method is unquestionably DV-2xAA, which outperforms (often by a very large margin) all others, for all numbers m of authors and for all numbers q of training examples per author.

The entire set of experiments run for this work is described in detail in [564], to which we refer the interested reader.

6.3.4. Conclusion

DVs are naturally geared towards solving the “same-author verification” (SAV) task, i.e., the binary task of deciding whether two documents have been written by the Same (possibly unknown) author or by Different authors. However, we have shown that both (i) (closed-set) authorship attribution (the task of predicting who among a given set of candidates is the true author of a given text), and (ii) authorship verification (the task of predicting whether a given author is or not the author of a given text), can be recast in terms of SAV; we have presented two original algorithms (*Lazy AA* and *Stacked AA*) that do this for both AA and AV.

In order to compare DV-based authorship identification methods with their counterparts based on “standard” vectors, we have carried out experiments on four datasets of texts labeled by author (one of which we have created ourselves and we here make publicly available for the first time) and representative





of different textual genres, lengths, and styles, and on three authorship identification tasks (SAV, AA, AV). Our experiments have shown that DV-based methods are particularly suited to some authorship identification tasks and are not suited to others. For instance, the results indicate that neither standard methods nor DV-based methods clearly outperform each other on open-set SAV. Instead, DV-based methods vastly outperform the competition on three important tasks, i.e., (a) on closed-set SAV, (b) on closed-set AA, and (c) on AV. As we have argued, these benefits derive from the fact that, in many cases, DV-based methods may exploit more training data than methods based on standard vectors, and that DVs may make training more robust also when the above is not the case.

6.3.5. Relevance to AI4Media use cases and media industry applications

While the experiments we have carried out concern authorship analysis, which is not featured in WP8 use cases, the methods we have discussed are obviously applicable to other text classification tasks. In the near future, we plan to test them on the task of classification by topic (e.g., classifying news articles according to classes such as **Home News**, **Sports**, **Lifestyles**, etc.), so as to check whether the advantages that DV-based methods have shown in authorship analysis tasks can also be enjoyed in contexts in which the dimension according to which texts are classified is not authorship.

6.3.6. Relevant publications

Silvia Corbara, Alejandro Moreo, and Fabrizio Sebastiani. “Same or different? Diff-Vectors for authorship analysis”. *ACM Transactions on Knowledge Discovery from Data* 18(1): Article 12, 2023. Available at <https://zenodo.org/records/10019527>

6.3.7. Relevant software/datasets/other outcomes

- The code to reproduce our experiments is open-source and available online at <https://github.com/AlexMoreo/diff-vectors>





7. Computationally Demanding Learning

7.1. Overview

Current state-of-the-art AI applications and training methods in most domains require an amount of computational resources that is not effectively obtainable by a majority of practitioners or interested industry members. Some of the most common bottlenecks in these trainings include the need for copious amounts of data, often requiring several terabytes of free space; or large training times due to the constantly growing scale of the models, only alleviated by the use of very big quantities of GPUs. In addition, in image-based domains, like medical imaging, autonomous driving or media content managing, most current approaches downsample the images to sizes that produce undesirable information losses. Handling this latter limitation is of special importance for media outlets, as high ($\approx 4K$) resolution media is becoming the current standard. In Task 5.5, “Computationally Demanding Learning”, of AI4Media, we explore ways of efficiently handling the scaling of neural networks to larger larger image resolutions while simultaneously studying methods for efficient DNN training and math.

7.2. Orthogonal SVD Covariance Conditioning and Latent Disentanglement

Contributing partner: UNITN

7.2.1. Introduction and methodology

The Singular Value Decomposition (SVD) can factorize a matrix into orthogonal eigenbases and non-negative singular values, serving as an essential step for many matrix operations. Recently in computer vision and deep learning, many approaches integrated the SVD as a meta-layer in the neural networks to perform some differentiable spectral transformations, such as the matrix square root and inverse square root. The applications arise in a wide range of methods, including Global Covariance Pooling (GCP) [569, 570, 571], decorrelated Batch Normalization (BN) [572, 573, 574], Whitening and Coloring Transform (WCT) for universal style transfer [575, 576, 577], and Perspective-n-Point (PnP) problems [578, 579, 580].

For the input feature map \mathbf{X} passed to the SVD meta-layer, one often first computes the covariance of the feature as $\mathbf{X}\mathbf{X}^T$. This can ensure that the covariance matrix is both symmetric and positive semi-definite, which does not involve any negative eigenvalues and leads to the identical left and right eigenvector matrices. However, it is observed that inserting the SVD layer into deep models would typically make the covariance very ill-conditioned [570], resulting in deleterious consequences on the stability and optimization of the training process. For a given covariance \mathbf{A} , its conditioning is measured by the condition number:

$$\kappa(\mathbf{A}) = \sigma_{max}(\mathbf{A})\sigma_{min}^{-1}(\mathbf{A}) \quad (104)$$

where $\sigma(\cdot)$ denotes the eigenvalue of the matrix. Mathematically speaking, the condition number measures how sensitive the SVD is to the errors of the input. Matrices with low condition numbers are considered **well-conditioned**, while matrices with high condition numbers are said to be **ill-conditioned**. Specific to neural networks, the ill-conditioned covariance matrices are harmful to the training process in several aspects, which we will analyze in detail later.

This phenomenon was first observed in the GCP methods by [570], and we found that it generally extrapolates to other SVD-related tasks, such as decorrelated BN. Figure 56 depicts the covariance conditioning of these two tasks throughout the training. As can be seen, the integration of the SVD layer makes the generated covariance very ill-conditioned ($\approx 1e12$ for decorrelated BN and $\approx 1e16$ for GCP). By contrast, the conditioning of the approximate solver, *i.e.*, Newton-Schulz iteration (NS iteration) [581], is about $1e5$ for decorrelated BN and is around $1e15$ for GCP, while the standard BN only has a condition number of $1e3$.



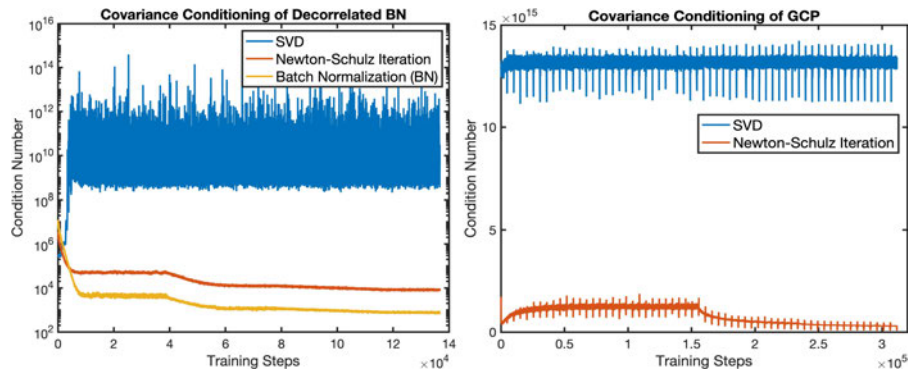


Figure 56. The covariance conditioning of the SVD meta-layer during the training process in the tasks of decorrelated BN (left) and GCP (Right). The decorrelated BN is based on ResNet-50 and CIFAR100, while ImageNet and ResNet-18 are used for the GCP.

Ill-conditioned covariance matrices can harm the training of the network in both the forward pass (FP) and the backward pass (BP). For the FP, mainly the SVD solver is influenced in terms of stability and accuracy. Since the ill-conditioned covariance has many trivially-small eigenvalues, it is difficult for an SVD solver to accurately estimate them and large round-off errors are likely to be triggered, which might hurt the network performances. Moreover, the very imbalanced eigenvalue distribution can easily make the SVD solver fail to converge and cause the training failure [582, 570]. For the BP, as pointed out in [583, 584, 572], the feature covariance is closely related to the Hessian matrix during the backpropagation. As the error curvature is given by the eigenvalues of the Hessian matrix [585], for the ill-conditioned Hessian, the Gradient Descent (GD) step would bounce back and forth in high curvature directions (large eigenvalues) and make slow progress in low curvature directions (small eigenvalues). As a consequence, the ill-conditioned covariance could cause slow convergence and oscillations in the optimization landscape. The generalization abilities of a deep model are thus harmed.

Due to the data-driven learning nature and the highly non-linear transform of deep neural networks, directly giving the analytical form of the covariance conditioning is intractable. Some simplifications have to be performed to ease the investigation. Since the covariance is generated and passed from the previous layer, the previous layer is likely to be the most relevant to the conditioning. Therefore, we naturally limit our focus to the Pre-SVD layer, *i.e.*, the layer before the SVD layer. To further simplify the analysis, we study the Pre-SVD layer in two consecutive training steps, which can be considered as a mimic of the whole training process. Throughout our research, we mainly investigate some meaningful manipulations on the weight, the gradient, and the learning rate of the Pre-SVD layer in two sequential training steps. *Under our Pre-SVD layer simplifications, one promising direction to improve the conditioning is enforcing orthogonality on the weights.* Orthogonal weights have the norm-preserving property, which could improve the conditioning of the feature matrix. This technique has been widely studied in the literature of stable training and Lipschitz networks [586, 587, 588]. We select some representative methods and validate their effectiveness in the task of decorrelated BN. Our experiment reveals that these orthogonal techniques can greatly improve the covariance conditioning, but could only bring marginal performance improvements and even slight degradation. *This indicates that when the representation power of weight is limited, the improved conditioning does not necessarily lead to better performance. Orthogonalizing only the weight is thus insufficient to improve the generalization.* Instead of seeking orthogonality constraints on the weights, we propose our Nearest Orthogonal Gradient (NOG) and Optimal Learning Rate (OLR). These two techniques explore the orthogonality possibilities about the learning rate and the gradient. More specifically, our NOG modifies the gradient of the Pre-SVD layer into its nearest-orthogonal form and keeps the GD direction unchanged. On the other hand, the proposed OLR dynamically changes the



learning rate of the Pre-SVD layer at each training step such that the updated weight is as close to an orthogonal matrix as possible. The experimental results demonstrate that the proposed two techniques not only significantly improve the covariance conditioning but also bring obvious improvements in the validation accuracy of both GCP and decorrelated BN. Moreover, when combined with the orthogonal weight treatments, the performance can have further improvements.

Besides the application on differentiable SVD, we propose that our orthogonality techniques can be also used for unsupervised latent disentanglement of Generative Adversarial Networks (GANs) [330]. Recent works [13, 12] revealed that the latent disentanglement of GANs is closely related to the gradient or weight of the first projector after the latent code. In particular, the eigenvectors of the gradient or weight can be viewed as closed-formed solutions of interpretable directions [12]. This raises the need for enforcing orthogonal constraints on the projector. *As shown in Figure 57, compared with non-orthogonal matrices, orthogonal matrices can lead to more disentangled representations and more precise attributes due to the property of equally-important eigenvectors.* Motivated by this observation, we propose to enforce our NOG and OLR as orthogonality constraints in generative models. Extensive experiments on various architectures and datasets demonstrate that our methods indeed improve the disentanglement ability of identifying semantic attributes and achieve state-of-the-art performance against other disentanglement approaches.

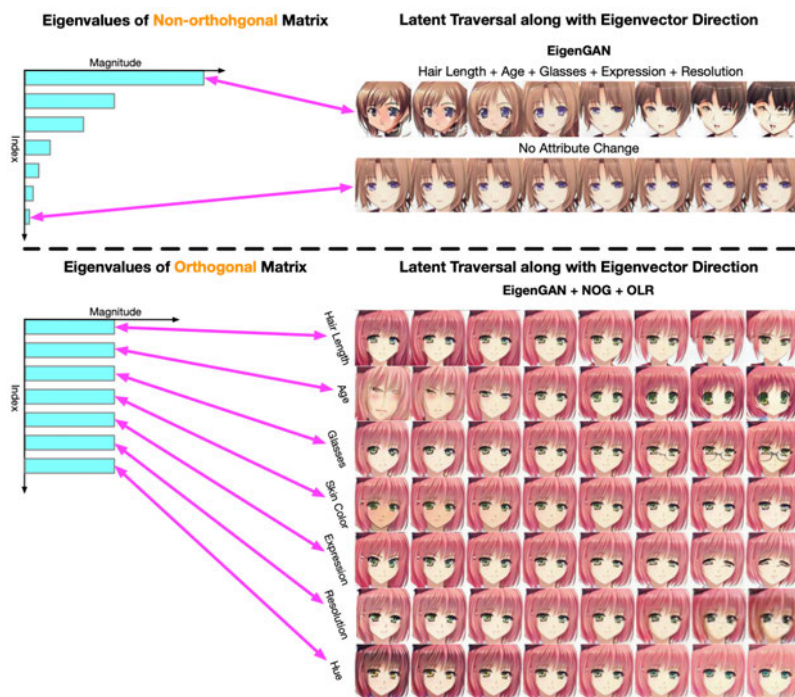


Figure 57. Illustration of the benefit of orthogonality in latent disentanglement. As revealed in [12, 13], the interpretable directions of latent codes are the eigenvectors of weight or gradient matrices. For non-orthogonal matrices, the principle eigenvector is of the most importance, which would make this direction correspond to many semantic attributes. The other eigenvectors might fail to capture any semantic information. By contrast, the eigenvectors of orthogonal matrices are equally important. The network with the orthogonal weight/gradient is likely to learn more disentangled representations.

This work is an extension of [589]. In [589], we proposed two orthogonality techniques and demonstrate that these methods can simultaneously improve the covariance conditioning and generalization abilities of the SVD meta-layer. The extension motivates and proposes that these techniques can be also applied in generative models for better latent disentanglement. This point is validated through extensive experiments on various generative architectures and datasets. Moreover, we also investigate



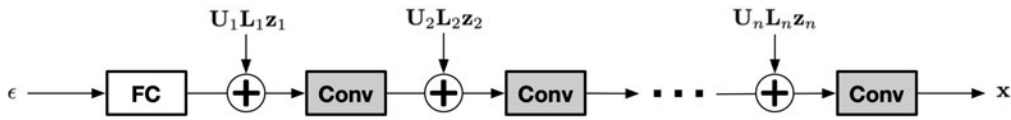


Figure 58. Overview of the EigenGAN architecture.



Figure 59. Latent traversal on AnimeFace [14]. The EigenGAN has entangled attributes in the identified interpretable directions, while our methods achieve better disentanglement and each direction corresponds to a unique attribute.

the probability of occurrence of our OLR throughout the training and show that the evaluation results agree well with our theoretical analysis.

7.2.2. Experiments on Latent Disentanglement

We validate the proposed approaches in two applications: GCP and decorrelated BN. These two tasks are very representative because they have different usages of the SVD meta-layer. The GCP uses the matrix square root, while the decorrelated BN applies the inverse square root. In addition, the models of decorrelated BN often insert the SVD meta-layer at the beginning of the network, whereas the GCP models integrate the layer before the FC layer. Due to space limitation, we refer the reader to the original publication. Instead, we present here the results on latent disentanglement.

7.2.2.1. Experimental Setup We show the evaluation of our methods on EigenGAN [590]. EigenGAN [590] is a particular GAN architecture dedicated to latent disentanglement. It progressively injects orthogonal subspaces into each layer of the generator, which can mine controllable semantic attributes in an unsupervised manner.

Datasets. For EigenGAN, we use AnimeFace [14] and FFHQ [591] datasets. AnimeFace [14] is comprised of 63,632 aligned anime faces with resolution varying from 90×90 to 120×120 . FFHQ [591] consists of 70,000 high-quality face images that have considerable variations in identifies and have good coverage in common accessories. We present results here on AnimeFace.

Metrics. We use Frechet Inception Distance (FID) [592] to quantitatively evaluate the quality of generate images. For the performance of latent disentanglement, we use Variational Predictability (VP) [593] as the quantitative metric. The VP metric adopts the few-shot learning setting to measure the generalization abilities of a simple neural network in classifying the discovered latent directions.

Baselines. For the EigenGAN model that already has inherent orthogonality constraints and good disentanglement abilities, we compare the ordinary EignGAN with the modified version augmented by our proposed orthogonal techniques (NOG and OLR).

7.2.2.2. EigenGAN Architecture and Modifications Figure 58 displays the overview of the EigenGAN. At each layer, the latent code z_i is multiplied with the orthogonal basis U_i and the diagonal importance matrix L_i to inject weighted orthogonal subspace for disentangled representation learning. The original EigenGAN [590] adopts the OL loss $\|U_i U_i^T - I\|_F$ to enforce *relaxed* orthogonality to each



Figure 60. Subtle semantic attributes mined by our method.

subspace U_i . Instead, we apply our NOG and OLR to achieve the weight and gradient orthogonality, respectively. Notice that when our NOG and OLR are applied, we do not use the OL loss of EigenGAN.

7.2.2.3. Qualitative Evaluation Results on EigenGAN Figure 59 compares the latent traversal results of the ordinary EigenGAN and our methods on AnimeFace. The interpretable direction of EigenGAN has many entangled attributes; the identity is poorly preserved during the latent traversal. By contrast, moving along with the discovered direction of our method would only introduce changes of a single semantic attribute. This demonstrates that our interpretable directions have more precisely-controlled semantics and our orthogonality techniques indeed help the model to learn more disentangled representations. Moreover, thanks to the power of orthogonality, our methods can mine many subtle and fine-grained attributes. Figure 60 displays such attributes that are precisely captured by our method but are not learned by EigenGAN. These attributes include very subtle local details of the image, such as facial blush, facial shadow, and mouth openness.

7.2.3. Conclusion

The main contributions of this work are as follows:

- We systematically study the problem of how to improve the covariance conditioning of the SVD meta-layer. We propose our Pre-SVD layer simplification to investigate this problem from the perspective of orthogonal constraints.
- We explore different techniques of orthogonal weights to improve the covariance conditioning. Our experiments reveal that these techniques could improve the conditioning but would harm the generalization abilities due to the limitation on the representation power of weight.
- We propose the nearest orthogonal gradient and optimal learning rate. The experiments on GCP and decorrelated BN demonstrate that these methods can attain better covariance conditioning and improved generalization. Their combinations with weight treatments can further boost the performance.



- We show that our proposed orthogonality approaches can be applied on the GANs projector for improved latent disentanglement ability of discovering precise semantic attributes, which opens the way for new applications of orthogonality techniques.

7.2.4. Relevant publications

- Y. Song, N. Sebe, and W. Wang, Orthogonal SVD Covariance Conditioning and Latent Disentanglement, IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(7): 8773-8786, July 2023. [594]
Zenodo record: <https://zenodo.org/record/8335410>

7.2.5. Relevant software/datasets/other outcomes

- The Pytorch implementation can be found in
<https://github.com/KingJamesSong/OrthoImproveCond>

7.2.6. Relevance to AI4Media use cases and media industry applications

The tools presented in this section are generic and can be applied to a large variety of applications. Evidence of the use of the tools in visual transformers and style transfer have been provided but their applicability is very large as SVD is used practically always when matrices are involved.

7.3. Masked Jigsaw Puzzle: A Versatile Position Embedding for Vision Transformers

Contributing partner: UNITN

7.3.1. Introduction

Transformers [595] demonstrated their overwhelming power on a broad range of language tasks (*e.g.*, text classification, machine translation, or question answering [595, 596]), and the vision community follows it closely and extends it for vision tasks, such as image classification [16, 15], object detection [597, 598], segmentation [599], and image generation [600, 601]. Most of the previous Vision Transformer(ViT)-based methods focus on designing different pre-training objectives [602, 603, 604] or variants of self-attention mechanisms [605, 606, 607]. By contrast, Position embeddings (PEs) receive less attention from the research community and have not been well studied yet. In fact, apart from the attention mechanism, how to embed the position information into the self-attention mechanism is also one indispensable research topic in Transformers. It has been demonstrated that without the PEs, the pure language Transformer encoders (*e.g.*, BERT [532] and RoBERTa [533]) may not well capture the meaning of positions [608]. As a consequence, the meaning of a sentence can not be well represented [609]. A similar phenomenon of PEs could also be observed in the computer vision community. Dosovitskiy *et al.* [16] reveals that removing PEs causes performance degradation. Moreover, Lu *et al.* [8] analyzed this issue from the perspective of user privacy and demonstrated that the PEs place the model at severe privacy risk since it leaks the clues of reconstructing sequential patches back to images. Hence, it is very interesting and necessary to understand how the PEs affect the accuracy, privacy, and consistency in computer vision tasks. Here the consistency means whether the predictions of the transformed/shuffled image are consistent with the ones of the original image.



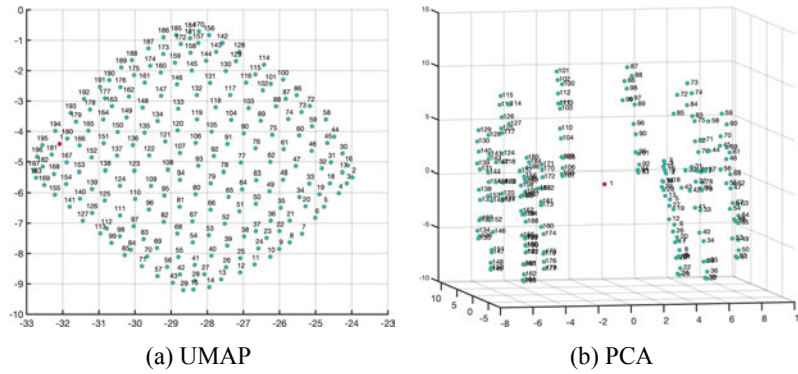


Figure 61. Low-dimensional projection of position embeddings from DeiT-S [15]. (a) The 2D UMAP projection, it shows that reverse diagonal indices have the same order as the input patch positions. (b) The 3D PCA projection, it also shows that the position information is well captured with PEs. Note that the embedding of index 1 (highlighted in red) corresponds to the [CLS] embedding that does not embed any positional information.

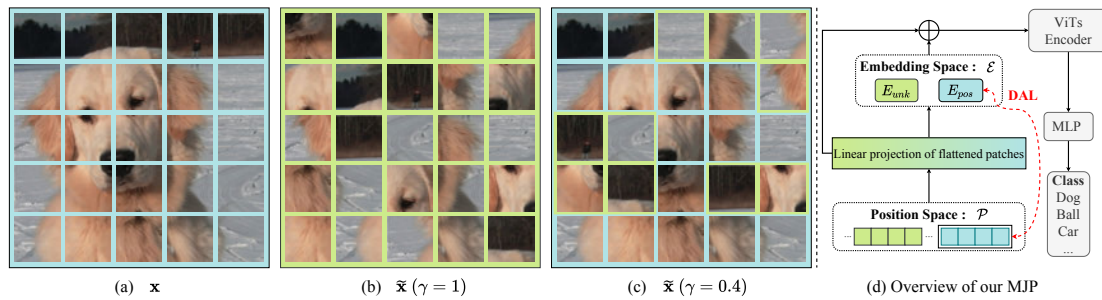


Figure 62. (a) The original input patches; (b) Totally random shuffled input patches; (c) Partially random shuffled input patches; (d) An overview of the proposed MJP. Note that we show the random shuffled patches and its corresponding unknown position embedding in green and the rest part in blue. DAL means the self-supervised dense absolute localization regression constraint.

7.3.2. Methodology

To study the aforementioned effects of PEs, the key is to figure out what explicitly PEs learn about positions from input patches. To answer this question, we project the high-dimensional PEs into the 2D and 3D spaces using Uniform Manifold Approximation & Projection (UMAP) [610] and PCA, respectively. Then for the first time, we visually demonstrate that the PEs can learn the 2D spatial relationship very well from the input image patches (the relation is visualized in Figure 61). We can see that the PEs are distributed in the same order as the input patch positions. Therefore, we can easily obtain the actual spatial position of the input patches by analyzing the PEs. Now it explains why PEs can bring the performance gain for ViTs [16]. This is because the spatial relation in ViTs works similar as the inherent intrinsic inductive bias in CNNs (*i.e.*, it models the local visual structure) [611]. However, these correctly learned spatial relations are unfortunately the exact key factor resulting in the privacy leakage [8].

Based on these observations, one straightforward idea to protect the user privacy is to provide ViTs with the randomly transformed (*i.e.*, shuffled) input data. The underlying intuition is that the original correct spatial relation within input patches will be violated via such a transformation. Therefore, we transform the previous visually recognizable input image patches \mathbf{x} shown in Figure 62(a) to its unrecognizable counterpart $\tilde{\mathbf{x}}$ depicted in Figure 62(b) during training. The experimental results show that such a strategy can effectively alleviate the privacy leakage problem. This is reasonable since the reconstruction of the original input data during the privacy attack is misled by the incorrect spatial relation.



However, the side-effect is that this leads to a severe accuracy drop.

Meanwhile, we noticed that such a naive transformation strategy actually boosts the **consistency** [612, 613, 614] albeit the accuracy drops. Note that here the consistency represents if the predictions of the original and transformed (*i.e.*, shuffled) images are consistent. Given the original input patches \mathbf{x} and its corresponding transformed (*i.e.*, shuffled) counterpart, we say that the predictions are consistent if $\operatorname{argmax}P(\mathcal{F}(\mathbf{x})) = \operatorname{argmax}P(\mathcal{F}(\tilde{\mathbf{x}}))$, where \mathcal{F} refers to the ViT models, and P denotes the predicted logits.

These observations hint that there might be a trade-off solution that makes ViTs take the best from both worlds (*i.e.*, both the accuracy and the consistency). Hence, we propose the Masked Jigsaw Puzzle (MJP) position embedding method. Specifically, there are four core procedures in the MJP: (1) We first utilize a block-wise masking method [615] to randomly select a partial of the input sequential patches; (2) Next, we apply jigsaw puzzle to the selected patches (*i.e.*, shuffle the orders); (3) After that, we use a shared *unknown* position embedding for the shuffled patches instead of using their original PEs; (4) To well maintain the position prior of the unshuffled patches, we introduce a *dense absolute localization* (DAL) regressor to strengthen their spatial relationship in a self-supervised manner. We simply demonstrate the idea of the first two procedures in Figure 62(c), and an overview of the proposed MJP method is available in Figure 62(d).

7.3.3. Experimental results

7.3.3.1. Privacy Preservation Experiments The fundamental principle of the gradient attack methods in federated learning is that each sample activates only a portion of content-related neurons in the deep neural networks, leading to one specific backward gradients for one related samples (*i.e.*, 1-to-1 mapping). Based on such an observation, we argue that feeding ViTs with input patches with permuted sequences may intuitively mislead the attack. This is because now both the original and the transformed inputs may be matched to the same backward gradients (*i.e.*, n -to-1 mapping).

To validate such an assumption, we utilize the public protocols³¹ to recover image with gradient updates in the privacy attack. In this privacy attack, we apply the Analytic Attack proposed in APRIL [8], which is designed for attacking the ViTs. We randomly sample 1K images from the validation set of ImageNet-1K (*i.e.*, one image per category). To evaluate the anti-attack performance of a model, we introduce image similarity metrics to account for pixel-wise mismatch, including Mean Square Error (MSE), Peak Signal-to-Noise Ratio (PSNR), cosine similarity in the Fourier space (FFT_{2D}), and Learned Perceptual Image Patch Similarity (LPIPS) [261]. Different from the evaluation in gradient attacks [8, 616, 617], we suppose a model is with better capacity of privacy preservation when the recovered images from its gradient updates are less similar to the ground truth images.

Given an image \mathbf{x} and its transformed (*i.e.*, patch shuffled) version $\tilde{\mathbf{x}}$, a ViT model \mathcal{M} , and automatic evaluation metrics ϕ , we conduct three different settings for fair comparisons: (a) $\phi(\nabla\mathcal{M}(\mathbf{x}), \mathbf{x})$, (b) $\phi(\nabla\mathcal{M}(\tilde{\mathbf{x}}), \tilde{\mathbf{x}})$, and (c) $\phi(\nabla\mathcal{M}(\tilde{\mathbf{x}}), \mathbf{x})$, where ∇ refers to recovering input image through gradient attacks. Table 63 shows the quantitative comparisons between our method and the original ViTs for batch gradient inversion on ImageNet-1K. APRIL [8] enables a viable, complete recovery of original images from the gradient updates of the original ViTs. However, it performs worse in recovering from “DeiT-S+MJP”, leading to best performances on all evaluation metrics and outperform others by a large margin.

More surprisingly, our proposed method makes APRIL yield unrecognizable images and fail in recovering the details in the original images (*i.e.*, noisy patches in the outputs), as shown in Figure 63. The left four columns in Figure 63 are tested on original images, where all PEs are standard and correspond to their patch embeddings. Meanwhile, the right four columns are tested with transformed ones, where the shuffled patches are with the shared unknown PEs. Both the visual and quantitative comparisons verify that our MJP alleviates the gradient leakage problem. We also notice that DeiT-S without using PEs is

³¹<https://github.com/JonasGeiping/breaching>





Table 63. Comparisons on gradient leakage by analytic attack [8] with ImageNet-1K validation set, where we test (1) ViT-S, DeiT-S and our model in the setting (a); (2) ViT-S, DeiT-S and our model in the setting (b) (i.e., MJP with $\gamma=0.27$); (3) ablation on without (w/o) using E_{unk} in setting (a); and (4) Our model in setting (c).

	Model	Set.	Acc. \uparrow	MSE \uparrow	FFT _{2D} \uparrow	PSNR \downarrow	SSIM \downarrow	LPIPS \uparrow
(1)	ViT-S [16]	a	78.1	.0278	.0039	19.27	.5203	.3623
	DeiT-S [15]		79.8	.0350	.0057	18.94	.5182	.3767
	DeiT-S (w/o PEs)		77.5	.0379	.0082	20.22	.5912	.2692
	DeiT-S+MJP		80.5	.1055	.0166	11.52	.4053	.6545
(2)	ViT-S [16]	b	18.7	.0327	.0016	18.44	.6065	.2836
	DeiT-S [15]		36.0	.0391	.0024	17.60	.5991	.3355
	DeiT-S (w/o PEs)		77.5	.0379	.0025	20.25	.6655	.2370
	DeiT-S+MJP		62.9	.1043	.0059	11.66	.4493	.6519
(3)	DeiT-S+MJP (w/o)	a	40.6	.1043	.0059	11.66	.4493	.6519
(4)	DeiT-S+MJP	c	62.9	.1706	.0338	8.07	.0875	.8945

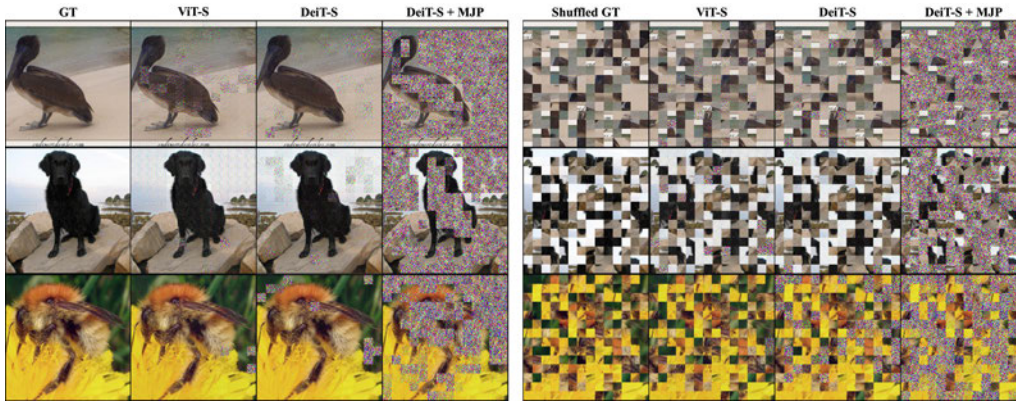


Figure 63. Visual comparisons on image recovery with gradient updates [8]. Our proposed DeiT-S+MJP model significantly outperforms the original ViT-S [16] and DeiT-S [15] models.

Table 64. Explained variance versus PCA projected dimensionality.

Projected Dimension	3	4	5	6	7
DeiT-S EV (%)	54.61	68.55	77.95	85.54	90.74
DeiT-S+MJP EV (%)	46.74	58.36	69.10	78.13	84.55

inclined to be at higher risk of privacy leakage (i.e., easier to be attacked by gradients). These promising results indicate that our MJP is a promising strategy to protect user privacy in federated learning.

7.3.3.2. PCA Projected Dimensionality. Table 64 presents the explained variance (EV) of our DeiT-S+MJP and DeiT-S versus different projection dimension. A low dimensionality can explain a large amount of information, which proves that the embedding matrix is sparse in nature. Moreover, to achieve the same explained variance ratio, our DeiT-S+MJP needs a large dimensionality than DeiT-S. This indicates that the positional embedding matrix of DeiT-S+MJP is less sparse but more informative.

7.3.4. Conclusion

The main contributions of this work are as follows:



- We demonstrate that although PEs can boost the accuracy, the consistency against image patch shuffling is harmed. Therefore, we argue that studying PEs is a valuable research topic for the community.
- We propose a simple yet efficient Masked Jigsaw Puzzle (MJP) position embedding method which is able to find a balance among accuracy, privacy, and consistency.
- Extensive experimental results show that MJP boosts the accuracy on regular large-scale datasets (*e.g.*, ImageNet-1K [341]) and the robustness largely on ImageNet-C [618], -A/O [619]. One additional bonus of MJP is that it can improve the privacy preservation ability under typical gradient attacks by a large margin.

7.3.5. Relevance to AI4media use cases and media industry applications

The presented approach for efficient training of visual transformers can be applied in all use cases where visual transformers could be applied. This can be the case of user stories in 3A3 (archive exploration), specifically 3A3-11 (Visual indexing and search), and 7A3 ((Re)organisation of visual content) by supporting the efficient training of image and video collections.

7.3.6. Relevant publications

- B. Ren, Y. Liu, Y. Song, W. Bi, R. Cucchiara, N. Sebe, and W. Wang, Masked Jigsaw Puzzle: A Versatile Position Embedding for Vision Transformers, CVPR 2023. [620]
Zenodo record: <https://zenodo.org/record/8337058>

7.3.7. Relevant software/datasets/other outcomes

- The Pytorch implementation can be found in <https://zenodo.org/record/8337058>

7.4. 4K Video Super-Resolution Detection

Contributing partners: BSC, RAI

7.4.1. Introduction

Digital content manipulation techniques, such as deepfakes, automatic colorization, or generative models, have garnered substantial attention in recent years. They have notably improved in quality and found numerous practical applications across diverse industries. Among these techniques is Super-Resolution (SR). Image enhancing applications are successfully applied in medical imaging [621] [622], security camera image footage [623] [624], remote sensing tasks [625] [626], gaming [627], and the entertainment industry [628] [629].

Modern SR models, particularly those focused on Video Super-Resolution (VSR), generally require high computational resources. This requirement is further amplified by the increasing tendency towards high-resolution 4K content. According to the Visual Networking Index by Cisco [630], it is estimated that, by 2023, two-thirds of the installed flat-panel TV sets will be UHD. As a result, many on-demand and streaming platforms have turned to SR techniques to upscale their content to 4K.

This tendency has led to an interest in the field of SR detection and also created a new set of challenges that threaten the authenticity of visual media, especially after the recent and socially impactful development of generative models. Digital forgeries, ranging from elementary manipulations like object cloning or removal to complex alterations involving deepfakes and SR pose substantial issues across different sectors, including digital forensics, cybersecurity, the legal system, media veracity, and privacy. Therefore, developing effective and reliable *forgery* detection mechanisms has become paramount.



Following this line of research, our contribution consists of two main components:

- We analyze the performance of diverse SR methods with objective and subjective metrics.
- We design, train, and evaluate a system that can accurately distinguish SR methods present within the training dataset.

7.4.2. Methodology

7.4.2.1. Data We have created two new datasets for the development of this work.

BVI-DVC-SR: Based on the existing BVI-DVC [631], published to train CNN-based video compression systems. It contains 200 original 4K videos from different sources. We extend the base database by upscaling the 200 1080p (downscaled from the original 4K counterparts) videos with different methods to create the BVI-DVC-SR dataset. The selected upscaling methods include one traditional technique and three DL-based video SR models: bicubic interpolation, Bilinear interpolation, Nearest-Neighbor Interpolation, BasicVSR [632], RealBasicVSR [633], RVRT [634], SwinIR-Classical [601], SwinIR-Real [601], and Real-ESRGAN [635]. This dataset is used to train the SR detection model.

BSC-4K: We present a dataset with paired video sequences at 1080p and 4K resolution recorded simultaneously. The dataset provides a valuable tool for analyzing the degradation nuances in the SR process by utilizing a unique camera setup to record the videos. It contains 33 4K and 33 1080p videos, cut to 64 frames each, recorded indoors and outdoors with a single DSLR camera. The dataset is used to evaluate the performance of current algorithms.

Lastly, we employ a set of videos recorded by RAI to perform a human quality assessment.

7.4.2.2. Performance Comparison We evaluate the effectiveness of various upscaling methods by measuring Full-Reference (FR) and No-Reference (NR) metrics across two separate datasets. The quantitative analysis across the BVI-DVC-SR and BSC-4K datasets reveals consistent trends among the evaluated methods (Table 65). RVRT and BasicVSR consistently exhibit high PSNR and SSIM values, indicating superior image quality and structural fidelity. RealBasicVSR stands out for its lower NIQE and BRISQUE scores, despite comparatively lower PSNR and SSIM. Bicubic interpolation shows moderate performance, maintaining good SSIM but lagging in PSNR and perceptual quality metrics.

In addition, we add the results from a subjective evaluation conducted by RAI (Table 66). Subjective evaluations were conducted at RAI’s laboratories following ITU BT500 recommendation with the collaboration of 10 experts. Surprisingly, Bicubic interpolation achieves the highest MOS for both urban and nature scenes, despite typically lower objective metrics. RealBasicVSR excels in urban scenes but underperforms in nature scenes. RVRT and SwinIR Real show consistently high performance across both scene types. BasicVSR, despite strong objective metrics in the previous table, receives lower subjective scores.

Dataset	Method	PSNR↑	SSIM↑	LPIPS↓	NIQE↓	BRISQUE↓
BVI-DVC-SR	RVRT	47.76	0.99	0.02	5.94	49.23
BVI-DVC-SR	BasicVSR	47.52	0.99	0.03	5.91	48.88
BVI-DVC-SR	Bicubic	47.24	0.99	0.04	6.68	54.62
BVI-DVC-SR	RealBasicVSR	30.49	0.89	0.33	4.14	25.00
BSC-4K	RVRT	33.14	0.96	0.07	5.76	45.77
BSC-4K	BasicVSR	33.46	0.96	0.06	5.93	46.43
BSC-4K	Bicubic	33.11	0.96	0.11	6.33	52.26
BSC-4K	RealBasicVSR	29.74	0.86	0.29	4.27	13.11

Table 65. Quantitative metrics for BVI-DVC-SR and BSC-4K datasets





SR Method	Urban MOS \uparrow	Nature MOS \uparrow
SwinIR Real	7.58	7.97
SwinIR Classical	5.1	6.57
BasicVSR	4.57	6.05
RealBasicVSR	8.53	5.95
RVRT	7.25	8.03
Bicubic	8.73	8.55

Table 66. Subjective Comparison with RAI’s dataset. Mean Opinion Score (MOS) is used, a subjective quality metric rated by human observers.

7.4.2.3. SR Detection We propose a network inspired by Lu et al.’s work [636] (BTURA). Our network’s architecture is based on the feature extractor, which processes the small patches in the training dataset. It consists of a ResNet-18 (pre-trained on ImageNet), where intermediate features are extracted from each block, grouped by a Global Average Pooling operation, and concatenated (Figure 64).

We incorporate two main new modules: First, the staircase structure, proposed in [637], attempts to fully utilize the visual information from low-level to high-level and learn the better feature representations for quality evaluation. The second module integrates a technique to combine local features from the patches and global features from the videos. We save the Discrete Cosine Transforms (DCT) features for each video in the dataset. Those features are concatenated with the local features from the feature extractor or staircase architecture.

We compare our results with three existing detection models, SRDM [638], TSARA [639], and [640] (Table 67).

Model	DCT	TSARA	SRDM-Patches	SUDDS (ours)
Original	0	0.72	0.88	0.94
SwinIR-Real	0.4	0	0.2	0.9
SwinIR-Classical	1	0.3	0.65	1
Real-ESRGAN	0	0	0.68	0.94
Nearest Neighbor.	1	0.05	0.4	0.9
BasicVSR	1	0.44	0.4	0.9*
Real-BasicVSR	0.06	0	0.59	1*
RVRT	1	0.36	0.42	0.8*
Bicubic	1	1	0.85	1*

Table 67. Accuracy Metrics for all studied SR and detection methods. * denotes SR methods that are in the training set

7.4.3. Conclusion

The subjective analysis (66) reveals a preference for Bicubic interpolation, especially above 720p, contrasting with previous Full-Reference metric assessments. Selected SR methods struggled with high-frequency artifacts and temporal consistency at higher resolutions, varying performance by video texture and domain. Evaluators generally preferred "Real" SR methods for video upscaling. Additionally, we present a super-resolution detection model for upscaling detection and recognition that outperforms existing methods, enhancing our understanding and evaluation of SR techniques. These findings highlight the importance of human perception in SR evaluation and offer new tools for analyzing upscaled content.



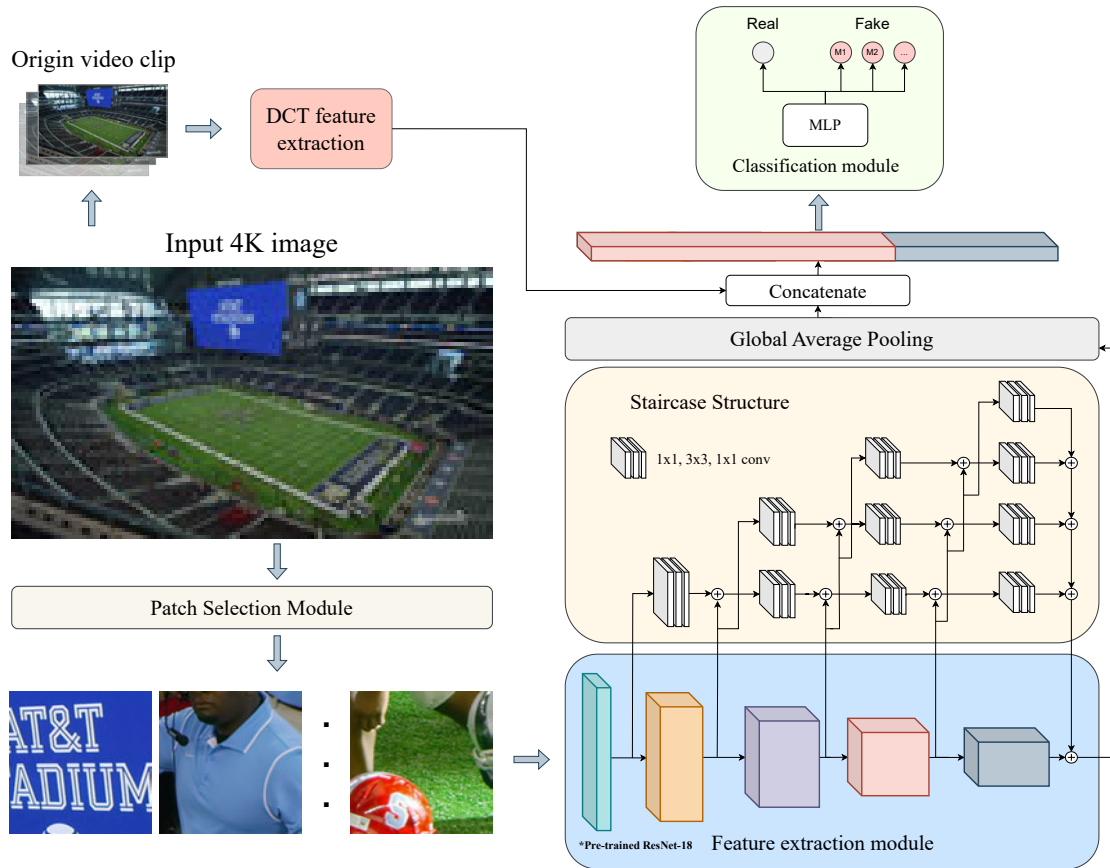


Figure 64. Proposed architecture of SR detection module for upscaling detection and recognition.

7.4.4. Relevance to AI4media use cases and media industry applications

This work leverages deep learning techniques to accurately detect upscaled 4K videos, ensuring that only genuine high-resolution content is delivered to audiences. By implementing this technology, media companies can maintain high-quality standards, enhance viewer trust, and protect their brand integrity. Furthermore, it allows for more efficient content management and quality control processes, ultimately contributing to a more reliable and satisfying viewing experience.

7.4.5. Relevant software/datasets/other outcomes

The detection model and the BVI-DVC-SR datasets are available at <https://github.com/Cuena/4k-vs-r-detection>



8. Music Annotation and Audio Provenance Analysis

8.1. Overview

AI-enabled music analysis is a topic of high industrial relevance that requires special attention. **Task 5.6 (T5.6)** “Music Annotation and Audio Provenance Analysis” of AI4Media dealt with automated music annotation and music similarity analysis, as well as with audio partial matching/reuse detection and audio phylogeny analysis, mainly using novel DNN-based methods. Music similarity analysis refers to the task of quantifying similarity between different music tracks and is particularly significant for the music replacement problem, i.e., when we search for a song as similar as possible to the query track. On the other hand, automated music annotation refers to methods that permit automatic production/extraction of annotation metadata for music tracks (e.g., for training DNNs in a supervised manner). Audio phylogeny implies the automatic detection of processing history relationships between audio items, while partial audio matching involves the detection and temporal localization of arbitrary partial matches between different audio items.

8.2. How reliable are posterior class probabilities in automatic music classification?

Contributing partners: FhG-IDMT

8.2.1. Introduction

Music genre and instrument classification are key tasks within Music Information Retrieval (MIR), both challenged by the ambiguity of categories and the similarities within them. Genre classification aims to label songs with a style (e.g., Rock, Pop, Jazz), while instrument classification identifies specific instruments in recordings. Both tasks suffer from overconfident predictions when using deep learning-based classifiers, which are the current standard [641, 642, 643, 644, 645]. This overconfidence complicates output interpretation and can reduce classification effectiveness [646, 647, 648]. Thus, establishing realistic confidence values is crucial for both genre and instrument classification tasks.

8.2.2. Methodology

The uncertainty of the classification decision can be quantified using a confidence measure which is a score that accompanies the decision and signifies its trustworthiness. A higher confidence corresponds to a more reliable decision.

The importance of confidence arises when it is necessary **(i)** to compare or merge classification decisions from different classifiers, **(ii)** to implement a reject option based on the confidence, or **(iii)** to interpret classification outcomes.

In this work, we define confidence as a value ranging from 0 to 1 that is associated with a classification decision and meets the criteria set forth by Duin and Tax [649]:

1. On average, a proportion c of all objects with a confidence of c should be classified accurately.
2. Objects that are classified reliably should possess higher confidences than objects near the decision boundary.

Confidences of this nature are simple to understand. For example, if we obtain 100 decisions with confidences around 0.7, we can anticipate approximately 70 of them to be accurate.





8.2.2.1. Deterministic Overconfidence For multi-class single-label tasks, the softmax activation function's output in the last layer is often misinterpreted as model confidence in class decisions, leading to *deterministic overconfidence* [646]. This overconfidence results from using point estimates rather than distributions, often causing inflated probabilities for both correct and incorrect classes. This phenomenon is exacerbated when data is far from the decision boundary or when ReLU activations are employed [648].

To mitigate deterministic overconfidence, *Temperature scaling* adjusts softmax outputs post-hoc by dividing the neural network logits by a temperature value T before the softmax function, effectively softening output probabilities for in-distribution data [647]. Another method, Monte Carlo (MC)-Dropout, introduces dropout during inference to model uncertainty and approximate Bayesian inference, varying outputs with each pass of the same input [646]. Additionally, deep ensembles, which utilize multiple independently trained networks, have proven effective in reducing overconfidence by leveraging diverse collective knowledge, particularly excelling in out-of-distribution scenarios [650, 651]. This approach has been shown to outperform MC-Dropout in uncertainty quantification and generalization across various tasks and datasets.

8.2.2.2. Datasets This work focuses on music genre classification using the FMA dataset [652] and instrument family classification with the NSynth dataset [653]. The FMA small dataset, a subset of the larger FMA, includes 8,000 tracks across eight genres, each 30 seconds long, with balanced training, validation, and evaluation splits. Zhao et al. [645] achieved a 56.4% accuracy using a Swin Transformer and self-supervised pre-training. Kostrzewa et al. [643] explored various architectures, achieving up to 56.39% accuracy with CNN ensembles.

The NSynth dataset comprises 300k musical notes from over 1k instruments, categorized into 10 families, recorded at 16 kHz over four seconds [654]. Advanced methods achieved up to 77.1% accuracy using a ResNet-based CNN with random image augmentations of log mel spectrograms [644].

8.2.3. Experimental Results

In this study, we focus on evaluating posterior class probabilities in automatic music genre and instrument family classification, utilizing temperature scaling and deep ensembles to achieve realistic confidence outputs. We employ two network architectures: a ResNet with 420k parameters [644] and a shallow Multi-Layer Perceptron (MLP) using OpenL3 embeddings [655], referred to as ResNet and OpenL3, respectively.

Both models are trained using the Adam optimizer at a learning rate of 10^{-3} for 100 epochs, with the ResNet applying random image augmentations to the mel spectrogram. For FMA, ResNet processes 3-second log mel spectrogram patches, while for NSynth, 4-second patches are used. OpenL3 utilizes audio embeddings trained with music data.

The models are trained and tested on subsets of the FMA small and NSynth datasets. During inference, softmax outputs are averaged over all patches to estimate class probabilities, assuming uniformity within each recording. Additionally, we implement deep ensembles by training the networks five times with random initialization, and calculate ensemble probabilities as the mean output across the models. Temperature scaling adjusts the logits before softmax activation to refine the class probabilities [650].

We investigated the calibration of single models and ensembles in music genre and instrument family classification, using the FMA and NSynth datasets. Table 68 summarizes the classification accuracies. Ensembles consistently improved accuracy across both datasets and architectures. For instance, ensemble accuracy for FMA using ResNet increased from 47.22% to 50.74%.

We also explored the discrepancy between softmax outputs (commonly interpreted as confidence) and estimated class probabilities. By analyzing reliability across confidence intervals, from high to low, we aimed to validate whether higher softmax outputs correlate with higher actual accuracies.

In Figure 65, reliability diagrams for each dataset and model combination display actual accuracies against expected accuracies for defined confidence intervals. The discrepancies observed prompt further investigation into refining model confidence assessments to align more closely with actual outcomes.



	Dataset	Architecture	Accuracies in %
Single models	FMA	ResNet	47.22 (0.78)
Ensemble	FMA	ResNet	50.74
Single models	FMA	OpenL3	45.57 (0.18)
Ensemble	FMA	OpenL3	46.70
Single models	NSynth	ResNet	79.96 (0.61)
Ensemble	NSynth	ResNet	81.49
Single models	NSynth	OpenL3	63.99 (0.17)
Ensemble	NSynth	OpenL3	65.32

Table 68. Accuracy values for both datasets and network architectures in %. The accuracy values for single models are provided as mean over all single models with the standard deviation in parentheses.

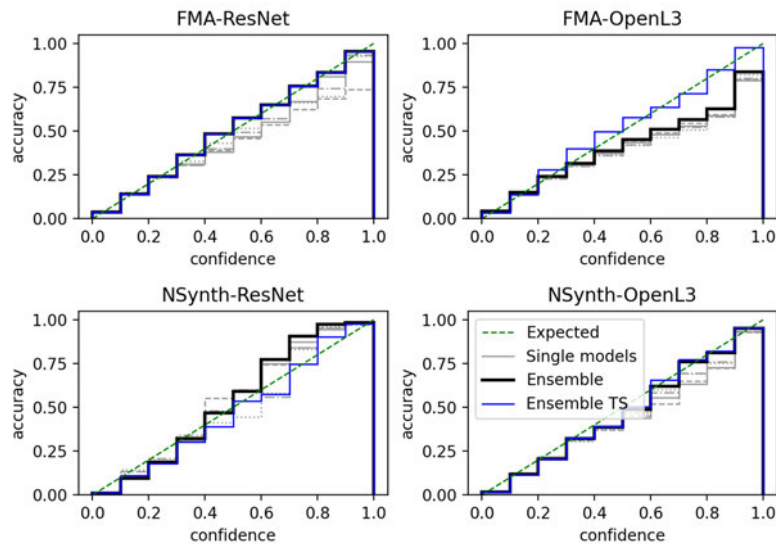


Figure 65. Reliability diagrams for all datasets and models

Figure 65 shows that single models, represented by grey lines, often display deterministic overconfidence. In contrast, ensembles (black lines) generally align closer to the ideal calibration (green dashed lines), especially noticeable in the “FMA–ResNet” and “NSynth–OpenL3” configurations.

Despite advancements, ensembles do not entirely resolve the issue of unreliable confidence outputs, prompting the exploration of temperature scaling as an additional calibration method. To enhance confidence calibration, we applied temperature scaling, which adjusts the logits before softmax activation. Reliability diagrams with optimal temperature settings for ensembles are depicted as blue lines in Figure 65, demonstrating improved alignment with expected accuracy. Please, refer to our publication [656] for detailed results.

8.2.4. Conclusion

This study examines the reliability of confidence values in automatic music classification tasks: music genre and instrument family classification, using a ResNet and a model with OpenL3 embeddings. We found that even advanced deep learning methods struggle with estimating realistic posterior class





probabilities. To address this, we implemented deep ensembles and temperature scaling, which improved reliability but required careful tuning specific to each dataset and model.

Our findings emphasize the importance of reliable classifier outputs in enhancing the accuracy and utility of music classification systems, guiding future advancements in the field.

8.2.5. Relevance to AI4media use cases and media industry applications

The approach is related to use case 5 (AI for Games), aiming at helping game audio designers to choose suitable music tracks for games.

8.2.6. Relevant Publications

- Hanna Lukashevich, Sascha Grollmisch, Jakob Abeßer, Sebastian Stober, and Joachim Bös. How reliable are posterior class probabilities in automatic music classification? In Proceedings of the Audio Mostly Conference, 2023 [656]

8.2.7. Relevant software/datasets/other outcomes

None

8.3. Free-form Text to Music Search Retrieval and Music Tagging

Contributing partner: FhG-IDMT

8.3.1. Introduction

Text-to-music retrieval involves retrieving music files in large repertoires that are most similar to the natural language query. This task is multi-modal, as it involves learning representations of the two modalities text and audio jointly in a common embedding space. This learned representation is then used to encode the text query and music recordings. Cosine similarity is used to identify the most similar music files related to a text query.

Audio classification and retrieval tasks typically require large data sets, whose annotation is labor intensive. Implementing training approaches with less supervision, such as based on self-supervised or unsupervised learning, remains a challenge. Training of such a multi-modal model is different from the state-of-the-art classification models that are typically trained with fixed categories and limited generality to new audio concepts. Contrastive learning offers a useful paradigm for training a model on large-scale noisy data collected from the Internet. This paradigm involves learning a low-dimension embedding representation of a particular entity (be it text or audio) by contrasting between similar and dissimilar pairs of entities such that similar pairs have a low distance and dissimilar pairs have a high distance in the embedding space.

LIAON-CLAP [657] is a pre-trained model that uses contrastive learning on a large collection of audio-text pairs of environmental, speech, and music data with a total number of 633,526 audio-text pairs. In our research, we fine-tuned this model with specific data for the music captioning task, which we obtained from two different sources.

Music tagging is the task of assigning a set of text tags to music clips. The most common tag categories are genres, instruments, and moods. We use the same fine-tuned models based on LIAON-CLAP for the music tagging task. Here, a prediction score is computed as the cosine similarity between the encoded representation of a music clip and the text embedding of the corresponding tags. Finally, tags with a high prediction score are assigned to music clips.





8.3.2. Methodology

We fine-tuned the LIAON-CLAP model on multiple datasets such as the MusiCaps dataset [291] and the LP-MusiCaps dataset [658], which are both open source. The MusicCaps dataset contains 5,521 music examples, each of which is labeled with an English aspect list and a free text caption written by musicians. An aspect list is a free text list of musical tags such as “pop, tinny wide hi-hats, mellow piano melody, high pitched female vocal melody, sustained pulsating synth lead”. The LP-MusiCaps dataset is a modified version of the Magnatagatune dataset [659] where multi-label tags related to each audio clip were converted to text captions using the GPT-3.5 Turbo Large Language Model. The Magnatagatune dataset consists of 26k music clips from 5,223 unique songs including genre, instrument, vocal, mood, perceptual tempo, origin, and sonority features. The LP-MusiCaps dataset utilizes 188 unique original tags from the Magnatagatune dataset to perform tag-to-caption generation.

To fine-tune the LIAON-CLAP model, we use the huggingface transformers [660] library. Transformers provides APIs and tools to easily download and train state-of-the-art pre-trained models. These models support common tasks from different modalities such as Natural Language Processing, Computer Vision, and Audio and Multi-modal Analysis. The tasks include optical character recognition, information extraction from scanned documents, video classification, and visual question answering.

Fine-tuned models were evaluated for both text-to-music retrieval and music tagging tasks. In order to evaluate text-to-music retrieval, we use the Song Describer dataset [661], which is another manually annotated dataset consisting of 706 music clips and their corresponding captions. This dataset is a crowd-sourced corpus of high-quality audio-caption pairs, which is designed for evaluation of music-language models. As evaluation metrics, we use the recall score and the retrieval rank, i. e., the median value of the rank for the retrieved results shown in the section 8.3.3.

We employ the model that performed best in the text-to-music retrieval experiment, to the music tagging task, which it was not originally trained for, effectively making this a zero-shot problem. This approach is similar to the method described in [662]. We used all available tags from the ground truth labels of the MagnaTagaTune dataset [659], which cover three broad categories genre, mood, and instrumentation. In order to streamline the tagging task, we manually divide the task into three separate tagging operations for each of the categories. The audio clips in the MagnaTagaTune dataset are 29 seconds long. We passed the entire audio clip through the audio encoder, and each category tag through the text encoder part of the model. For each audio clip, the cosine similarity score is computed between the audio clip and the category tags, and the tags with the higher similarity score are taken as a prediction. The number of tags to predict is arbitrarily chosen based on a threshold value of the maximum similarity score, and evaluation metrics like precision and recall are computed to assess the performance and are currently part of the ongoing work.

8.3.3. Experimental Results

Table 69 shows the results of the model evaluation metrics in the Song Describer dataset after fine-tuning. The first result is the huggingface baseline model that was trained on music and speech data, for which the recall@10 score is 0.247. After model fine-tuning, we observe an improved Recall@10 score of 0.362, which increased by 10 %, although the MusiCaps dataset is relatively small with only 5k+ samples. On the other hand, the LP-MusiCaps dataset with pseudo-labels generated by the GPT3.5 Turbo model did not improve the metrics as much even with a significantly larger dataset with 20k+ audio-text pairs.

8.3.4. Conclusion

This task overall encompasses using audio-language pre-trained models to perform two downstream tasks that are relevant in the music information retrieval domain. The first is the text-to-music retrieval task and the second is the music tagging task. Both tasks utilize the pre-trained joint embeddings of





	Recall@10	MRR (Median Retrieval Rank)
LAION-CLAP-music-and-speech (baseline)	0.247	34.999
LAION-CLAP-music-and-speech (fine-tuned on MusicCaps)	0.362	23.999
LAION-CLAP-music-and-speech (fine-tuned on -LPMusicCaps)	0.281	33.999

Table 69. Performance Metrics for Different Models

audio and text in order to achieve a task the model was not originally trained on. Our results show that further fine-tuning on relevant datasets can improve the results in downstream tasks.

8.3.5. Relevance to AI4media use cases and media industry applications

The approach is related to use case 5 (AI for Games), aiming at helping game audio designers to choose suitable music tracks for games.

8.4. Audio Provenance Analysis in Heterogeneous Media Content Sets

Contributing partners: FhG-IDMT

8.4.1. Introduction and methodology

Verifying the reliability and origin, or provenance, of audio files is crucial in combating disinformation and ensuring the integrity of media content. This is especially important in fields such as journalism and law enforcement, where validating audio material can be pivotal for investigations or fact-checking efforts.

Journalists and law enforcement agencies often need to examine media files to trace their distribution and identify the earliest or least altered versions. This process is essential for verifying content authenticity, identifying information sources, and unraveling distribution patterns. The complexity increases with extensive sets of audio files, where manipulated or decontextualized materials may incorporate segments from genuine sources, and identical content may spread across multiple platforms. Thus, distinguishing between derived and original or first-published versions and detecting the transformations applied is essential.

Detecting content similarity and transformations remains a challenge in the current SOTA, especially within heterogeneous sets of media content from various internet sources or devices, and lacking detailed content information. In our latest work [17], we introduced *Audio Provenance Analysis* to address these challenges by mapping the directed relationships among media files focusing on reused audio segments. The goal is to identify near-duplicate audio sets, reconstruct their lineage in directed acyclic graphs, and highlight partial content reuses contributing to new compositions (see Figure 66).

Our analysis of this complex problem led us to identify two critical tasks essential for an effective audio provenance framework: Provenance Clustering and Provenance Graph Building.

The goal of the **Provenance Clustering** task, illustrated in Figure 66, involves initially applying Partial Audio Matching to determine which audio items are related, followed by a Near Duplicate Clustering process. This process aims to group near-duplicate items in clusters and identify connections between non-near-duplicates.

The Partial Audio Matching, illustrated in Figure 67, utilizes an audio matching approach we proposed in [663, 664]. This method introduces an advanced retrieval algorithm tailored to detect and localize reused audio segments as short as 3 seconds, meeting our requirements for precision and reliability in segment localization, making it ideal for provenance clustering.



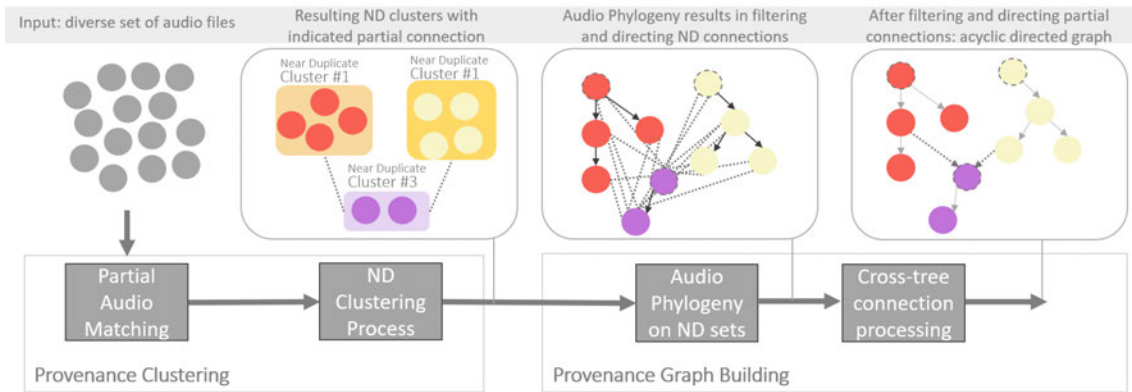


Figure 66. Audio Provenance Analysis workflow proposed in [17]

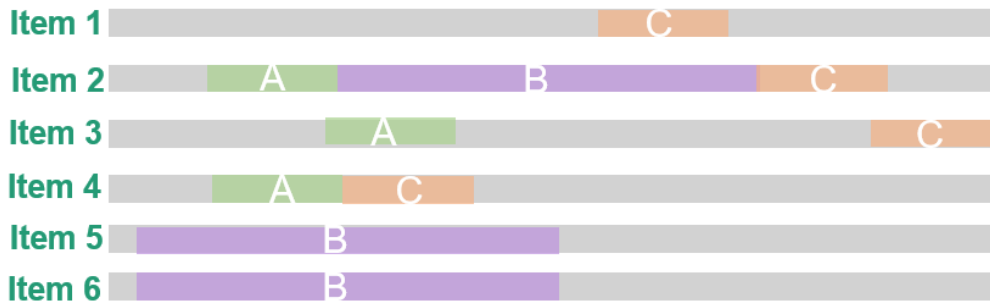


Figure 67. Partial audio matching focuses on identifying reused or recurring segments, sometimes just a few seconds long, within datasets or streams without any prior knowledge of the segments' existence, duration, or frequency of reuse. The image illustrates a dataset containing six audio items, where partial matching successfully detected three different recurring segments, despite having no prior knowledge of the quantity or length of the recurring content.

As part of Task 5.6 related to Provenance Clustering, we have improved the existing tool for partial audio matching and, together with our AI4Media partners, implemented it directly in applications for asset management in the media research domain.

The goal of the **Provenance Graph Building** task is depicted in Figure 66. First, we transform the clusters of near-duplicates into sets of phylogeny trees, i.e., directed graphs indicating provenance. Next, we process the cross-tree partial connections, i.e., the partial matching between disjoint phylogeny trees, to pinpoint specific files acting as donors in creating derived content. The key component for successful provenance graph building is the Audio Phylogeny Analysis.

Audio Phylogeny Analysis aims to detect relationships and transformations within a set of near-duplicate audio items. This involves computing a dissimilarity matrix between each pair of near-duplicates, which is then transformed into a directed phylogeny tree using the Oriented Kruskal algorithm [665]. While several methods for audio phylogeny exist, they detect only a limited set of transformations [19, 18, 666]. Extending this set significantly increases complexity. Hence, our main focus within Task 5.6 was developing a method for Audio Phylogeny analysis as a fundamental part of the Audio Provenance Analysis Framework. This novel method uses a neural network to detect the most probable transformation between each input pair of near-duplicates, presented in [667]. This approach offers high computational efficiency, enabling detection of specific transformations between pairs of files and allowing for the expansion of the set of potentially detected transformations with relative ease.



Efficient phylogeny analysis involves determining the most likely transformation τ_b between each pair of files (a,b) in the analysis set \mathcal{A} . The seminal work by Nucci et al. [19] realized this step through an exhaustive search, which is highly demanding and nearly unfeasible for large datasets. This issue was partially addressed by Maksimovic et al. [18], who proposed a two-step procedure based on a first coarse search followed by refinement to reduce the required computation.

In our latest work [667], we address transformation estimation in a single step. Given a pair of input audio files (a,b) , we interpret transformation estimation as a closed-set classification problem, where each class represents one possible transformation τ_b . The probability of each transformation is computed by reading the b -th output of a neural network $DNN(\cdot)$ trained ad-hoc.

More in detail, Figure 68 shows the phylogeny analysis process for one pair of audio files: potential parent a and potential child audio file b . Mel-spectrograms of these two audio files are given as input to the network, which extracts ResNet50 embeddings that are then fed to a feed-forward classification network to compute class probabilities. The output layer is interpreted as class probabilities, using one-hot encoding for each transformation in the set and training the network with Binary Cross Entropy (BCE) loss. This network outputs the best suitable transformation that should be applied to potential parent a to get the closest version possible to b .

After applying the top two transformations selected by the network on audio file a , three dissimilarity values are calculated between the original and transformed versions of a against the potential child b . The lowest of these three values is saved in the dissimilarity matrix, which holds dissimilarity values between every pair of files in the analyzed set. The Oriented Kruskal algorithm is then used to reconstruct a phylogeny tree from the given dissimilarity matrix.

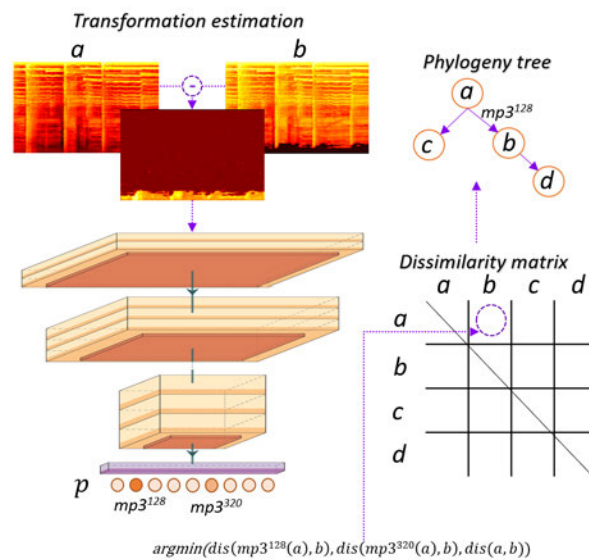


Figure 68. Complete audio phylogeny analysis system with transformation prediction via DNN classifier, dissimilarity calculation, and tree reconstruction

8.4.2. Experimental Results

Here we detail the experiment with a focus on our audio phylogeny approach from [667] as the most prominent part of our audio provenance analysis framework developed within Task 5.6. These experiments were done in two phases: In the first phase, we compared the performance and scalability of the proposed approach against state-of-the-art methods, using a base set of transformations the pre-existing algorithms





have been designed for. In the second phase, we tested the adaptability of the proposed approach to new demands by extending the set of considered transformations and evaluating the resulting performance.

To evaluate our method against the existing SOTA, we considered the following base set of transformations:

$$\mathcal{T}_b = \{\text{none}, \text{mp3}_{320}, \text{mp3}_{192}, \text{mp3}_{128}, \text{aac}_{320}, \text{aac}_{192}, \text{aac}_{128}, \text{fade}, \text{trim}\}, \quad (105)$$

Using this set of transformations, we created an evaluation dataset containing 60 audio phylogeny trees with 20 nodes each which has been made available in [668].

The evaluation metrics used are the ones originally proposed in [665] and then adopted as standard for evaluating a reconstruction of phylogeny trees. The amount of correctly reconstructed roots R , edges E (parent-child links), leaves L (nodes with no children), and ancestry A (lists of all children derived from every node) have been compared between ground truth Audio Phylogeny Tree APT_{gr} and reconstructed one APT_r .

Figures 69 to 71 show the results on this evaluation set for the proposed approach and for the state-of-the-art methods by Maksimovic et al. [18], and Nucci et al. [19].

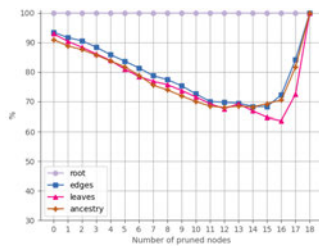


Figure 69. Reconstructed phylogeny trees results for own approach.

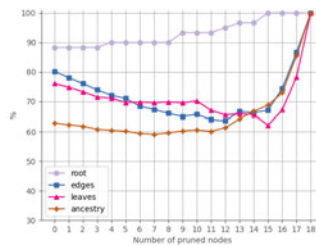


Figure 70. Reconstructed phylogeny trees results for method from [18].

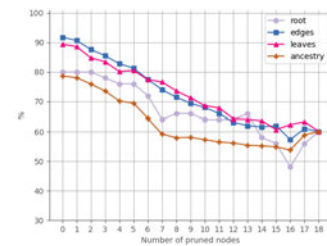


Figure 71. Reconstructed phylogeny trees results for method from [19].

Unlike the existing state-of-the-art methods, our algorithm was able to identify correctly the root of all phylogeny trees in the evaluation set independently from the amount of nodes which were pruned. The amount of edges and leaves which were identified correctly is systematically higher than in [18], and decrease at a slower pace than in [19], even though the pre-existing proposal is based on an exhaustive search. Lastly, our method retrieves the highest amount of parent-child relations across generations, as reflected by the ancestry measure being the highest.

The experiments we conducted with the extended set of transformations including in addition pitch shift and time stretch, proved the extensibility of our network for a minimal cost of retraining the network while the performances stay stable, see Figure 72.

8.4.3. Conclusion

The presented approach to audio phylogeny outperformed the current SOTA while maintaining computational efficiency, and retained its performance after expanding the initial set of transformations, showing that it can be extended at a minimal cost. Thanks to its transformation detection performance, we believe that it can support many applications. Hence, it is an ideal choice for the provenance graph building task of the overall audio provenance analysis framework, providing a robust tool for verifying the authenticity and lineage of audio files.

8.4.4. Relevance to AI4media use cases and media industry applications

The proposed Audio Provenance Analysis method contributes to use case 1 by providing tools for audio verification, and to use case 4 by providing tools for comparison of audio objects in archives.



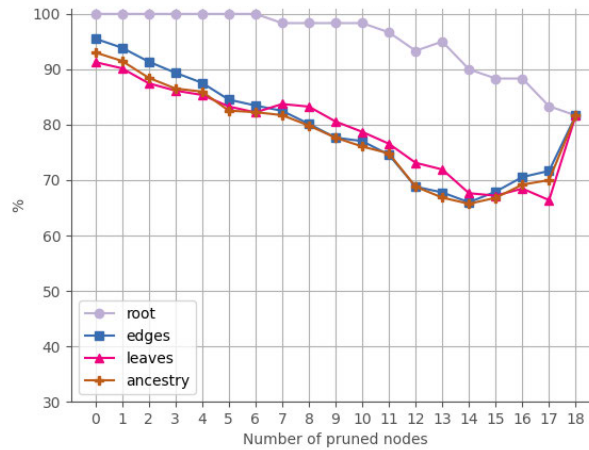


Figure 72. Reconstructed phylogeny trees results for own approach with extended set of transformations

8.4.5. Relevant Publications

- M. Gerhardt, L. Cuccovillo and P. Aichroth, "Advancing Audio Phylogeny: A Neural Network Approach for Transformation Detection," 2023 IEEE International Workshop on Information Forensics and Security (WIFS), Nürnberg, Germany, 2023. [667]
Zenodo record: <https://zenodo.org/records/10124333>

8.4.6. Relevant software/datasets/other outcomes

- IDMT Audio Phylogeny Dataset [668]
Zenodo record: <https://zenodo.org/records/8135331>





9. Research on Large Language Models for the media industry

9.1. Overview

There has been an explosion of Large Language Models (LLMs) recently. Following this trend, **Task 5.7 (T5.7)** “Research on Large Language Models for the media industry” is focused on new research exploring different aspects of LLM use in the media industry. An internal open call was organized where AI4Media beneficiaries were able to submit proposals for LLM-focused mini-projects. An internal evaluation committee evaluated the submitted proposals and selected three of them for funding. The rest of this section is a detailed description of the challenge that each project tried to tackle and an extensive report about methodology and results.

9.2. LLMs for media content editorial segmentation

Contributing partner: RAI

9.2.1. Challenge

The advent of Large Language Models and their wide availability both as proprietary solutions and as openly available models has represented a revolution for many business fields. The media sector is no exception as the large and ever-growing amount of media content produced and distributed every day poses serious challenges for ensuring their findability and accessibility. One of the key unsolved problems in this context is the ability to find relevant parts, e.g. short clips or larger segments (for example the individual news stories of a newscast or sub-clips at different points in a programme where the same topic is addressed) that can have an independently exploitable nature on publication platforms and that can be identified following multiple segmentation criteria (e.g. topic-, event- or editorial-based).

Approaches to automated multimedia segmentation vary according to the content genre, e.g., newscasts, movies, fiction, documentary, and are often based on genre-local heuristics or on some axiomatic definition of the atomic unit of which content is composed of (e.g., steps of a recipe). The main criticism is that most of the efforts focused on a mono-modal analysis (mainly visual shots), as if the underlying assumption was that segments are distinguishable by visual features only and address the segmentation problem using predefined static criteria (e.g. movie scenes). Though research on this specific task using existing and emerging multi-modal LLMs (like GPT4V or LLaVA) is still lacking to fully assess their performance, these seem overly complex and resource consuming, as well as mostly proprietary and difficult to refine.

Differently from existing mono-modal mono-criteria approaches, in this work we will firstly merge information coming from different channels (visual, aural, textual) in a unified textual domain and then use the LLMs to extract/abstract information from this merged domain under several dynamic criteria. We call this approach trans-modal, since the idea is that of transforming/translating different media channels into the textual domain where to exploit LLMs’ power to process both structured and non-structured text. In other words, while recent more advanced approaches are multimodal (performed e.g. through Llava, GPT4o, Gemini), i.e. they rely on independent processing pipelines to extract information from multiple channels (e.g., textual, acoustic, visual) and then elaborate the resulting data in a common hybrid token space normally providing a final textual output, transmodal analysis is an alternative approach that operates by generating textual descriptions conveying contributions from all available channels and making LLMs process them towards different objectives (see Figure 73).

The outcomes of this work are a set of novel methods aimed at integrating and utilizing LLMs for the multi-criteria content segmentation task. This is substantiated by a set of AI models made available to the community (see table 80), produced through zero-shot, few-shot or finetuning approaches.



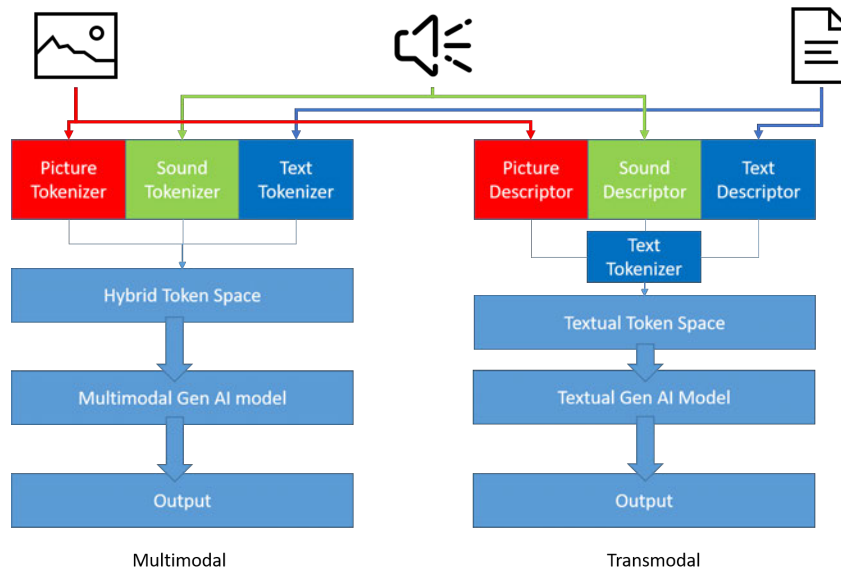


Figure 73. The concept of transmodality.

9.2.2. Related Work

The research work on multimedia content segmentation has a long history. The work presented in [669] offers a good survey of the “pre” deep learning efforts until 2013 by grouping onto seven categories the approaches based on the combination of three classes of low-level features – visual, audio and textual. For the sake of relevance and compactness, we will focus our related work search to the most significant latest developments, all exploiting the power of deep-learning techniques. We also exclude works that employ content recognition techniques (e.g., video or audio instance recognition) to identify predefined audio-visual patterns since - although useful in many practical cases - they do not generalise and need continuous integration of the reference data.

In general terms, the approaches vary according to the content genre, e.g., newscasts, movies, fiction, documentary and are often based on genre-local heuristics or on some axiomatic definition of a scene, i.e. the atomic unit of which longer content is composed of. In many works, once a scene is defined by some agreed characterisation, the technical approach translates this characterisation into some algorithmic form detecting the corresponding pattern. The work in [670] uses a siamese network to learn a discriminating metrics between consecutive shots. In [671] authors use an automated image captioning tracking approach to group keyframes in scenes, thus considering only the visual channel as carrier of the segmentation cues. This is philosophically similar to the work in [672], where authors use detected objects in the visual channel to perform matching among shots. Authors of [673] define a scene as "a plot-based semantic unit, where a certain activity takes place among a certain group of characters" and use a combination of local and global visual shot analysis to optimise scene boundary detection. Although the work shows a good generalisation performance towards other genres, the development is heavily grounded on assumptions derived from working on a movie-only dataset. Authors of [674] propose a self-supervised shot contrastive learning approach (ShotCoL) to learn a shot representation that maximizes the similarity between nearby shots compared to randomly selected shots. The very recent work in [675] uses a two-stage approach in which a video shot representational stage is followed by a scene segmentation stage. In both these last two cases, training and optimisation are performed on a movie dataset. In [676], authors achieve good cross-domain generalisation approaching the problem of long video segmentation through a combination



of short-range and long-range analysis using state-space transformers, but only considering the visual channel as information input for the segmentation task.

Other works like [677] focus more on a topic based segmentation of media content, aiming to obtain a segmentation in which each segment is semantically homogeneous and different from the previous and the upcoming. The idea behind these works is to use the same approaches used to detect topic in texts in order to segment transcript coming from media content according to the topic treated. The aforementioned types of text segmentations were first done via “simple” statistical methods like [678]. Then approaches based on probabilistic topic models like Latent Dirichlet Allocation [679] were used to analyse the changing of topic within a text, see [680] for an example. Currently these methods have been improved using textual embeddings; indeed, in works like [681, 677] the text (transcription of content) is divided in groups of sentences and then, after an analysis of the similarity between adjacent groups, changes of topic are detected providing a topic based segmentation.

Approaches departing from mono-modality are few, like for example [682], in which they rely on a combination of visual and textual features augmented with temporal information to improve shot clustering. A similar approach is that presented in [683], which is based on audio-visual deep features for shot genre prediction and successive aggregation. Authors of [684] apply learnable Optimal Sequential Grouping downstream of a video and audio embedding extraction scheme. Still, the evaluation is done on a limited-size movie only dataset.

9.2.3. Objectives

Editorial segmentation of media content is a complex process in which together with operational purposes and specific criteria, cultural and societal aspects are involved as well, which are difficult to isolate and rigorously define. The initial part of the project was dedicated to understanding the many aspects related to the target research and formulating a consistent research problem showing – at the same time – a promise for concrete and useful outcomes. The issue of editorial segmentation has been addressed by research literature across many decades, although not being one of the fields with major breakthroughs. In fact, most of the efforts have been spent into defining and testing specific, mostly heuristic-based, algorithms able to tackle the issue in particular domains (e.g., news, movies, online instructional videos) but no general approach has proven successful so far. The clear drawback of such heuristic approaches at segmentation is that they cannot generalise out of the original domain in which are crafted, therefore every system based on such approaches is inherently non-scalable. The segmentation process is performed by documentalists until now. Turning the project solely into a method to emulate this process was readily identified as an uninteresting direction to pursue in favour of a more general one, namely that of constructing a segmentation framework rather than a single tool. With this goal in mind, the overall research objective has been formulated as follows: study of research and experimental methods aimed at building an AI-supported framework for media editorial segmentation, taking into account purposes and subsequent segmentation criteria. The main requirements of this framework can be summarised as follows:

- media genre independence: the framework shall be able to operate across several content genres without specific or explicit configuration by the user;
- heuristics-free operation: the framework shall not depend on external or aprioristic rulesets dictating what observable features are direct hints for segmentation points;
- segmentation purpose adaptability: the framework shall flexibly adapt segmentation criteria to adhere to a user-defined segmentation purpose.

The following sections are organised as follows: Section 9.2.4 describes the adopted methodology, including several fundamental definitions useful to understand the theoretical formulation. Section 9.2.5 introduces and discusses the metrics used to evaluate the various segmentation methods. Section 9.2.6 briefly describes some initial approaches at introducing learning in the envisaged framework. Finally, Section 9.2.7 reports experimental results.





9.2.4. Methodology

In this work we opted to experiment what we call a *transmodal* approach. Let's start with some fundamental definitions and terminology.

Definition 9.1 (Edit Unit). An edit unit is an arbitrary atomic piece of media content rendered in time. For example, the rendered sound and pictures from second 00:01 to second 00:02, or from frame 115 to frame 125.

Definition 9.2 (Timeline). A timeline t is the ordered sequence of edit units of a piece of media content C . For example the entire play of all frames of a media item. An edit unit t_i is adjacent to t_j when it follows t_j in the timeline.

Definition 9.3 (Segmentation). A segmentation σ is a proper partition of a timeline t . Each element s of σ is an ordered set containing only adjacent edit units. By construction, each element $s \in \sigma$ is a subset of the timeline t .

Definition 9.4 (Segment Description). Each segment $s \in \sigma$ can be associated to a generic set of descriptive features $f_s \in \Phi$, so that for each element of σ there is a corresponding element in Φ . We call this relation Description, $D: \sigma \rightarrow \Phi$.

Definition 9.5 (Segmentation Inclusion). A segmentation σ_i is included in σ_j , and we write $\sigma_i \triangleleft \sigma_j$, if $\exists s_a \in \sigma_i: \exists s_b \in \sigma_j: s_a \subset s_b$ and $\nexists s_c \in \sigma_j: \exists s_d \in \sigma_i: s_c \subset s_d$.

Definition 9.6 (Transmodal Trail). A Transmodal Trail is any segmentation σ_T for which $\forall s \in \sigma_T: D(s) = f_s$ is a descriptive feature that combines information coming from one or more content tracks. We call *scenelets* the elements of a Transmodal Trail.

In general terms, the developed approach can be seen as a system which provides a succession of segmentations $\Sigma = \{\sigma_0, \sigma_1, \dots, \sigma_N, \dots\}$ of a content C following a purpose π :

$$\sigma_0 = S_0(C, \pi) \quad (106)$$

$$\sigma_1 = S_1(\sigma_0, C, \pi) \quad (107)$$

...

$$\sigma_N = S_N(\sigma_{N-1}, C, \pi) \quad (108)$$

...

where σ_0 is the fundamental segmentation and $\sigma_1, \dots, \sigma_N$ are aggregative segmentations, defined through a hierarchical segmentation scheme as illustrated in Algorithm 1.

The purpose π , expressed in generic natural language, conveys the intention for which the segmentation is being performed. In fact, depending on the target application domain, the most appropriate segmentation can vary considerably to the extent that a reference segmentation valid for a wide range of possibilities is not really conceivable. The purpose π is exactly intended to model this situation, acting as a global parameter governing the automated segmentation process.

The key idea is that an optimal implementation of functions S_i can be achieved through the integration of LLMs due to their flexibility in processing textual inputs of various nature and structure. In the following paragraphs, we introduce the key definitions and the fundamental functional relations of the process.

Definition 9.7 (Genre Extraction Function). We define a genre extraction function $\gamma(C)$ as a function associating genre information to a media content C . Genre information can be expressed in any textual format as long as it conveys the key descriptive information that defines a media genre.





Algorithm 1 Hierarchical segmentation algorithm

Input: C, π

Output: Σ

$i \leftarrow 0$

$\Sigma = \emptyset$

$\sigma_i \leftarrow S_0(C, \pi)$

while $i=0$ OR $(\sigma_{i-1} \triangleleft \sigma_i)$ **do**

$\Sigma \leftarrow \Sigma \cup \{\sigma_i\}$

$i \leftarrow i+1$

$\sigma_i \leftarrow S_i(\sigma_{i-1}, C, \pi)$

end while

SegmentXX, audio type: reading, main image content: zodiac signs, speaker YY says :

Relazioni che devono essere sempre gestite con prudenza, perché ricordate che l'acquario detesta la morbosità, le persone troppo appiccicose. Si avvicina anche un weekend importante e poi il sagittario. Cielo molto valido, chissà che già la prossima settimana non ci sia stato un qualcosa di più. Innovazioni e devo dire anche amore, amore positivo, incontri che valgono, idee vincenti. Vi abbraccio e vi aspetto.

Table 70. Example of a scenelet. The orange text is the segment label (corresponding to an edit unit in the programme's timeline). The green and cyan text represent the audio and video classification as detected by two state-of-the-art zero-shot audio and image classifiers, respectively. The yellow text labels the speaker as per the output of speaker diarization. The plain text is the audio transcription.

Definition 9.8 (Topics Extraction Function). We define a topics extraction function $T(C)$ as a function associating topic information to a media content C . Topic information can be expressed in any textual format as long as it conveys the key descriptive information that defines a topic which is being addressed in the content.

Definition 9.9 (Transmodal Generation function). We call Transmodal Generation Function a generic function extracting a Transmodal Trail from a media item C . We denote a generic Transmodal Function as $\tau(C)$. Descriptors f_s of segments of Transmodal Trails can be expressed in different formats, including structured textual descriptions. Table 70 reports an example of an element of a Transmodal Trail (scenelet). The textual information collects aspects coming from several tracks of the original content.

Definition 9.10 (Criteria Generation Function). We define a segmentation criteria generation function $\Gamma_0 = \Gamma_0(T, \gamma, \pi)$ as a function that associates a list of segmentation criteria depending on the content topics $T(C)$, the genre $\gamma(C)$ and on the segmentation purpose π . Segmentation criteria can be expressed in natural language.

Definition 9.11 (Fundamental Segmentation Function S_0). Given a Transmodal Trail σ_T of a media item C , we define ϕ_0 as a function that filters σ_T by selecting elements representing a change according to a criteria set Γ_0 . We can then write:

$$\tilde{\sigma}_0 = \phi_0(\tau(C), \Gamma_0(T(C), \gamma(C), \pi)) \quad (109)$$

We have therefore that $\tilde{\sigma}_0 \subseteq \tau(C)$. However, $\tilde{\sigma}_0$ is not yet a proper segmentation, because it is not a proper partition of the timeline of C . To obtain the proper partition σ_0 , it is sufficient that we modify each segment of $\tilde{\sigma}_0$ so that its ending element coincides with the previous element of the starting element of the following segment (gap-filling). Finally we have:

$$\sigma_0 = S_0(C, \pi) = \text{gapfill}(\tilde{\sigma}_0) \quad (110)$$





It is clear that the function $\tau(C)$ is key to determine the way in which the process works, since it is a parameter for both criteria generation and segment filtering (Equation 109). To implement a function able to generate scenelets like that of Table 70, for example, we need to implement audio and image classification, speech transcription and audio diarization. The formal functional dependency between these data and τ is not reported here for the sake of simplicity, however we will describe a concrete implementation in Section 9.2.7.

Once the fundamental segmentation σ_0 is produced, the subsequent part of the algorithm is made up of a bottom-up aggregative process, as illustrated in Algorithm 1. Differently from $S_0()$, however, each segmentation function $S_i()$ operates on the intermediate segmentation σ_{i-1} rather than on $\tau(C)$. Furthermore, it uses C to extract descriptions for elements $s_{i-1,j} \in \sigma_{i-1}$ and topics $T(C)$, that are used to obtain segmentation refinement criteria.

Definition 9.12 (Refinement Criteria Generation Function). We define a segmentation refinement criteria generation function of order i , $\Gamma_i^{ref}(\sigma_{i-1}, \gamma(C), T(C), \pi)$ as a function that associates a list of segmentation refinement criteria depending on a given segmentation σ_{i-1} , the topics $T(C)$, the genre $\gamma(C)$ and on the segmentation purpose π . Segmentation criteria can be expressed in natural language.

Definition 9.13 (Merge Decision Function). A merge decision function $\delta(\Gamma, s_k, s_{k+1})$ associates a true boolean value to a couple of adjacent segments s_k, s_{k+1} if they are considered mergeable under criteria Γ , and a false boolean value otherwise.

Each S_i works by iteratively considering groups of subsequent segments in σ_{i-1} , that are either individual segments or partial aggregation of segments, as illustrated in Algorithm 2, where:

- function $join(s_i, s_k)$ returns a segment whose start is the start of s_i and whose end is the end of s_k ;
- function $describe_i(c, \sigma)$ returns a described version of σ based on metadata extracted from C , according to Definition 9.4. The function depends on the aggregation level i .

Algorithm 2 Segmentation refinement function S_i

Input: $\sigma_{i-1} = [s_{i-1,1}, s_{i-1,2}, \dots, s_{i-1,L_{i-1}}], C, \pi$

Output: σ_i

```

 $\gamma \leftarrow \gamma(C)$ 
 $T \leftarrow T(C)$ 
 $k \leftarrow 2$ 
 $\sigma_i = \emptyset$ 
 $\sigma_{i-1} = describe_i(C, \sigma_{i-1})$ 
 $\Gamma_i = \Gamma_i^{ref}(\sigma_{i-1}, \gamma, T, \pi)$ 
 $s_{curr} = s_{i-1,1}$ 
while  $k < L_{i-1}$  do
  if  $\delta(\Gamma_i, s_{curr}, s_k)$  then
     $s_{curr} = join(s_{curr}, s_k)$ 
  else
     $\sigma_i \leftarrow \sigma_i \cup \{s_{curr}\}$ 
     $s_{curr} = s_k$ 
  end if
   $k \leftarrow k + 1$ 
end while

```

It is clear that the algorithms and functions defined earlier are quite high-level and hide a certain non-trivial amount of implicit complexity. The conjecture (or thesis) of this work is that LLMs offer ways to





solve this complexity and provide adequate implementations of these functions for the specific task of media segmentation. In fact, empirical evidence gained in the early phases of this project showed that state-of-the-art LLMs (like GPT4 Turbo) are able to implement $\gamma(C)$, $T(C)$, Γ_0 , Γ_i^{ref} , $\delta(\Gamma, s_i, s_k)$ and $\phi_0(\tau, \Gamma)$ quite straightforwardly, after several rounds of prompt refinement, and with good quality and stability of results.

9.2.5. Metrics

One of the biggest challenges in media segmentation is assessing whether one segmentation is “better” than another. Assuming we have a ground truth segmentation σ^{gt} for a given media content C , the core issue lies in identifying a method to compare different segmentations against σ^{gt} to determine which one is the most similar. In the literature this is done in various ways, all based on discretized versions of the segmentations. We briefly recall these methods below and then propose a couple of different ways to directly compare σ and σ^{gt} .

9.2.5.1. Discretize the segmentation To discretize content C means dividing the interval $[0, T_C]$, corresponding to its duration T_C , into M consecutive subintervals of time (not necessarily equal), denoted as $\{\iota_i\}_{i=1}^M$. The segmentation σ of C can then be mapped into a binary sequence $\tilde{\sigma} \in \{0, 1\}^M$ (discretized segmentation) by assigning a label of 1 to each subinterval ι_i in which a segment $s \in \sigma$ starts, and 0 otherwise. There are primarily two types of discretization strategies: those based only on duration, such as discretizing the content second by second, and those based on the events in C , such as discretizing sentence by sentence ([681]) or according to the speaker. We chose to consider discretization on a per-second basis. This means that if the content C lasts M seconds, the segmentation $\tilde{\sigma}$ is a sequence of 0s and 1s of length M . We made this choice because it is a strategy adaptable to any content, as it does not depend on C , and it also appears to be a discretization that closely approximates the real segmentation.

The problem of comparing σ^{gt} with σ is then translated into comparing two binary sequences of equal length, $\tilde{\sigma}$ and $\tilde{\sigma}^{gt}$. This becomes a classic classification problem, where standard metrics such as precision, recall, accuracy, and F1-score can be used to compare the sequences. Note that the data is typically heavily unbalanced, with many 0s and few 1s, making the F1-score the most significant metric among those mentioned. The limitation of these “metrics” is that they rely on the exact matching of the labels in $\tilde{\sigma}$ and $\tilde{\sigma}^{gt}$, without considering that for the task considered, a label that is slightly misplaced is not as incorrect as one that is misplaced by a larger margin.

To address this issue, the authors in [685] defined a new metric called p_k that does not consider near misses as completely wrong. Later, in [686], a slight modification of p_k was introduced, usually denoted as W_d , which addresses some of the limitations of p_k . Both these quantities behave like true metrics, meaning that $p_k(\sigma, \sigma) = W_d(\sigma, \sigma) = 0$, and their values increase (up to 1) as the compared segmentations become more different. We refer the interested reader to [686] for more details. These two metrics have been widely used in the literature to evaluate segmentations, see for example [681], [680], and [677].

9.2.5.2. Comparing σ and σ^{gt} directly In order to avoid the dependence on the discretization strategy, we defined two additional quantities to compare segmentations. The first one, denoted by IoU , inspired by the intersection over union metric commonly used in computer vision tasks, is defined as follows: for each segments in σ^{gt} , we consider the maximum of all the values of intersection over union with respect to all the segments in σ ; the value of $IoU(\sigma^{gt}, \sigma)$ is the average of these maximums, i.e.

$$IoU(\sigma^{gt}, \sigma) := \frac{1}{|\sigma^{gt}|} \sum_{s \in \sigma^{gt}} \max \left\{ \frac{s \cap s'}{s \cup s'} : s' \in \sigma \right\}, \quad (111)$$

where $|\sigma^{gt}|$ is the number of segments in σ^{gt} . If the segmentations match perfectly the value of IoU is 1, while its minimum value is (theoretically) 0.





Note that IoU is not symmetric and so it is not a metric. Therefore, we considered mapping the segmentations into a space where it would be easy to use a real metric. In particular, we decided to map any segmentation σ of a content C into an element f_σ in the space $[0,1]^{[0,1]}$ of functions from $[0,1]$ to $[0,1]$, and on this space, we use a metric equivalent to the standard \mathbb{L}^1 metric on $[0,1]^{[0,1]}$. The bijective map from σ to f_σ is defined as

$$f_\sigma(x) = \sum_{s \in \sigma} \frac{i}{|\sigma|} \mathbb{1}_s(x \cdot T_C) \text{ for any } x \in [0,1], \quad (112)$$

where T_C is the duration of C and $\mathbb{1}_s(x) = 1$ if $x \in s$ and 0 otherwise. The function f_σ is a monotone increasing step function with positions, amplitude and number of steps depending on σ . Now we can define a metric d that establishes the distance between two segmentations σ, σ' using a variation of the standard \mathbb{L}_1 norm $\|\cdot\|_1$, defined as

$$d(\sigma, \sigma') = d(f_\sigma, f_{\sigma'}) := 2 \int_0^1 |f_\sigma(x) - f_{\sigma'}(x)| dx = 2 \|f_\sigma - f_{\sigma'}\|_1, \quad (113)$$

The factor 2 in eq. 113 is chosen to ensure that the distance between a segmentation consisting of a single segment and one composed of infinitely many infinitesimally small equal segments is 1.

9.2.5.3. “Metrics” Evaluation In this paragraph, we compare the different benchmarking techniques defined above. The strategy is to simulate a large number of segmentations and altering each of them modifying the number and/or the length of the segments. Then by using the “metrics” to compare the original and the perturbed segmentations, we can conclude that a metric is consistent if it evaluates as more different from the original ones for the segmentations which undergo stronger perturbations.

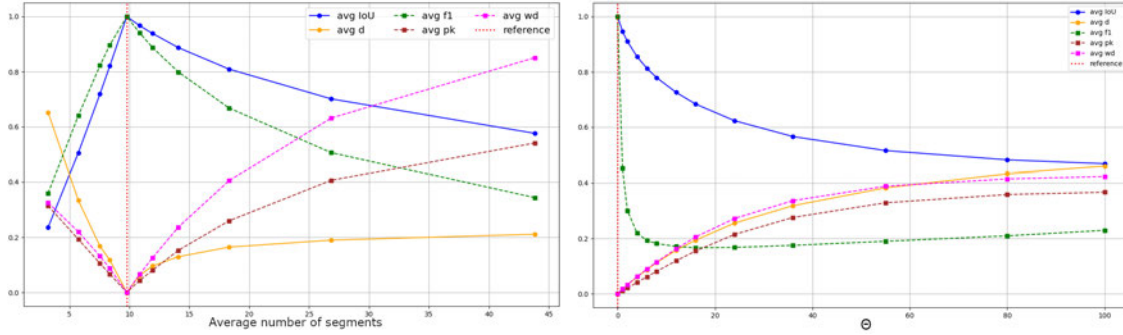
After a thorough analysis of the test set available at https://huggingface.co/datasets/raicrits/YouTube_RAI_dataset, we concluded that segmentations, intended simply as partitions of a time interval $[0, T]$, could be simulated as realizations of a Poisson process with parameter λ on $[0, T]$. Specifically, we can identify changes of segments within $[0, T]$ as occurrence times in the Poisson process. These simulated segmentations exhibit statistical similarity to those found in the real dataset. We chose $\lambda = \bar{\lambda} \approx 0.0097$ and $T = \bar{T} = 800$ ³² to generate a simulated dataset of 1000 segmentations originated from the same stochastic process on the same time interval. Afterward, we perturbed these 1000 segmentations in two ways:

- Adding and removing segments. This perturbation is performed for each reference segmentation simulating another Poisson process on $[0, \bar{T}]$ of arbitrary parameter λ . For adding segments we add to the reference segmentation points of segment change corresponding to the occurrences of the new simulated process, while for removing segments we remove randomly a number of segment changes from the reference segmentation equal to the number of occurrences in the new simulated process.
- Changing the length of the original segments, i.e. changing the positions of the points of segment change. This perturbation is performed by adding to the original position of each time instant in which there is a segment change a random value coming from a Gaussian random variable of mean 0 and standard deviation $\theta \in \mathbb{R}$.

It is clear from Figure 74 that the “metrics” are all consistent with the perturbations applied, except for the $F1$ score in the case of shrinking/stretching the segments as it is natural by its definition. We also experiment combining the two perturbations considered and the results are coherent. The important thing to notice is that each metric considered reacts in a different way to the perturbations, for example over-segmentations seem to be under penalized by d while they are strongly penalized by W_d . This does not permit to conclude that one benchmarking quantity is better or worse than the others. It would

³² \bar{T} and $\bar{\lambda}$ are approximations of respectively, the average duration of the contents in the real dataset and inverse of the average duration of the segments of each video (see [687] for details on the properties of Poisson processes).





(a) Effect of adding/removing segments on metrics

(b) Effect of shrinking/stretching segments on metrics

Figure 74. Average values of the metrics evaluated on the 1000 simulated reference segmentations against their perturbed versions. In (a) we added/removed segments while in (b) we changed the position of the points of segment change applying a Gaussian perturbation with standard deviation θ to them.

be necessary to perform extended user evaluations in order to see which one of the “metrics” defined (or combination of them) correlate the most with the human’s feedback.

9.2.6. Learning Paradigms

In this Section, we will explore how the presented algorithm can exploit learning to achieve its results. Section 9.2.4 described a completely self-contained solution, able to produce a segmentation succession Σ from the only inputs represented by the content itself C and a segmentation purpose π . In particular, the process flow is based on the Segmentation Criteria Generation Function Γ_0 , which formulates segmentation criteria based on the specific topics $T(C)$ and information about the content genre expressed by $\gamma(C)$. As such, the formulation is substantially based on aprioristic knowledge about media genres that might have been absorbed in their training phase by the LLMs implementing Γ_0 , contextualised by the specific topical content of C . To overcome this limitation, which might be a source of bias as well, we defined a learning paradigm with the objective of deriving segmentation criteria and corresponding purposes, thanks to the availability of a good amount of manual segmentations for which - however - original segmentation criteria and purpose are no longer available.

Let $\Delta = \{\Sigma_1, \Sigma_2, \dots, \Sigma_M\}$ be a set of manually generated segmentation successions of a content set $\mathbb{C} = \{C_1, \dots, C_M\}$ ³³. Let $\Delta^i = \{\sigma_{1,i}, \dots, \sigma_{M,i}\}$ be the set of manually generated segmentations of level $i \in \{0, \dots, I\}$. The idea is to try to distil criteria and purposes from the analysis of these data. For this, we then define a Segmentation Criteria Synthesis Function and a Purpose Inference Function as follows.

Definition 9.14 (Segmentation Criteria Synthesis Function). The segmentation criteria synthesis function $\tilde{\Gamma}_i = \tilde{\Gamma}_i(\Delta^i)$ is a function that distils criteria based on the observation of a collection of segmentations Δ^i .

Definition 9.15 (Purpose Inference Function). A purpose inference function $\Pi^i = \Pi^i(\tilde{\Gamma}_i)$ is a function that maps segmentation purposes to criteria $\tilde{\Gamma}_i$ in a way that criteria generally satisfy purposes.

As immediately observable, and already pointed out when discussing the other functions introduced earlier, these two functions are very abstract and imply the solution of very complex matters like criteria synthesis from data and purpose abstraction. Again, the conjecture underlying this approach is that LLMs (the means through which these two additional functions are implemented) are able to address this complexity and provide useful outputs.

³³The assumption, for the sake of simplicity, is that there is 1-1 relation between elements of \mathbb{C} and Δ , but the method can be generalised to a situation in which more than one segmentation is available for the same content item.



Thus, the execution of $\tilde{\Gamma}_i$ and Π^i produces instances of purposes π and criteria Γ_i , usable as learned parameters in Algorithms 1 and 2.

In particular, in a minimal configuration, i.e. when the only available manual segmentation data are at level 0 (fundamental segmentation), eq. 109 becomes:

$$\tilde{\sigma}_0 = \phi_0(\tau(C), \tilde{\Gamma}_0) \quad (114)$$

and line 7 of Algorithm 2 becomes $\Gamma_i = \Gamma_i^{ref}(\sigma_{i-1}, \gamma, T, \Pi^0)$. Table 71 reports an example of a learned purpose and corresponding criteria. The two rows were respectively learned from a dataset of talk shows and of newscasts.

Table 71. Examples of learned purposes and criteria.

Purpose Π^0	Description	Criteria $\tilde{\Gamma}_0$
Content Analysis	Segmenting the program to analyze content structure, diversity, and transitions for academic research or production review.	Speaker Change (Identifying when different individuals contribute to the discourse) Topic Shift (Determining the range of topics and their transitions within the program) Narrative Progression (Understanding how the narrative develops over time).
Editing and Post-production	To facilitate the editing process by marking points for potential cuts or transitions between different types of content.	Change in Reporting Style (Marks a transition between live and pre-recorded content, helping editors to arrange the sequence.), Shift from National to International Focus (Identifies a change in focus that might require different graphical overlays or contextual setup.), Transition to recorded segment (Identifies transitions to pre-recorded segments, which may be edited differently than live content.)

9.2.7. Test and Validation

9.2.7.1. Approach In order to validate the identified process, we chose to compare an implementation of our transmodal approach with several reference monomodal approaches on a collection of datasets. Below a list of these kind of approaches that we experimented with the respective identifier that we will use in the presentation of the results (when relevant).

- **STC**: a standard topic-change detection approach based on similarity between sentences, developed following the approach presented in [681]³⁴.

³⁴https://github.com/gdamaskinos/unsupervised_topic_segmentation








- **RTC**³⁵: A novel topic-change detection approach based on the fine-tuning of BERT model on the Sequence Classification downstream task, trained on a the test set of the YTDataset³⁶.
- **LTC**³⁷: A topic-change detection approach similar to the previous one but based on the fine-tuning of Meta Llama 3 model [688], using LoRa technique [689], on the same dataset used for RTC.
- **ATC**³⁸: Analogue to RTC but based on RoBERTa and trained on a different dataset mainly composed of news transcripts.
- **HEU**: A heuristic method developed for the internal purpose of segmenting news programmes in Rai, based mainly on the recurrent features occurring during segment changes in these particular kind of contents.
- **SDT**: Scene detection based on a Python library³⁹ used as a benchmark to compare methods aimed at segmenting content form a semantic point of view with a low-level visual-only structural segmentation approach.

9.2.7.2. Implementation Figures 75, 76 and 77 illustrate the implementation of our method, which we call *SegSmith*. In the pictures, blue blocks represent the implementation of the various functions introduced in earlier sections, green ellipses are the inputs and brown ellipses are data generated or transformed in the process. Table 72 reports the meanings of the icons decorating the functional blocks of the mentioned Figures. Blocks marked with the generative icon have been implemented with a LLM. Table 73 reports the component used to implement each of the functional blocks. Figure 75 includes a couple of functional blocks useful to build the Transmodal Trail like the one in the example of scenelet of Table 70, namely Diarization, Transcription, Role Inference, Img and Sound Class Inference, Img and Sound Classification. In particular, the Img and Sound Class Inference blocks infer what visual and audio classes are likely to appear in the content based on its genres. These classes are then fed to the two zero-shot classification blocks.

Table 72. Legenda for graphic labels in Figures 75, 76 and 77

Symbol	Name	Description
	Descriptive	It produces structured or unstructured descriptions of objects or phenomena based on models built on examples of pre-existing descriptions relating to a number of such objects (e.g. image classification)
	Transformative	It produces, starting from objects or phenomena of a certain type, equivalent objects or phenomena of different types (for example the production of the transcribed text starting from speech or vice versa)
	Generative	It produces, possibly conditioned to certain inputs (so-called «prompts»), instances of objects (e.g. images, text) on the basis of probabilistic models that represent in a compact way the essential characteristics of the sets of such objects (e.g. Large Language Models)

³⁵https://huggingface.co/raicrits/BERT_ChangeOfTopic

³⁶https://huggingface.co/datasets/raicrits/YouTube_RAI_dataset

³⁷https://huggingface.co/raicrits/Llama3_ChangeOfTopic

³⁸https://huggingface.co/raicrits/topicChangeDetector_v1

³⁹<https://www.scenesdetect.com>





Table 73. Implementation details

Figure	Tool	Component
Figure 75	Diarization	pyannote ⁴⁰
Figure 75	Transcription	whisperX ⁴¹
Figure 75	Role Inference	GPT4 Turbo
Figure 75	Genre Inference	GPT4 Turbo
Figure 75	Img and sound class Inference	GPT4 Turbo
Figure 75	Topics Extraction	GPT4 Turbo
Figure 75	Img Classification	laion/CLIP-ViT-bigG-14-laion2B-39B-b160k ⁴²
Figure 75	Sound Classification	laion/larger_clap_general ⁴³
Figure 75	Transmodal Trail Creation	procedural
Figure 76	Segmentation Criteria Inference	GPT4 Turbo
Figure 76	Segmentation	GPT4 Turbo
Figure 77	Refinement Criteria Inference	GPT4 Turbo
Figure 77	Segmentation Aggregation	GPT4 Turbo
Figure 77	$\sigma_{i-1} < \sigma_i?$	procedural

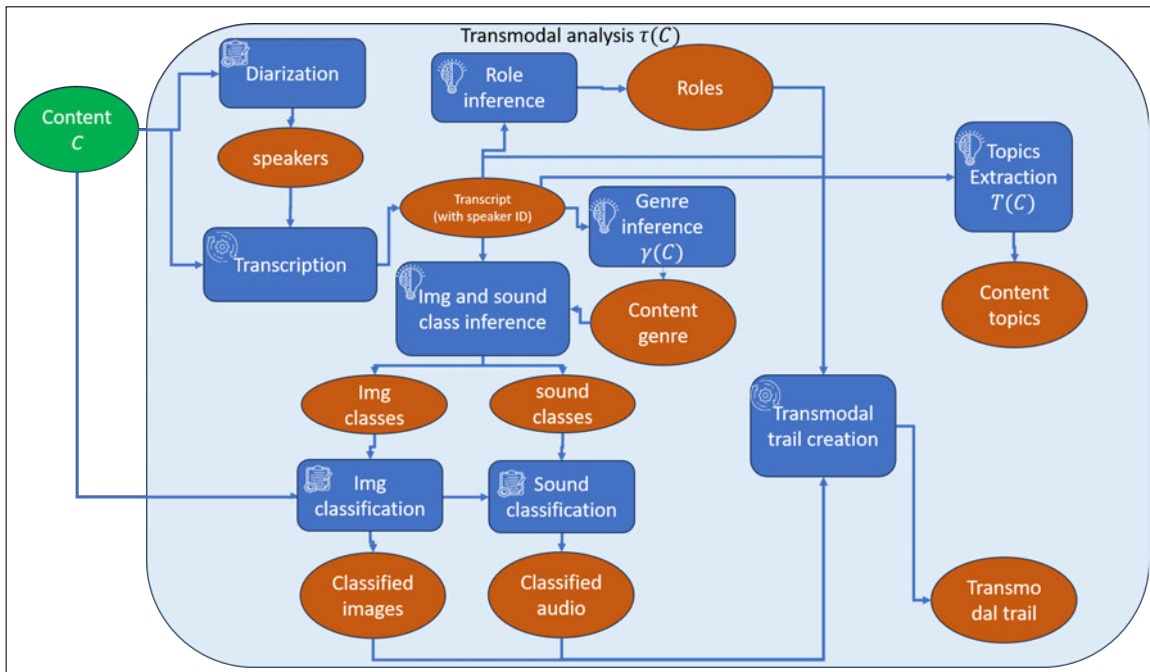


Figure 75. SegSmith: Implementation of the Transmodal Analysis.

⁴⁰<https://huggingface.co/pyannote/speaker-diarization-3.1>

⁴¹<https://github.com/m-bain/whisperX>

⁴²<https://huggingface.co/laion/CLIP-ViT-bigG-14-laion2B-39B-b160k>

⁴³https://huggingface.co/laion/larger_clap_general



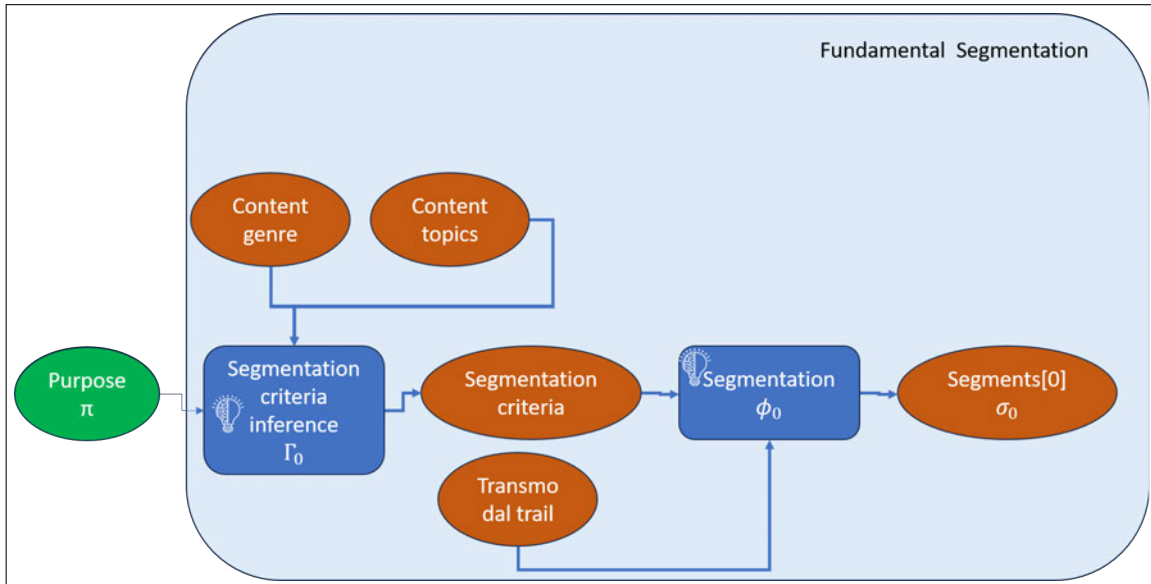


Figure 76. SegSmith: Implementation of the Fundamental Segmentation S_0 .

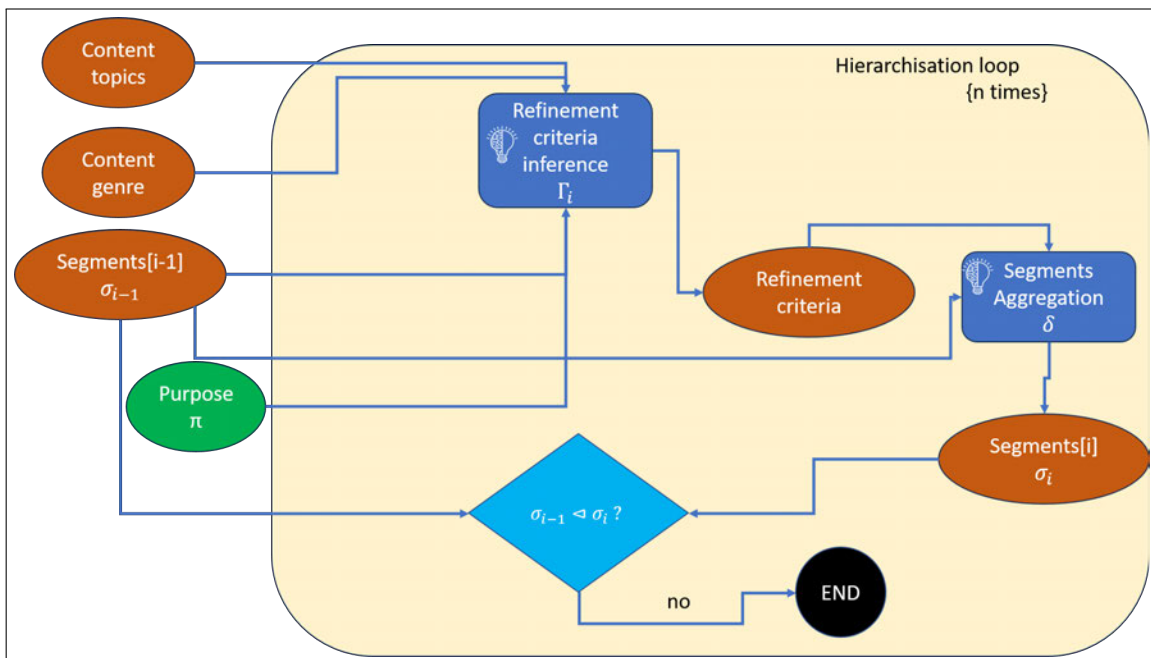


Figure 77. SegSmith: Implementation of the Hierarchical Segmentation S_i .

9.2.7.3. Data Collection and Analysis To test our approach, we collected a series of datasets composed of material coming from RAI archives and RAI’s public social media services, to test possible solutions on segmentation cases whose guiding purpose was varying. To accommodate the genre independence requirement, we collected a number of programmes of several different genres. The amount and genre of these programmes is summarised in Table 74.



Genre	Short Name	Nr. of Programmes	Total Duration
Culture, Talk Show, In-depth news, Investigative journalism, Morning Show	CMMDataset	70	93 h
News bulletin	ANTSDataset	32	15 h
Talk Show, Interviews, Lifestyle, Game Show, Cooking, Politics debate, Astrology Show	YTDataset	2169	458 h
Cultural, travel, and historical documentaries, True Crime	CMMDocDataset	16	18 h

Table 74. Experimental datasets

To better characterise the features of the available segmentation ground truth we performed a detailed analysis on the CMM Dataset. The CMM Dataset, where CMM stands for *Catalogo Multimediale* in Italian (*Multimedia Catalogue* in English), refers to the multimedia catalogue of Rai’s programmes. A total of 70 episodes were randomly selected from 7 programmes aired in 2023, within the date range of January 3, 2023, to December 20, 2023. The genres of these programmes include Journalistic Insight, Inquiry, Society and Customs, and News. More detailed information about the dataset is provided in Table 75.

The programmes were segmented by human annotators. Each programme consists of segments defined by a *start cut* and an *end cut*, which correspond to specific moments within the programme. These segments can be contiguous, where the *end cut* of one segment matches the *start cut* of the next, or non-contiguous.

Figure 78 shows examples of segmentation for 5 episodes. The solid line segments indicate parts of the programme separated by vertical bars representing the start and end cuts. In contrast, the dotted line segments indicate portions of the programme that do not belong to any segment. These dotted segments typically correspond to advertisements or nested programmes within the main programme.

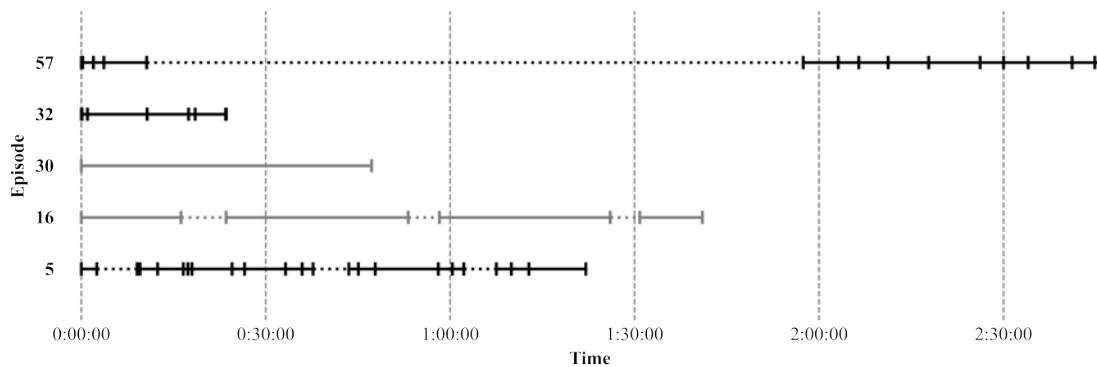


Figure 78. Segmentation examples (ground truth).

For instance, Episode 57 has 3 segments followed by a long dotted portion, likely referring to a nested program given its length. In Episode 32, there are no dotted parts, indicating that all segments are contiguous. Episode 30 consists of a single long segment. Episode 16 is composed entirely of non-contiguous segments. The lines for Episodes 30 and 16 are colored grey because they represent cases





Program	Type	Episodes Count	Segments Count	Average Length (min)	Date Range
Report	Journalistic insight	10 (2 with FGS)	74 (56 with FGS)	98.00 (14.37 s.t.d.)	2023-01-07 to 2023-08-21
Agorà	Journalistic insight	10 (10 with FGS)	191 (191 with FGS)	60.40 (37.18 s.t.d.)	2023-01-03 to 2023-08-28
Petrolio	Inquiry	10 (1 with FGS)	26 (14 with FGS)	58.40 (35.92 s.t.d.)	2023-06-30 to 2023-11-04
Unomattina	Society and customs	10 (1 with FGS)	48 (16 with FGS)	120.30 (56.09 s.t.d.)	2023-01-21 to 2023-09-16
Porta a porta	Journalistic insight	10 (1 with FGS)	55 (19 with FGS)	101.30 (12.53 s.t.d.)	2023-01-10 to 2023-09-14
Tg Parlamento	News	10 (9 with FGS)	51 (50 with FGS)	4.50 (0.92 s.t.d.)	2023-01-30 to 2023-12-20
Presa diretta	Journalistic insight	10 (6 with FGS)	151 (144 with FGS)	102.90 (14.73 s.t.d.)	2023-02-18 to 2023-09-25

Table 75. CMMDataset detailed composition.

where segmentation is either done only with non-contiguous segments or consists of a single segment. This analysis pointed out that this scenario occurs in 40 out of 70 programs, while the remaining 30 have a more fine-grained annotation, as seen in Episodes 57, 32, and 5. For a more comprehensive view of this phenomenon, refer to the complete chart attached in Figure 79.

A conclusion that can be drawn from this analysis is that there is heterogeneity in the annotations both *inter* and *intra*-programme. This variability could be due to different annotators or varying annotation methods by the same annotator at different times, or even due to varying time constraints annotators were given during the completion of their task. However, since we do not have data on who annotated each programme, we can only state that the criteria by which the programmes in the dataset have been segmented is highly heterogeneous and inconsistent. Furthermore, the lack of contiguity in segmentation (the dotted lines of Figures 78 and 79) must be taken into account in metrics computation since the algorithms must be evaluated only on the parts in which the annotators provided the explicit segmentation. These observations, although limited to a single instance of data, pointed out the risk that the quality of ground truth segmentation may be sometimes not adequate to effectively measure the performance of automatic algorithms. This led us to build other experimental datasets quite carefully: both the CMMDocDataset and ANTSDataset contain programmes which do not suffer from the aforementioned issues.

9.2.7.4. Experimental Results Let's now see how the models introduced in 9.2.7.1 perform compared to our method *SegSmith*. Our method was applied using different configurations of its parameters to see how these parameters change its behavior and performances.





We analyzed the datasets described in Section 9.2.7.3 separately, segmenting their items and evaluating the segmentations obtained with the considered methods against the ground truth available using the quantities described in Section 9.2.5. As we anticipated in that Section, for the metrics computed on the discretized segmentations we considered the items as discretized on a second by second basis. Notice that we did not consider all the methods available for every dataset but only the ones that seemed to be more significant, for example we considered HEU only on the ANTSDataset since for news programmes an efficient empirically-validated heuristics is available⁴⁴.

The results are summarized in Tables 76⁴⁵, 77, 78, 79 where for each segmentation method used it is reported the average value of the metrics considered. Notice that the column "#items" indicates how many content items were segmented with each technique. Numbers lesser than the total number of videos in the dataset for *SegSmith* are due to different reasons: provider's filter errors, timeout errors due to the high amount of calls to the OpenAI services or, in the case of the refined segmentations, no further refinement proposed by *SegSmith*. The values in the column " Δ segments" indicate the average difference of number of segments for the segmentations generated and the ground truth ones (positive values indicates over-segmentations, negatives under-segmentations). These values are useful for interpreting the values of the metrics considering what we observed in Section 9.2.5.3.

Table 76 includes the visual features - based shot detection model SDT, which shows the poorest performance except for one metric. This widely foreseeable result stresses the difference between the complexity of the editorial segmentation task w.r.t. the plain detection of low-level structural patterns. All Tables report alternative methods as labelled in Section 9.2.7.1 and transmodal methods labelled according to "<mode>_refined_<level>" pattern. The <mode> element is either "learnedcrits" or "abstractcrits", respectively indicating configurations in which purpose and criteria have been learned from data according to methods illustrated in Section 9.2.6 or in which the purpose is the fixed string "identifying topics and subtopics". In this second case, segmentation criteria have been generated by *SegSmith*. The <level> digit indicates the level of aggregative segmentation, as defined in Algorithm 2.

It is difficult to conclude from the experimental results if one model is better than another. As we already pointed in Section 9.2.5.3, each metric takes in account different factors in establishing how a segmentation is close to the reference one. Moreover, as observed in details in Section 9.2.7.3, the criteria used to provide reference segmentations by human annotators are not so homogeneous, even within the same dataset. This confirms that to really assess if one of the methods considered is significantly better than the others it would be necessary to perform an extensive user evaluation.

9.2.8. Potential impact on AI research/media industry/society

This project aimed at defining the overall concept of a genre-independent user-centric automatic media segmentation framework based on the integration of several AI tools, most importantly LLMs. We believe that this objective stands as an innovation mark w.r.t. previous state of the art in media segmentation for its generality and comprehensiveness, going beyond stereotyped research on few well-known media genres and datasets. The overall theoretical setting on which this work is based is still expression of an early conceptualisation, but we believe it can represent a reference for future work in the field of media segmentation. The implemented framework, named *SegSmith*, is an initial attempt at materializing the research and proved to be a sufficiently powerful software framework on which to build future versions of the algorithms.

The developed framework is relevant to a wide number of media applications and use cases, namely all those which benefit from chapterisation of longer content into smaller coherent units. Examples

⁴⁴For the sake of space it is impossible to fully account for the heuristics. Indicatively, it is based on the elicitation of the anchorperson among the speakers detected via speaker diarization, and then considering segments as starting every time the anchorperson starts speaking again.

⁴⁵In the experiments conducted on this dataset, we selected programmes which were not affected by the heterogeneity problems discussed earlier.





	d	p_k	W_d	IoU	$F1$	Δ segments	#items
learnedcrits_refined_0	0.179	0.452	0.629	0.543	0.117	43.458	24
learnedcrits_refined_1	0.205	0.486	0.64	0.458	0.124	31.462	13
learnedcrits_refined_2	0.202	0.413	0.556	0.447	0.124	23.5	12
abstractcrits_refined_0	0.157	0.451	0.615	0.563	0.144	30.391	23
abstractcrits_refined_1	0.173	0.369	0.476	0.515	0.185	12.412	17
abstractcrits_refined_2	0.233	0.346	0.438	0.483	0.182	7.333	12
LTC	0.23	0.403	0.471	0.425	0.166	5.3	30
RTC	0.249	0.393	0.464	0.412	0.178	4.633	30
STC	0.308	0.387	0.414	0.312	0.218	-6.833	30
SDT	0.19	0.629	0.983	0.289	0.035	569.467	30

Table 76. Average metrics values on CMMDataset. In green, yellow and red the best, second best and worst values.

	d	p_k	W_d	IoU	$F1$	Δ segments	#items
learnedcrits_refined_0	0.179	0.571	0.865	0.394	0.05	58.6	10
learnedcrits_refined_1	0.169	0.406	0.58	0.468	0.062	23.5	8
learnedcrits_refined_2	0.16	0.389	0.531	0.46	0.083	8.0	4
abstractcrits_refined_0	0.15	0.504	0.794	0.467	0.047	57.0	10
abstractcrits_refined_1	0.242	0.416	0.561	0.464	0.072	14.25	8
abstractcrits_refined_2	0.314	0.408	0.5	0.366	0.08	5.125	8
LTC	0.197	0.429	0.525	0.423	0.089	3.5	16
RTC	0.202	0.486	0.572	0.428	0.068	5.562	16
STC	0.237	0.477	0.505	0.354	0.092	-4.5	16

Table 77. Average metrics values on CMMDocDataset. In green, yellow and red the best, second best and worst values.

of such situations among AI4Media use cases are UC1, UC2, UC3 and UC7. In general, foreseen advantages range from media documentation, annotation and indexing to impact & interaction analytics and marketing where the ability to associate key observations to coherent media segments enhances their interpretability and actionability.

9.2.9. Assets released to the community

As part of the work, we released several artifacts (software, dataset), summarised in Table 80. We are also planning to release a version of the implemented software framework.

9.2.10. Conclusions/future work

This work contributed to innovate the approach addressing the complex problem of genre-independent user-centric media segmentation task. Differently from previous approaches, this work introduced multimodality, in its here originally introduced transmodal conception, at the core of the envisaged solution for the task, and it based the development of the approach on a general theoretical/algorithmical formulation. As part of the effort, a study was conducted on the reliability of existing metrics, introducing a novel one too. Possible learning paradigms have been also explored. The provided implementation, that we named *SegSmith*, is complete from the point of view of the foundational theoretical setting and provides a first software framework to build future developments upon. The experimental results are not





	d	p_k	W_d	IoU	$F1$	Δ segments	#items
learnedcrits_refined_0	0.151	0.285	0.387	0.648	0.179	11.621	29
learnedcrits_refined_1	0.138	0.273	0.337	0.626	0.182	3.333	21
learnedcrits_refined_2	0.215	0.305	0.348	0.519	0.192	-3.714	14
abstractcrits_refined_0	0.19	0.209	0.25	0.646	0.201	3.000	30
abstractcrits_refined_1	0.185	0.242	0.275	0.549	0.199	-3.500	22
abstractcrits_refined_2	0.195	0.257	0.288	0.528	0.182	-6.000	10
HEU	0.08	0.124	0.137	0.769	0.25	-2.781	32
LTC	0.14	0.39	0.413	0.465	0.067	-3.938	32
RTC	0.188	0.432	0.453	0.389	0.069	-6.562	32
ATC	0.102	0.284	0.31	0.6	0.086	-3.719	32
STC	0.237	0.422	0.426	0.281	0.093	-14.188	32

Table 78. Average metrics values on ANTS Dataset. In green, yellow and red the best, second best and worst values.

	d	p_k	W_d	IoU	$F1$	Δ segments	#items
abstractcrits_refined_0	0.238	0.487	0.562	0.512	0.155	11.483	539
abstractcrits_refined_1	0.329	0.452	0.566	0.469	0.199	2.38	390
abstractcrits_refined_2	0.483	0.447	0.572	0.377	0.223	-0.623	244
LTC	0.361	0.407	0.444	0.425	0.211	-1.703	609
RTC	0.306	0.440	0.483	0.461	0.186	-0.376	609
ATC	0.628	0.425	0.434	0.286	0.251	-3.954	609
STC	0.311	0.457	0.484	0.436	0.215	-1.39	609

Table 79. Average metrics values on YT Dataset. In green, yellow and red the best, second best and worst values.

yet fully satisfying since they do not allow to draw very firm conclusions. However, the number of implied hyperparameters is still too high to make them statistically representative of the ability of the framework to fully address the original problem also because detailed analyses of available ground truth segmentation pointed out several quality issues. Future developments will include further studies on the effects of the various models' hyperparameters on the experimental results as well as a thorough user evaluation on several aspects including no-reference segmentation quality and segmentation purpose alignment.





Name	Type	Link
BERT-CTC	BERT Change of Topic Classifier	https://huggingface.co/raicrits/BERT_ChangeOfTopic
LLAMA-CTC	LLAMA 3 Change of Topic Classifier	https://huggingface.co/raicrits/Llama3_ChangeOfTopic
YTDataset	Media Segmentation Dataset	https://huggingface.co/datasets/raicrits/YouTube_RAI_dataset

Table 80. Assets delivered to the community from LLM research on editorial segmentation.



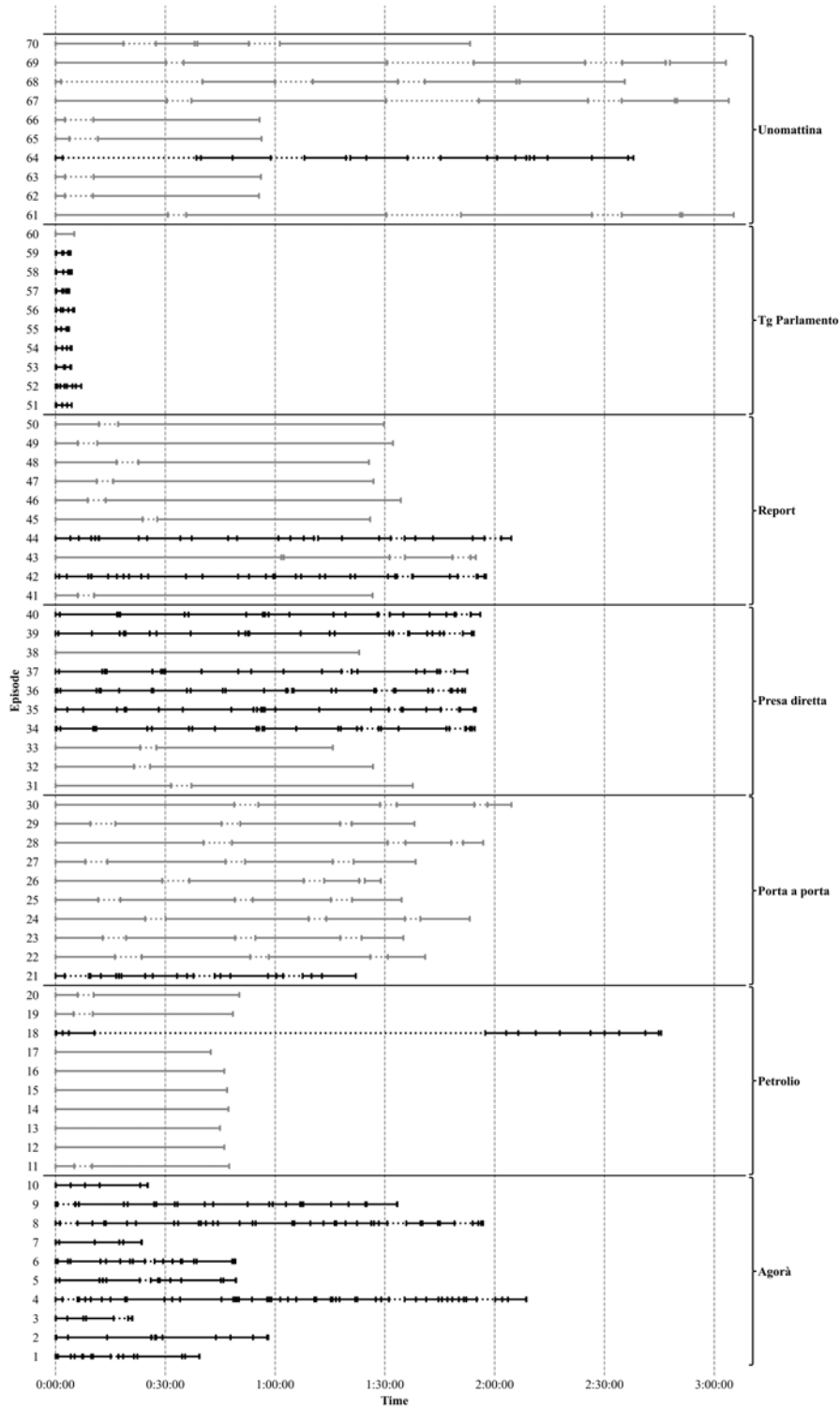


Figure 79. CMM dataset segments.





9.3. Evaluating LMMs on common sense and factuality

Contributing partner: CNR

9.3.1. Challenge

Despite the astonishing emerging capabilities of Large Language Models (LLMs) and, more recently, of Large Multimodal Models (LMMs), recent research is discovering many limitations of such models in comprehensively understanding complex multimedia data in a human-like fashion. While most of the recent effort has been put into LMMs that can digest still images to solve many vision-language downstream tasks, little attention has been paid to the more challenging video modality, with the time dimension strongly connecting the sequence of images at various levels. Furthermore, it is well known that these kinds of networks suffer from strong cultural biases and are therefore prone to better understand the Western world, with many drawbacks when the same models are deployed in multi-cultural scenarios.

Many downstream applications have started to require generative methods that can understand and digest long and unannotated streams of video data, as well as be resilient to specific cultural shifts. For example, this is of utmost importance for improving the value, accessibility, and protection of large-scale historical audiovisual archives, with impactful returns on cultural heritage accessibility and innovation in the cultural and creative sector. The study and the integration of LMMs in this domain are oriented towards developing novel tools that can automatically and effectively understand, extract information, and index large audiovisual archives, with the final result of increasing their accessibility to final users, their re-use and utility, even when dealing with complex and articulated semantic domains like the one of a national cultural patrimony.

Another important use case where video understanding is a key element is in the organization of personal multimedia archives, where people may like to retrieve shots and pose questions about long and untrimmed amateur or egocentric videos. This necessity is also demonstrated by the increasing interest in lifelong search challenges (LSC [690]) and egocentric tasks – like moment retrieval [691, 692] – requiring the understanding of long-range time dependencies across different portions of a raw video, as well as robustness to different intercultural scenarios.

As of now, these applications benefit from the development, in the last years, of cross-modal retrieval deep learning methods, which are able to retrieve videos or images most relevant to a given textual query [693]. Despite their large employment in large video search challenges like VBS [694] or TrecVID [695], these technologies cannot handle more complex browsing and search schema in which the user interacts with the system by asking complex questions on the video collection concerning long-range temporal dependencies (e.g., *Find the color of the jacket of the person taking the bus and commuting few minutes after at the main station*), or factual data (e.g., *What is the name of the famous building I encountered after having breakfast at the Bodeguita café?*).

CLIP-like retrieval models, while being extremely efficient at retrieval time, cannot handle free-form questions or perpetuate complex temporal or factual reasoning. On the other hand, while being promising, LMMs like LLaVa [696] or [697] (especially with RAG [693] capabilities) cannot be easily deployed on large-scale collections due to their inability to process long video streams and perform complex temporal and factual reasoning to come to a reasonable conclusion.

In this activity, we move the first steps towards better benchmarking LMMs for understanding long untrimmed videos and their capability of handling multi-cultural data, having in mind the above-mentioned scenarios.

9.3.2. Objectives

Although LLMs achieved remarkable performance in many classical downstream tasks like image captioning [698, 699] or image question answering [696, 700, 697], we believe that a careful evaluation of





such models on tasks concerning the understanding of untrimmed long videos is never been done in the literature. Some recent approaches [701, 702, 703, 704, 705, 706] try to answer similar questions. However, the sources of the videos are mostly TV shows and movies, making the available data skewed toward this domain. In light of this, in this activity we aim to understand the capabilities of the most promising LMMs in understanding long-range temporal dependencies in untrimmed videos and possibly understand their ability to be robust to multi-cultural scenarios. This is intended to be a first step toward developing more time-aware and bias-mitigated LMMs, which can process and understand videos efficiently and effectively.

To reach this goal, we designed a two-stage plan. Firstly, we propose a new benchmark consisting of relatively long video shots, with each video associated with a sentence involving objects and facts happening distant in time, which can be *True*, *False* or, in the edge case, *Ambiguous*. The videos are obtained from different places in the world to capture as much as possible multiple and diverse cultures. The sentences are generated in a completely automated way using existing LMMs, and their ground truth answers are obtained through crowdsourcing platforms that can offer efficient and reliable manual labeling. The proposed benchmark is constructed around already available footage, which is publicly accessible and usable. Secondly, we test some state-of-the-art LMMs, such as VideoLLaVa [707] and [697] on this novel constructed benchmark to effectively show the actual limitations of advanced multi-modal models on this challenging task.

9.3.3. State of the Art

Large Language Models (LLMs) have shown impressive performances in several Natural Language Processing (NLP) tasks. They are able to solve, in a 0- or few-shot setting, several NLP tasks that previously could exclusively be solved with specialized models [708].

Following this success, several approaches have been developed to let LLMs take images as input along text. These models are also known as Visual Language Models (VLMs) or, more generically, Large Multimedia Models (LMMs). These models often rely on a pre-trained LLM that is then further trained to process images as well [696, 709, 710, 711]. VLMs show strong performances on several Visual Question Answering Benchmarks as well as Image Captioning and more and are currently the solution behind several state-of-the-art solutions for vision and language understanding.

To further improve on integrating language modeling with the ability to process different kinds of inputs, novel models train on sequences of images to be able to process videos and create Large Multimodal Models (LMMs) [712, 707, 713, 714]. While video language models show some degree of temporal understanding, researchers are still working on optimal solutions to make these models proficiently understand temporal and physical relations in videos.

New benchmarks have been developed recently to measure how well LMMs perform in understanding videos. TVQA [701] is a Video Question Answering (VQA) dataset based on popular TV shows where questions involve localizing objects and understanding subtitle-based information. TVQA+ [702] is an extension of TVQA with bounding boxes and more fine-grained spatio-temporal questions. How2QA and How2R [703] are humanly annotated benchmarks – the former for multiple choice question answering and the second for image-video retrieval, based on HowTo100M [715]. NextQA [704] is a multiple-choice benchmark meant to assess LMMs temporal understanding. They focus on testing if LMMs can understand actions following each other in a video. AGQA [705] focuses on measuring how well LMMs can understand actions in videos and on carefully annotating the dataset so that different mistakes made by these models are not considered as equal when measuring LMMs understanding ability. STAR [706] is a benchmark meant to study the LMMs situational understanding, which is their ability to understand and answer questions requiring the understanding of the context presented in a video.

All these benchmarks tackle a similar problem to the one presented in this study, that is, the ability of LMMs to understand temporal relations in videos, which are the most prominent extension between understanding images and understanding videos. However, each of them is focused on specific domains and on specific ways to measure this complex ability of LMMs. Differently from the aforementioned





Figure 80. Examples of egocentric raw videos captured in different cities from the City Videos collection.

datasets, we focus on longer videos in the urban outdoor domain to understand how LMMs can generalize to these complex scenarios, requiring the different architectures to be robust to heterogeneous geographical – and therefore cultural – features.

9.3.4. Methodology

In this section, we carefully detail all the steps we followed to prepare our video benchmark. The overall preparation process can be divided into two main stages. Within the first stage, we aim to produce reasonable and unambiguous sentences concerning the content and events in the provided video that can be further evaluated through human judgment. This is obtained through carefully prompting existing LMMs and then further filtering their outcomes. The second stage, instead, consists of setting up a crowdsourcing platform for obtaining ground-truth answers from human annotators for the previously generated sentences. We detail these steps in the following sections and summarize the whole process in Figure 81.

9.3.4.1. Data Source As the primary source of visual data, we focused on videos from [716], which capture an egocentric perspective of a person walking through famous cities (City Videos) like Venice, Bangkok, Zurich, etc. We report some reference frames from some of the locations within the collection in Figure 80. These videos satisfy our needs in the following ways:

- the environment is mostly controlled as there is the presence of many everyday life objects and concepts, and it is free of unsafe content;
- as the cities shown in the video are well known, it is relatively easy to ground shots from the video into recognizable named places (famous squares, buildings, monuments, etc.), which can be used to test the ability of retrieval-augmented LMMs to access information outside the knowledge present in their weights;
- the video is a raw single-shot stream without any cuts, where events happen in large temporal windows, allowing us to test the ability of LMMs to process and understand long video sequences;
- being videos captured in diverse worldwide scenarios, we can later employ this benchmark to also quantify the cultural biases of the model.

9.3.4.2. Data pre-processing After defining the data sources, we performed initial data collection steps. The City Videos are currently focused on 10 cities worldwide and offer variegated data sources referring to different geographical locations and cultures. We focus on 8 cities: Amsterdam, Bangkok,



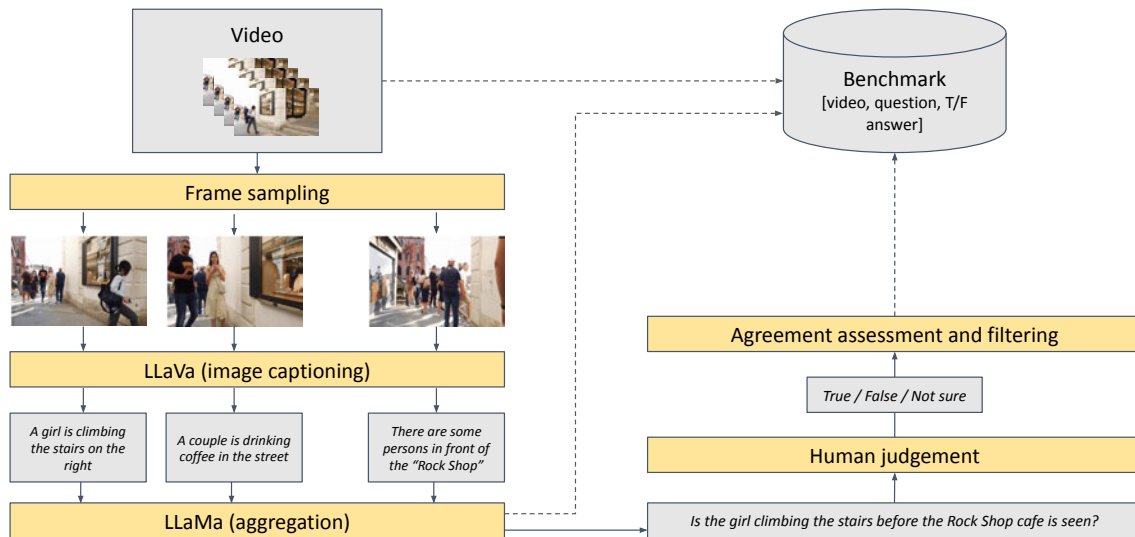


Figure 81. Overall annotation pipeline. On the left, we depict the automatic generation of sentences through off-the-shelf LMMs and LLMs. On the right, we show the manual judgment performed through crowdsourcing platforms.

Chiang Mai, Istanbul, Singapore, Stockholm, Venice, and Zürich. Although the final objective is to obtain a video-wise annotation, the pipeline is frame-based in many of its steps, as i) current state-of-the-art architectures that extract meaningful content from the visuals are tailored to analyzing still images and not videos, and ii) it is easier for the annotators that have to check the produced sentences to work on still images. For this reason, we extract frames from videos every 7 seconds. This amount of time seemed the right compromise for avoiding redundancy while capturing all the events happening in the video. The varying lengths of the videos provide us with a fair amount of images from each city.

The images from the videos contain several landmarks present within each city (see Figure 80 for example images) and provide a good starting point for video-level questions. We collect a total of 6,295 frames. Figure 82 shows how these images are divided among each of the cities.

9.3.4.3. Landmark recognition We experimented with the automatic extraction of landmarks, i.e., notable and easily identifiable locations within a city environment, e.g., statues, squares, and famous buildings. The idea is that landmarks can be a cornerstone for a retrieval-based approach to answering questions over hours-long videos. This was done through the use of an API provided by Google that allows the automatic identification of notable places based on the content of an image (we tested with all the frames from a video). However, in this way, we retrieve both landmarks in the sense we define above, as well as commercial places, such as bars and restaurants. This second kind of landmark is less interesting for our use case and provides too much noise to be reliably employed in our pipeline. Moreover, most of our frames are generic views of the cities, with no landmarks. In these cases, Google’s answers are mostly incorrect, with the service trying to guess a landmark from the visual similarity of images and returning a list of unrelated places, typically commercial. This is likely due to the assumption of Google’s service about the intent of the query, i.e., recognizing a landmark that is assumed to be present in the image, while our intent is to tell what landmark we are seeing only if the image depicts a landmark. This mismatch in detecting the presence or absence of a landmark causes a flood of false positives. We also tried setting up a crowdsourcing activity to filter out such false positives, but it turned out that relevant landmarks retrieved using this method were a minimal part of the data, and the cost

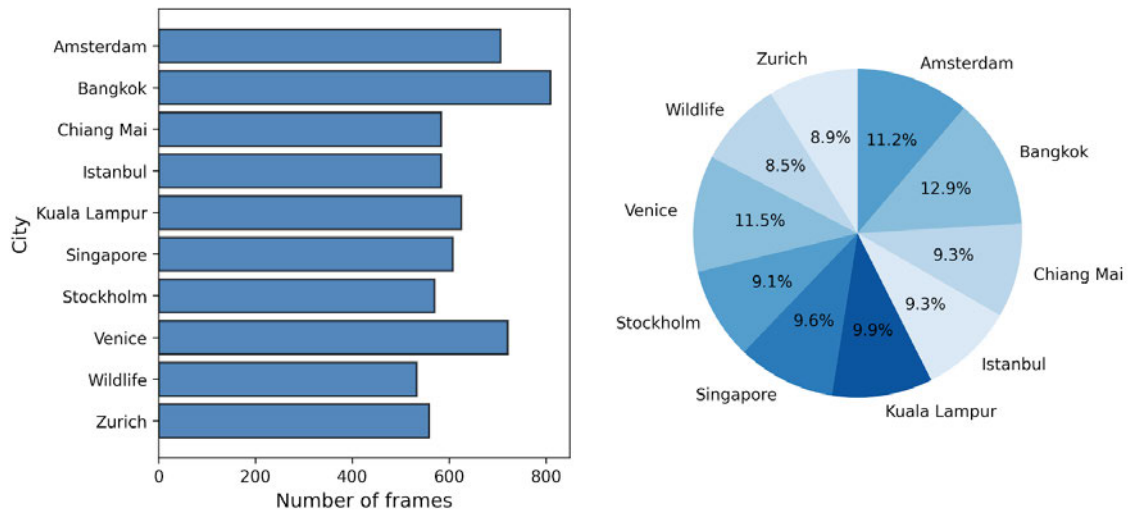


Figure 82. Number of frames taken from each City Video and percentages over the total.

of the crowdsourcing activity would have required most of our budget.

Given the excessive number of false positives produced with the above landmark extraction procedure, we were not able to incorporate this into the main dataset construction pipeline. However, we are still working to reduce the number of false positive, by experimenting with different approaches for the identification of landmarks in the video. In particular, we are employing image instance similarity methods to visually match the content of a frame to a known point of interest found in the video frame. Reference images are extracted from trustworthy collections, which largely diminish the possibility of finding false positives – e.g. Wikipedia pages. These images will then be used as queries for a retrieval system indexing the full video so that we can later provide this information to the LMM, which will consequently include also geographical landmarks within the descriptions used to create the questions.

9.3.4.4. Automatic questions generation After the data definition, we define procedures for automatically generating questions from the video that can be later evaluated as true or false by human annotators. This step is required to have better control of the content of the questions and also to better exploit the valuable and limited resource of human annotation. In fact, we assume that the validation of automatically generated data is usually simpler and more controllable than directly requesting annotators to caption a frame or write questions/answers from scratch, as this process would input a lot of unwanted biases and variability, other than harming annotators with a long and complex task.

To extract high-level semantic information from the video, we start by following a frame-centric approach. Specifically, we employ state-of-the-art computer vision models that can produce natural language descriptions of the frame. More specifically, we rely on large multimodal models (i.e., LLaVA [696]) to provide each image with a textual description of its most peculiar information. We designed a prompt for the model to force them to describe the scene from different perspectives and create a list of possible captions, grounded as much as possible to the depicted scene. In order to avoid strong hallucinations and therefore ensure a good quality of the produced caption, we enforce the model through appropriate prompting not to produce either low-level descriptions (e.g., concerning color palettes) or too high-level captions (e.g., concerning the user sentiment). The top row of Table 82 shows the prompt used to obtain the single frames descriptions.

To obtain a question concerning the whole video shot, we proceed by aggregating the captions obtained





City	Video Length (seconds)	N. Frames	N. Subvideos	N. Questions
Amsterdam	4912	706	311	1555
Bangkok	10499	809	202	1005
Chiang Mai	4075	583	145	725
Istanbul	4080	583	156	760
Singapore	5800	607	151	755
Stockholm	3989	570	142	710
Venice	6599	721	180	900
Zurich	3899	558	139	695

Table 81. Relevant amounts of information about the raw data extrapolated from the video of each city.



Figure 83. Example of frames extracted from a video over which the following questions is asked: "Is it true or false that the statue of the seated figure, possibly a Buddha, appears in the video before the blue plastic chair with a simple design?"

from different frames by employing the sentence construction and summarization capabilities of a text-only LLM (i.e., LLaMa). Specifically, knowing the temporal positioning of two frames within the video, we can ask an LLM to join the respective captions using temporal indications like "after" or "before" to obtain questions like "Is it true or false that the statue of the seated figure, possibly a Buddha, appears in the video before the blue plastic chair with a simple design?" (As shown in the frame sequence in Figure 83).

To avoid feeding the human annotators with too long videos, we only feed LLaMa with captions coming from frames within a 24-second wide temporal window to generate the final questions. We also try to reduce the text model hallucinations through appropriate prompting and reject some generated questions using a rule-based approach, which removes some unclear phrase constructions. The bottom row of Table 82 shows the prompt used to obtain the questions over the whole video.

In total, we consider 8 cities: *Amsterdam*, *Bangkok*, *Chiang Mai*, *Istanbul*, *Singapore*, *Stockholm*, *Venice* and *Zurich*. Table 81 shows how many frames per video we have and how many short videos we extract from each longer city video. From this table, we notice that we end up with a set of approximately 7,000 questions that require manual annotation. Moreover, by resampling both the frame captions and the aggregation over the video, we can further expand our raw dataset pool to provide it to human annotators.

9.3.4.5. Human Judgement In order to recruit human annotators, we rely on the *prolific* platform⁴⁶, shown in Figure 84a. Prolific is a platform designed to facilitate the recruitment of participants for various types of research, including surveys, behavioral studies, and experiments, which is often used to recruit human annotators for AI projects. It is characterized by the possibility of having diverse demographics – given that it boasts a large and diverse participant pool from around the world – and control over annotation quality through attention checks and historical performance data.

In our scenario, annotators are presented with videos and questions built as described in Section

⁴⁶<https://www.prolific.com>





LLava Prompt	Write a list of extensive and detailed descriptions of at most three elements in this photo choosing among people, objects, vehicles, signs, writings, bars, restaurants or others. Describe only one of each kind, focus on remarkable things and avoid those that are colored black or white. Describe them one by one, for each of them give as many details as possible in a few sentences, such as colors, shapes, activities and more, so it can't be mistaken for another one. If there are writings on walls or signs report them as well. Add geometric information if possible, such as sign shape and similar. Avoid talking about the sky or the weather in general, don't mention signs that are upside down and don't mention the color of the sky. Do not talk about reflections or left and right and never mention foreground or background and never use the word remarkable.
LLama Prompt	Given the following descriptions about different events in a video ask 5 true/false questions about the relative occurrence in time of two of these events. For example, if a specific sign appears before a vehicle, or a specific person enters the video before another and more. Make the questions precise and non ambiguous and ask questions about the whole video without referring to the single events. The questions should only be about events that happen in the video, not about them happening in general, for example "Is the woman with black top and blue jeans blonde before a starbucks sign appears on the wall?" should be asked in the form "Is it true or false that the blonde woman with black top and blue jeans appears in the video before the starbucks sign?" Keep all the details present in the descriptions you are provided, don't make up new ones but don't omit any of the information you were given. The following are the Events:infos. Please never mention the words frames, photos, scenes, event, events, Event and Events and only write the questions.

Table 82. Prompts used to generate the frames descriptions with llava and to generate the questions with llama.

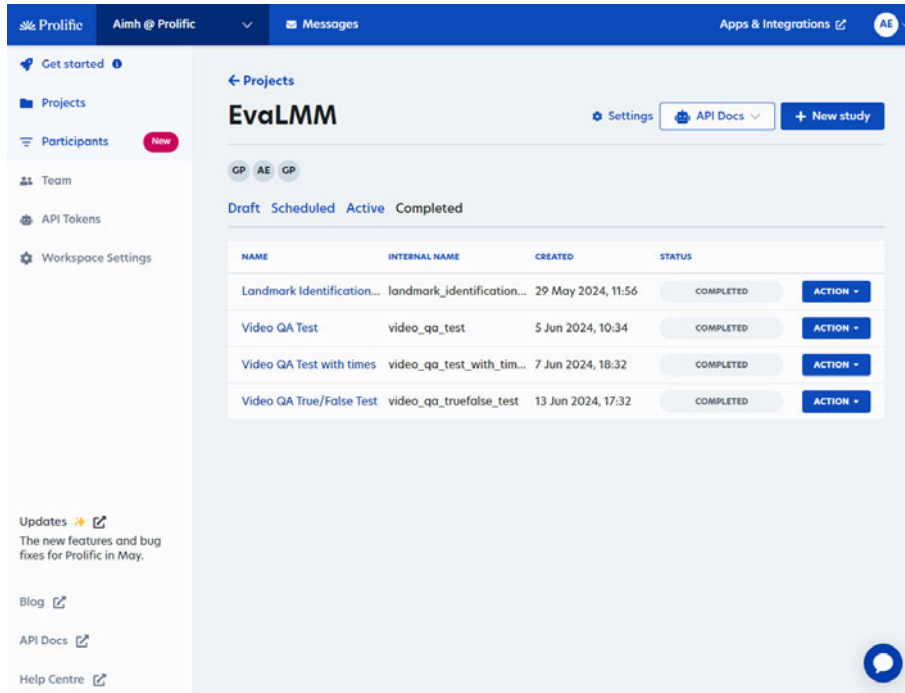
9.3.4.4 and are asked to answer each question by choosing one out of three possibilities:

- **True:** if the answer to the question is true;
- **False:** if the answer to the question is false;
- **Don't know:** if the question is ambiguous in any way, e.g. one of the objects mentioned is not there, or the question can't be answered certainly or any other possible unclear case.

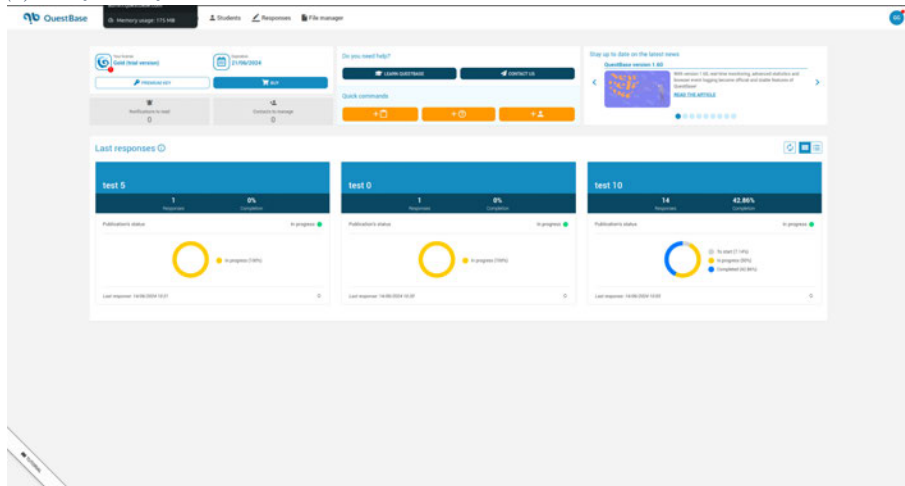
The annotators are presented with the videos and the questions through the *questbase* platform, shown in Figure 84b. The Questbase platform enables the creation of simple multimedia forms built of text, images, or videos, allowing the annotators to watch the video and answer the questions in a single interface. Each video is shown as in Figure 85. We first show the annotator the list of questions, then the video, and consequently, the annotator is presented again with the question, having to answer each question individually. Reading the questions before looking at the video gives the annotator a hint on what to look for in the video, easing the annotation process.

Each single questionnaire is composed of 50 questions about 10 videos, 5 questions per video. We have each questionnaire taken by 5 different annotators. We estimate the time needed to complete one of the surveys based on the time needed by reliable internal annotators sampled from our laboratory and conclude that 25 minutes is the average time needed. We pay 3.13£ for each questionnaire, which amounts to 7.51£/hour (set as a *fair* compensation on the prolific platform). Of the 7,000 available questions, we have had 800 questions annotated by human raters. The remaining ones will be annotated using different language models to generate the questions, to increase the diversity of the benchmark.





(a) Prolific interface.



(b) Questbase interface.

Figure 84. Platform screenshots.

9.3.4.6. Agreement assessment and Filtering Figure 86 measures the agreement between annotators in terms of Fleiss' Kappa [546, 717], which measures agreement taking into account chance agreement, i.e., agreement that occurs even in the case of random choices. Fleiss' Kappa values range from -1 to 1. In literature, commonly accepted values for Fleiss' Kappa in the case of two labels and two raters are: 0.01-0.20 as slight agreement, 0.21-0.40 as fair agreement, 0.41-0.60 as moderate agreement, 0.61-0.80 as substantial agreement and 0.81-1.00 as almost perfect agreement. A negative value indicates that an agreement is present, yet it is lower than the one observable by chance. A value of -1 indicates no agreement at all.





Here there are 5 questions about the video below, after reading them, watch the video and then follow the instructions.

1. Is it true or false that the dark-colored SUV with an antenna on the roof appears in the video before the red-roofed traditional-style building?
2. Is it true or false that the large, green trash can appears in the video after the dark-colored SUV parked on the street?
3. Is it true or false that the dark-colored SUV parked in front of the building with a red roof appears in the video before the green tree in the background?
4. Is it true or false that the outdoor dining area with a red roof and white walls appears in the video after the covered walkway with a red roof?
5. Is it true or false that the dark-colored SUV parked on the side of the street appears in the video before the large, green tree in the background?

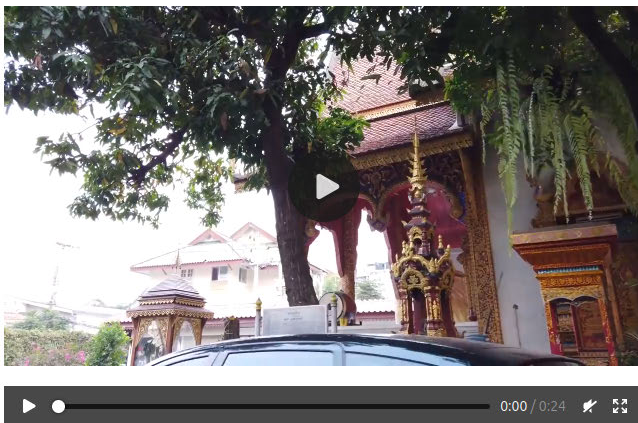


Figure 85. Example of the interface shown to the annotators.

Since we are measuring agreement in a case involving three labels (given that we have the "Don't know" option for those cases when the question is not appropriate for the video) and five raters, the values in Figure 86 can be still considered in the agreement range. Having more labels makes the expected value of agreement lower and the low values we attain are motivated by this, as well as by the naturally challenging dataset we are building, as also shown by the successive experiments. Istanbul is the city with the lowest agreement, slightly below chance agreement, and is in contrast with the others. After inspecting the videos, we believe this is the case because the part of the video that is exposed to raters showcases extremely crowded streets. This makes all questions about people very difficult to answer and it makes several details hard to see, since the viewpoint of the videos is at a person height.

Every video we study has several instances of the same objects, e.g. more people, more vehicles, etc., that naturally occur multiple times in an urban environment. Human annotators are faced with the difficult challenge of identifying which instances are referred by the questions they have to answer and often the ambiguity of this task is a source of a lower level of agreement.

Thus, we also perform additional filtering of the outcome of this human evaluation by manually revising those samples where at least 4 raters agree, and they agree on either "True" or "False". Finally,



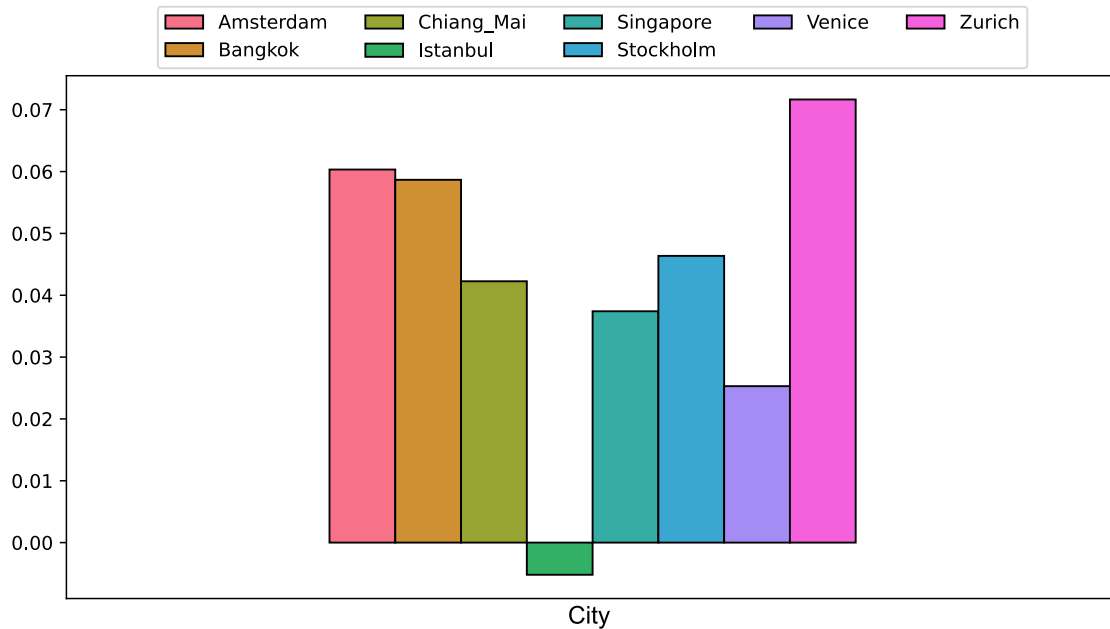


Figure 86. Fleiss kappa agreement score for all the questions used to evaluate raters.

we keep a total of 266 questions about 165 unique videos.

As of now, the number of collected examples is quite limited. This comes from the fact that we decided to prioritize quality over quantity, as the collected data serves as a preliminary benchmark and the overall proposed annotation framework serves as a working pipeline for producing high-quality long video annotations. Thanks to the developed pipeline, all these procedures can be scaled up to tens of thousands of examples, with the final objective of collecting a gold set of large-scale, high-quality human-annotated video data samples requiring long-term temporal reasoning to be processed.

9.3.5. Experimental results

In this section, we carefully set up some baselines to show to which extent current state-of-the-art methods can provide correct answers to the novel crafted benchmark. Therefore, we define inference procedures to adapt current methods to this novel benchmark. On properly introduced evaluation metrics, we perform a final assessment that helps us understand the ability of current models to understand events happening in potentially long video streams.

9.3.5.1. Methods The baselines we probe are either non-generative methods like CLIP [718, 719, 720], as well as state-of-the-art large multimodal models able to process videos, like VideoLLaVa [707].

The clip models are tested through a similarity approach; for each video, we encode 8 frames and compute their vector representation using the image tower of a CLIP-like model. Since all questions are in the form of a before/after statement concerning two events, event A and event B, happening in the video, we compute the similarity with 2 versions of the text, one saying that Event A happens *before* Event B and the reverse statement saying that it happened *after*, we then compute the similarity between each of the 8 frames and each of the 2 statements. We then average over the frames and keep, as the correct answer between the *before* and *after* statements, the one having the higher similarity with the visual feature.

On the other hand, generative video language models are evaluated by encoding the video frames and





	Accuracy	Amst.	Bang.	Chia.	Ista.	Sing.	Stoc.	Veni.	Zuri.
N samples	266	46	10	58	21	28	27	29	47
SO400M-14-SigLIP	0.451	0.477	0.200	0.518	0.476	0.286	0.444	0.500	0.463
H-14-378-quickgelu_dfn5b	0.549	0.489	0.444	0.561	0.550	0.423	0.600	0.577	0.638
H-14-CLIPA-336_laion2b	0.541	0.628	0.600	0.554	0.550	0.520	0.259	0.407	0.681
H-14_laion2b_s32b_b79k	0.486	0.455	0.444	0.518	0.300	0.536	0.370	0.667	0.500
H-14-quickgelu_dfn5b	0.576	0.477	0.500	0.500	0.810	0.667	0.625	0.808	0.467
g-14_laion2b_s34b_b88k	0.529	0.413	0.600	0.537	0.632	0.520	0.462	0.448	0.674
SO400M-14-SigLIP-384	0.522	0.477	0.600	0.491	0.619	0.400	0.462	0.607	0.587
g-14_laion2b_s12b_b42k	0.451	0.500	0.200	0.389	0.333	0.538	0.654	0.483	0.391
avg	0.500	0.489	0.426	0.497	0.510	0.474	0.462	0.561	0.526

Table 83. Accuracy in a 0-shot setting for 7 different CLIP-like models.

	Full Score	Amst.	Bang.	Chia.	Ista.	Sing.	Stoc.	Veni.	Zuri.
N samples	266	46	10	58	21	28	27	29	47
Video-LLaVA-7B	0.526	0.500	0.800	0.586	0.381	0.643	0.407	0.483	0.511

Table 84. Accuracy in a 0-shot setting for the LLaVA model.

then evaluating their answers as True or False using the probability of each token and taking the higher after normalizing over the whole set of answers. This is meant to moderate average biases present in the models that generally make them more inclined to say "Yes" and "True" rather than "No" or "False".

We measure accuracy, precision, and recall as performance scores since the dataset is balanced between positive and negative examples.

9.3.5.2. Results Table 83 shows the accuracy achieved by several CLIP-like models on the dataset. As we can notice, the proposed task is very challenging for these models, as they can't achieve more than 55% accuracy, only slightly above random.

We also break down the results by city. However, given that the number of samples is scarce for some cities, it is not possible to draw strong conclusions on the distribution of the accuracies varying different cultural scenarios. However, we can notice how even Western cities – where most of the training data comes from – maintain a near-random performance on this task, meaning that it is likely that the problem resides in the temporal understanding abilities of these networks rather than in their cultural biases.

Table 84 shows the accuracy achieved in a 0-shot setting using generative approaches. Again, even for these models trained on billions of data samples and fine-tuned using instruction tuning, our benchmark is very challenging, and models are only able to perform slightly above random choice. This is a good indication that our benchmark can be useful over time and that current video language models are not yet able to achieve long-range temporal reasoning.

9.3.6. Assets released to the community

We release the benchmark developed in this study, the short videos with questions and correct answers that have undergone human evaluation as well as the code we used to generate the larger datasets exposed to human evaluation.

First, the actual benchmark that we have developed and is available here <https://zenodo.org/records/13379851> this is the benchmark we developed and it paves the way towards the assessment of Large Vision Language Models temporal understanding in complex scenarios.

Second, the code used to create the videos with the automatically generated captions, available here <https://github.com/gpuce/EvalLMM/> that was used to create the initial pool of videos that have



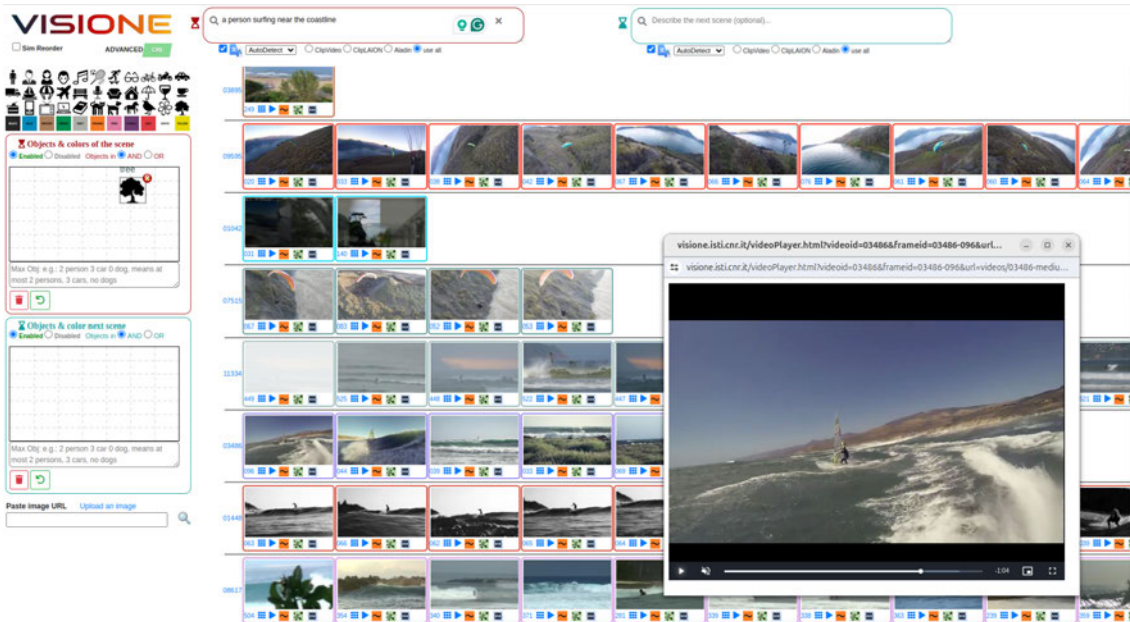


Figure 87. VISIONE software interface, developed by ISTI-CNR in collaboration with RAI as a demonstrator of UC3. This video search and browsing tool can greatly benefit from LMMs capable of understanding long-range temporal dependencies and answering complex factual questions concerning events happening in hours-long videos.

undergone human evaluation. This is useful for the future development of new benchmarks built from an automatic approach similar to ours.

9.3.7. Potential impact on AI research/media industry/society

This activity stands to make a significant impact on the field of AI research, particularly in the domains of large multimodal models (LMMs) and long-duration and long-tail video understanding. In particular, it implements a strategic task within AI4Media (especially tasks 5.1 and 5.4), specifically through the VISIONE large-scale video search tool (Figure 87), developed by ISTI in collaboration with RAI as a demonstrator of UC3 (AI in Vision – High Quality Video Production & Content Automation), where the final aim is to provide users tools for searching and browsing large video collections. By developing and introducing a novel benchmark specifically designed for long-range video understanding and factual reasoning, we provide the research community with a critical tool for evaluating and improving the capabilities of existing text-video models. This benchmark highlights the current limitations and guides future advancements, potentially leading to the development of more sophisticated, temporally-aware models that can reason on long videos and answer complex questions that also require access to external knowledge. Furthermore, the insights gained from the whole activity will contribute to a deeper understanding of how LMMs process and interpret complex, untrimmed video data, thus pushing the boundaries of what these models can achieve.

The result of this activity can impact how large-scale audiovisual archives are managed and accessed in the media industry. With the proposed advancements in video understanding, media organizations will be able to index, browse, and retrieve information from vast amounts of video data more efficiently and accurately. This will enhance the value and usability of historical archives, making them more accessible to researchers, historians, and the general public. Additionally, improved video analysis tools can streamline content creation and curation processes, enabling more dynamic and contextually rich



media experiences. This can lead to innovative applications in entertainment, journalism, and digital storytelling, where a nuanced understanding of video content is crucial.

As a consequence, in general, the outcomes of this activity may show promising effects on society, particularly in the context of cultural heritage preservation and personal media management. By improving the accessibility of large-scale audiovisual archives, we facilitate greater public engagement with historical and cultural resources, promoting education and cultural appreciation. This democratization of access can have a lasting impact on societal knowledge and cultural preservation. Additionally, in the realm of personal media management, enhanced video understanding capabilities will allow individuals to organize and retrieve personal video content more effectively, making it easier to preserve and share personal histories and experiences. This can also extend to applications in security, where better video analysis can improve surveillance systems and incident investigation.

Overall, this activity not only addresses current limitations in AI but also paves the way for future innovations that can significantly enhance the utility and accessibility of video content across various domains, ultimately benefiting AI research, the media industry, and society as a whole.

9.3.8. Conclusions/future work

In this activity, we have taken significant strides toward addressing the challenging task of understanding long-range temporal dependencies in untrimmed multicultural videos using large multimodal models (LMMs). By developing a novel benchmark that incorporates true/false questions concerning multiple time-spanning events within a video, we have provided a robust framework for evaluating the current capabilities and limitations of state-of-the-art LMMs on the processing of challenging raw egocentric video data. Our benchmark, coupled with automated sentence generation and reliable human labeling, offers a comprehensive evaluation tool that can reveal the nuanced deficiencies in existing models' abilities to handle complex video understanding tasks.

The preliminary testing of state-of-the-art models – among which LMMs such as LLaVa, or CLIP-based architectures like SigLIP [720] – has uncovered critical insights into their performance. These findings underscore the need for more time-aware and contextually sophisticated models to address the intricate requirements of video understanding, particularly in scenarios involving extensive temporal dependencies and multicultural contexts. Our work highlights the necessity of further research and development in this domain, preparing the stage for future advancements that can overcome these challenges.

We are already working on an extension of the presented benchmark, which includes geographical factual data to allow for more complicated temporal reference points in the question construction. For example, it could be interesting to also handle questions like *"Is the person eating a sandwich before the visit to the Rialto Bridge?"*. This kind of apparently easy question is actually very complex for non-retrieval-augmented vision-language models, given that they likely do not have access to the visual features of the Rialto Bridge to ground it within the video.

Using the proposed framework, the future steps will possibly include the collection of a full-sized dataset composed of a heterogeneous set of questions (not necessarily accepting only binary answers) that could be employed to fine-tune a state-of-the-art LMM model to better attend to long temporal dependencies and multicultural data. The obtained models could then be implemented within large-scale video browsing software like VISIONE to understand their usability in different applicative real scenarios, like the ones requiring handling large audiovisual archives or managing private user video collections.

9.4. Use of LLMs for co-creative human-computer interfaces for game design

Contributing partner: UM





9.4.1. Challenge

Large Language Models (LLMs), and in general foundation models (FMs) as media generators, have permeated our everyday lives [721] and discourse [722, 723, 724] due to their accessibility and low barrier of entry. Both amateurs and professionals use LLMs and FMs to create new forms of art [725]. However, controlling such models is rarely user-friendly, achieved by changing values via UI buttons and sliders with unclear effects [726, 727, 728], or by editing code via APIs [729, 660]. More importantly, these models are often “one-shot” and lack an output refining process, requiring trial-and-error of different prompts to direct the generation towards a user’s goal. While this process may be fun for inspiration or follow-up manual editing [730], it proves challenging when more structured output is required, such as game content [731] which needs to be functional (playable). To this point, LLMs struggle to produce structured data, as hallucinations [732] and incomplete outputs lead to results that require repairing or that are simply unusable. Video games are a pertinent field of applications for LLMs. Unlike traditional media, games are multi-faceted creative artifacts which hinge not only on text (e.g. narrative, dialogue), visual or audio, but also architectural layouts (levels) and functional gameplay loops [731, 733]. This makes LLMs and other FMs of particular interest to the game industry [734], and some promising research has already explored (unstructured) game content as a target domain [735, 736]. However, current LLM applications to games lack any human-in-the-loop feedback and treat the LLM model as a black-box without any sort of output validation and explainability [737].

LLMAKER targets the issue of controllability in LLMs and FMs with a specific context and goal: *the design of a game through a primarily chat-based interface*. To achieve this, LLMAKER builds an ecosystem where multiple foundation models interact with each other and with a human user. While existing platforms may generate images, videos, or audio tracks individually [734], LLMAKER envisages a functional game content generation pipeline that intertwines FMs and LLMs controlled by user requests. This pipeline raises important questions regarding (a) controllability and explainability of LLMs and FMs, (b) LLMs as generators of structured (game) data via function calls, and (c) cohesion between different (generated) modalities of game content such as visuals, audio, text descriptions, and game rules [731]. LLMAKER addresses these via a chat-based interaction loop for designing missions in a side-scrolling dungeon crawler video game, with enemy encounters and loot. By harnessing the pattern completion capabilities of LLMs, LLMAKER transforms the fuzzy and ambiguous user requests into precise, correctly formatted, and context-aware function calls for integration into an existing game design system. LLM function calling provides the basis for a more verifiable and explainable generation process, and functional errors [738] will be added in the feedback to the LLM as a way of refining interactions and improving explainability for failed user requests.

9.4.2. Related Work

This section provides an overview on large language models with different prompting techniques and evaluation methods, stable diffusion models, and AI-assisted design tools for video games.

9.4.2.1. Large Language Models Broadly speaking, LLMs are models trained on a corpora of text with the goal of generating words that most likely would follow a starting prompt. In recent years, however, LLMs are closely associated with the transformers architecture [739], particularly popularized by OpenAI with the introduction of GPT-2 [740], and more recently GPT-3 [741] and GPT-4 [742]. The power of these generative pre-trained transformers (GPT) lies in both the backbone architecture (the attention transformer) as well as the large training corpus. Being trained on a vast amount of text, spanning multiple diverse fields, GPT models acquire an illusion of reasoning. This renders them, at times, indistinguishable from a human typing from the other side of the screen.

Text generation via an LLM starts from an initial text prompt (the “system” prompt), which usually defines some rules the model should follow, an optional history of the conversation, and the latest user





message. This is the *zero shot* prompting: no additional domain knowledge is provided to the LLM, therefore all responses are based on either information provided by the user during the conversation, or present in the training data. The introduction of additional domain knowledge, oftentimes presented via examples to the language model, allows for *few-shot* reasoning [743, 744]. Learning by examples is a simple way to add domain knowledge, however it can lead to wrong assumptions made by the model. To alleviate this issue, prompting with *chain-of-thought* examples [745, 746] simulates the reasoning behind the decision making process, increasing the accuracy in responses given by the LLM on a variety of tasks. However, even this prompting technique can sometimes fail [747, 748]. The problem of hallucinating responses, paired with the extreme confidence LLMs seem to possess even when the information generated is wrong, pushed for the introduction of another prompting technique: *function calling* [749, 750]. In this case, the LLM relies on a separate system to obtain data for their responses, drastically reducing the possibility of hallucinations [751, 745]. It is important to note that LLMs are not usually trained on data that allows for function calling, therefore they need to be specifically fine-tuned for it [752].

Evaluation of different prompting techniques is carried out on now-standardized benchmarks, such as GLUE [753], the Stanford Question Answering Dataset (SQuAD) [754], and SNLI [755]. These benchmarks evaluate the ability of a language model to complete or classify human-written sentences. Measuring the accuracy of the responses can be carried out via exact matches [756]. Alternatively, as LLMs generate responses that can vary slightly from the target text, a semantic similarity accuracy measure [756] can be used. In this work however, we can not rely on either measure, and instead we devised our own accuracy measure based on domain features (see Section 9.4.5).

9.4.2.2. Stable Diffusion Diffusion models [757, 758] are a class of generative models that create images conditioned by a textual prompt, starting from random noise. This is achieved by training these models to learn a parametrized function that can iteratively reverse the additive noise from the initially noisy image. Training these models starts by adding disruptive Gaussian noise to “clean” images progressively over multiple timesteps. Then, the model is trained to apply the correct denoising diffusion step to obtain back the original image [758]. The neural network architecture most commonly used in such settings is the U-Net [242], suitable for image-to-image translation tasks. Such a generative approach has been successfully applied to image synthesis [759], achieving better image fidelity and mode coverage than GANs- and VAEs-powered methods.

Stable Diffusion (SD) models [760] are a popular class of latent diffusion models that can generate images from different input modalities, such as text only or a combination of text and images. This allows for different image editing and generation paradigms that better suit specific user needs; for example, ControlNet [761] allows for the conditional generation of images starting from a text description and a “control” image that the model uses as reference or starting point for the subsequent generation.

As these models are trained on a large corpus of images and text pairs, they rarely correctly capture certain niche styles or subjects. However, instead of retraining from scratch these models with hundreds of thousands of data points, one can use low-rank matrix adaptation (LORA) [762] to instill new knowledge into the models, requiring only a few (ten to twenty) images, making it extremely easy to personalize these models.

9.4.2.3. AI-assisted Design for Video Games Automated generation of content for video games is a matter of computational creativity [763]. The entire field of procedural content generation (PCG) is focused on producing functional content for video games that is also enjoyable to the players. The content generated can span from weapons, as in Galactic Arms Race [764], to game mechanics [765], to the game as a whole [766, 767, 768].

Tools that generate content for video games can require minimal user input, often just a couple of parameters [768], such as in the *ANGELINA* system [769]. More interestingly, content generation can be





guided by human designers. The mixed-initiative content creation [770] framework lets a human designer interact with an automated PCG system to quickly iterate over content design while respecting designer agency and authorship. Content creation systems are extremely versatile for level design [771, 772, 773], in-game content [774], and game mechanics [775].

Applications of large language models for design tools have already shown promising results [776]. In this work, however, the LLM is implemented to make the back-end system as *transparent* as possible to the user, meaning that the designers need not concern themselves with every minutia of the domain while designing, as the LLM takes care of interfacing with it.

9.4.3. Game Domain: Dungeon Despair

To test LLMAKER functionalities (see Section 9.4.5) for creating semantically consistent and visually coherent game content, it was necessary to design and develop an appropriate testbed game domain for our use case. Towards this end, we designed and developed *Dungeon Despair*, a game domain that follows as closely as possible the design concepts of the *Darkest Dungeon* [777] video game. *Dungeon Despair* is a reversed dungeon-crawler video game, where the player acts as the dungeon keeper, setting up traps and directly controlling groups of monsters to fend off progressively stronger parties of human heroes. Upgrading, reinforcing, and expanding the dungeon itself is achieved via the in-game currency: *stress*. Stress is built up by the heroes as they struggle while exploring the dungeon, and is lowered by positive events, such as defeating monsters or looting treasure chests. As stress is a resource that depends on the difficulty of the dungeon itself, the player must balance their gameplay to maximize how much they can accumulate before eliminating the heroes party. *Dungeon Despair* draws inspiration from *Darkest Dungeon*, where instead the player controls the heroes and must minimize the stress to avoid dangerous and negative effects to appear during their missions. We draw inspiration from the graphical style of *Darkest Dungeon*, which we try to mimic in our assets generation. We show a demo of our *Dungeon Despair* in Figure 88. The game is built using the PyGame [778] library, and all assets were generated using our framework, LLMAKER, which we will explain further in Section 9.4.5.1.

In the current state of *Dungeon Despair*, levels follow a clearly defined parametric, context-aware generative grammar [779, 780, 781]. Context-awareness is necessary for defining constraints, and parameters are used when assigning properties to each element in the level. Generative grammars have been leveraged in the context of procedural content generation for video games via mixed-initiative design [782, 783], learned from examples [784, 785], or via answer-set programming [786].

A level starts with the root node *Level*, which only has one production rule: $Level \rightarrow Room$. A *Room* can then be expanded to a sequence $Room:Corridor:Room$ if the current room has less than 4 corridors connected to it. Each *Room* contains an *Encounter*. Each *Corridor* may contain between 1 and 4 *Encounters*. An *Encounter* can be *Empty* or contain up to 4 *Enemy* entities, a *Trap*, or a *TreasureChest*. It is important to note that a *Trap* can only be placed in an *Encounter* contained in a *Corridor*. An *Enemy* is defined by its *Name*, *Description*, *Species*, and combat-oriented statistics: *HealthPoints*, *Dodge*, *Protection*, *Speed*—each defined within a range of positive values. A *Trap* is defined by its *Name*, *Description*, and *Effect* of activating. A treasure chest is defined by its *Name*, *Description*, and *Loot* that it contains. Unlike *Darkest Dungeon*, here each room and each entity has to be uniquely identified by its name (i.e., there can not be two enemies with same *Name*). In the context of this domain, a valid level is therefore a level that complies to the above described grammar, and satisfies all specified constraints.

9.4.4. Objectives

As presented in Section 9.4.1, the goals of this project test (a) the controllability and explainability of LLMs and FMs, (b) the applicability of LLMs as generators of structured (game) data via function calls,



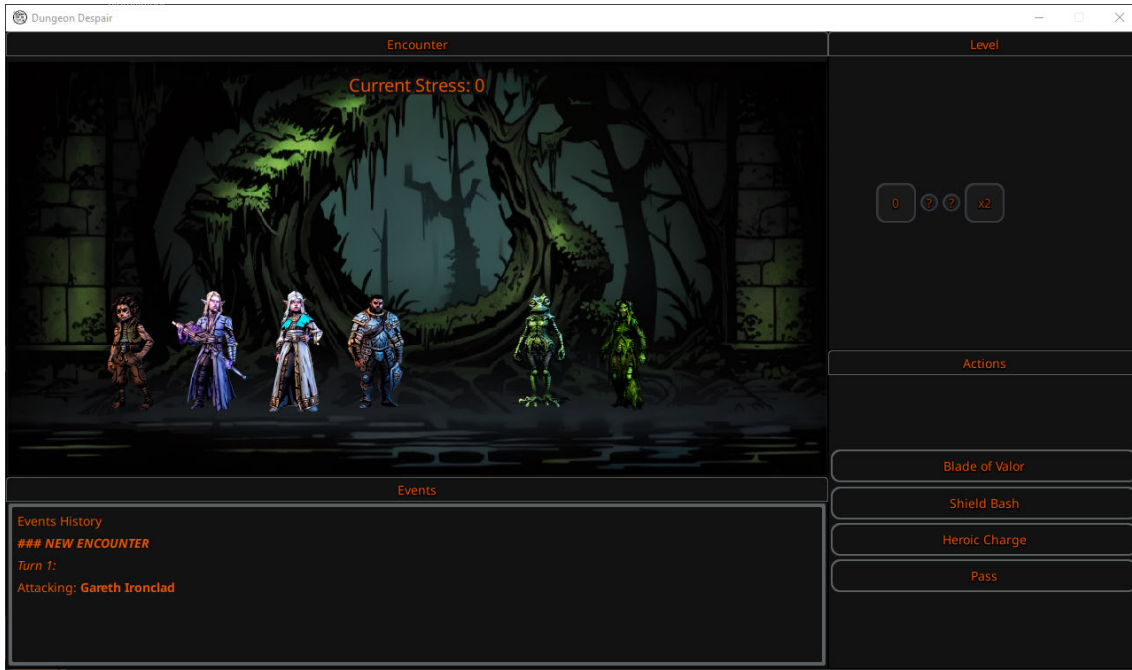


Figure 88. A screenshot of our Dungeon Despair demo video game. In the “Encounter” tab, the user can see the room, the heroes party (left) and the enemies (right), as well as hovering over their sprites to learn more about each of them. In the “Events” tab, the user can see the combat history as well as any other additional messages. In the “Level” tab, a preview of the map is shown, with information about encounters and other dangers the heroes party may face. Finally, in the “Actions” tab, a series of possible attacks are displayed for the user to choose and progress through the encounter.

and (c) the cohesion between different (generated) modalities of game content such as visuals, audio, text descriptions, and game rules [731]. Over the course of the LLMAKER project, and the formalization of the user experience described in Section 9.4.3, these objectives have crystallized into questions of (A) controllable game content design via function calling LLMs, and (B) visually consistent (but unique) game art via Foundation Models.

Based on the above, we formulate the following research questions which have been tested throughout this project and are validated in experiments reported in Section 9.4.6:

- RQA.1 Does function calling produce more valid artifacts than other prompt engineering methods?
- RQA.2 Does function calling result in faster responses to a designer’s query?
- RQB.1 Does an increasing level of context result in more consistent entities sprites generated in a room?
- RQB.2 Does an increasing level of context result in more visible entities sprites generated in a room?
- RQB.3 Does an increasing level of context result in more diverse enemies of the same type across different rooms?

9.4.5. Methodology

In this section, we report implementation details of the main application, the LLMAKER, and the experimental protocols for both the LLM prompt engineering study and the effects of context on images generation study.





9.4.5.1. General LLMAKER architecture and interface We develop LLMAKER as a Windows application. LLMAKER allows the user to design a level for a hypothetical dungeon crawler video game, in the vein of Darkest Dungeon [777], called “Dungeon Despair”⁴⁷. Generation is driven exclusively via natural language instructions. A LLM, in this case GPT 3.5 Turbo, interprets the request and, via function calling [752], generates as response the function name and parameters that will be executed on our back-end system. Parameters for the function are filled out either via extrapolation from the user request, or are generated by the LLM directly. For example, the user can specify the “name” of an enemy to create, and the LLM will use it in the function call, while generating the enemy’s “description” accordingly. As we leverage a back-end system, we can enforce constraints that the LLM is forced to adhere to. The back-end functions affects the level by:

1. adding, removing, or modifying a room;
2. adding or removing a corridor;
3. adding, removing or altering an enemy, a trap, or a treasure chest.

Once the function is executed, the LLM provides a short summary of the changes to the user. In case a function fails to execute, a functional error [787] is returned to the LLM, which can decide whether to try calling the function again with different parameters, or simply inform the user of the problem.

While the level is described in its entirety in a structured text format, here JSON, we extend the capabilities of LLMAKER to support the generation of graphical assets (backdrops for the room or corridor, and sprites for enemies, traps, and treasure chests). We use Stable Diffusion models [788]. We refer to the explanation of the “fine-tuned SD” in section 9.4.5.3 for further details on the configuration of SD models used in the current LLMAKER application.

We can identify three main components the user can interact with in the main LLMAKER interface, as shown in Figure 89. On the right side of the interface, we can find the “Chat Area”. This is where the user can ask questions about the current level, or request changes (as mentioned above). To the right of the interface, the current room or corridor is displayed as its image background along with any enemies, traps, and treasure chests. Hovering over any of these entities will make a tool-tip appear, summarizing the properties of the entity. Finally, a mini-map at the bottom of the interface shows rooms and corridors on a tile-based grid. Users can move from one room or corridor to another simply by clicking on the mini-map.

The entire application is built with Python using PyQt for the user interface. LLMAKER is powered by GPT 3.5 Turbo, a proprietary LLM provided by OpenAI. LLMAKER interacts with the model via API calls using Python. The functions in our back-end system allow for creation, removal, or editing of rooms, corridors, enemies, treasures, and traps. The description of these methods, necessary for function calling, is included in each API call following YAML formatting, as recommended in the OpenAI API developer guidelines.

9.4.5.2. Controllable game content design via function calling LLMs As introduced in Section 9.4.4, we are interested in evaluating the efficacy of the function calling approach over alternative prompt engineering methods (RQA.1 and RQA.2). We test our function calling configuration against different prompting techniques on 5 separate test cases (T1-T5) available in this project repository⁴⁸. We report one test case in Table 85. The test cases used in this work have a different number of (artificial) user requests. They all represent realistic use-cases for a designer faced with the task of creating a game level. Each test case has a different number of target rooms, enemies, traps, and treasures for the system to achieve. Other than the diversity in content generated, the test cases also have different levels of specificity

⁴⁷Source code available at <https://github.com/gallorob/dungeon-despair>

⁴⁸We release the source code for this project at https://github.com/gallorob/llmaker_functioncalling



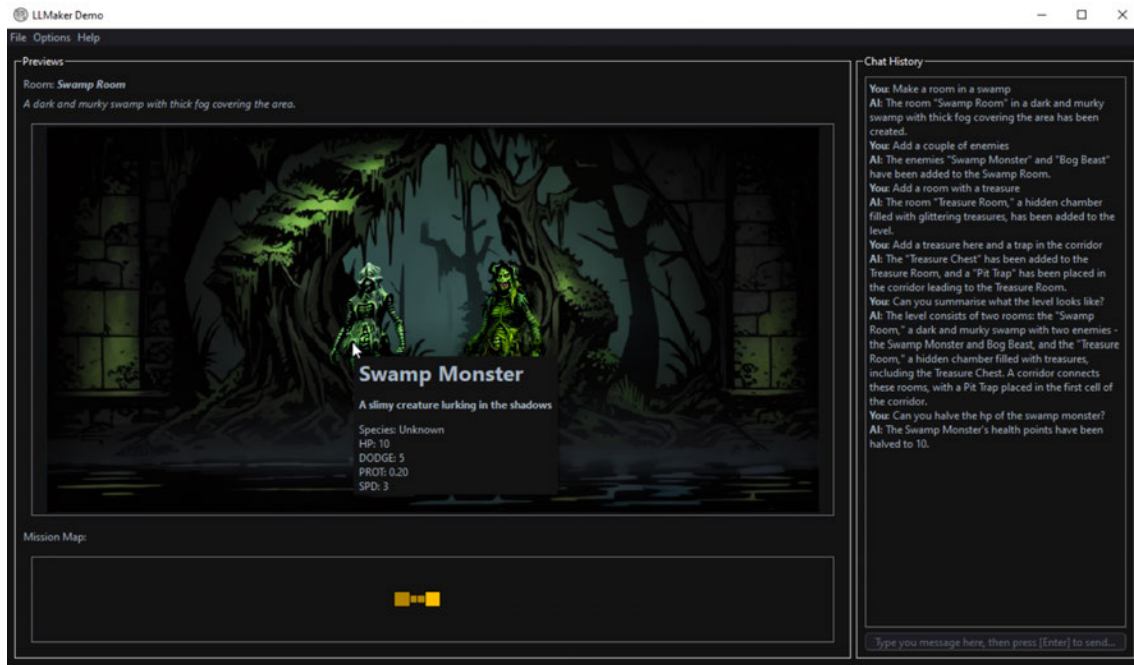


Figure 89. A screenshot of our chat-based level design interface, LLMAKER. On the upper left pane, the preview of the currently selected room. On the lower left pane, the generated level layout, with rooms (larger squares) and corridors (smaller squares). On the right pane, the chat area with the conversation between designer and LLM.

in requests (for example, T2 and T4 are not very specific in what the user wants), as well as different design control flow (for example, T3 and T5 are not ordered and instead jump from one room to another).

We run each test case 10 times, with seed randomization. We test the following prompting techniques, introduced in 9.4.2.1, assigning the respective prompts⁴⁹:

- **Zero-shot**: the LLM has only basic knowledge of the level grammar;
- **Few-shots** [743]: the LLM has a basic knowledge of the level grammar, and the “Additional Information” consists of a few example interactions presented in the system prompt;
- **Chain-of-thought** (few-shot) [746]: the LLM has basic knowledge of the level grammar, and the “Additional Information” consists of a few example interactions with explanations in natural language of how the changes to the level are made presented in the system prompt; and
- **Function Calling** [745]: the LLM has a basic knowledge of the level grammar, and the “Additional Information” consists of a description of functions it can access, and the corresponding arguments per function.

Zero-shot, Few-shots, and Chain-of-Thought are our baseline techniques. Along with the system prompt, each interaction with the LLM also includes a JSON representation of the current level (empty at the beginning of each test case). We do not include the history of the conversation (i.e., past interactions between user and system), as it is not required to fulfil the user requests. The function calling LLM output does not contain a representation of the level, as it updates the current level directly via the functions it calls. The other models instead output directly a JSON-formatted level, which will be parsed to tentatively update the current level.

We identify four possible outcomes whenever an LLM responds to a user request to change the level:

⁴⁹Prompts used in this project are available at https://github.com/gallorob/llmaker_functioncalling/tree/main/prompts



Table 85. User requests for test case T5. Each request is submitted sequentially via LLMaker.

Create 3 rooms, each connected to the next one, all set in a different European city
Add a goblin archer in the first room
Also add two zombies
Now generate a room connected to the first one, set in underground Atlantis
Put a couple of evil mermaids in Atlantis
Place multiple ocean-themed traps in the corridor to Atlantis
Place a single treasure chest in all rooms, each containing a piece of a treasure map
Remove the chest containing the second piece of the treasure map
Add another room connected to Atlantis, set in Hell
Place two fallen angels armed with flaming swords
Change one of the angels to a capybara monster
Set the health of the capybara to 1000
Make the capybara a punker, with pink spiky hair

- **Parser Fail:** the produced output is not a parseable level, which can happen when the JSON is ill-formatted or missing entirely;
- **Domain Fail:** the produced output is a parseable level, but it is not a *valid* level (i.e., at least one of the grammar constraints is not satisfied);
- **Design Fail:** the produced output is a parseable level, but its contents have not been changed as the user requested; and
- **Success:** the produced output is a parseable level and its contents reflect the user requests.

A function calling LLM will never produce a level that is not valid, as the back-end system applies changes that always adhere to domain constraints. However, it can misinterpret or fail to accommodate all user requests. At each request, we check that the updates carried out by the system on the level reflect what the user expected. More precisely, we check that objectively defined requests are correctly implemented in the level (e.g., a new room or a new enemy has been added, the health points of an enemy are set to a specific value, etc.), and define an acceptance interval for subjective requests (e.g., when checking for requests such as “Add a couple of enemies”, we check that there are more than one enemy, and less than the maximum number allowed per encounter). If during the test case a parser, a domain, or a design fail is raised, we track which request triggered it. Once a fail occurs, we terminate the execution of the test case. During the execution of a test case, we also track changes to the level itself, which allows for easier visual debugging. Finally, we note that different prompting methods result in different number of tokens generated by the language model and, therefore, different compute time elapsed to obtain a response. In this work, we track the average time per request the LLM required to produce a response per test case.

9.4.5.3. Visually consistent (but unique) game art via Foundation Models Similar to Section 9.4.5.2, to evaluate the algorithms we need a hand-crafted and curated set of test cases that encompass realistic designer request. We generate, via LLMAKER, 5 thematically unique rooms and, for each room, 5 thematically coherent enemies. We then proceed to generate all possible combinations of enemy per room, over 10 separate and reproducible runs. We share our source code for this set of experiments at https://github.com/gallorob/llmaker_sd_context.





We test two different Stable Diffusion (SD) models:

- **Vanilla SD**: the off-the-shelf Stable Diffusion v1.5 from RunwayML available online⁵⁰; and
- **Fine-tuned SD**, specifically the A-Zovya RPG Artist Tools v4 model⁵¹. The fine-tuned SD is further enhanced by the application of two distinct LORAs: Necro Sketcher⁵² and DarkestDungeon⁵³. While both are tuned to mimic the artistic style of Darkest Dungeon, the former is specific for entities (such as monsters and objects) found in the game, whereas the latter is specific for environmental backgrounds and landscapes. We use Necro Sketcher only when generating entities for Dungeon Despair, and DarkestDungeon only when generating rooms and corridors backdrops. Further, we employ a custom variational auto-encoder (VAE) from StabilityAI⁵⁴, as recommended in the user guide of A-Zovya RPG Artist Tools v4.

Quantifying the effects of different levels of context in the image generation process requires first the definition of multiple metrics. Measuring the consistency of an entity sprite within a room (RQB.1) can be defined as the percentage of shared colors between the two images (the entity sprite and the room background). We define the consistency C as

$$C = \frac{\|\gamma(e) \cap \gamma(r)\|}{\|\gamma(r)\|} \cdot 100, \quad (115)$$

where $\gamma(\cdot)$ is a function that extract the ordered list of colors in the quantized input image down to a 32 colors palette, e is the entity sprite image, and r is the room background image. As it's defined, an entity image that scores 100% consistency will share at most 32 colors with the background image, and a score of 0% implies that the two images do not share any color.

While consistency between rooms and entities is important, we always want to avoid generating entities that get lost in the room. To be more precise, while we want the entities to look as if they “belong” in the room, we also want them to “pop” visually (RQB.2). We measure the effects of different levels of context on this by defining a simple visibility metric V , computed as the difference between the (LUMA) gray-scaled images, i.e.:

$$V = g(r \cup e) - g(r), \quad (116)$$

where g is the grey-scale transformation applied to input images, e is the entity sprite image, and r is the room background image.

To answer RQB.3, we instead rely on the Learned Perceptual Image Patch Similarity (LPIPS) metric [789]. We employ LPIPS to measure the diversity between different entities. We identify three different types of diversity we are interested in:

1. Context ($d_{context}$): here we compare, for the same run, same entity, and same room, the different character sprites generated by different levels of context;
2. Runs (d_{runs}): here we compare, for the same entity, same room, and same context level, the diversity resulting from a different initialization in the foundation model; and
3. Room (d_{rooms}): here we compare, for the same run, same entity, and same context level, the diversity that arises when looking at different rooms.

While the above metrics are used to answer the aforementioned research questions, we also care about the quality of the generated character images. We measure the effects of different levels of context on this metric via BRISQUE [790, 791]. We show example values of all the presented metrics in Table 86.

⁵⁰<https://huggingface.co/runwayml/stable-diffusion-v1-5>

⁵¹<https://civitai.com/models/81247?modelVersionId=250344>

⁵²<https://civitai.com/models/70147/darkest-dungeon-style-or-necro-sketcher-or-lora>

⁵³<https://civitai.com/models/65324/darkestdungeon>

⁵⁴vae-ft-mse-840000-ema-pruned, available at <https://huggingface.co/stabilityai/sd-vae-ft-mse-original/tree/main>



Metric	None	Semantics
Quality	36.71	15.55
Complexity	35%	58%
Colourfulness	3%	7%
Visibility	4.1×10^{-5}	2.4×10^{-5}
Consistency	50%	50%
Diversity	0.37	

Table 86. Example of metrics values for “Mad Tinkerer” in “Steam Engine Room” at different levels of context.

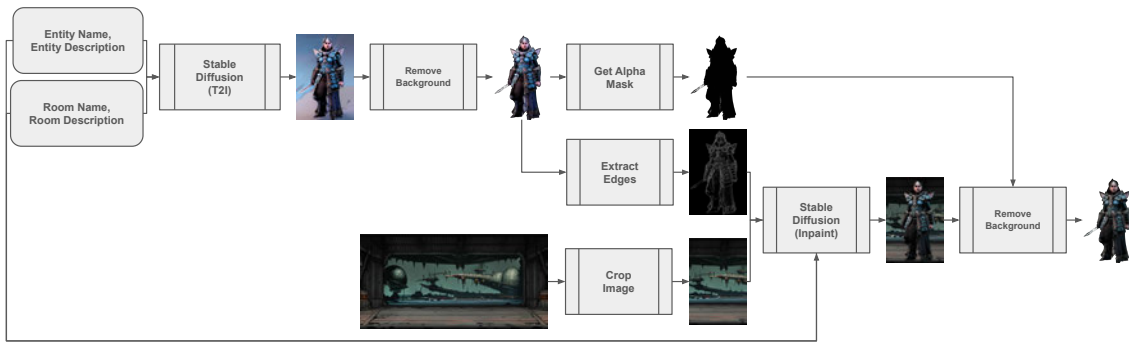


Figure 90. Pipeline diagram for the generation of an entity leveraging both semantics (room name and description) and image context. We show interim outputs and final output for “Faerie Queen’s Guard”: “Elite warriors sworn to protect the faerie queen, armed with enchanted blades and shields, and capable of flight” in the “Airship Docking Bay”: “A vast hangar housing airships of various sizes, bustling with activity as crews prepare for departure”. The full prompt, with Compel weighting, for the in-painting step is “darkest dungeon, (full body)+++ faerie queen’s guard: (elite warriors sworn to protect the faerie queen, armed with enchanted blades and shields, and capable of flight)++, set in airship docking bay: a vast hangar housing airships of various sizes, bustling with activity as crews prepare for departure, masterpiece++, highly detailed+”.

We measure the colorfulness of images being generated following [792] $M^{(3)}$ formula of colorfulness, defined as:

$$\begin{aligned}
 M^{(3)} &= \sigma_{rgyb} + 0.3 \cdot \mu_{rgyb} \\
 \sigma_{rgyb} &= \sqrt{\sigma_{R-G}^2 + \sigma_{0.5 \cdot (R+G) - B}^2} \\
 \mu_{rgyb} &= \sqrt{\mu_{R-G}^2 + \mu_{0.5 \cdot (R+G) - B}^2}
 \end{aligned}
 \tag{117}$$

We track the complexity of the images being generated. We are interested in this as we expect more details provided to the foundation model result in more complex images. We measure this as the percentage of edges detected via the holistically-nested edge detection (HED) [793] detector in the image.

We analyse the effects on the above metrics by varying levels of context. Context is defined as additional information available to the foundation model whilst generating the entity sprite. All entities are generated using the same base information: the entity name and physical description (e.g. “Mischievous Imp” as entity name and “Small, agile creatures known for their trickery and deception, wielding enchanted daggers and arcane spells” as physical description). We identify the following levels of context:



1. *None*: The entity sprite is generated using only the base information, and no additional context is provided.
2. *Colours*: We extract the principal colours of the room, provided with their names to the foundation model. We only pass up to 8 colours to avoid prompt bloating.
3. *Semantics*: We use the name and the description of the room the entity would be in.
4. *Semantics & Colours*: We combine methods (2) and (3).
5. *Semantics & Image*: Along with the room semantics (3), we pass a cropped section of the image.
6. *Caption*: We use the BLIP model [697] to generate a caption of the room the entity would be in.
7. *Caption & Colours*: We combine methods (2) and (6).
8. *Caption & Image*: Along with the room caption (6), we pass a cropped section of the image.

“Colours” are extracted by first quantising the room image (with a depth of 32), and then ordering the RGB pixel values based on their frequency. We create the list of colours as strings by finding the closest RGB value for which we know the HTML names.

When using “Image”, for either “Semantic & Image” or “Caption & Image”, we first generate the entity image using only the semantics or the caption information. We then apply an edge detection filter, and pass it to a SD model that performs in-painting over a cropped portion of the room image. The transparent entity image is then obtained by masking the output image of the in-painting with the alpha channel of the original (semantics- or caption-context) image. We illustrate this pipeline in Figure 90 for better clarity.

9.4.6. Experimental results

In this section, we present the results obtained from our experiments. The research questions A and B elaborated in Section 9.4.4 are addressed in the below subsections:

9.4.6.1. Controllable game content design via function calling LLMs Table 87 summarises the metrics for all configurations for each test case (T1-T5) from 10 runs with seed randomisation. Overall, other prompting techniques fail in a way that makes the system unable to move forward, often after as few as 1 or 2 responses, in the case of Zero-shot and Chain-of-Thought. None of the baseline methods ever completes a test case without failing. Conversely, function calling never raises a parser or domain fail, while it can still incur design fails. Additionally, it achieves the highest average number of responses (reaching the end of each test case) with some of the lowest per-request elapsed time. We further perform Wilcoxon signed-rank test [794] with Bonferroni correction on each test case for each methods pair to assess which configuration is significantly faster than the others and completes more requests. Significance is established at $p < 0.05$.

In our experiments, we find that parser fails exclusively occur when the language model ignores the prompt request to always generate a JSON in its response, and instead inquires the user for more information. For example, for requests where a new room should be generated (e.g.: T1.3, T2.3, T3.4, and T4.3), the model asks for the new room properties instead of generating a new one. While this behaviour could be acceptable in a more interactive setting, we still mark this as a failure since part of the prompt is being ignored.

Failing to comply to the domain grammar makes up the majority of failures that the baseline models run into. In almost all cases, the LLM fails to create corridors when a new room is created, even though this is described in the level grammar and included in the system prompt. Another, less common,





Table 87. Results for different prompting methods on all test cases averaged from 10 independent runs. Fails measure the number of instances of 10 runs that failed, while Responses and Time (per Request) are averaged from 10 runs and include the 95% Confidence Interval. Responses and Time values with * indicate significantly outperforming all other configurations on this Test Case.

Prompting	Test Case	Fails ↓			Responses ↑	Time (s) ↓
		Pars.	Dom.	Des.		
Zero Shot	T1	0	10	0	3±0.0	6.1±0.1
	T2	0	10	0	3±0.0	10.2±0.2
	T3	0	10	0	2±0.0	7.6±0.3
	T4	10	0	0	1.1±0.2	2.8±1.5
	T5	7	3	0	3±0.0	26.6±0.3
Few Shot	T1	0	10	0	4±0.0	15.6±0.8
	T2	0	10	0	4±0.0	12.7±0.2
	T3	0	10	0	7±0.0	21.4±0.2
	T4	0	10	0	3±0.0	5.9±0.1
	T5	0	10	0	3±0.0	26.9±2.6
Chain of Thought	T1	0	10	0	3±0.0	16.4±0.4
	T2	0	10	0	3±0.7	13.7±3.2
	T3	0	10	0	2±0.0	11.9±0.3
	T4	0	10	0	3±0.0	12.8±0.2
	T5	10	0	0	1.5±0.9	67.3±4.8
Function Calling	T1	0	0	7	7±0.0*	9.6±0.6
	T2	0	0	0	9±0.0*	6.9±0.4*
	T3	0	0	0	10±0.0*	5.7±0.0*
	T4	0	0	1	11±0.0*	4.9±0.1
	T5	0	0	0	13±0.0*	8.5±0.1*

domain failure occurred when the models ignored part of the level grammar, such as defining the loot of a treasure chest as a list of items instead of a single string.

Function calling cannot, by design, run into parser or domain failures. Yet, design failures can still occur, albeit more rarely. In our experiments, we see this in T1, where the model only adds one trap in the corridor instead of at least two as requested by the user. When looking at the conversation logs, we actually observe that the LLM attempts to add two traps to the same encounter, but the second attempt always fails by design and the model does not try again in a different encounter of the corridor. In T4, instead, the model attempts to add a new room with the same name as an existing room, which is not allowed, and the model does not try again by giving it a different name.

When looking at the elapsed time per request, we observe that the baseline models scale with the complexity of the level, which is expected as they need to regenerate the entire JSON description of the level in their responses. For all baseline methods, we see that T5 is the test case that results in the highest elapsed time—with Chain-of-Thought achieving the highest value (67.3 seconds). On average, even though Chain-of-Thought fails after 2 or 3 requests in all test cases, its response time is between 100% (T1) and 600% (T5) slower than function calling, which manages to always complete at least 7 requests per test case. We believe this is because, along with the JSON description of the level, Chain-of-Thought also needs to generate the decision-making thought process undertaken to alter the level. The elapsed time in T5 for Chain-of-Thought is particularly interesting, as it seems almost an outlier compared to the other tests,





especially given that it fails after 1.5 responses on average. However, the first request of T5 is to generate three rooms at once, therefore the response is noticeably long. When inspecting the output for T5.1, we see that Chain-of-Thought provided multiple JSON representations of the level, one per room being added.

Other baseline methods also underperform when compared to function calling. The response time for Zero-Shot is at worst 200% (T5) slower than function calling, and for Few-Shot it is at worst 300% (T3) slower. Additionally, Zero-Shot is unable to complete more than 3 responses before failing, whereas Few-Shot performs slightly better, completing as many as 7 responses in T3.

Function calling is agnostic to the complexity of the level, as it only needs to generate the function calling data (which can happen multiple times per request), and a final short summary text response. Although this means that there are always at least two responses being generated per user request, the average time is almost always consistently lower than the baselines on each test case, with the exception of Zero-Shot on T4, which achieves a lower elapsed time compared to function calling, albeit not significantly. However, the responses from Zero-Shot in T4 lacked the required JSON structure entirely.

9.4.6.2. Visually consistent (but unique) game art via Foundation Models Table 88 reports the results for each metric for sprites generated using the Vanilla SD and fine-tuned SD models. Results are collected from 250 tests, with 5 different entities placed in 5 different rooms each and images generated for each combination 10 times. In these tables we indicate the number of times a context level yields significantly better results for a specific metric compared to the other context levels. We perform a Welch T-test with Bonferroni correction to determine significance of results at $p < 0.05$.

From our results, it is clear that the additional styling choice greatly affects all metrics, to the point where no context level is clearly superior to another for the fine-tuned SD model. However, it seems that “Caption & Colours” is the better context level for the vanilla SD model. A collection of sprites with a single prompt, “Mischievous Imp”⁵⁵ generated in two separate rooms can be found in Table 89. One of the drawbacks of the vanilla SD model, which can be seen in this example, is that sometimes the generated entity sprite is not recognized as foreground and therefore is partly removed along with the background. Using a specific LORA instead, we found that this is much less frequent.

The quality of the images generated using semantic context and their combinations is generally lower, while captions and colors are among the better context levels for this metric. Intriguingly, the complexity of the images generated with the image inpainting is much higher only in the case of the fine-tuned SD model, whereas it’s much lower for the vanilla SD model. Just as surprisingly, we find that the consistency of the generated images for semantics variants is much lower than even the baseline. When looking at the visibility, in the case of fine-tuned SD-generated images, we find that unsurprisingly no additional context makes them “pop” more from their background, whereas in the case of the vanilla SD model the caption with colors is the context level that mostly results in images that stand out. Colorfulness is also different for the two styles: in vanilla stable diffusion, caption with colors generates much more colorful images than the other context levels, whereas in the fine-tuned SD model purely semantics is enough to generate more colorful images.

Results seem to suggest that, in both cases, the use of semantics levels of context is actually detrimental for some of the key metrics of interest, namely the consistency, the visibility and, surprisingly, the diversity across rooms (d_{rooms}).

9.4.7. Assets released to the community

As part of the work, we released several artifacts (software), summarised in Table 90. We are also planning to release a full version of LLMAKER and publish an additional paper on the generation of game sprites via FMs.

⁵⁵A “Mischievous Imp” is defined as a “Small, agile creatures known for their trickery and deception, wielding enchanted





Table 88. Summary table from 250 generation tests per context level; results include both a vanilla SD model and a fine-tuned SD model for the task at hand. The number under each column indicates the times a context level yields significantly higher value in the row’s metric, compared to the other context levels. The best context level per metric appears in bold.

Metric	None (<i>baseline</i>)	Colours	Semantics	Semantics & Colour	Semantics & Image	Caption	Caption & Colours	Caption & Image
Fine-tuned SD model								
Quality	5	6	2	3	0	4	6	1
Visibility	7	1	4	0	0	2	2	3
Consistency	4	6	1	3	0	5	7	2
Complexity	0	1	3	4	6	1	3	6
Colorfulness	0	3	6	1	3	1	3	5
$d_{context}$	3	0	5	4	6	6	0	2
d_{runs}	5	2	4	3	0	7	1	4
d_{rooms}	3	5	2	1	0	7	6	4
Vanilla SD model								
Quality	4	4	1	3	0	1	4	4
Visibility	2	4	4	3	0	0	6	5
Consistency	4	5	1	2	0	5	7	3
Complexity	4	3	2	6	5	5	1	0
Colorfulness	0	5	4	2	4	1	7	2
$d_{context}$	1	5	0	3	5	4	7	2
d_{runs}	1	5	4	3	5	2	7	0
d_{rooms}	2	6	3	1	0	4	7	5

9.4.8. Potential impact on AI research/media industry/society


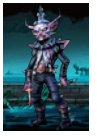
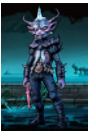

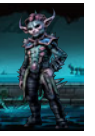

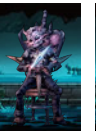
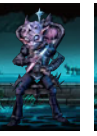
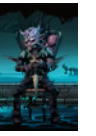






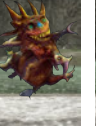











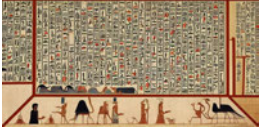





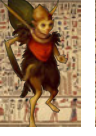


LLMAKER directly contributes to AI4Media Use Case 5 (AI for Games): LLMAKER’s focus on controllability and cohesion is directly applicable for many procedural content generation tasks (see Section 9.4.2) where structured content is important—such as level generation [763]. In addition, LLMAKER provides practical insights to AI4Media’s T4.3 (Novel methods for explainable and interpretable AI), as the LLM constantly provides feedback of its additions to the level design to the user (see Figure 89). LLMAKER directly contributes to AI4Media’s T5.2 (Media content production) as it produces both structured content described textually (as room and enemy descriptions, as well as stat blocks) and visually (as enemy sprites, as per Table 89). Extensions to this work, and its broader applications to AI research and the (designer-focused) applications for the media industry are described in Section 9.4.9.

daggers and arcane spells”





Table 89. Sprites generated for the entity “Mischievous Imp” using the fine-tuned SD model and the vanilla SD model. The two rooms tested are “Submerged Arena” (top two rows) and “Hieroglyphic Hallway” (bottom two rows) and are similarly generated via the respective SD model.

SD model	Room	None	Colours	Semantics	Semantics & Colours	Semantics & Image	Caption	Caption & Colours	Caption & Image
Fine-tuned									
Vanilla									
Fine-tuned									
Vanilla									

9.4.9. Conclusions/future work

LLMAKER is posed as an innovative tool for co-creative video game content design empowered by large language models. In LLMAKER, the interaction between designer and system is entirely based on natural language, with the LLM translating user queries into properly formatted requests to a back-end system via function calling. We also proposed a pipeline using stable diffusion models to generate the graphical assets that represent the content being idealized by the user.

In Section 9.4.6.1, we demonstrated that the function calling approach is superior to other LLM-based methods for generating content in terms of prompt adherence and domain constraints satisfaction. Additionally, LLMAKER consistently processes user requests in a few seconds, serving the user with the updated content almost in real-time. LLMAKER demonstrates how function calling for LLMs can be efficiently implemented in a content design tool.

Experiments in Section 9.4.6.1 focused only on content consistency, i.e. whether the generated content adhered to both domain specifications and reflected user requests correctly. However, other aspects of LLMAKER should be considered going forward. So far, we did not focus on how useful the responses were to a human designer, as we put an emphasis on a JSON representation of the level. This is an important direction for future research, as better responses would improve the usability of the application overall.

Experiments in Section 9.4.6.2 used automated metrics for calculating image quality, consistency, and variety. While a diverse set of metrics (popular in the bibliography) were explored, it is worth noting that these metrics are not tailored to the task of visual design of game assets. Therefore, the quality and diversity of assets produced (e.g. in Table 89) could be additionally validated via human viewers,





Name	Link
Function Calling Benchmark	https://github.com/gallorob/llmaker_functioncalling
Game Sprites Generation Framework	https://drive.google.com/drive/folders/1SNmGMj0pTZGTSvgMu7jj1irvdRn10nWYT?usp=drive_link ⁵⁶
Dungeon Despair (Domain)	https://github.com/gallorob/dungeon-despair-domain
Dungeon Despair (Game)	https://github.com/gallorob/dungeon-despair
LLMAKER (Demo)	https://github.com/gallorob/llmaker/tree/cog24-demo

Table 90. Assets delivered to the community from LLMAKER activities.

e.g. in a survey similar to [795].

Overall, from experiments in Section 9.4.6.2 it is unclear which of these context levels would be best for LLMAKER. The in-painting step included in the “Semantic & Image” and “Caption & Image” context levels requires additional computation, which would affect the real-time applicability of the program, and it does not seem to lead to any clear benefits as identified by our metrics. While semantic information is easily provided by the LLM, it also does not perform well. On the other hand, the caption information requires a forward pass of the BLIP model, which again would impact performances. A small user study would thus also indicate the tradeoffs of quality, diversity, and response time.

Another research direction could address the lack of proactive assistance in the current implementation of LLMAKER: the tool simply implements the changes requested by the user, but never tries to suggest changes of its own. This is a known problem in the field that is still under active research [796]. Suggested changes by a proactive AI co-creator could include adapting enemies’ combat statistics to a specific play-style [797], or altering the layout of the level based on a difficulty scale [798]. Additionally, the current implementation does not guarantee that the level can be completed—indeed, no completion criteria are included in this version of LLMAKER. Adding such constraint checks would ensure the playability and balance of generated levels.

Finally, we note that LLMAKER is the first procedural content design assistant that is in constant dialogue with the designer, opening up future work for mixed-initiative tools where interaction between human and machine is based exclusively on natural language. LLMAKER offers new opportunities for assisted design, but comes with new challenges of cognitive demand. Future work should evaluate the chat-based interaction of LLMAKER with actual designers in user studies. Studies on user interfaces in computer-aided design tools are quite common also in the arts industry [799]. A similar study with LLMAKER should aim to better understand how the system helps designers by identifying its strength and weaknesses, and most importantly evaluate the creative process based entirely on the communication via natural language paradigm.





10. Conclusion

D5.4 is the final deliverable of WP5 and presents progress in the different WP5 tasks since the submission of previous deliverables, i.e. D5.1-D5.3. The deliverable presents the latest research results of WP5 regarding Content-centered AI, specifically on the tasks: T5.1 “Media analysis and summarisation”, T5.2 “Media content production”, T5.3 “Learning with scarce data”, T5.4 “Language analysis in Media”, T5.5 “Computationally demanding Learning”, T5.6 “Music Annotation and Audio Provenance Analysis” and T5.7 “Research on Large Language Models for the media industry”.

Several new methodologies bringing novel solutions and state-of-the-art results are presented, while also, multiple datasets were produced. Approaches that fall under T5.1, include, but are not limited to, novel methods and software for efficient and aesthetically pleasing video summarization, algorithms for video analysis for multimodal gesture recognition, face labeling, shot detection and character objectification detection. The presented works are particularly relevant to the AI4Media use cases, since they can be integrated in media outlet workflows to automatically support content organisation (UC7), event detection in long videos (UC3) and information discovery from multimedia data (UC3, UC4, UC7).

Under T5.2, we presented research on Robot Systems for automatic visual target detection that can lead to aesthetically pleasing and efficient UAV cinematography. Moreover, we presented generative approaches that attempted to enhance digitized music scores with human and instrument-like characteristics without supervision. Advances in both video and music production are particularly useful in the media sector, and clearly align with AI4Media use cases like UC3, UC5 and UC6 where content generation is the main focus.

A plethora of important learning algorithms were developed in T5.3 for learning from scarce data that have led to multiple publications in top conferences and journals of Computer Vision and Machine Learning. Bioinspired DNN learning approaches were researched, as well as, automatic video search software systems that automatically annotate visual data, and unsupervised domain adaptation methods for detection of events in different types of images and videos. Moreover, novel state-of-the-art methods were studied for representation learning for a variety of tasks, such as reducing the need for annotated data, information retrieval from videos, human face awareness and understanding and noisy data training among others.

Regarding T5.4, the work presented in this deliverable was mainly focused on language model adaptation to specialized domains and experimentation with alternative vectorial representations of text data. Specifically, we presented the first large multilingual dataset for sentiment-classification that aligns sentiments expressed toward given entities, across different languages. This research results aligns with use-cases that include the usage of AI against misinformation in the news (UC1) and can be used in the context of journalism or/and news research/analysis in different countries and languages (UC2, UC4). Moreover, we tested a novel, “contrastive” type of vectorial representations of texts, suited to classifiers that decide whether two texts belong to the same class or not. While the test was conducted on authorship analysis tasks, the agnostic nature of these representations allows them to be used in other text classification tasks, such as classification by topic.

Under T5.5, we presented novel research works that handle efficient training methods and mathematical computations in DNNs with matrix factorization layers that achieve semantic-rich feature representations, positional embedding methodologies that enhance Transformer performance in classification tasks while preserving privacy, and Super-Resolution evaluation datasets and baselines. These results are useful in multiple Deep Learning scenarios where Transformers are utilized and matrix multiplications are performed.

Important developments in AI-enabled music analysis were included in T5.6. Works presented include studies for the reliability and realism of DNN confidence in automatic music classification, development of pre-trained DNN fine-tuning methodologies to novel music-relevant domains for music tagging and music information retrieval, and finally a novel audio provenance analysis framework. These outcomes are useful in use cases where music or audio analysis is the focus.





Finally, in this deliverable the outcomes of three mini-projects implemented by RAI, CNR and UM were reported, focusing on the use of LLMs for different media industry applications as part of T5.7. The first work tackled the challenging problem of editorial media segmentation through the creation of a novel, multimodal LLM-based framework, to be able to find relevant parts, e.g. short clips or larger segments, in multimedia data that can have an independently exploitable nature on publication platforms and that can be identified following multiple segmentation criteria. The developed framework is relevant to a wide number of media applications and use cases, namely all those which benefit from chapterisation of longer content into smaller coherent units, like UC1, UC2, UC3 and UC7 in AI4Media. The second work focused on creating a benchmark for LLMs to evaluate their performance in understanding long, untrimmed videos, in a human-like fashion, without being hindered by original domain differences such as cultural biases. This work can impact how large-scale audiovisual archives are managed and accessed in the media industry, being relevant to UC2, UC3, UC4, and UC7. Finally, an innovative tool was developed for co-creative video game content design empowered by LLMs to help the seamless interaction between designer and system, while also a generative pipeline was proposed to generate the graphical assets for the video-game. This research result directly contributes to game development use cases like UC5.

As the AI4Media project reaches its conclusion, WP5 “Content-Centered AI” is also completed. WP5 has been a key work package within AI4Media, with numerous partners contributing through research and the development of AI technologies designed to generate positive social, ethical, and economic impacts for the media sector. Despite the challenges, WP5 has yielded significant outcomes, including the creation of novel software tools, research published in top-tier venues, and publicly available datasets aimed at benchmarking and advancing AI research. These achievements have been closely aligned with carefully designed Use Cases that drive impactful research tailored to the needs of the media industry. Additionally, WP5 has fostered exceptional cooperation across multiple levels: between partners, across work packages, and among various European organizations. Given these accomplishments, we firmly believe that the outstanding work produced in WP5 will have a lasting influence on the research community and media organizations. We are eager to see how the connections and innovations developed throughout WP5, and AI4Media as a whole, will continue to evolve and shape the future.





References

- [1] F. Patrona, I. Mademlis, and I. Pitas, “An overview of hand gesture languages for autonomous UAV handling,” *Aerial Robotic Systems Physically Interacting with the Environment (AIRPHARO)*, 2021.
- [2] A. G. Perera, Y. Wei Law, and J. Chahl, “UAV-GESTURE: A dataset for UAV control and gesture recognition,” in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018.
- [3] L. Galteri, L. Seidenari, M. Bertini, and A. Del Bimbo, “Deep universal generative adversarial compression artifact removal,” *IEEE Transactions on Multimedia*, vol. 21, no. 8, pp. 2131–2145, 2019.
- [4] M. Cornia, M. Stefanini, L. Baraldi, and R. Cucchiara, “Meshed-memory transformer for image captioning,” in *Proc. of CVPR*, pp. 10578–10587, 2020.
- [5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, 2012.
- [6] X. Chen and K. He, “Exploring simple siamese representation learning,” in *CVPR*, 2021.
- [7] A. Fan, S. Bhosale, H. Schwenk, Z. Ma, A. El-Kishky, S. Goyal, M. Baines, O. Celebi, G. Wenzek, V. Chaudhary, N. Goyal, T. Birch, V. Liptchinsky, S. Edunov, M. Auli, and A. Joulin, “Beyond english-centric multilingual machine translation,” *Journal of Machine Learning Research*, vol. 22, no. 107, pp. 1–48, 2021.
- [8] J. Lu, X. S. Zhang, T. Zhao, X. He, and J. Cheng, “April: Finding the achilles’ heel on privacy for vision transformers,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10051–10060, 2022.
- [9] R. Plutchik, “Chapter 1-a general psychoevolutionary theory of emotion,” 1980.
- [10] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms,” 2017.
- [11] Y. Le and X. S. Yang, “Tiny imagenet visual recognition challenge,” 2015.
- [12] Y. Shen and B. Zhou, “Closed-form factorization of latent semantics in gans,” in *CVPR*, 2021.
- [13] J. Zhu, R. Feng, Y. Shen, D. Zhao, Z.-J. Zha, J. Zhou, and Q. Chen, “Low-rank subspaces in gans,” *NeurIPS*, 2021.
- [14] B. Chao, “Anime face dataset: a collection of high-quality anime faces.,” 2019.
- [15] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jegou, “Training data-efficient image transformers & distillation through attention,” in *International Conference on Machine Learning (ICML)*, 2021.
- [16] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” in *ICLR*, 2020.
- [17] M. Gerhardt, L. Cuccovillo, and P. Aichroth, “Audio provenance analysis in heterogeneous media sets,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 4387–4396, June 2024.





- [18] M. Maksimovic, L. Cuccovillo, and P. Aichroth, “Phylogeny analysis for MP3 and AAC coding transformations,” in *ICME*, 2017.
- [19] M. Nucci, M. Tagliasacchi, and S. Tubaro, “A phylogenetic analysis of near-duplicate audio tracks,” in *MMSP*, 2013.
- [20] H. Lian *et al.*, “Automatic video thumbnail selection,” in *2011 Int. Conf. on Multimedia Technology*, pp. 242–245, 2011.
- [21] W. Zhang *et al.*, “A novel framework for web video thumbnail generation,” in *2012 Int. Conf. on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 343–346, 2012.
- [22] J. Choi *et al.*, “A framework for automatic static and dynamic video thumbnail extraction,” *Multimedia Tools and Applicat.*, vol. 75, no. 23, p. 15975–15991, 2016.
- [23] Y. Song *et al.*, “To click or not to click: Automatic selection of beautiful thumbnails from videos,” in *25th ACM Int. on Conf. on Information and Knowledge Management*, p. 659–668, ACM, 2016.
- [24] C. Tsao *et al.*, “Thumbnail image selection for VOD services,” in *Proc. of the 2019 IEEE Conf. on Multimedia Information Processing and Retrieval*, pp. 54–59, 2019.
- [25] Y. Chen *et al.*, “Mobile media thumbnailing,” in *Int. Conf. on Multimedia Retrieval (ICMR)*, p. 665–666, ACM, 2015.
- [26] H. Gu *et al.*, “From thumbnails to summaries - A single deep neural network to rule them all,” in *IEEE Int. Conf. on Multimedia and Expo (ICME)*, pp. 1–6, 2018.
- [27] B. Zhao *et al.*, “Automatic generation of informative video thumbnail,” in *8th Int. Conf. on Digital Home*, pp. 254–259, 2020.
- [28] E. Apostolidis *et al.*, “Combining adversarial and reinforcement learning for video thumbnail selection,” in *Int. Conf. on Multimedia Retrieval (ICMR)*, p. 1–9, ACM, 2021.
- [29] A. B. Vasudevan *et al.*, “Query-adaptive video summarization via quality-aware relevance estimation,” in *25th ACM Int. Conf. on Multimedia (ACM MM)*, p. 582–590, ACM, 2017.
- [30] Y. Yuan *et al.*, “Sentence specified dynamic video thumbnail generation,” in *27th ACM Int. Conf. on Multimedia (ACM MM)*, p. 2332–2340, ACM, 2019.
- [31] M. Rochan *et al.*, “Sentence guided temporal modulation for dynamic video thumbnail generation,” in *British Machine Vision Conf. (BMVC)*, 2020.
- [32] K. Apostolidis *et al.*, “Image aesthetics assessment using Fully Convolutional Neural Networks,” in *25th Int. Conf. on MultiMedia Modeling (MMM)*, pp. 361–373, Springer International Publishing, 2019.
- [33] N. Murray *et al.*, “AVA: A large-scale database for aesthetic visual analysis,” in *IEEE Conf. on Computer Vision and Patt. Recog. (CVPR)*, pp. 2408–2415, 2012.
- [34] C. Szegedy *et al.*, “Going deeper with convolutions,” in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015.
- [35] A. Krizhevsky *et al.*, “Imagenet classification with Deep Convolutional Neural Networks,” *Communications of the ACM*, vol. 60, no. 6, p. 84–90, 2017.





- [36] R. J. Williams, “Simple statistical gradient-following algorithms for connectionist reinforcement learning,” *Machine Learning*, vol. 8, no. 3–4, p. 229–256, 1992.
- [37] E. Apostolidis *et al.*, “AC-SUM-GAN: Connecting Actor-Critic and Generative Adversarial Networks for unsupervised video summarization,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 31, no. 8, pp. 3278–3292, 2021.
- [38] E. Apostolidis *et al.*, “Summarizing videos using concentrated attention and considering the uniqueness and diversity of the video frames,” in *Int. Conf. on Multimedia Retrieval (ICMR)*, p. 407–415, ACM, 2022.
- [39] C. Collyda, K. Apostolidis, E. Apostolidis, E. Adamantidou, A. I. Metsai, and V. Mezaris, “A web service for video summarization,” in *ACM Int. Conf. on Interactive Media Experiences (IMX)*, pp. 148–153, 2020.
- [40] T. Souček and J. Lokoč, “Transnet V2: An effective deep network architecture for fast shot transition detection,” *arXiv preprint arXiv:2008.04838*, 2020.
- [41] E. Apostolidis *et al.*, “Video summarization using deep neural networks: A survey,” *Proceedings of the IEEE*, vol. 109, no. 11, pp. 1838–1863, 2021.
- [42] E. Apostolidis, E. Adamantidou, A. I. Metsai, V. Mezaris, and I. Patras, “Unsupervised video summarization via attention-driven adversarial learning,” in *Proc. of the 26th Int. Conf. on Multimedia Modeling (MMM 2020)*, (Cham), pp. 492–504, Springer International Publishing, 2020.
- [43] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool, “Creating summaries from user videos,” in *Proc. of the European Conference on Computer Vision – ECCV 2014* (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), (Cham), pp. 505–520, Springer International Publishing, 2014.
- [44] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes, “TVSum: Summarizing web videos using titles,” in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 5179–5187, 2015.
- [45] S. E. F. De Avila, A. P. B. Lopes, A. da Luz Jr, and A. de Albuquerque Araújo, “VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method,” *Pattern recognition letters*, vol. 32, no. 1, pp. 56–68, 2011.
- [46] X. He, Y. Hua, T. Song, Z. Zhang, Z. Xue, R. Ma, N. Robertson, and H. Guan, “Unsupervised video summarization with attentive conditional generative adversarial networks,” in *Proc. of the 27th ACM Int. Conf. on Multimedia (MM ’19)*, (New York, NY, USA), pp. 2296–2304, ACM, 2019.
- [47] B. Zhao, H. Li, X. Lu, and X. Li, “Reconstructive sequence-graph network for video summarization,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.
- [48] T. Liu, Q. Meng, J.-J. Huang, A. Vlontzos, D. Rueckert, and B. Kainz, “Video summarization through reinforcement learning with a 3D spatio-temporal U-Net,” *Trans. on Image Processing*, vol. 31, p. 1573–1586, 2022.
- [49] A. Phaphuangwittayakul, Y. Guo, F. Ying, W. Xu, and Z. Zheng, “Self-attention recurrent summarization network with reinforcement learning for video summarization task,” in *Proc. of the 2021 IEEE Int. Conf. on Multimedia and Expo (ICME)*, pp. 1–6, 2021.
- [50] H. Min, H. Ruimin, W. Zhongyuan, X. Zixiang, and Z. Rui, “Spatiotemporal two-stream lstm network for unsupervised video summarization,” *Multimedia Tools and Applications*, vol. 81, pp. 40489–40510, 2022.





- [51] G. Liang, Y. Lv, S. Li, S. Zhang, and Y. Zhang, “Video summarization with a convolutional attentive adversarial network,” *Pattern Recognition*, vol. 131, p. 108840, 2022.
- [52] P. Li, Q. Ye, L. Zhang, L. Yuan, X. Xu, and L. Shao, “Exploring global diverse attention via pairwise temporal relation for video summarization,” *Pattern Recognition*, vol. 111, p. 107677, 2021.
- [53] M. Rochan, L. Ye, and Y. Wang, “Video summarization using fully convolutional sequence networks,” in *European Conf. on Computer Vision (ECCV) 2018* (V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, eds.), (Cham), pp. 358–374, Springer International Publishing, 2018.
- [54] K. Apostolidis and V. Mezaris, “A web service for video smart-cropping,” in *2021 IEEE Int. Symposium on Multimedia (ISM)*, pp. 25–26, IEEE, 2021.
- [55] F. Hu, S. Palazzo, F. P. Salanitri, G. Bellitto, M. Moradi, C. Spampinato, and K. McGuinness, “Tinyhd: Efficient video saliency prediction with heterogeneous decoders using hierarchical maps distillation,” in *Proc. of the IEEE/CVF Winter Conf. on Applications of Computer Vision*, pp. 2051–2060, 2023.
- [56] K. Apostolidis and V. Mezaris, “A fast smart-cropping method and dataset for video retargeting,” in *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, pp. 1956–1960, 2021.
- [57] E. Apostolidis, G. Balaouras, V. Mezaris, and I. Patras, “Combining global and local attention with positional encoding for video summarization,” in *2021 IEEE International Symposium on Multimedia (ISM)*, pp. 226–234, 2021.
- [58] M. Narasimhan, A. Rohrbach, and T. Darrell, “Clip-it! language-guided video summarization,” in *Proceedings of the 35th International Conference on Neural Information Processing Systems, NIPS ’21*, (Red Hook, NY, USA), Curran Associates Inc., 2024.
- [59] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, “Learning transferable visual models from natural language supervision,” in *Proc. of the 38th Int. Conf. on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event* (M. Meila and T. Zhang, eds.), vol. 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763, PMLR, 2021.
- [60] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, “Video Summarization with Long Short-Term Memory,” in *Proc. of the European Conf. on Computer Vision 2016 (ECCV)* (B. Leibe, J. Matas, N. Sebe, and M. Welling, eds.), (Cham), pp. 766–782, Springer International Publishing, 2016.
- [61] E. Apostolidis, E. Adamantidou, A. I. Metsai, V. Mezaris, and I. Patras, “Performance over Random: A Robust Evaluation Protocol for Video Summarization Methods,” in *Proc. of the 28th ACM Int. Conf. on Multimedia (MM ’20)*, (New York, NY, USA), p. 1056–1064, ACM, 2020.
- [62] B. He, J. Wang, J. Qiu, T. Bui, A. Shrivastava, and Z. Wang, “Align and attend: Multimodal summarization with dual contrastive losses,” in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14867–14878, 2023.
- [63] J. Qiu, J. Zhu, W. Han, A. Kumar, K. Mittal, C. Jin, Z. Yang, L. Li, J. Wang, D. Zhao, B. Li, and L. Wang, “Mmsum: A dataset for multimodal summarization and thumbnail generation of videos,” 2023.
- [64] H. Hua, Y. Tang, C. Xu, and J. Luo, “V2xum-llm: Cross-modal video summarization with temporal prompt instruction tuning,” 2024.





- [65] D. M. Argaw, S. Yoon, F. C. Heilbron, H. Deilamsalehy, T. Bui, Z. Wang, F. Dernoncourt, and J. S. Chung, “Scaling up video summarization pretraining with large language models,” 2024.
- [66] S. H. Keller, K. S. Pedersen, and F. Lauze, “Detecting interlaced or progressive source of video,” in *MMSP*, 2005.
- [67] M. Pindoria and T. Borer, “Automatic interlace or progressive video discrimination,” in *SMPTE*, 2012.
- [68] T. Kroeger and R. Timofte, “Fast optical flow using dense inverse search,” in *ECCV 2016*, 2016.
- [69] B. Mahasseni, M. Lam, and S. Todorovic, “Unsupervised video summarization with adversarial lstm networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [70] A. Kulesza and B. Taskar, “Determinantal Point Processes for machine learning,” *arXiv preprint arXiv:1207.6083*, 2012.
- [71] E. Apostolidis, A. I. Metsai, E. Adamantidou, V. Mezaris, and I. Patras, “A stepwise, label-based approach for improving the adversarial training in unsupervised video summarization,” in *Proceedings of the International Workshop on AI for Smart TV Content Production, Access and Delivery*, 2019.
- [72] E. Apostolidis, E. Adamantidou, A. I. Metsai, V. Mezaris, and I. Patras, “AC-SUM-GAN: connecting actor-critic and generative adversarial networks for unsupervised video summarization,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 8, pp. 3278–3292, 2020.
- [73] M. Schilling, A. Melnik, F. W. Ohl, H. J. Ritter, and B. Hammer, “Decentralized control and local information for robust and adaptive decentralized Deep Reinforcement Learning,” *Neural Networks*, vol. 144, pp. 699–725, 2021.
- [74] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor,” in *Proceedings of the International Conference on Machine Learning*, PMLR, 2018.
- [75] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes, “TVSum: Summarizing web videos using titles,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [76] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, “Video summarization with Long Short-Term Memory,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, Springer, 2016.
- [77] Y. Zhang, X. Liang, D. Zhang, M. Tan, and E. P. Xing, “Unsupervised object-level video summarization with online motion auto-encoder,” *Pattern Recognition Letters*, vol. 130, pp. 376–385, 2020.
- [78] M. Rochan, L. Ye, and Y. Wang, “Video summarization using fully convolutional sequence networks,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, Springer, 2018.
- [79] K. Zhou, Y. Qiao, and T. Xiang, “Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.
- [80] N. Gonuguntla, B. Mandal, and N. Puhan, “Enhanced deep video summarization network,” in *Proceedings of the British Machine Vision Conference (BMVC)*, 2019.
- [81] M. Rochan and Y. Wang, “Video summarization by learning from unpaired data,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.





- [82] B. Zhao, X. Li, and X. Lu, “Property-constrained dual learning for video summarization,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 10, pp. 3989–4000, 2019.
- [83] X. He, Y. Hua, T. Song, Z. Zhang, Z. Xue, R. Ma, N. Robertson, and H. Guan, “Unsupervised video summarization with attentive conditional Generative Adversarial Networks,” in *Proceedings of the ACM International Conference on Multimedia*, 2019.
- [84] Y. Jung, D. Cho, D. Kim, S. Woo, and I. S. Kweon, “Discriminative feature learning for unsupervised video summarization,” in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2019.
- [85] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool, “Creating summaries from user videos,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, Springer, 2014.
- [86] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.
- [87] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.
- [88] G. F. Elsayed, I. Goodfellow, and J. Sohl-Dickstein, “Adversarial reprogramming of neural networks,” in *International Conference on Learning Representations*, 2018.
- [89] Y. Zheng, X. Feng, Z. Xia, X. Jiang, A. Demontis, M. Pintor, B. Biggio, and F. Roli, “Why adversarial reprogramming works, when it fails, and how to tell the difference,” *Information Sciences*, vol. 632, pp. 130–143, 2023.
- [90] M. Englert and R. Lazic, “Adversarial reprogramming revisited,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 28588–28600, 2022.
- [91] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *AISTATS*, 2010.
- [92] M. Narasimhan, A. Rohrbach, and T. Darrell, “CLIP-It! language-guided video summarization,” *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, vol. 34, 2021.
- [93] E. Apostolidis, G. Balaouras, V. Mezaris, and I. Patras, “Combining global and local attention with positional encoding for video summarization,” in *Proceedings of the IEEE International Symposium on Multimedia (ISM)*, 2021.
- [94] Z. Ji, K. Xiong, Y. Pang, and X. Li, “Video summarization with attention-based encoder-decoder networks,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 6, pp. 1709–1717, 2019.
- [95] W. Zhu, J. Lu, J. Li, and J. Zhou, “DSNet: A flexible detect-to-summarize network for video summarization,” *IEEE Transactions on Image Processing*, vol. 30, pp. 948–962, 2020.
- [96] J. Fajtl, H. S. Sokeh, V. Argyriou, D. Monekosso, and P. Remagnino, “Summarizing videos with attention,” in *Computer Vision–ACCV 2018 Workshops: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers 14*, pp. 39–54, Springer, 2019.
- [97] Z. Ji, F. Jiao, Y. Pang, and L. Shao, “Deep attentive and semantic preserving video summarization,” *Neurocomputing*, vol. 405, pp. 200–207, 2020.





- [98] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, “Real-time multi-person 2D pose estimation using part affinity fields,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [99] I. Mademlis, A. Tefas, N. Nikolaidis, and I. Pitas, “Multimodal stereoscopic movie summarization conforming to narrative characteristics,” *IEEE Transactions on Image Processing*, vol. 25, no. 12, pp. 5828–5840, 2016.
- [100] I. Mademlis, A. Tefas, N. Nikolaidis, and I. Pitas, “Compact video description and representation for automated summarization of human activities,” in *Proceedings of the INNS Conference on Big Data*, Springer, 2017.
- [101] C. Symeonidis, I. Mademlis, I. Pitas, and N. Nikolaidis, “Neural attention-driven Non-Maximum Suppression for person detection,” *IEEE Transactions on Image Processing*, 2023.
- [102] F. Yang, Y. Wu, S. Sakti, and S. Nakamura, “Make skeleton-based action recognition model smaller, faster and better,” in *Proceedings of the ACM Multimedia Asia*, 2019.
- [103] C. Papaioannidis, D. Makrygiannis, I. Mademlis, and I. Pitas, “Learning fast and robust gesture recognition,” in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, 2021.
- [104] J. Kittler, M. Hatef, R. P. Duin, and J. Matas, “On combining classifiers,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 20, no. 3, pp. 226–239, 1998.
- [105] L. Lam and S. Suen, “Application of majority voting to pattern recognition: an analysis of its behavior and performance,” *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 27, no. 5, pp. 553–568, 1997.
- [106] L. Lamport, R. Shostak, and M. Pease, “The byzantine generals problem,” in *Concurrency: the works of leslie lamport*, pp. 203–226, 2019.
- [107] A. Bessani, J. Sousa, and E. E. Alchieri, “State machine replication for the masses with bft-smart,” in *2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, pp. 355–362, IEEE, 2014.
- [108] “Sex and gender.” (last accessed june 2024).
- [109] D. Doukhan, G. Poels, Z. Rezgui, and J. Carrive, “Describing gender equality in french audiovisual streams with a deep learning approach,” *VIEW Journal of European Television History and Culture*, vol. 7, no. 14, pp. 103–122, 2018.
- [110] M. Bazin and C. Méadel, “Les SHS dans le projet Gender Equality Monitoring,” tech. rep., GEM, oct 2022. (last accessed oct 2023).
- [111] P. Terhörst, D. Fährmann, N. Damer, F. Kirchbuchner, and A. Kuijper, “Beyond identity: What information is stored in biometric face templates?,” in *2020 IEEE international joint conference on biometrics (IJCB)*, pp. 1–10, IEEE, 2020.
- [112] A. Swaminathan, M. Chaba, D. K. Sharma, and Y. Chaba, “Gender classification using facial embeddings: A novel approach,” *Procedia Computer Science*, vol. 167, pp. 2634–2642, 2020.
- [113] Y. Lin and H. Xie, “Face gender recognition based on face recognition feature vectors,” in *2020 IEEE 3rd International conference on information systems and computer aided education (ICISCAE)*, pp. 162–166, IEEE, 2020.





- [114] M. Farzaneh, “Arcface knows the gender, too!” *arXiv preprint arXiv:2112.10101*, 2021.
- [115] T. Kim, “Generalizing MLPs with dropouts, batch normalization, and skip connections,” *arXiv preprint arXiv:2108.08186*, 2021.
- [116] M. Kuprashevich and I. Tolstykh, “Mivolo: Multi-input transformer for age and gender estimation,” *arXiv preprint arXiv:2307.04616*, 2023.
- [117] E. Bonet Cervera, “Age & gender recognition in the wild,” B.S. thesis, Universitat Politècnica de Catalunya, 2022.
- [118] A. Hast, “Sex Classification of Face Images using Embedded Prototype Subspace Classifiers,” *Computer Science Research Notes*, vol. 3301, pp. 43–52, 2023.
- [119] “Initial report on multimedia summarisation and analysis (d5.1).” (last accessed june 2024).
- [120] G. Chen, P. Chen, Y. Shi, C.-Y. Hsieh, B. Liao, and S. Zhang, “Rethinking the usage of batch normalization and dropout in the training of deep neural networks,” *arXiv preprint arXiv:1905.05928*, 2019.
- [121] L. Van der Maaten and G. Hinton, “Visualizing data using t-sne.,” *Journal of machine learning research*, vol. 9, no. 11, 2008.
- [122] R. Rothe, R. Timofte, and L. Gool, “Imdb-wiki-500k+ face images with age and gender labels,” *Online] URL: <https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki>*, vol. 4, 2015.
- [123] E. Eiding, R. Enbar, and T. Hassner, “Age and gender estimation of unfiltered faces,” *IEEE Transactions on information forensics and security*, vol. 9, no. 12, pp. 2170–2179, 2014.
- [124] (last accessed june 2024).
- [125] (last accessed june 2024).
- [126] (last accessed june 2024).
- [127] B. Huurnink, L. Hollink, W. Van Den Heuvel, and M. De Rijke, “Search behavior of media professionals at an audiovisual archive: A transaction log analysis,” *Journal of the American society for information science and technology*, vol. 61, no. 6, pp. 1180–1197, 2010.
- [128] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, “Places: A 10 million image database for scene recognition,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 6, pp. 1452–1464, 2017.
- [129] R. P. Mihail, S. Workman, Z. Bessinger, and N. Jacobs, “Sky segmentation in the wild: An empirical study,” in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1–6, 2016. Acceptance rate: 42.3%.
- [130] Q. Wang, J. Xie, W. Zuo, L. Zhang, and P. Li, “Deep cnns meet global covariance pooling: Better representation and generalization,” *IEEE transactions on pattern analysis and machine intelligence*, 2020.
- [131] H. Qassim, A. Verma, and D. Feinzimer, “Compressed residual-vgg16 cnn model for big data places image recognition,” in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 169–175, IEEE, 2018.





- [132] T. Xiao, P. Dollar, M. Singh, E. Mintun, T. Darrell, and R. Girshick, “Early convolutions help transformers see better,” *Advances in Neural Information Processing Systems*, vol. 34, 2021.
- [133] M. Savardi, A. Signoroni, P. Migliorati, and S. Benini, “Shot scale analysis in movies by convolutional neural networks,” in *2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 2620–2624, IEEE, 2018.
- [134] H.-Y. Bak and S.-B. Park, “Comparative study of movie shot classification based on semantic segmentation,” *Applied Sciences*, vol. 10, no. 10, p. 3390, 2020.
- [135] A. Rao, J. Wang, L. Xu, X. Jiang, Q. Huang, B. Zhou, and D. Lin, “A unified framework for shot type classification based on subject centric lens,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pp. 17–34, Springer, 2020.
- [136] P. Cheng and J. Zhou, “Automatic season classification of outdoor photos,” in *2011 Third International Conference on Intelligent Human-Machine Systems and Cybernetics*, vol. 1, pp. 46–49, IEEE, 2011.
- [137] G. Awad, C. G. Snoek, A. F. Smeaton, and G. Quénot, “Trecvid semantic indexing of video: A 6-year retrospective,” *ITE Transactions on Media Technology and Applications*, vol. 4, no. 3, pp. 187–208, 2016.
- [138] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, “Youtube-8m: A large-scale video classification benchmark,” *arXiv preprint arXiv:1609.08675*, 2016.
- [139] ETSI, “Ts 102 822-3-1 v1.9.1 - broadcast and on-line services: Search, select, and rightful use of content on personal storage systems (tv-anytime); part 3: Metadata; sub-part 1: Phase 1 - metadata schemas,” tech. rep., 2015.
- [140] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, “Scaled-yolov4: Scaling cross stage partial network,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13029–13038, 2021.
- [141] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “Yolov4: Optimal speed and accuracy of object detection,” *ArXiv*, vol. abs/2004.10934, 2020.
- [142] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou, “Retinaface: Single-stage dense face localisation in the wild,” *arXiv preprint arXiv:1905.00641*, 2019.
- [143] S. Yang, P. Luo, C.-C. Loy, and X. Tang, “Wider face: A face detection benchmark,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5525–5533, 2016.
- [144] Y. Sun, Q. Bao, W. Liu, Y. Fu, M. J. Black, and T. Mei, “Monocular, one-stage, regression of multiple 3d people,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11179–11188, 2021.
- [145] T. von Marcard, R. Henschel, M. Black, B. Rosenhahn, and G. Pons-Moll, “Recovering accurate 3d human pose in the wild using imus and a moving camera,” in *European Conference on Computer Vision (ECCV)*, sep 2018.
- [146] I. Kissos, L. Fritz, M. Goldman, O. Meir, E. Oks, and M. Klinger, “Beyond weak perspective for monocular 3d human pose estimation,” in *European Conference on Computer Vision (ECCV)*, pp. 541–554, 01 2020.





- [147] D. Arijon, *Grammar of the film language*. Silman-James Press, 1991.
- [148] Q. Galvane, *Automatic Cinematography and Editing in Virtual Environments*. PhD thesis, Université Grenoble Alpes (ComUE), 2015.
- [149] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, “SMPL: A skinned multi-person linear model,” *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, vol. 34, pp. 248:1–248:16, Oct. 2015.
- [150] A. T. Fairbanks and E. F. Fairbanks, *Human proportions for artists*. Fairbanks Art and Books, 2005.
- [151] K. E. Trenberth, “What are the seasons?,” *Bulletin of the American Meteorological Society*, vol. 64, no. 11, pp. 1276–1282, 1983.
- [152] S. Boggs, “Seasonal variations in daylight, twilight, and darkness,” *Geographical Review*, vol. 21, no. 4, pp. 656–659, 1931.
- [153] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *International Conference on Machine Learning*, pp. 6105–6114, PMLR, 2019.
- [154] R. Wightman, “Pytorch image models.” <https://github.com/rwightman/pytorch-image-models>, 2019.
- [155] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, “Masked autoencoders are scalable vision learners,” *arXiv preprint arXiv:2111.06377*, 2021.
- [156] J. Tores, L. Sassatelli, H.-Y. Wu, C. Bergman, L. Andolfi, V. Ecrement, F. Precioso, T. Devars, M. Guaresi, V. Julliard, and S. Lecossais, “Visual objectification in films: Towards a new ai task for video interpretation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10864–10874, June 2024.
- [157] Y. Mathet, A. Widlöcher, and J.-P. Métivier, “The unified and holistic method gamma (γ) for inter-annotator agreement measure and alignment,” *Computational Linguistics*, vol. 41, no. 3, pp. 437–479, 2015.
- [158] M. Yuksekgonul, M. Wang, and J. Zou, “Post-hoc concept bottleneck models,” in *ICLR 2022 Workshop on PAIR²Struct: Privacy, Accountability, Interpretability, Robustness, Reasoning on Structured Data*, 2022.
- [159] B. Kim, M. Wattenberg, J. Gilmer, C. J. Cai, J. Wexler, F. B. Viégas, and R. Sayres, “Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav),” in *International Conference on Machine Learning*, 2017.
- [160] M. Quigley, K. Conley, B. Gerkey, and J. e. a. Faust, “Ros: an open-source robot operating system,” 01 2009.
- [161] S. Shah, D. Dey, C. Lovett, and A. Kapoor, “Airsim: High-fidelity visual and physical simulation for autonomous vehicles,” *CoRR*, vol. abs/1705.05065, 2017.
- [162] I. Mademlis, N. Nikolaidis, A. Tefas, I. Pitas, T. Wagner, and A. Messina, “Autonomous uav cinematography: A tutorial and a formalized shot-type taxonomy,” *ACM Computing Surveys (CSUR)*, vol. 52, no. 5, pp. 1–33, 2019.
- [163] X. Ren and X. Wang, “Look outside the room: Synthesizing a consistent long-term 3d scene video from a single image,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.





- [164] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *CVPR*, 2018.
- [165] P. Nousi, I. Mademlis, I. Karakostas, A. Tefas, and I. Pitas, “Embedded uav real-time visual object detection and tracking,” in *2019 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pp. 708–713, IEEE, 2019.
- [166] I. Karakostas, V. Mygdalis, A. Tefas, and I. Pitas, “Occlusion detection and drift-avoidance framework for 2d visual object tracking,” *Signal Processing: Image Communication*, vol. 90, p. 116011, 2021.
- [167] E. Patsiouras, V. Mygdalis, and I. Pitas, “Whitening transformation inspired self-attention for powerline element detection,” in *2022 26th International Conference on Pattern Recognition (ICPR)*, pp. 4844–4849, IEEE, 2022.
- [168] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, J.-K. Kamarainen, H. J. Chang, M. Danelljan, L. Čehovin Zajc, A. Lukežič, O. Drbohlav, J. Bjorklund, Y. Zhang, Z. Zhang, S. Yan, W. Yang, D. Cai, C. Mayer, and G. Fernandez, “The tenth visual object tracking vot2022 challenge results,” 2022.
- [169] A. He, C. Luo, X. Tian, and W. Zeng, “A twofold siamese network for real-time object tracking,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4834–4843, 2018.
- [170] H. Caesar, J. Uijlings, and V. Ferrari, “Coco-stuff: Thing and stuff classes in context,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1209–1218, 2018.
- [171] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, NanoCode012, Y. Kwon, K. Michael, TaoXie, J. Fang, imyhxy, Lorna, Z. Yifu, C. Wong, A. V, D. Montes, Z. Wang, C. Fati, J. Nadar, Laughing, UnglvKitDe, V. Sonck, tkianai, yxNONG, P. Skalski, A. Hogan, D. Nair, M. Strobel, and M. Jain, “ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation,” Nov. 2022.
- [172] Q. Guimard, L. Sassatelli, F. Marchetti, F. Becattini, L. Seidenari, and A. D. Bimbo, “Deep variational learning for multiple trajectory prediction of 360 head movements,” in *Proceedings of the 13th ACM Multimedia Systems Conference*, pp. 12–26, 2022.
- [173] Q. Guimard, L. Sassatelli, F. Marchetti, F. Becattini, L. Seidenari, and A. D. Bimbo, “Deep variational learning for 360 adaptive streaming,” *ACM Transactions on Multimedia Computing, Communications and Applications*, 2024.
- [174] F. Marchetti, F. Becattini, L. Seidenari, and A. Del Bimbo, “Explainable sparse attention for memory-based trajectory predictors,” in *European Conference on Computer Vision*, pp. 543–560, Springer, 2022.
- [175] F. Marchetti, F. Becattini, L. Seidenari, and A. Del Bimbo, “Smemo: social memory for trajectory forecasting,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- [176] A. Ciamarra, F. Becattini, L. Seidenari, and A. Del Bimbo, “Forecasting future instance segmentation with learned optical flow and warping,” in *International Conference on Image Analysis and Processing*, pp. 349–361, Springer, 2022.
- [177] A. Ciamarra, F. Becattini, L. Seidenari, and A. Del Bimbo, “Flodcast: Flow and depth forecasting via multimodal recurrent architectures,” *Pattern Recognition*, p. 110337, 2024.





- [178] D. Pucci, F. Becattini, and A. Del Bimbo, “Joint-based action progress prediction,” *Sensors*, vol. 23, no. 1, p. 520, 2023.
- [179] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, “Deeper depth prediction with fully convolutional residual networks,” in *2016 Fourth international conference on 3D vision (3DV)*, pp. 239–248, IEEE, 2016.
- [180] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, “Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data,” in *European Conference on Computer Vision*, pp. 683–700, Springer, 2020.
- [181] T. Gu, G. Chen, J. Li, C. Lin, Y. Rao, J. Zhou, and J. Lu, “Stochastic trajectory prediction via motion indeterminacy diffusion,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17113–17122, 2022.
- [182] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofghi, and S. Savarese, “Sophie: An attentive gan for predicting paths compliant to social and physical constraints,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1349–1358, 2019.
- [183] F. Marchetti, F. Becattini, L. Seidenari, and A. Del Bimbo, “MANTRA: Memory augmented networks for multiple trajectory prediction,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.
- [184] J. Li, F. Yang, M. Tomizuka, and C. Choi, “Evolvegraph: Multi-agent trajectory prediction with dynamic relational reasoning,” *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2020.
- [185] B. Pang, T. Zhao, X. Xie, and Y. N. Wu, “Trajectory prediction with latent belief energy-based model,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11814–11824, 2021.
- [186] A. Bhattacharyya, M. Hanselmann, M. Fritz, B. Schiele, and C.-N. Straehle, “Conditional flow variational autoencoders for structured sequence prediction,” 2020.
- [187] J. Sun, Y. Li, H.-S. Fang, and C. Lu, “Three steps to multimodal trajectory prediction: Modality clustering, classification and synthesis,” *arXiv preprint arXiv:2103.07854*, 2021.
- [188] N. Deo and M. M. Trivedi, “Trajectory forecasts in unknown environments conditioned on grid-based plans,” *arXiv preprint arXiv:2001.00735*, 2020.
- [189] C. Xu, W. Mao, W. Zhang, and S. Chen, “Remember intentions: Retrospective-memory-based trajectory prediction,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6488–6497, 2022.
- [190] P. Dendorfer, A. Osep, and L. Leal-Taixe, “Goal-gan: Multimodal trajectory prediction based on goal position estimation,” in *Proceedings of the Asian Conference on Computer Vision (ACCV)*, November 2020.
- [191] W. Mao, C. Xu, Q. Zhu, S. Chen, and Y. Wang, “Leapfrog diffusion model for stochastic trajectory prediction,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5517–5526, 2023.
- [192] Z. He and R. P. Wildes, “Where are you heading? dynamic trajectory prediction with expert goal examples,” in *Proceedings of the International Conference on Computer Vision (ICCV)*, Oct. 2021.





- [193] K. Mangalam, Y. An, H. Girase, and J. Malik, “From goals, waypoints & paths to long term human trajectory forecasting,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15233–15242, 2021.
- [194] J. Liang, L. Jiang, and A. Hauptmann, “Simaug: Learning robust representations from simulation for trajectory prediction,” in *European Conference on Computer Vision*, pp. 275–292, Springer, 2020.
- [195] K. Mangalam, H. Girase, S. Agarwal, K.-H. Lee, E. Adeli, J. Malik, and A. Gaidon, “It is not the journey but the destination: Endpoint conditioned trajectory prediction,” *arXiv preprint arXiv:2004.02025*, 2020.
- [196] X. Qi, Z. Liu, Q. Chen, and J. Jia, “3d motion decomposition for rgbd future dynamic scene synthesis,” in *Proceedings of Conference on Computer Vision and Pattern Recognition*, pp. 7673–7682, 2019.
- [197] A. Hu, F. Cotter, N. Mohan, C. Gurau, and A. Kendall, “Probabilistic future prediction for video scene understanding,” in *European Conference on Computer Vision*, pp. 767–785, Springer, 2020.
- [198] J. Sun, J. Xie, J.-F. Hu, Z. Lin, J. Lai, W. Zeng, and W.-S. Zheng, “Predicting future instance segmentation with contextual pyramid convlstm,” in *Proceedings of the 27th acm international conference on multimedia*, pp. 2043–2051, 2019.
- [199] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, “Digging into self-supervised monocular depth estimation,” in *Proceedings of International Conference on Computer Vision*, pp. 3828–3838, 2019.
- [200] S. Nag, N. Shah, A. Qi, and R. Ramachandra, “How far can i go?: A self-supervised approach for deterministic video depth forecasting,” *arXiv preprint arXiv:2207.00506*, 2022.
- [201] L. Berlincioni, S. Berretti, M. Bertini, and A. D. Bimbo, “4dsr-gcn: 4d video point cloud upsampling using graph convolutional networks,” in *Proceedings of the 1st International Workshop on Multimedia Content Generation and Evaluation: New Methods and Practice*, pp. 57–65, 2023.
- [202] F. Principi, S. Berretti, C. Ferrari, N. Otberdout, M. Daoudi, and A. Del Bimbo, “The florence 4d facial expression dataset,” in *2023 IEEE 17th International Conference on Automatic Face and Gesture Recognition (FG)*, pp. 1–6, IEEE, 2023.
- [203] L. Berlincioni, L. Cultrera, C. Albisani, L. Cresti, A. Leonardo, S. Picchioni, F. Becattini, and A. Del Bimbo, “Neuromorphic event-based facial expression recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4108–4118, 2023.
- [204] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “Pointnet++: Deep hierarchical feature learning on point sets in a metric space,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.
- [205] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, “Dynamic graph cnn for learning on point clouds,” *Acm Transactions On Graphics (tog)*, vol. 38, no. 5, pp. 1–12, 2019.
- [206] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, “Graph attention networks,” *arXiv preprint arXiv:1710.10903*, 2017.
- [207] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems*, vol. 27, 2014.





- [208] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, “Least squares generative adversarial networks,” *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2813–2821, 2016.
- [209] T. Wu, L. Pan, J. Zhang, T. Wang, Z. Liu, and D. Lin, “Density-aware chamfer distance as a comprehensive metric for point cloud completion,” *arXiv preprint arXiv:2111.12702*, 2021.
- [210] A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black, “Generating 3D faces using convolutional mesh autoencoders,” in *European Conf. on Computer Vision (ECCV)*, pp. 725–741, 2018.
- [211] I. Daz Productions, “Daz 3D,” 2022.
- [212] R. LLC, “R3DS wrap3,” 2022.
- [213] P. Ekman, “Universal facial expressions in emotion,” *Studia Psychologica*, vol. 15, no. 2, p. 140, 1973.
- [214] W.-J. Yan, Q. Wu, Y.-J. Liu, S.-J. Wang, and X. Fu, “Casm database: A dataset of spontaneous micro-expressions collected from neutralized faces,” in *2013 10th IEEE international conference and workshops on automatic face and gesture recognition (FG)*, pp. 1–7, IEEE, 2013.
- [215] A. K. Davison, C. Lansley, N. Costen, K. Tan, and M. H. Yap, “Samm: A spontaneous micro-facial movement dataset,” *IEEE transactions on affective computing*, vol. 9, no. 1, pp. 116–129, 2016.
- [216] H. Rebecq, D. Gehrig, and D. Scaramuzza, “ESIM: an open event camera simulator,” *Conf. on Robotics Learning (CoRL)*, Oct. 2018.
- [217] A. Bulat and G. Tzimiropoulos, “How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks),” in *International Conference on Computer Vision*, 2017.
- [218] J. Redmon and A. Farhadi, “Yolo9000: better, faster, stronger,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271, 2017.
- [219] C. Vondrick, D. Patterson, and D. Ramanan, “Efficiently scaling up crowdsourced video annotation: A set of best practices for high quality, economical video labeling,” *International journal of computer vision*, vol. 101, pp. 184–204, 2013.
- [220] X. Lagorce, G. Orchard, F. Galluppi, B. E. Shi, and R. B. Benosman, “Hots: A hierarchy of event-based time-surfaces for pattern recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 7, pp. 1346–1359, 2017.
- [221] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017.
- [222] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3d convolutional networks,” in *Proceedings of the IEEE international conference on computer vision*, pp. 4489–4497, 2015.
- [223] “T-c3d: Temporal convolutional 3d network for real-time action recognition,” vol. 32.
- [224] Y. Fan, X. Lu, D. Li, and Y. Liu, “Video-based emotion recognition using cnn-rnn and c3d hybrid networks,” in *Proceedings of the 18th ACM International Conference on Multimodal Interaction, ICMI '16*, (New York, NY, USA), p. 445–450, Association for Computing Machinery, 2016.



- [225] A. Montes, A. Salvador, S. Pascual, and X. Giro-i Nieto, “Temporal activity detection in untrimmed videos with recurrent neural networks,” in *1st NIPS Workshop on Large Scale Computer Vision Systems*, December 2016.
- [226] S. U. Innocenti, F. Becattini, F. Pernici, and A. Del Bimbo, “Temporal binary representation for event-based action recognition,” in *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 10426–10432, IEEE, 2021.
- [227] T.-A. Vu, D. T. Nguyen, B.-S. Hua, Q.-H. Pham, and S.-K. Yeung, “Rfnet-4d: Joint object reconstruction and flow estimation from 4d point clouds,” in *European Conf. on Computer Vision (ECCV)*, pp. 36–52, 2022.
- [228] H. Fan, H. Su, and L. J. Guibas, “A point set generation network for 3d object reconstruction from a single image,” in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [229] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, “Occupancy networks: Learning 3D reconstruction in function space,” in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 4460–4470, 2019.
- [230] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger, “Occupancy flow: 4D reconstruction by learning particle dynamics,” in *IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, Oct. 2019.
- [231] J. Tang, D. Xu, K. Jia, and L. Zhang, “Learning parallel dense correspondence from spatio-temporal descriptors for efficient and robust 4d reconstruction,” *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 6018–6027, 2021.
- [232] B. Jiang, Y. Zhang, X. Wei, X. Xue, and Y. Fu, “Learning compositional representation for 4d captures with neural ode,” in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 5340–5350, June 2021.
- [233] G. Bouritsas, S. Bokhnyak, S. Ploumpis, S. Zafeiriou, and M. Bronstein, “Neural 3D morphable models: Spiral convolutional networks for 3D shape representation learning and generation,” in *IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, pp. 7212–7221, 2019.
- [234] N. Otberdout, C. Ferrari, M. Daoudi, S. Berretti, and A. D. Bimbo, “Sparse to dense dynamic 3d facial expression generation,” in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 20385–20394, IEEE, 2022.
- [235] C. Ferrari, G. Lisanti, S. Berretti, and A. Del Bimbo, “Dictionary learning based 3d morphable model construction for face recognition with varying expression and pose,” in *IEEE Int. Conf. on 3D Vision*, pp. 509–517, 2015.
- [236] T. Li, T. Bolkart, M. Julian, H. Li, and J. Romero, “Learning a model of facial shape and expression from 4D scans,” *ACM Trans. on Graphics, (Proc. SIGGRAPH Asia)*, vol. 36, no. 6, 2017.
- [237] C. Ferrari, S. Berretti, P. Pala, and A. Del Bimbo, “A sparse and locally coherent morphable face model for dense semantic correspondence across heterogeneous 3D faces,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2021.
- [238] R. A. Potamias, J. Zheng, S. Ploumpis, G. Bouritsas, E. Ververas, and S. Zafeiriou, “Learning to generate customized dynamic 3D facial expressions,” in *European Conf. on Computer Vision (ECCV)*, pp. 278–294, 2020.



- [239] L. N. Smith and N. Topin, “Super-convergence: Very fast training of neural networks using large learning rates,” in *Artificial intelligence and machine learning for multi-domain operations applications*, vol. 11006, pp. 369–386, SPIE, 2019.
- [240] L. Galteri, L. Seidenari, P. Bongini, M. Bertini, and A. D. Bimbo, “Lanbique: Language-based blind image quality evaluation,” *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 18, no. 2s, pp. 1–19, 2022.
- [241] L. Agnolucci, L. Galteri, M. Bertini, and A. Del Bimbo, “Perceptual quality improvement in videoconferencing using keyframes-based gan,” *IEEE Transactions on Multimedia*, 2023.
- [242] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Proceedings of the International Conference on Medical image computing and computer-assisted intervention*, 2015.
- [243] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [244] X. Huang and S. Belongie, “Arbitrary style transfer in real-time with adaptive instance normalization,” in *Proc. of European Conference on Computer Vision (ECCV)*, 2017.
- [245] X. Wang, K. Yu, C. Dong, and C. C. Loy, “Recovering realistic texture in image super-resolution by deep spatial feature transform,” in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [246] L. Galteri, M. Bertini, L. Seidenari, T. Uricchio, and A. Del Bimbo, “Increasing Video Perceptual Quality with GANs and Semantic Coding,” in *Proc. of ACM International Conference on Multimedia (ACM MM)*, 2020.
- [247] B. Dogan, S. Gu, and R. Timofte, “Exemplar guided face image super-resolution without facial landmarks,” in *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2019.
- [248] X. Li, M. Liu, Y. Ye, W. Zuo, L. Lin, and R. Yang, “Learning Warped Guidance for Blind Face Restoration,” in *Proc. of European Conference on Computer Vision (ECCV)*, 2018.
- [249] X. Li, W. Li, D. Ren, H. Zhang, M. Wang, and W. Zuo, “Enhanced blind face restoration with multi-exemplar images and adaptive spatial feature fusion,” in *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [250] S. Schaefer, T. McPhail, and J. Warren, “Image deformation using Moving Least Squares,” in *Proc. of ACM SIGGRAPH*, 2006.
- [251] L. Galteri, L. Seidenari, M. Bertini, and A. Del Bimbo, “Deep generative adversarial compression artifact removal,” in *Proc. of ICCV*, pp. 4826–4835, 2017.
- [252] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, “Bottom-up and top-down attention for image captioning and visual question answering,” in *Proc. of CVPR*, pp. 6077–6086, 2018.
- [253] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2015.





- [254] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, *et al.*, “Visual genome: Connecting language and vision using crowdsourced dense image annotations,” *International Journal of Computer Vision*, vol. 123, no. 1, pp. 32–73, 2017.
- [255] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: a method for automatic evaluation of machine translation,” in *Proc. of ACL*, pp. 311–318, July 2002.
- [256] S. Banerjee and A. Lavie, “Meteor: An automatic metric for mt evaluation with improved correlation with human judgments,” in *Proc. of ACL workshop*, pp. 65–72, 2005.
- [257] C.-Y. Lin, “ROUGE: A package for automatic evaluation of summaries,” in *Proc. of ACL*, pp. 74–81, July 2004.
- [258] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, “CIDEr: Consensus-based image description evaluation,” in *Proc. of CVPR*, pp. 4566–4575, 2015.
- [259] P. Anderson, B. Fernando, M. Johnson, and S. Gould, “Spice: Semantic propositional image caption evaluation,” in *Proc. of ECCV*, pp. 382–398, Springer, 2016.
- [260] X. Wang, L. Xie, C. Dong, and Y. Shan, “Real-ESRGAN: Training real-world blind super-resolution with pure synthetic data,” in *Proc. of ICCV*, pp. 1905–1914, 2021.
- [261] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [262] G. Meseguer-Brocal, A. Cohen-Hadria, and G. Peeters, “Creating dali, a large dataset of synchronized audio, lyrics, and notes,” *Transactions of the International Society for Music Information Retrieval*, vol. 3, no. 1, 2020.
- [263] C. E. Cancino-Chacón, M. Grachten, W. Goebel, and G. Widmer, “Computational models of expressive music performance: A comprehensive and critical review,” *Frontiers in Digital Humanities*, vol. 5, 2018.
- [264] I. Malik and C. H. Ek, “Neural translation of musical style,” in *Workshop on Machine Learning for Creativity and Design, Neural Information Processing Systems (NIPS)*, (Long Beach, California, USA), Dec. 8, 2017.
- [265] F. J. Muneratti Ortega *et al.*, *A machine learning approach to computer modeling of musical expression for performance learning and practice*. PhD thesis, Universitat Pompeu Fabra, 2022.
- [266] H.-W. Dong, C. Zhou, T. Berg-Kirkpatrick, and J. McAuley, “Deep performer: Score-to-audio music performance synthesis,” in *Proc. of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 951–955, IEEE, 2022.
- [267] S. Rhyu, S. Kim, and K. Lee, “Sketching the expression: Flexible rendering of expressive piano performance with self-supervised learning,” in *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, (Bengaluru, India), Dec. 4-8, 2022.
- [268] A. Maezawa, K. Yamamoto, and T. Fujishima, “Rendering music performance with interpretation variations using conditional variational RNN,” in *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, (Delft, The Netherlands), pp. 855–861, Nov. 4-8, 2019.





- [269] D. Jeong, T. Kwon, Y. Kim, K. Lee, and J. Nam, “VirtuosoNet: A Hierarchical RNN-based System for Modeling Expressive Piano Performance,” in *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, (Delft, The Netherlands), pp. 908–915, 2019.
- [270] D. Jeong, T. Kwon, Y. Kim, and J. Nam, “Graph neural network for music score data and modeling expressive piano performance,” in *Proc. of the International Conference on Machine Learning (ICML)*, (Long Beach, California, USA), pp. 3060–3070, PMLR, June. 9-15, 2019.
- [271] D. Jeong, T. Kwon, Y. Kim, and J. Nam, “Score and performance features for rendering expressive music performances,” in *Proc. of the Music Encoding Conference*, (Vienna, Austria), May 2019.
- [272] F. Foscarin, E. Karystinaios, S. D. Peter, C. Cancino-Chacón, M. Grachten, and G. Widmer, “The match file format: Encoding alignments between scores and performances,” in *Proc. of the Music Encoding Conference*, (Halifax, Canada), May. 19-22, 2022.
- [273] E. Nakamura, K. Yoshii, and H. Katayose, “Performance error detection and post-processing for fast and accurate symbolic music alignment,” in *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, (Suzhou, China), pp. 347–353, Oct. 2017.
- [274] F. Foscarin, A. Mcleod, P. Rigaux, F. Jacquemard, and M. Sakai, “ASAP: a dataset of aligned scores and performances for piano transcription,” in *Proc. of the International Society for Music Information Retrieval (ISMIR)*, (Montreal / Virtual, Canada), Oct. 2020.
- [275] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, “Enabling factorized piano music modeling and generation with the MAESTRO dataset,” in *Proc. of the International Conference on Learning Representations (ICLR)*, (New Orleans, Louisiana, USA), OpenReview.net, May 2019.
- [276] Y. Pang, J. Lin, T. Qin, and Z. Chen, “Image-to-image translation: Methods and applications,” *IEEE Transactions on Multimedia*, vol. 24, pp. 3859–3881, 2022.
- [277] A. Wright, V. Välimäki, and L. Juvela, “Adversarial guitar amplifier modelling with unpaired data,” in *Proc. of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (Rhodes Island, Greece), IEEE, June. 4-10, 2023.
- [278] G. Brunner, Y. Wang, R. Wattenhofer, and S. Zhao, “Symbolic music genre transfer with CycleGAN,” in *International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 786–793, IEEE, 2018.
- [279] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. C. Courville, “Melgan: Generative adversarial networks for conditional waveform synthesis,” in *Advances in Neural Information Processing Systems (NIPS)* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), vol. 32, Curran Associates, Inc., 2019.
- [280] L. Liu, V. Morfi, and E. Benetos, “ACPAS: a dataset of aligned classical piano audio and scores for audio-to-score transcription,” in *Late-Breaking Demos of the International Society for Music Information Retrieval Conference (ISMIR)*, 2021.
- [281] J. Engel, L. H. Hantrakul, C. Gu, and A. Roberts, “Ddsp: Differentiable digital signal processing,” in *International Conference on Learning Representations*, July 2020.
- [282] X. Serra and J. Smith, “Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition,” *Computer Music Journal*, vol. 14, no. 4, pp. 12–24, 1990.





- [283] E. Cooper, X. Wang, and J. Yamagishi, “Text-to-Speech Synthesis Techniques for MIDI-to-Audio Synthesis,” in *Proc. 11th ISCA Speech Synthesis Workshop (SSW 11)*, pp. 130–135, 2021.
- [284] F. Rigaud, B. David, and L. Daudet, “A parametric model of piano tuning,” in *Proc. of the 14th Int. Conf. on Digital Audio Effects (DAFx-11)*, pp. 393–399, 2011.
- [285] H. Hahn and A. Roebel, “Joint F0 and Inharmonicity Estimation using Second Order Optimization,” in *SMC Sound and Music Computing Conference 2013*, (Stockholm, Sweden), pp. 695–700, July 2013.
- [286] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. C. Courville, “FiLM: Visual reasoning with a general conditioning layer,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 3942–3951, AAAI press, 9 2018.
- [287] C. Hawthorne, I. Simon, A. Roberts, N. Zeghidour, J. Gardner, E. Manilow, and J. Engel, “Multi-instrument music synthesis with spectrogram diffusion,” in *Proceedings of the International Society of Music Information Retrieval (ISMIR)*, (Bengaluru, India), pp. 337–344, December 2022.
- [288] S. Lee, H.-S. Choi, and K. Lee, “Differentiable artificial reverberation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 2541–2556, July 2022.
- [289] G. D. Santo, K. Prawda, S. Schlecht, and V. Välimäki, “Differentiable feedback delay network for colorless reverberation,” in *Proceedings of the International Conference on Digital Audio Effects (DAFx23)*, pp. 244–251, September 2023.
- [290] B. Bank and J. Chabassier, “Model-based digital pianos: From physics to sound synthesis,” *IEEE Signal Processing Magazine*, vol. 36, pp. 103–114, January 2019.
- [291] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, M. Sharifi, N. Zeghidour, and C. Frank, “Musiclm: Generating music from text,” 2023.
- [292] A. Gupta, P. Dollar, and R. Girshick, “Lvis: A dataset for large vocabulary instance segmentation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5356–5364, 2019.
- [293] L. Karlinsky, J. Shtok, S. Harary, E. Schwartz, A. Aides, R. Feris, R. Giryes, and A. M. Bronstein, “Repmet: Representative-based metric learning for classification and few-shot object detection,” in *Proc. CVPR*, 2019.
- [294] B. Singh, H. Li, A. Sharma, and L. S. Davis, “R-FCN-3000 at 30fps: Decoupling detection and classification,” in *Proc. CVPR*, 2018.
- [295] B. Kang, Z. Liu, X. Wang, F. Yu, J. Feng, and T. Darrell, “Few-shot object detection via feature reweighting,” in *Proc. ICCV*, 2019.
- [296] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in *Advances in neural information processing systems*, pp. 91–99, 2015.
- [297] X. Wang, T. Huang, J. Gonzalez, T. Darrell, and F. Yu, “Frustratingly simple few-shot object detection,” in *International Conference on Machine Learning*, pp. 9919–9928, PMLR, 2020.
- [298] G. Huang, I. Laradji, D. Vazquez, S. Lacoste-Julien, and P. Rodriguez, “A survey of self-supervised and few-shot object detection,” *arXiv preprint arXiv:2110.14711*, 2021.





- [299] H. Zhang, F. Chen, Z. Shen, Q. Hao, C. Zhu, and M. Savvides, “Solving missing-annotation object detection with background recalibration loss,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1888–1892, IEEE, 2020.
- [300] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.
- [301] Z. Wu, N. Bodla, B. Singh, M. Najibi, R. Chellappa, and L. S. Davis, “Soft sampling for robust object detection,” in *30th British Machine Vision Conference 2019, BMVC 2019, Cardiff, UK, September 9-12, 2019*, p. 225, BMVA Press, 2019.
- [302] J. H. Pollard and E. J. Valkovics, “The gompertz distribution and its applications,” *Genus*, pp. 15–28, 1992.
- [303] T. Wang, T. Yang, J. Cao, and X. Zhang, “Co-mining: Self-supervised learning for sparsely annotated object detection,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 2800–2808, 2021.
- [304] M. Xu, Z. Zhang, H. Hu, J. Wang, L. Wang, F. Wei, X. Bai, and Z. Liu, “End-to-end semi-supervised object detection with soft teacher,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3060–3069, 2021.
- [305] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li, “Fixmatch: Simplifying semi-supervised learning with consistency and confidence,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 596–608, 2020.
- [306] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *European conference on computer vision*, pp. 740–755, Springer, 2014.
- [307] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, “Greedy layer-wise training of deep networks,” in *Advances in neural information processing systems*, pp. 153–160, 2007.
- [308] H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin, “Exploring strategies for training deep neural networks.,” *Journal of machine learning research*, vol. 10, no. 1, 2009.
- [309] J. Weston, F. Ratle, H. Mobahi, and R. Collobert, “Deep learning via semi-supervised embedding,” in *Neural networks: Tricks of the trade*, pp. 639–655, Springer, 2012.
- [310] D. P. Kingma, S. Mohamed, D. Jimenez Rezende, and M. Welling, “Semi-supervised learning with deep generative models,” *Advances in neural information processing systems*, vol. 27, pp. 3581–3589, 2014.
- [311] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko, “Semi-supervised learning with ladder networks,” in *Advances in neural information processing systems*, pp. 3546–3554, 2015.
- [312] Y. Zhang, K. Lee, and H. Lee, “Augmenting supervised neural networks with unsupervised objectives for large-scale image classification,” in *International conference on machine learning*, pp. 612–621, 2016.
- [313] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *International conference on machine learning*, pp. 1597–1607, PMLR, 2020.
- [314] S. Haykin, *Neural networks and learning machines*. Pearson, 3 ed., 2009.





- [315] W. Gerstner and W. M. Kistler, *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.
- [316] R. C. O'Reilly and Y. Munakata, *Computational explorations in cognitive neuroscience: Understanding the mind by simulating the brain*. MIT press, 2000.
- [317] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.
- [318] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *CVPR 2011*, pp. 1521–1528, 2011.
- [319] V. S. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*, pp. 1324–1332, Curran Associates, Inc., 2010.
- [320] L. Ciampi, V. Zeni, L. Incrocci, A. Canale, G. Benelli, F. Falchi, G. Amato, and S. Chessa, "Pest Sticky Traps: a dataset for Whitefly Pest Population Density Estimation in Chromotropic Sticky Traps," Zenodo, Apr. 2023.
- [321] L. Ciampi, P. Foszner, N. Messina, M. Staniszewski, C. Gennaro, F. Falchi, G. Serao, M. Cogiell, D. Golba, A. Szczesna, and G. Amato, "Bus violence: An open benchmark for video violence detection on public transport," *Sensors*, vol. 22, p. 8345, oct 2022.
- [322] S. Akti, F. Ofli, M. Imran, and H. K. Ekenel, "Fight detection from still images in the wild," in *2022 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, IEEE, jan 2022.
- [323] S. Zini, A. Gomez-Villa, M. Buzzelli, B. Twardowski, A. D. Bagdanov, and J. van de Weijer, "Planckian jitter: countering the color-crippling effects of color jitter on self-supervised training," *arXiv preprint arXiv:2202.07993*, 2022.
- [324] B. Bosquet, D. Cores, L. Seidenari, V. M. Brea, M. Mucientes, and A. Del Bimbo, "A full data augmentation pipeline for small object detection based on generative adversarial networks," *Pattern Recognition*, vol. 133, p. 108998, 2023.
- [325] G. D. Finlayson and G. Schaefer, "Solving for colour constancy using a constrained dichromatic reflection model," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 127–144, 2001.
- [326] S. Tominaga, S. Ebisui, and B. A. Wandell, "Color temperature estimation of scene illumination," in *Color and Imaging Conference*, vol. 1999, pp. 42–47, Society for Imaging Science and Technology, 1999.
- [327] D. G. Andrews, *An introduction to atmospheric physics*. Cambridge University Press, 2010.
- [328] G. Wyszecki and W. S. Stiles, *Color science*, vol. 8. Wiley New York, 1982.
- [329] J. von Kries, "Theoretische studien über die umstimmung des sehorgans," *Festschrift der Albrecht-Ludwigs-Universität*, pp. 145–158, 1902.
- [330] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2014.





- [331] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 770–778, 2016.
- [332] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” in *Eur. Conf. Comput. Vis. (ECCV)*, pp. 630–645, 2016.
- [333] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, “Spectral normalization for generative adversarial networks,” in *ICLR*, 2018.
- [334] K. Turkowski, “Filters for common resampling tasks,” in *Graphics Gems*, pp. 147–165, 1990.
- [335] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “GANs trained by a two time-scale update rule converge to a local nash equilibrium,” in *Proc. of NIPS*, vol. 30, pp. 6629–6640, 2017.
- [336] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: Common objects in context,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.
- [337] H. Yu, G. Li, W. Zhang, Q. Huang, D. Du, Q. Tian, and N. Sebe, “The unmanned aerial vehicle benchmark: Object detection, tracking and baseline,” *International Journal of Computer Vision*, vol. 128, no. 5, pp. 1141–1159, 2020.
- [338] P. Zhu *et al.*, “VisDrone-VID2019: The vision meets drone object detection in video challenge results,” in *IEEE Int. Conf. Comput. Vis. Workshops*, 2019.
- [339] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2015.
- [340] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *CVPR*, 2016.
- [341] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, “Imagenet large scale visual recognition challenge,” *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
- [342] A. Bulat, J. Yang, and G. Tzimiropoulos, “To learn image super-resolution, use a gan to learn how to do image degradation first,” in *Eur. Conf. Comput. Vis. (ECCV)*, pp. 185–200, 2018.
- [343] A. Shocher, N. Cohen, and M. Irani, ““zero-shot” super-resolution using deep internal learning,” in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 3118–3126, 2018.
- [344] B. Bosquet, M. Mucientes, and V. M. Brea, “STDnet: Exploiting high resolution feature maps for small object detection,” *Eng. App. Artif. Intell.*, vol. 91, p. 103615, 2020.
- [345] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017.
- [346] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, “CenterNet: Keypoint triplets for object detection,” in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 6569–6578, 2019.
- [347] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, “Generative image inpainting with contextual attention,” in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 5505–5514, 2018.





- [348] M. Kisantal, Z. Wojna, J. Murawski, J. Naruniec, and K. Cho, “Augmentation for small object detection,” *arXiv preprint arXiv:1902.07296*, 2019.
- [349] H. Touvron, A. Sablayrolles, M. Douze, M. Cord, and H. Jégou, “Graftit: Learning fine-grained image representations with coarse labels,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 874–884, 2021.
- [350] Y. Xu, Q. Qian, H. Li, R. Jin, and J. Hu, “Weakly supervised representation learning with coarse labels,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10593–10601, 2021.
- [351] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, “Supervised contrastive learning,” *Advances in neural information processing systems*, vol. 33, pp. 18661–18673, 2020.
- [352] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese, “Deep metric learning via lifted structured feature embedding,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4004–4012, 2016.
- [353] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *ICML*, 2020.
- [354] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in *CVPR*, 2020.
- [355] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, “Emerging properties in self-supervised vision transformers,” in *ICCV*, 2021.
- [356] T. Han, W. Xie, and A. Zisserman, “Self-supervised co-training for video representation learning,” *NeurIPS*, 2020.
- [357] R. Qian, T. Meng, B. Gong, M.-H. Yang, H. Wang, S. Belongie, and Y. Cui, “Spatiotemporal contrastive video representation learning,” in *CVPR*, 2021.
- [358] G. Kordopatis-Zilos, S. Papadopoulos, I. Patras, and I. Kompatsiaris, “ViSiL: Fine-grained spatio-temporal video similarity learning,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019.
- [359] G. Kordopatis-Zilos, C. Tzelepis, S. Papadopoulos, I. Kompatsiaris, and I. Patras, “DnS: Distill-and-Select for Efficient and Accurate Video Indexing and Retrieval,” *IJCV*, 2022.
- [360] Y.-G. Jiang, Y. Jiang, and J. Wang, “VCDB: A large-scale database for partial copy detection in videos,” in *ECCV*, 2014.
- [361] G. Kordopatis-Zilos, S. Papadopoulos, I. Patras, and I. Kompatsiaris, “FIVR: Fine-grained Incident Video Retrieval,” *IEEE TMM*, 2019.
- [362] J. Revaud, M. Douze, C. Schmid, and H. Jégou, “Event retrieval in large video collections with circulant temporal encoding,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2459–2466, IEEE, 2013.
- [363] E. Pizzi, S. D. Roy, S. N. Ravindra, P. Goyal, and M. Douze, “A self-supervised descriptor for image copy detection,” in *CVPR*, 2022.



- [364] L. Baraldi, M. Douze, R. Cucchiara, and H. Jégou, “LAMV: Learning to align and match videos with kernelized temporal layers,” in *CVPR*, 2018.
- [365] C. Jiang, K. Huang, S. He, X. Yang, W. Zhang, X. Zhang, Y. Cheng, L. Yang, Q. Wang, F. Xu, *et al.*, “Learning segment similarity and alignment in large-scale content based video retrieval,” in *ACM MM*, 2021.
- [366] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, “Hierarchical attention networks for document classification,” in *nAC-ACL*, 2016.
- [367] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, “RandAugment: Practical automated data augmentation with a reduced search space,” in *CVPRW*, 2020.
- [368] A. v. d. Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” in *arXiv:1807.03748*, 2018.
- [369] A. Sablayrolles, M. Douze, C. Schmid, and H. Jégou, “Spreading vectors for similarity search,” in *ICLR*, 2018.
- [370] S. He, X. Yang, C. Jiang, G. Liang, W. Zhang, T. Pan, Q. Wang, F. Xu, C. Li, J. Liu, *et al.*, “A large-scale comprehensive dataset and copy-overlap aware evaluation protocol for segment-level video copy detection,” in *CVPR*, 2022.
- [371] G. Kordopatis-Zilos, S. Papadopoulos, I. Patras, and I. Kompatsiaris, “Near-duplicate video retrieval with deep metric learning,” in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 347–356, IEEE, 2017.
- [372] J. Shao, X. Wen, B. Zhao, and X. Xue, “Temporal context aggregation for video retrieval with contrastive learning,” 2021.
- [373] X. He, Y. Pan, M. Tang, Y. Lv, and Y. Peng, “Learn from unlabeled videos for near-duplicate video retrieval,” in *ACM SIGIR*, 2022.
- [374] S. S. Shapiro and M. B. Wilk, “An analysis of variance test for normality (complete samples),” *Biometrika*, vol. 52, no. 3/4, pp. 591–611, 1965.
- [375] A. Krizhevsky, “Learning multiple layers of features from tiny images,” 2009.
- [376] L. Deng, “The mnist database of handwritten digit images for machine learning research [best of the web],” *IEEE signal processing magazine*, vol. 29, no. 6, pp. 141–142, 2012.
- [377] Y. Tian, D. Krishnan, and P. Isola, “Contrastive representation distillation,” in *International Conference on Learning Representations, ICLR 2010*, 2019.
- [378] K. Q. Weinberger and L. K. Saul, “Distance metric learning for large margin nearest neighbor classification,” *Journal of machine learning research*, vol. 10, no. 2, 2009.
- [379] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, “Learning fine-grained image similarity with deep ranking,” *Proceedings of the IEEE conference on computer vision and pattern recognition*, vol. 1386–1393, 2014.
- [380] H. Lai, Y. Pan, Y. Liu, and S. Yan, “Simultaneous feature learning and hash coding with deep neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3270–3278, 2015.



- [381] A. Hermans, L. Beyer, and B. Leibe, “In defense of the triplet loss for person re-identification,” *arXiv preprint arXiv:1703.07737*, 2017.
- [382] O. Parkhi, A. Vedaldi, and A. Zisserman, “Deep face recognition,” in *BMVC 2015-Proceedings of the British Machine Vision Conference 2015*, British Machine Vision Association, 2015.
- [383] C. Yu, X. Zhao, Q. Zheng, P. Zhang, and X. You, “Hierarchical bilinear pooling for fine-grained visual recognition,” in *ECCV*, 2018.
- [384] G. Hinton, O. Vinyals, J. Dean, *et al.*, “Distilling the knowledge in a neural network,” *arXiv preprint arXiv:1503.02531*, vol. vol. 2, no. no. 7, 2015.
- [385] M.-E. Nilsback and A. Zisserman, “Automated flower classification over a large number of classes,” in *2008 Sixth Indian conference on computer vision, graphics & image processing*, pp. 722–729, IEEE, 2008.
- [386] J. Johnson, M. Douze, and H. Jégou, “Billion-scale similarity search with GPUs,” *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2019.
- [387] Y. Zheng, H. Yang, T. Zhang, J. Bao, D. Chen, Y. Huang, L. Yuan, D. Chen, M. Zeng, and F. Wen, “General facial representation learning in a visual-linguistic manner,” in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18676–18688, 2022.
- [388] Y. Liu, W. Wang, Y. Zhan, S. Feng, K. Liu, and Z. Chen, “Pose-disentangled contrastive learning for self-supervised facial representation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9717–9728, 2023.
- [389] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, *et al.*, “Bootstrap your own latent—a new approach to self-supervised learning,” *NeurIPS*, 2020.
- [390] B. Cheng, A. Schwing, and A. Kirillov, “Per-pixel classification is not all you need for semantic segmentation,” in *Advances in Neural Information Processing Systems* (M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, eds.), vol. 34, pp. 17864–17875, Curran Associates, Inc., 2021.
- [391] H. Li, N. Wang, X. Ding, X. Yang, and X. Gao, “Adaptively learning facial expression representation via c-f labels and distillation,” *IEEE Transactions on Image Processing*, vol. 30, pp. 2016–2028, 2021.
- [392] Y. Zhang, C. Wang, and W. Deng, “Relative uncertainty learning for facial expression recognition,” in *Advances in Neural Information Processing Systems* (M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, eds.), vol. 34, pp. 17616–17627, Curran Associates, Inc., 2021.
- [393] Y. Zhang, C. Wang, X. Ling, and W. Deng, “Learn from all: Erasing attention consistency for noisy label facial expression recognition,” in *Computer Vision – ECCV 2022* (S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, eds.), (Cham), pp. 418–434, Springer Nature Switzerland, 2022.
- [394] X. Zhang, T. Wang, X. Li, H. Yang, and L. Yin, “Weakly-supervised text-driven contrastive learning for facial behavior understanding,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20751–20762, October 2023.



- [395] J. Wang, H. Dai, T. Chen, H. Liu, X. Zhang, Q. Zhong, and R. Lu, “Toward surface defect detection in electronics manufacturing by an accurate and lightweight yolo-style object detector,” *Scientific Reports*, vol. 13, no. 1, p. 7062, 2023.
- [396] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, “Unsupervised learning of visual features by contrasting cluster assignments,” *NeurIPS*, 2020.
- [397] A. Bulat, S. Cheng, J. Yang, A. Garbett, E. Sanchez, and G. Tzimiropoulos, “Pre-training strategies and datasets for facial representation learning,” in *Computer Vision – ECCV 2022* (S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, eds.), (Cham), pp. 107–125, Springer Nature Switzerland, 2022.
- [398] L. Huang, S. You, M. Zheng, F. Wang, C. Qian, and T. Yamasaki, “Learning where to learn in cross-view self-supervised learning,” in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14431–14440, 2022.
- [399] J. Xia, W. Qu, W. Huang, J. Zhang, X. Wang, and M. Xu, “Sparse local patch transformer for robust face alignment and landmarks inherent relation learning,” in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4042–4051, 2022.
- [400] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVPR*, 2016.
- [401] H. Li, Z. Guo, S. Rhee, S. Han, and J.-J. Han, “Towards accurate facial landmark detection via cascaded transformers,” in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4166–4175, 2022.
- [402] J. Li, H. Jin, S. Liao, L. Shao, and P.-A. Heng, “Repformer: Refinement pyramid transformer for robust facial landmark detection,” in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22* (L. D. Raedt, ed.), pp. 1088–1094, International Joint Conferences on Artificial Intelligence Organization, 7 2022. Main Track.
- [403] Y. Huang, H. Yang, C. Li, J. Kim, and F. Wei, “Adnet: Leveraging error-bias towards normal direction in face alignment,” in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3060–3070, 2021.
- [404] J. Yang, Q. Liu, and K. Zhang, “Stacked hourglass network for robust facial landmark localisation,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2025–2033, 2017.
- [405] Z. Zhou, H. Li, H. Liu, N. Wang, G. Yu, and R. Ji, “Star loss: Reducing semantic ambiguity in facial landmark detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15475–15484, June 2023.
- [406] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” in *International Conference on Learning Representations*, 2021.
- [407] E. Barsoum, C. Zhang, C. C. Ferrer, and Z. Zhang, “Training deep networks for facial expression recognition with crowd-sourced label distribution,” in *Proceedings of the 18th ACM International Conference on Multimodal Interaction, ICMI '16*, (New York, NY, USA), p. 279–283, Association for Computing Machinery, 2016.





- [408] S. Li, W. Deng, and J. Du, “Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2584–2593, 2017.
- [409] W. Wu, C. Qian, S. Yang, Q. Wang, Y. Cai, and Q. Zhou, “Look at boundary: A boundary-aware face alignment algorithm,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2129–2138, 2018.
- [410] C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, “300 faces in-the-wild challenge: Database and results,” *Image and vision computing*, vol. 47, pp. 3–18, 2016.
- [411] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, “300 faces in-the-wild challenge: The first facial landmark localization challenge,” in *2013 IEEE International Conference on Computer Vision Workshops*, pp. 397–403, 2013.
- [412] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, “A semi-automatic methodology for facial landmark annotation,” in *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 896–903, 2013.
- [413] X. Chen, H. Fan, R. Girshick, and K. He, “Improved baselines with momentum contrastive learning,” *arXiv preprint arXiv:2003.04297*, 2020.
- [414] J. Li, P. Zhou, C. Xiong, and S. Hoi, “Prototypical contrastive learning of unsupervised representations,” in *International Conference on Learning Representations*, 2021.
- [415] Y. Guo, M. Xu, J. Li, B. Ni, X. Zhu, Z. Sun, and Y. Xu, “Hsc: Hierarchical contrastive selective coding,” in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9696–9705, 2022.
- [416] S. Gidaris, A. Bursuc, G. Puy, N. Komodakis, M. Cord, and P. Pérez, “Obow: Online bag-of-visual-words generation for self-supervised learning,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6830–6840, 2021.
- [417] M. Zheng, S. You, F. Wang, C. Qian, C. Zhang, X. Wang, and C. Xu, “Rssl: Relational self-supervised learning with weak augmentation,” in *Advances in Neural Information Processing Systems* (M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, eds.), vol. 34, pp. 2543–2555, Curran Associates, Inc., 2021.
- [418] M. Zheng, S. You, F. Wang, C. Qian, C. Zhang, X. Wang, and C. Xu, “Rssl: Relational self-supervised learning with weak augmentation,” *arXiv preprint arXiv:2107.09282*, 2021.
- [419] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, and A. Zisserman, “With a little help from my friends: Nearest-neighbor contrastive learning of visual representations,” *arXiv preprint arXiv:2104.14548*, 2021.
- [420] H. Alwassel, D. Mahajan, B. Korbar, L. Torresani, B. Ghanem, and D. Tran, “Self-supervised learning by cross-modal audio-video clustering,” in *Advances in Neural Information Processing Systems*, vol. 33, 2020.
- [421] X. Peng, Y. Wei, A. Deng, D. Wang, and D. Hu, “Balanced multimodal learning via on-the-fly gradient modulation,” in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8228–8237, 2022.



- [422] A. Ghosh, H. Kumar, and P. S. Sastry, “Robust loss functions under label noise for deep neural networks,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, 2017.
- [423] Z. Zhang and M. R. Sabuncu, “Generalized cross entropy loss for training deep neural networks with noisy labels,” *arXiv preprint arXiv:1805.07836*, 2018.
- [424] Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi, and J. Bailey, “Symmetric cross entropy for robust learning with noisy labels,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 322–330, 2019.
- [425] J. Goldberger and E. Ben-Reuven, “Training deep neural-networks using a noise adaptation layer,” 2016.
- [426] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama, “Co-teaching: Robust training of deep neural networks with extremely noisy labels,” *arXiv preprint arXiv:1804.06872*, 2018.
- [427] X. Yu, B. Han, J. Yao, G. Niu, I. Tsang, and M. Sugiyama, “How does disagreement help generalization against label corruption?,” in *International Conference on Machine Learning*, pp. 7164–7173, PMLR, 2019.
- [428] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, “Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels,” in *International Conference on Machine Learning*, pp. 2304–2313, PMLR, 2018.
- [429] E. Malach and S. Shalev-Shwartz, “Decoupling" when to update" from" how to update",” *arXiv preprint arXiv:1706.02613*, 2017.
- [430] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. Raffel, “Mixmatch: A holistic approach to semi-supervised learning,” *arXiv preprint arXiv:1905.02249*, 2019.
- [431] J. Li, R. Socher, and S. C. Hoi, “Dividemix: Learning with noisy labels as semi-supervised learning,” *arXiv preprint arXiv:2002.07394*, 2020.
- [432] D. Ortego, E. Arazo, P. Albert, N. E. O’Connor, and K. McGuinness, “Multi-objective interpolation training for robustness to label noise,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6606–6615, 2021.
- [433] R. Sachdeva, F. R. Cordeiro, V. Belagiannis, I. Reid, and G. Carneiro, “Evidentialmix: Learning with combined open-set and closed-set noisy labels,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3607–3615, 2021.
- [434] Z.-F. Wu, T. Wei, J. Jiang, C. Mao, M. Tang, and Y.-F. Li, “Ngc: A unified framework for learning with open-world noisy data,” *arXiv preprint arXiv:2108.11035*, 2021.
- [435] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang, “Learning from massive noisy labeled data for image classification,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2691–2699, 2015.
- [436] W. Li, L. Wang, W. Li, E. Agustsson, and L. Van Gool, “Webvision database: Visual learning and understanding from web data,” *arXiv preprint arXiv:1708.02862*, 2017.
- [437] H. Song, M. Kim, and J.-G. Lee, “Selfie: Refurbishing unclean samples for robust deep learning,” in *International Conference on Machine Learning*, pp. 5907–5915, PMLR, 2019.



- [438] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, and L. Qu, “Making deep neural networks robust to label noise: A loss correction approach,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1944–1952, 2017.
- [439] K. Yi and J. Wu, “Probabilistic end-to-end noise correction for learning with noisy labels,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7017–7025, 2019.
- [440] E. Arazo, D. Ortego, P. Albert, N. O’Connor, and K. McGuinness, “Unsupervised label noise modeling and loss correction,” in *International Conference on Machine Learning*, pp. 312–321, PMLR, 2019.
- [441] S. Liu, J. Niles-Weed, N. Razavian, and C. Fernandez-Granda, “Early-learning regularization prevents memorization of noisy labels,” *arXiv preprint arXiv:2007.00151*, 2020.
- [442] J. Li, C. Xiong, and S. Hoi, “Learning from noisy data with robust representation learning,” 2020.
- [443] K. Nishi, Y. Ding, A. Rich, and T. Hollerer, “Augmentation strategies for learning with noisy labels,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8022–8031, 2021.
- [444] E. Zheltonozhskii, C. Baskin, A. Mendelson, A. M. Bronstein, and O. Litany, “Contrast to divide: Self-supervised pre-training for learning with noisy labels,” *arXiv preprint arXiv:2103.13646*, 2021.
- [445] S. Arora, H. Khandeparkar, M. Khodak, O. Plevrakis, and N. Saunshi, “A theoretical analysis of contrastive unsupervised representation learning,” *arXiv preprint arXiv:1902.09229*, 2019.
- [446] J. Li, P. Zhou, C. Xiong, and S. C. Hoi, “Prototypical contrastive learning of unsupervised representations,” *arXiv preprint arXiv:2005.04966*, 2020.
- [447] C.-Y. Chuang, J. Robinson, L. Yen-Chen, A. Torralba, and S. Jegelka, “Debiased contrastive learning,” *arXiv preprint arXiv:2007.00224*, 2020.
- [448] T. Huynh, S. Kornblith, M. R. Walter, M. Maire, and M. Khademi, “Boosting contrastive self-supervised learning with false negative cancellation,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2785–2795, 2022.
- [449] T.-S. Chen, W.-C. Hung, H.-Y. Tseng, S.-Y. Chien, and M.-H. Yang, “Incremental false negative detection for contrastive learning,” *arXiv preprint arXiv:2106.03719*, 2021.
- [450] C. Wei, H. Wang, W. Shen, and A. Yuille, “Co2: Consistent contrast for unsupervised visual representation learning,” 2020.
- [451] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, “Deep clustering for unsupervised learning of visual features,” in *ECCV*, 2018.
- [452] Y. M. Asano, C. Rupprecht, and A. Vedaldi, “Self-labelling via simultaneous clustering and representation learning,” *arXiv preprint arXiv:1911.05371*, 2019.
- [453] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, “Unsupervised feature learning via non-parametric instance discrimination,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3733–3742, 2018.





- [454] C. Zhuang, A. L. Zhai, and D. Yamins, “Local aggregation for unsupervised learning of visual embeddings,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6002–6012, 2019.
- [455] X. Wang and G.-J. Qi, “Contrastive learning with stronger augmentations,” *arXiv preprint arXiv:2104.07713*, 2021.
- [456] P. Zhou, L. Du, and X. Li, “Self-paced consensus clustering with bipartite graph,” in *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pp. 2133–2139, 2021.
- [457] N. Dvornik, C. Schmid, and J. Mairal, “Diversity with cooperation: Ensemble methods for few-shot classification,” in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3723–3731, 2019.
- [458] Y. Tian, Y. Wang, D. Krishnan, J. B. Tenenbaum, and P. Isola, “Rethinking few-shot image classification: a good embedding is all you need?,” in *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pp. 266–282, Springer, 2020.
- [459] A. Afrasiyabi, H. Larochelle, J.-F. Lalonde, and C. Gagné, “Matching feature sets for few-shot image classification,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9014–9024, 2022.
- [460] Á. Casado-García and J. Heras, “Ensemble methods for object detection,” in *ECAI 2020*, pp. 2688–2695, IOS Press, 2020.
- [461] J. Lee, S.-K. Lee, and S.-I. Yang, “An ensemble method of cnn models for object detection,” in *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 898–901, IEEE, 2018.
- [462] M. Carranza-García, P. Lara-Benítez, J. García-Gutiérrez, and J. C. Riquelme, “Enhancing object detection for autonomous driving by optimizing anchor generation and addressing class imbalance,” *Neurocomputing*, vol. 449, pp. 229–244, 2021.
- [463] A. Bar, X. Wang, V. Kantorov, C. J. Reed, R. Herzig, G. Chechik, A. Rohrbach, T. Darrell, and A. Globerson, “Detreg: Unsupervised pretraining with region priors for object detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14605–14615, 2022.
- [464] F. Liu, X. Zhang, Z. Peng, Z. Guo, F. Wan, X. Ji, and Q. Ye, “Integrally migrating pre-trained transformer encoder-decoders for visual object detection,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6825–6834, 2023.
- [465] Y. Xiao, V. Lepetit, and R. Marlet, “Few-shot object detection and viewpoint estimation for objects in the wild,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3090–3106, 2022.
- [466] C. D. Manning, P. Raghavan, and H. Schütze, “Introduction to information retrieval,” *Cambridge, UK: Cambridge Univ. Press*, 2008.
- [467] Y. Li, Z. Miao, J. Wang, and Y. Zhang, “Nonlinear embedding neural codes for visual instance retrieval,” *Neurocomputing*, vol. 275, pp. 1275–1281, 2018.





- [468] D. K. Iakovidis, N. Pelekis, E. E. Kotsifakos, I. Kopanakis, H. Karanikas, and Y. Theodoridis, "A pattern similarity scheme for medical image retrieval," *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, no. 4, pp. 442–450, 2008.
- [469] R. Datta, J. Li, and J. Z. Wang, "Content-based image retrieval: approaches and trends of the new age," in *Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval*, pp. 253–262, 2005.
- [470] M. Alkhawlan, M. Elmogy, and H. El Bakry, "Text-based, content-based, and semantic-based image retrievals: a survey," *Int. J. Comput. Inf. Technol.*, vol. 4, no. 01, pp. 58–66, 2015.
- [471] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 12, pp. 1349–1380, 2000.
- [472] Y. Cao, C. Wang, L. Zhang, and L. Zhang, "Edgel index for large-scale sketch-based image search," in *CVPR 2011*, pp. 761–768, IEEE, 2011.
- [473] X.-Y. Wang, B.-B. Zhang, and H.-Y. Yang, "Content-based image retrieval by integrating color and texture features," *Multimedia tools and applications*, vol. 68, no. 3, pp. 545–569, 2014.
- [474] C. Wengert, M. Douze, and H. Jégou, "Bag-of-colors for improved image search," in *Proceedings of the 19th ACM international conference on Multimedia*, pp. 1437–1440, 2011.
- [475] T. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," *Advances in neural information processing systems*, vol. 11, 1998.
- [476] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2, pp. 1150–1157, Ieee, 1999.
- [477] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, pp. 91–110, 2004.
- [478] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1, pp. 886–893, Ieee, 2005.
- [479] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*, pp. 2564–2571, Ieee, 2011.
- [480] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *Lecture notes in computer science*, vol. 3951, pp. 404–417, 2006.
- [481] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern recognition*, vol. 29, no. 1, pp. 51–59, 1996.
- [482] A. Babenko and V. Lempitsky, "Aggregating local deep features for image retrieval," in *Proceedings of the IEEE international conference on computer vision*, pp. 1269–1277, 2015.
- [483] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part I 13*, pp. 584–599, Springer, 2014.
- [484] G. Tolias, R. Sivic, and H. Jégou, "Particular object retrieval with integral max-pooling of cnn activations," *arXiv preprint arXiv:1511.05879*, 2015.





- [485] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, “Large-scale image retrieval with attentive deep local features,” *ICCV*, 2018.
- [486] K. He, Y. Lu, and S. Sclaroff, “Local descriptors optimized for average precision,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 596–605, 2018.
- [487] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk, “Learning local feature descriptors with triplets and shallow convolutional neural networks.,” in *Bmvc*, vol. 1, p. 3, 2016.
- [488] Y. Ke and R. Sukthankar, “Pca-sift: A more distinctive representation for local image descriptors,” in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 2, pp. II–II, IEEE, 2004.
- [489] L. Jing and Y. Tian, “Self-supervised visual feature learning with deep neural networks: A survey,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 11, pp. 4037–4058, 2020.
- [490] A. Zhai and H.-Y. Wu, “Classification is a strong baseline for deep metric learning,” *arXiv preprint arXiv:1811.12649*, 2018.
- [491] H. Jun, B. Ko, Y. Kim, I. Kim, and J. Kim, “Combination of multiple global descriptors for image retrieval,” *arXiv:1903.10663*, 2020.
- [492] A. El-Nouby, N. Neverova, I. Laptev, and H. Jégou, “Training vision transformers for image retrieval,” *arXiv preprint arXiv:2102.05644*, 2021.
- [493] F. Radenović, G. Tolias, and O. Chum, “Fine-tuning cnn image retrieval with no human annotation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 7, pp. 1655–1668, 2018.
- [494] Y. Gu, C. Li, and J. Xie, “Attention-aware generalized mean pooling for image retrieval,” *arXiv:1811.00202*, 2019.
- [495] B. Cao, A. Araujo, and J. Sim, “Unifying deep local and global features for image search,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pp. 726–743, Springer, 2020.
- [496] O. Siméoni, Y. Avrithis, and O. Chum, “Local features and visual words emerge in activations,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11651–11660, 2019.
- [497] M. Teichmann, A. Araujo, M. Zhu, and J. Sim, “Detect-to-retrieve: Efficient regional aggregation for image search,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5109–5118, 2019.
- [498] M. Yang, D. He, M. Fan, B. Shi, X. Xue, F. Li, E. Ding, and J. Huang, “Dolg: Single-stage image retrieval with deep orthogonal fusion of local and global features,” in *Proceedings of the IEEE/CVF International conference on Computer Vision*, pp. 11772–11781, 2021.
- [499] X. An, J. Deng, K. Yang, J. Li, Z. Feng, J. Guo, J. Yang, and T. Liu, “Unicom: Universal and compact representation learning for image retrieval,” *arXiv preprint arXiv:2304.05884*, 2023.
- [500] S. Gkelios, Y. Boutalis, and S. A. Chatzichristofis, “Investigating the vision transformer model for image retrieval tasks,” in *2021 17th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pp. 367–373, IEEE, 2021.



- [501] C. H. Song, J. Yoon, S. Choi, and Y. Avrithis, “Boosting vision transformers for image retrieval,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 107–117, 2023.
- [502] S. Bai, P. Tang, P. H. Torr, and L. J. Latecki, “Re-ranking via metric fusion for object retrieval and person re-identification,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 740–749, 2019.
- [503] P. Indyk and R. Motwani, “Approximate nearest neighbors: towards removing the curse of dimensionality,” in *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pp. 604–613, 1998.
- [504] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, “Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 12, pp. 2916–2929, 2012.
- [505] H. Jegou, M. Douze, and C. Schmid, “Product quantization for nearest neighbor search,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 1, pp. 117–128, 2010.
- [506] M. Kaya and H. Ş. Bilge, “Deep metric learning: A survey,” *Symmetry*, vol. 11, no. 9, p. 1066, 2019.
- [507] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.
- [508] X. Jian, S. Cunzhao, Q. Chengzuo, C. Wang, and X. Baihua, “Unsupervised part-based weighting aggregation of deep convolutional features for image retrieval,” *AAAI*, 2018.
- [509] T.-T. Do, T. Hoang, D.-K. L. Tan, H. Le, T. V. Nguyen, and N.-M. Cheung, “From selective deep convolutional features to compact binary representations for image retrieval,” *ACM Trans. Multimedia Comput. Commun. Appl. (TOMM)*, 2019.
- [510] H. Jégou, M. Douze, C. Schmid, and P. Pérez, “Aggregating local descriptors into a compact image representation,” *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 3304–3311, 2010.
- [511] H. Fang, P. Xiong, L. Xu, and Y. Chen, “Clip2video: Mastering video-text retrieval via image clip,” *arXiv preprint arXiv:2106.11097*, 2021.
- [512] N. Messina, M. Stefanini, M. Cornia, L. Baraldi, F. Falchi, G. Amato, and R. Cucchiara, “Aladin: Distilling fine-grained alignment scores for efficient image-text matching and retrieval,” *arXiv preprint arXiv:2207.14757*, 2022.
- [513] H. Zhang, Y. Wang, F. Dayoub, and N. Sunderhauf, “VarifocalNet: An IoU-aware dense object detector,” in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, jun 2021.
- [514] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.
- [515] J. Van De Weijer, C. Schmid, J. Verbeek, and D. Larlus, “Learning color names for real-world applications,” *IEEE Transactions on Image Processing*, vol. 18, no. 7, pp. 1512–1523, 2009.
- [516] R. Benavente, M. Vanrell, and R. Baldrich, “Parametric fuzzy sets for automatic color naming,” *JOSA A*, vol. 25, no. 10, pp. 2582–2593, 2008.

- [517] J. Revaud, J. Almazan, R. Rezende, and C. de Souza, “Learning with average precision: Training image retrieval with a listwise loss,” in *International Conference on Computer Vision*, pp. 5106–5115, IEEE, 2019.
- [518] G. Amato, P. Bolettieri, F. Carrara, F. Debole, F. Falchi, C. Gennaro, L. Vadicamo, and C. Vairo, “The VISIONE video search system: exploiting off-the-shelf text search engines for large-scale video retrieval,” *Journal of Imaging*, vol. 7, no. 5, p. 76, 2021.
- [519] G. Amato, F. Carrara, F. Falchi, C. Gennaro, and L. Vadicamo, “Large-scale instance-level image retrieval,” *Information Processing & Management*, p. 102100, 2019.
- [520] F. Carrara, L. Vadicamo, C. Gennaro, and G. Amato, “Approximate Nearest Neighbor Search on Standard Search Engines,” in *Similarity Search and Applications* (T. Skopal, F. Falchi, J. Lokoč, M. L. Sapino, I. Bartolini, and M. Patella, eds.), (Cham), pp. 214–221, Springer International Publishing, 2022.
- [521] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, “Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2916–2929, 2012.
- [522] J. Song, Y. Yang, Z. Huang, H. T. Shen, and J. Luo, “Effective multiple feature hashing for large-scale near-duplicate video retrieval,” *IEEE TMM*, 2013.
- [523] L. Yuan, T. Wang, X. Zhang, F. E. Tay, Z. Jie, W. Liu, and J. Feng, “Central similarity quantization for efficient image and video retrieval,” in *CVPR*, 2020.
- [524] Y. Cai, L. Yang, W. Ping, F. Wang, T. Mei, X.-S. Hua, and S. Li, “Million-scale near-duplicate video retrieval system,” in *Proceedings of the ACM international conference on Multimedia*, ACM, 2011.
- [525] G. Kordopatis-Zilos, S. Papadopoulos, I. Patras, and I. Kompatsiaris, “Near-duplicate video retrieval by aggregating intermediate cnn layers,” in *Proceedings of the International Conference on Multimedia Modeling*, pp. 251–263, Springer, 2017.
- [526] D. Liang, L. Lin, R. Wang, J. Shao, C. Wang, and Y.-W. Chen, “Unsupervised teacher-student model for large-scale video retrieval,” in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019.
- [527] S. Poullot, S. Tsukatani, A. Phuong Nguyen, H. Jégou, and S. Satoh, “Temporal matching kernel with explicit feature maps,” in *ACM MM*, 2015.
- [528] H.-K. Tan, C.-W. Ngo, R. Hong, and T.-S. Chua, “Scalable detection of partial near-duplicate videos by visual-temporal consistency,” in *ACM MM*, 2009.
- [529] C.-L. Chou, H.-T. Chen, and S.-Y. Lee, “Pattern-based near-duplicate video retrieval and localization on web-scale videos,” *IEEE TMM*, 2015.
- [530] K.-H. Wang, C.-C. Cheng, Y.-L. Chen, Y. Song, and S.-H. Lai, “Attention-based deep metric learning for near-duplicate video retrieval,” in *ICPR*, 2021.
- [531] S. Liang and P. Wang, “An efficient hierarchical near-duplicate video detection algorithm based on deep semantic features,” in *Proceedings of the International Conference on Multimedia Modeling*, 2020.



- [532] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Association for Computational Linguistics, June 2019.
- [533] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “Roberta: A robustly optimized bert pretraining approach,” 2019.
- [534] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, “Xlnet: Generalized autoregressive pretraining for language understanding,” *Advances in neural information processing systems*, vol. 32, 2019.
- [535] G. Brauwerters and F. Frasincaer, “A survey on aspect-based sentiment classification,” *ACM Comput. Surv.*, vol. 55, nov 2022.
- [536] A. Nazir, Y. Rao, L. Wu, and L. Sun, “Issues and challenges of aspect-based sentiment analysis: a comprehensive survey,” *IEEE Transactions on Affective Computing*, 2020.
- [537] A. Balahur and M. Turchi, “Comparative experiments using supervised learning and machine translation for multilingual sentiment analysis,” *Computer Speech & Language*, vol. 28, no. 1, pp. 56–75, 2014.
- [538] V. Barriere and A. Balahur, “Improving sentiment analysis over non-English tweets using multilingual transformers and automatic translation for data-augmentation,” in *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 266–271, International Committee on Computational Linguistics, Dec. 2020.
- [539] K. Cortis and B. Davis, “A dataset of multidimensional and multilingual social opinions for malta’s annual government budget,” *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 15, pp. 971–981, May 2021.
- [540] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. Al-Smadi, M. Al-Ayyoub, Y. Zhao, B. Qin, O. De Clercq, V. Hoste, M. Apidianaki, X. Tannier, N. Loukachevitch, E. Kotelnikov, N. Bel, S. M. Jiménez-Zafra, and G. Eryigit, “SemEval-2016 task 5: Aspect based sentiment analysis,” in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pp. 19–30, Association for Computational Linguistics, June 2016.
- [541] A. Severyn, A. Moschitti, O. Uryupina, B. Plank, and K. Filippova, “Multi-lingual opinion mining on youtube,” *Information Processing & Management*, vol. 52, no. 1, pp. 46–60, 2016.
- [542] P. Nakov, A. Ritter, S. Rosenthal, F. Sebastiani, and V. Stoyanov, “SemEval-2016 task 4: Sentiment analysis in Twitter,” in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pp. 1–18, Association for Computational Linguistics, June 2016.
- [543] F. Hamborg and K. Donnay, “NewsMTSC: A dataset for (multi-)target-dependent sentiment classification in political news articles,” in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Association for Computational Linguistics, 2021.
- [544] A. Akbik, D. Blythe, and R. Vollgraf, “Contextual string embeddings for sequence labeling,” in *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 1638–1649, Association for Computational Linguistics, Aug. 2018.



- [545] L. Wu, F. Petroni, M. Josifoski, S. Riedel, and L. Zettlemoyer, “Scalable zero-shot entity linking with dense entity retrieval,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6397–6407, Association for Computational Linguistics, Nov. 2020.
- [546] J. L. Fleiss, “Measuring nominal scale agreement among many raters.,” *Psychological bulletin*, vol. 76, no. 5, p. 378, 1971.
- [547] J. Cao, R. Liu, H. Peng, L. Jiang, and X. Bai, “Aspect is not you need: No-aspect differential sentiment framework for aspect-based sentiment analysis,” in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1599–1609, Association for Computational Linguistics, July 2022.
- [548] R. Seoh, I. Birle, M. Tak, H.-S. Chang, B. Pinette, and A. Hough, “Open aspect target sentiment classification with natural language prompts,” in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, (Online and Punta Cana, Dominican Republic), pp. 6311–6322, Association for Computational Linguistics, Nov. 2021.
- [549] Y. Song, J. Wang, T. Jiang, Z. Liu, and Y. Rao, “Attentional encoder network for targeted sentiment classification,” *arXiv preprint arXiv:1902.09314*, 2019.
- [550] F. Stahlberg, “Neural machine translation: A review,” *Journal of Artificial Intelligence Research*, vol. 69, pp. 343–418, 2020.
- [551] S. M. Mohammad, M. Salameh, and S. Kiritchenko, “How translation alters sentiment,” *Journal of Artificial Intelligence Research*, vol. 55, pp. 95–130, 2016.
- [552] M. Koppel, S. Argamon, and A. R. Shimoni, “Automatically categorizing written texts by author gender,” *Literary and Linguistic Computing*, vol. 17, no. 4, pp. 401–412, 2002.
- [553] T. Gollub, M. Potthast, A. Beyer, M. Busse, F. M. Rangel Pardo, P. Rosso, E. Stamatatos, and B. Stein, “Recent trends in digital text forensics and its evaluation: Plagiarism detection, author identification, and author profiling,” in *Proceedings of the 4th International Conference of the CLEF Initiative (CLEF 2013)*, (Valencia, ES), pp. 282–302, 2013.
- [554] J. R. Tetreault, D. Blanchard, A. Cahill, and M. Chodorow, “Native tongues, lost and found: Resources and empirical evaluations in native language identification,” in *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, (Mumbai, IN), pp. 2585–2602, 2012.
- [555] S. Argamon, M. Koppel, J. W. Pennebaker, and J. Schler, “Automatically profiling the author of an anonymous text,” *Communications of the ACM*, vol. 52, no. 2, pp. 119–123, 2009.
- [556] E. Stamatatos, “Authorship verification: A review of recent advances,” *Research in Computing Science*, vol. 123, pp. 9–25, 2016.
- [557] M. Koppel, J. Schler, and S. Argamon, “Computational methods in authorship attribution,” *Journal of the American Society for Information Science and Technology*, vol. 60, no. 1, pp. 9–26, 2009.
- [558] E. Stamatatos, “A survey of modern authorship attribution methods,” *Journal of the American Society for Information Science and Technology*, vol. 60, no. 3, pp. 538–556, 2009.
- [559] M. Koppel and Y. Winter, “Determining if two documents are written by the same author,” *Journal of the Association for Information Science and Technology*, vol. 65, no. 1, pp. 178–187, 2014.





- [560] S. Corbara, A. Moreo, F. Sebastiani, and M. Tavoni, “The Epistle to Cangrande through the lens of computational authorship verification,” in *Proceedings of the 1st International Workshop on Pattern Recognition for Cultural Heritage (PatReCH 2019)*, (Trento, IT), pp. 148–158, 2019.
- [561] J. Kabala, “Computational authorship attribution in medieval Latin corpora: The case of the Monk of Lido (ca. 1101–08) and Gallus Anonymous (ca. 1113–17),” *Language Resources and Evaluation*, vol. 54, no. 1, pp. 25–56, 2020.
- [562] S. Larner, *Forensic authorship analysis and the World Wide Web*. Heidelberg, DE: Springer, 2014.
- [563] A. Rocha, W. J. Scheirer, C. W. Forstall, T. Cavalcante, A. Theophilo, B. Shen, A. Carvalho, and E. Stamatatos, “Authorship attribution for social media forensics,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 5–33, 2017.
- [564] S. Corbara, A. Moreo, and F. Sebastiani, “Same or different? Diff-vectors for authorship analysis,” *ACM Transactions on Knowledge Discovery from Data*, vol. 18, no. 1, p. Article 12, 2023.
- [565] M. Eder, “Style-markers in authorship attribution: A cross-language study of the authorial fingerprint,” *Studies in Polish Linguistics*, vol. 6, no. 1, pp. 99–114, 2011.
- [566] P. Juola, “Authorship attribution,” *Foundations and Trends in Information Retrieval*, vol. 1, no. 3, pp. 233–334, 2006.
- [567] M. Kestemont, M. Tschuggnall, E. Stamatatos, W. Daelemans, G. Specht, B. Stein, and M. Potthast, “Overview of the author identification task at PAN-2018: Cross-domain authorship attribution and style change detection,” in *Working Notes of the 2018 Conference and Labs of the Evaluation Forum (CLEF 2018)*, (Avignon, FR), pp. 1–25, 2018.
- [568] M. Kestemont, E. Stamatatos, E. Manjavacas, W. Daelemans, M. Potthast, and B. Stein, “Overview of the cross-domain authorship attribution task at PAN-2019,” in *Working Notes of the 2019 Conference and Labs of the Evaluation Forum (CLEF 2019)*, (Lugano, CH), pp. 1–15, 2019.
- [569] P. Li, J. Xie, Q. Wang, and W. Zuo, “Is second-order information helpful for large-scale visual recognition?,” in *ICCV*, 2017.
- [570] Y. Song, N. Sebe, and W. Wang, “Why approximate matrix square root outperforms accurate svd in global covariance pooling?,” in *ICCV*, 2021.
- [571] Z. Gao, Q. Wang, B. Zhang, Q. Hu, and P. Li, “Temporal-attentive covariance pooling networks for video recognition,” in *NeurIPS*, 2021.
- [572] L. Huang, D. Yang, B. Lang, and J. Deng, “Decorrelated batch normalization,” in *CVPR*, 2018.
- [573] L. Huang, Y. Zhou, L. Liu, F. Zhu, and L. Shao, “Group whitening: Balancing learning efficiency and representational capacity,” in *CVPR*, 2021.
- [574] Y. Song, N. Sebe, and W. Wang, “Fast differentiable matrix square root,” in *ICLR*, 2022.
- [575] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, “Universal style transfer via feature transforms,” in *NeurIPS*, 2017.
- [576] T.-Y. Chiu, “Understanding generalized whitening and coloring transform for universal style transfer,” in *ICCV*, 2019.





- [577] Z. Wang, L. Zhao, H. Chen, L. Qiu, Q. Mo, S. Lin, W. Xing, and D. Lu, “Diversified arbitrary style transfer via deep feature perturbation,” in *CVPR*, 2020.
- [578] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother, “Dsac-differentiable ransac for camera localization,” in *CVPR*, 2017.
- [579] D. Campbell, L. Liu, and S. Gould, “Solving the blind perspective-n-point problem end-to-end with robust differentiable geometric optimization,” in *ECCV*, 2020.
- [580] Z. Dang, K. M. Yi, Y. Hu, F. Wang, P. Fua, and M. Salzmann, “Eigendecomposition-free training of deep networks for linear least-square problems,” *TPAMI*, 2020.
- [581] N. J. Higham, *Functions of matrices: theory and computation*. SIAM, 2008.
- [582] W. Wang, Z. Dang, Y. Hu, P. Fua, and M. Salzmann, “Robust differentiable svd,” *TPAMI*, 2021.
- [583] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, “Efficient backprop,” in *Neural networks: Tricks of the trade*, pp. 9–48, Springer, 2012.
- [584] S. Wiesler and H. Ney, “A convergence analysis of log-linear training,” *NeurIPS*, 2011.
- [585] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, “On the importance of initialization and momentum in deep learning,” in *ICML*, 2013.
- [586] D. Mishkin and J. Matas, “All you need is a good init,” *ICLR*, 2016.
- [587] J. Wang, Y. Chen, R. Chakraborty, and S. X. Yu, “Orthogonal convolutional neural networks,” in *CVPR*, 2020.
- [588] S. Singla and S. Feizi, “Skew orthogonal convolutions,” in *ICML*, 2021.
- [589] Y. Song, N. Sebe, and W. Wang, “Improving covariance conditioning of the svd meta-layer by orthogonality,” in *ECCV*, 2022.
- [590] Z. He, M. Kan, and S. Shan, “Eigengan: Layer-wise eigen-learning for gans,” in *CVPR*, 2021.
- [591] V. Kazemi and J. Sullivan, “One millisecond face alignment with an ensemble of regression trees,” in *CVPR*, 2014.
- [592] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “Gans trained by a two time-scale update rule converge to a local nash equilibrium,” *NeurIPS*, 2017.
- [593] X. Zhu, C. Xu, and D. Tao, “Learning disentangled representations with latent variation predictability,” in *ECCV*, Springer, 2020.
- [594] Y. Song, N. Sebe, and W. Wang, “Orthogonal svd covariance conditioning and latent disentanglement,” *Journal of Field Robotics*, vol. 45, no. 7, pp. 8773–8786, 2023.
- [595] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *NeurIPS*, 2017.
- [596] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, “Transformers in vision: A survey,” *ACM computing surveys (CSUR)*, vol. 54, no. 10s, pp. 1–41, 2022.
- [597] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in *European conference on computer vision*, pp. 213–229, Springer, 2020.





- [598] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, “Deformable {detr}: Deformable transformers for end-to-end object detection,” in *International Conference on Learning Representations*, 2021.
- [599] L. Ye, M. Rochan, Z. Liu, and Y. Wang, “Cross-modal self-attention network for referring image segmentation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10502–10511, 2019.
- [600] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao, “Pre-trained image processing transformer,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [601] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, “Swinir: Image restoration using swin transformer,” *arXiv preprint arXiv:2108.10257*, 2021.
- [602] Y. Fang, B. Liao, X. Wang, J. Fang, J. Qi, R. Wu, J. Niu, and W. Liu, “You only look at one sequence: Rethinking transformer in vision through object detection,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 26183–26197, 2021.
- [603] S. Gidaris, P. Singh, and N. Komodakis, “Unsupervised representation learning by predicting image rotations,” in *International Conference on Learning Representations*, 2018.
- [604] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman, “Video action transformer network,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 244–253, 2019.
- [605] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin transformer: Hierarchical vision transformer using shifted windows,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.
- [606] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, “Pvt v2: Improved baselines with pyramid vision transformer,” *Computational Visual Media*, vol. 8, no. 3, pp. 415–424, 2022.
- [607] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, “Pyramid vision transformer: A versatile backbone for dense prediction without convolutions,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 568–578, 2021.
- [608] Y.-A. Wang and Y.-N. Chen, “What do position embeddings learn? an empirical study of pre-trained language model positional encoding,” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.
- [609] P. Dufter, M. Schmitt, and H. Schütze, “Position information in transformers: An overview,” *Computational Linguistics*, vol. 48, no. 3, pp. 733–763, 2022.
- [610] L. McInnes, J. Healy, and J. Melville, “Umap: Uniform manifold approximation and projection for dimension reduction,” *arXiv preprint arXiv:1802.03426*, 2018.
- [611] Y. Xu, Q. Zhang, J. Zhang, and D. Tao, “Vitae: Vision transformer advanced by exploring intrinsic inductive bias,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 28522–28535, 2021.
- [612] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li, “Fixmatch: Simplifying semi-supervised learning with consistency and confidence,” *Advances in neural information processing systems*, vol. 33, pp. 596–608, 2020.





- [613] Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le, “Unsupervised data augmentation for consistency training,” *Advances in neural information processing systems*, vol. 33, pp. 6256–6268, 2020.
- [614] S. B. Rangrej, C. L. Srinidhi, and J. J. Clark, “Consistency driven sequential transformers attention model for partially observable scenes,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2518–2527, 2022.
- [615] H. Bao, L. Dong, and F. Wei, “Beit: Bert pre-training of image transformers,” in *International Conference on Learning Representations (ICLR)*, 2022.
- [616] A. Hatamizadeh, H. Yin, H. Roth, W. Li, J. Kautz, D. Xu, and P. Molchanov, “Gradvit: Gradient inversion of vision transformers,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [617] H. Yin, A. Mallya, A. Vahdat, J. M. Alvarez, J. Kautz, and P. Molchanov, “See through gradients: Image batch recovery via gradinversion,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [618] D. Hendrycks and T. Dietterich, “Benchmarking neural network robustness to common corruptions and perturbations,” *arXiv preprint arXiv:1903.12261*, 2019.
- [619] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song, “Natural adversarial examples,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [620] B. Ren, Y. Liu, W. Bi, R. Cucchiara, N. Sebe, and W. Wang, “Masked jigsaw puzzle: A versatile position embedding for vision transformers,” in *CVPR*, 2023.
- [621] W. Ahmad, H. Ali, Z. Shah, and S. Azmat, “A new generative adversarial network for medical images super resolution,” *Scientific Reports*, vol. 12, no. 1, p. 9533, 2022.
- [622] Y. Chen, Y. Xie, Z. Zhou, F. Shi, A. G. Christodoulou, and D. Li, “Brain mri super resolution using 3d deep densely connected neural networks,” in *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*, pp. 739–742, IEEE, 2018.
- [623] A. Aakerberg, K. Nasrollahi, and T. B. Moeslund, “Real-world super-resolution of face-images from surveillance cameras,” *IET Image Processing*, vol. 16, no. 2, pp. 442–452, 2022.
- [624] S. P. Mudunuri and S. Biswas, “Low resolution face recognition across variations in pose and illumination,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 5, pp. 1034–1040, 2015.
- [625] D. Zhang, J. Shao, X. Li, and H. T. Shen, “Remote sensing image super-resolution via mixed high-order attention network,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 6, pp. 5183–5196, 2020.
- [626] J. Gu, X. Sun, Y. Zhang, K. Fu, and L. Wang, “Deep residual squeeze and excitation network for remote sensing image super-resolution,” *Remote Sensing*, vol. 11, no. 15, p. 1817, 2019.
- [627] NVIDIA, “Pixel perfect: Rtx video super resolution now available | nvidia blog.” <https://blogs.nvidia.com/blog/2023/02/28/rtx-video-super-resolution/>, 2023.
- [628] V. Vavilala and M. Meyer, “Deep learned super resolution for feature film production,” in *Special Interest Group on Computer Graphics and Interactive Techniques Conference Talks*, pp. 1–2, 2020.





- [629] SAMSUNG, “How to use the intelligent mode of samsung qled tv | samsung ca.” <https://www.samsung.com/ca/support/tv-audio-video/how-to-use-the-intelligent-mode-of-samsung-qled-tvs/>.
- [630] U. Cisco, “Cisco annual internet report (2018–2023) white paper,” *Cisco: San Jose, CA, USA*, vol. 10, no. 1, pp. 1–35, 2020.
- [631] D. Ma, F. Zhang, and D. Bull, “Bvi-dvc: a training database for deep video compression,” *IEEE Transactions on Multimedia*, 2021.
- [632] K. C. Chan, X. Wang, K. Yu, C. Dong, and C. C. Loy, “Basicvsr: The search for essential components in video super-resolution and beyond,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2021.
- [633] K. C. Chan, S. Zhou, X. Xu, and C. C. Loy, “Investigating tradeoffs in real-world video super-resolution,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2022.
- [634] J. Liang, Y. Fan, X. Xiang, R. Ranjan, E. Ilg, S. Green, J. Cao, K. Zhang, R. Timofte, and L. Van Gool, “Recurrent video restoration transformer with guided deformable attention,” *arXiv preprint arXiv:2206.02146*, 2022.
- [635] X. Wang, L. Xie, C. Dong, and Y. Shan, “Real-esrgan: Training real-world blind super-resolution with pure synthetic data,” in *International Conference on Computer Vision Workshops (ICCVW)*.
- [636] W. Lu, W. Sun, X. Min, W. Zhu, Q. Zhou, J. He, Q. Wang, Z. Zhang, T. Wang, and G. Zhai, “Deep neural network for blind visual quality assessment of 4k content,” *IEEE Transactions on Broadcasting*, vol. 69, no. 2, pp. 406–421, 2022.
- [637] W. Sun, H. Duan, X. Min, L. Chen, and G. Zhai, “Blind quality assessment for in-the-wild images via hierarchical feature fusion strategy,” in *2022 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pp. 01–06, IEEE, 2022.
- [638] V. Meshchaninov, I. Molodetskikh, and D. Vatolin, “Combining contrastive and supervised learning for video super-resolution detection,” *arXiv preprint arXiv:2205.10406*, 2022.
- [639] R. R. Shah, V. A. Akundy, and Z. Wang, “Real versus fake 4k-authentic resolution assessment,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2185–2189, IEEE, 2021.
- [640] Z. Yang, Y. Dong, L. Song, R. Xie, L. Li, and Y. Feng, “Native resolution detection for 4k-uhd videos,” in *2020 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pp. 1–5, IEEE, 2020.
- [641] Y. Han, J. Kim, and K. Lee, “Deep Convolutional Neural Networks for predominant instrument recognition in polyphonic music,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, pp. 208–221, 2017.
- [642] M. Taenzer, J. Abeßer, S. I. Mimitakis, C. Weiß, H. Lukashevich, and M. Müller, “Investigating CNN-based instrument family recognition for western classical music recordings,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, (Delft, The Netherlands), pp. 612–619, 2019.





- [643] D. Kostrzewa, P. Kaminski, and R. Brzeski, “Music genre classification: Looking for the perfect network,” in *Proc. of the 21st International Conference in Computational Science (ICCS)*, pp. 55–67, 2021.
- [644] S. Grollmisch and E. Cano, “Improving semi-supervised learning for audio classification with fixmatch,” *Electronics*, vol. 10, no. 15, p. 1807, 2021.
- [645] H. Zhao, C. Zhang, B. Zhu, Z. Ma, and K. Zhang, “S3T: Self-supervised pre-training with swin transformer for music classification,” in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 606–610, 2022.
- [646] Y. Gal and Z. Ghahramani, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning,” in *Proc. of International conference on machine learning (ICML)*, pp. 1050–1059, 2016.
- [647] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On calibration of modern neural networks,” in *Proc. of International conference on machine learning (ICML)*, pp. 1321–1330, 2017.
- [648] M. Hein, M. Andriushchenko, and J. Bitterwolf, “Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate the problem,” in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [649] R. P. Duin and D. M. Tax, “Classifier conditional posterior probabilities,” in *Proceedings of the Joint IAPR International Workshops SSPR’98 and SPR’98*, (Sydney, Australia), pp. 611–619, 1998.
- [650] B. Lakshminarayanan, A. Pritzel, and C. Blundell, “Simple and scalable predictive uncertainty estimation using deep ensembles,” *Advances in neural information processing systems*, vol. 30, 2017.
- [651] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. V. Dillon, B. Lakshminarayanan, and J. Snoek, “Can you trust your model’s uncertainty? Evaluating predictive uncertainty under dataset shift,” in *Proc. of the 33rd Conference on Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.
- [652] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, “FMA: A dataset for music analysis,” in *Proc. of the 18th International Society for Music Information Retrieval Conference*, 2017.
- [653] J. Engel, C. Resnick, A. Roberts, S. Dieleman, M. Norouzi, D. Eck, and K. Simonyan, “Neural audio synthesis of musical notes with WaveNet autoencoders,” in *Proceedings of the International Conference on Machine Learning (ICML)*, (Sydney, Australia), pp. 1068–1077, 2017.
- [654] A. Ramires and X. Serra, “Data augmentation for instrument classification robust to audio effects,” in *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, (Birmingham, United Kingdom), 2019.
- [655] J. Cramer, H.-H. Wu, J. Salamon, and J. P. Bello, “Look, listen, and learn more: Design choices for deep audio embeddings,” in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3852–3856, 2019.
- [656] H. Lukashevich, S. Grollmisch, J. Abefter, S. Stober, and J. Bös, “How reliable are posterior class probabilities in automatic music classification?,” in *Proceedings of the 18th International Audio Mostly Conference*, pp. 45–50, 2023.





- [657] Y. Wu, K. Chen, T. Zhang, Y. Hui, M. Nezhurina, T. Berg-Kirkpatrick, and S. Dubnov, “Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation,” 2024.
- [658] S. Doh, K. Choi, J. Lee, and J. Nam, “Lp-musiccaps: Llm-based pseudo music captioning,” 2023.
- [659] A. E. Gencer, T. Güngör, A. Gürer, and A. S. Özsoy, “Input-evaluation: A new mechanism for collecting data using games with a purpose,” in *2012 IEEE Symposium on Computers and Communications (ISCC)*, pp. 000239–000244, 2012.
- [660] “Hugging face.” <https://huggingface.co>.
- [661] I. Manco, B. Weck, S. Doh, M. Won, Y. Zhang, D. Bogdanov, Y. Wu, K. Chen, P. Tovstogan, E. Benetos, E. Quinton, G. Fazekas, and J. Nam, “The song describer dataset: a corpus of audio captions for music-and-language evaluation,” 2023.
- [662] Q. Huang, A. Jansen, J. Lee, R. Ganti, J. Y. Li, and D. P. W. Ellis, “Mulan: A joint embedding of music audio and natural language,” 2022.
- [663] M. Maksimović, P. Aichroth, and L. Cuccovillo, “Detection and localization of partial audio matches,” in *International Conference on Content-Based Multimedia Indexing (CBMI)*, pp. 1–6, 2018.
- [664] M. Maksimović, P. Aichroth, and L. Cuccovillo, “Detection and localization of partial audio matches in various application scenarios,” *Multimedia Tools and Applications*, vol. 80, no. 1, pp. 22619–22641, 2021.
- [665] Z. Dias, A. Rocha, and S. Goldenstein, “Image phylogeny by minimal spanning trees.,” *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 774–788, 2012.
- [666] S. Verde, S. Milani, P. Bestagini, and S. Tubaro, “Audio phylogenetic analysis using geometric transforms,” in *2017 IEEE Workshop on Information Forensics and Security (WIFS)*, pp. 1–6, 2017.
- [667] M. Gerhardt, L. Cuccovillo, and P. Aichroth, “Advancing audio phylogeny: A neural network approach for transformation detection,” in *2023 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–6, 2023.
- [668] M. Gerhardt and L. Cuccovillo, “IDMT audio phylogeny dataset,” 2023.
- [669] M. Del Fabro and L. Böszörményi, “State-of-the-art and future challenges in video scene detection: a survey,” *Multimedia Systems*, vol. 19, pp. 427–454, 2013.
- [670] L. Baraldi, C. Grana, and R. Cucchiara, “A deep siamese network for scene detection in broadcast videos,” in *Proceedings of the 23rd ACM International Conference on Multimedia*, p. 1199–1202, Association for Computing Machinery, 2015.
- [671] H. Ji, D. Hooshyar, K. Kim, and H. Lim, “A semantic-based video scene segmentation using a deep neural network,” *Journal of Information Science*, vol. 45, no. 6, pp. 833–844, 2019.
- [672] I. U. Haq, K. Muhammad, T. Hussain, S. Kwon, M. Sodanil, S. W. Baik, and M. Y. Lee, “Movie scene segmentation using object detection and set theory,” *International Journal of Distributed Sensor Networks*, vol. 15, no. 6, 2019.
- [673] A. Rao, L. Xu, Y. Xiong, G. Xu, Q. Huang, B. Zhou, and D. Lin, “A local-to-global approach to multi-modal movie scene segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.





- [674] S. Chen, X. Nie, D. Fan, D. Zhang, V. Bhat, and R. Hamid, “Shot contrastive self-supervised learning for scene boundary detection,” in *CVPR 2021*, 2021.
- [675] H. Wu, K. Chen, Y. Luo, R. Qiao, B. Ren, H. Liu, W. Xie, and L. Shen, “Scene consistency representation learning for video scene segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14021–14030, June 2022.
- [676] M. M. Islam and G. Bertasius, “Long movie clip classification with state-space video models,” in *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXV*, p. 87–104, Springer-Verlag, 2022.
- [677] I. Harrando and R. Troncy, “and cut!” exploring textual representations for media content segmentation and alignment,” in *DataTV-2021, 2nd International Workshop on Data-driven Personalisation of Television*, 2021.
- [678] M. A. Hearst, “Texttiling: segmenting text into multi-paragraph subtopic passages,” *Comput. Linguist.*, vol. 23, no. 1, p. 33–64, 1997.
- [679] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.
- [680] M. Riedl and C. Biemann, “Text Segmentation with Topic Models ,” *Journal for Language Technology and Computational Linguistics (JLCL)*, vol. 27, no. 47-69, pp. 13–24, 2012.
- [681] A. Solbiati, K. Heffernan, G. Damaskinos, S. Poddar, S. Modi, and J. Cali, “Unsupervised topic segmentation of meetings with bert embeddings. arxiv,” *arXiv preprint arXiv:2106.12978*, 2021.
- [682] A. Berhe, C. Barras, and C. Guinaudeau, “Video scene segmentation of tv series using multimodal neural features,” *Series - International Journal of TV Serial Narratives*, vol. 5, p. 59–68, Jan. 2019.
- [683] M. Bouyahi and Y. B. Ayed, “Video scenes segmentation based on multimodal genre prediction,” *Procedia Computer Science*, vol. 176, pp. 10–21, 2020. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 24th International Conference KES2020.
- [684] D. Rotman, Y. Yaroker, E. Amrani, U. Barzelay, and R. Ben-Ari, “Learnable optimal sequential grouping for video scene detection,” in *Proceedings of the 28th ACM International Conference on Multimedia*, MM ’20, p. 1958–1966, Association for Computing Machinery, 2020.
- [685] D. Beeferman, A. Berger, and J. Lafferty, “Statistical models for text segmentation,” *Machine learning*, vol. 34, pp. 177–210, 1999.
- [686] L. Pevzner and M. A. Hearst, “A critique and improvement of an evaluation metric for text segmentation,” *Computational Linguistics*, vol. 28, no. 1, pp. 19–36, 2002.
- [687] P. Billingsley, *Probability and Measure*. John Wiley and Sons, second ed., 1986.
- [688] AI@Meta, “Llama 3 model card,” 2024.
- [689] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, “Lora: Low-rank adaptation of large language models,” *arXiv preprint arXiv:2106.09685*, 2021.
- [690] *LSC ’23: Proceedings of the 6th Annual ACM Lifelog Search Challenge*, (New York, NY, USA), Association for Computing Machinery, 2023.



- [691] X. Yang, S. Wang, J. Dong, J. Dong, M. Wang, and T.-S. Chua, “Video moment retrieval with cross-modal neural architecture search,” *IEEE Transactions on Image Processing*, vol. 31, pp. 1204–1216, 2022.
- [692] S. K. Ramakrishnan, Z. Al-Halah, and K. Grauman, “Naq: Leveraging narrations as queries to supervise episodic memory,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6694–6703, 2023.
- [693] Z. Yang, W. Ping, Z. Liu, V. Korthikanti, W. Nie, D.-A. Huang, L. Fan, Z. Yu, S. Lan, B. Li, M. Shoeybi, M.-Y. Liu, Y. Zhu, B. Catanzaro, C. Xiao, and A. Anandkumar, “Re-ViLM: Retrieval-augmented visual language model for zero and few-shot image captioning,” in *Findings of the Association for Computational Linguistics: EMNLP 2023* (H. Bouamor, J. Pino, and K. Bali, eds.), (Singapore), pp. 11844–11857, Association for Computational Linguistics, Dec. 2023.
- [694] K. Schoeffmann, J. Lokoč, and W. Bailer, “10 years of video browser showdown,” in *Proceedings of the 2nd ACM International Conference on Multimedia in Asia, MMAsia ’20*, (New York, NY, USA), Association for Computing Machinery, 2021.
- [695] G. Awad, K. Curtis, A. A. Butt, J. Fiscus, A. Godil, Y. Lee, A. Delgado, E. Godard, L. Diduch, D. Gupta, D. D. Fushman, Y. Graham, and G. Quénot, “Trecvid 2023 - a series of evaluation tracks in video understanding,” in *Proceedings of TRECVID 2023*, NIST, USA, 2023.
- [696] H. Liu, C. Li, Q. Wu, and Y. J. Lee, “Visual instruction tuning,” in *Advances in Neural Information Processing Systems* (A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, eds.), vol. 36, pp. 34892–34916, Curran Associates, Inc., 2023.
- [697] J. Li, D. Li, C. Xiong, and S. Hoi, “BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation,” in *Proceedings of the International Conference on Machine Learning*, 2022.
- [698] J. Yu, Z. Wang, V. Vasudevan, L. Yeung, M. Seyedhosseini, and Y. Wu, “Coca: Contrastive captioners are image-text foundation models,” *Transactions on Machine Learning Research*, 2022.
- [699] W. Kuo, A. Piergiovanni, D. Kim, xiyang luo, B. Caine, W. Li, A. Ogale, L. Zhou, A. M. Dai, Z. Chen, C. Cui, and A. Angelova, “MaMMUT: A simple architecture for joint learning for multimodal tasks,” *Transactions on Machine Learning Research*, 2023.
- [700] C. Team, “Chameleon: Mixed-modal early-fusion foundation models,” 2024.
- [701] J. Lei, L. Yu, M. Bansal, and T. Berg, “TVQA: Localized, compositional video question answering,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, (Brussels, Belgium), pp. 1369–1379, Association for Computational Linguistics, Oct.-Nov. 2018.
- [702] J. Lei, L. Yu, T. Berg, and M. Bansal, “TVQA+: Spatio-temporal grounding for video question answering,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, (Online), pp. 8211–8225, Association for Computational Linguistics, July 2020.
- [703] L. Li, Y.-C. Chen, Y. Cheng, Z. Gan, L. Yu, and J. Liu, “HERO: Hierarchical encoder for Video+Language omni-representation pre-training,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Online), pp. 2046–2065, Association for Computational Linguistics, Nov. 2020.

- [704] J. Xiao, X. Shang, A. Yao, and T.-S. Chua, “Next-qa: Next phase of question-answering to explaining temporal actions,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9777–9786, June 2021.
- [705] M. Grunde-McLaughlin, R. Krishna, and M. Agrawala, “Agqa: A benchmark for compositional spatio-temporal reasoning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.
- [706] B. Wu, S. Yu, Z. Chen, J. B. Tenenbaum, and C. Gan, “STAR: A benchmark for situated reasoning in real-world videos,” in *Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS)*, 2021.
- [707] B. Lin, B. Zhu, Y. Ye, M. Ning, P. Jin, and L. Yuan, “Video-llava: Learning united visual representation by alignment before projection,” *arXiv preprint arXiv:2311.10122*, 2023.
- [708] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom, “Llama 2: Open foundation and fine-tuned chat models,” 2023.
- [709] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millicah, M. Reynolds, R. Ring, E. Rutherford, S. Cabi, T. Han, Z. Gong, S. Samangooei, M. Monteiro, J. Menick, S. Borgeaud, A. Brock, A. Nematzadeh, S. Sharifzadeh, M. Binkowski, R. Barreira, O. Vinyals, A. Zisserman, and K. Simonyan, “Flamingo: a visual language model for few-shot learning,” in *Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS ’22*, (Red Hook, NY, USA), Curran Associates Inc., 2024.
- [710] J. Li, D. Li, C. Xiong, and S. Hoi, “BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation,” in *Proceedings of the 39th International Conference on Machine Learning* (K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, eds.), vol. 162 of *Proceedings of Machine Learning Research*, pp. 12888–12900, PMLR, 17–23 Jul 2022.
- [711] J. Li, D. Li, S. Savarese, and S. Hoi, “Blip-2: bootstrapping language-image pre-training with frozen image encoders and large language models,” in *Proceedings of the 40th International Conference on Machine Learning, ICML’23*, JMLR.org, 2023.
- [712] B. Zhu, B. Lin, M. Ning, Y. Yan, J. Cui, H. Wang, Y. Pang, W. Jiang, J. Zhang, Z. Li, *et al.*, “Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment,” *arXiv preprint arXiv:2310.01852*, 2023.
- [713] M. Maaz, H. Rasheed, S. Khan, and F. S. Khan, “Video-chatgpt: Towards detailed video understanding via large vision and language models,” in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024)*, 2024.
- [714] M. Maaz, H. Rasheed, S. Khan, and F. S. Khan, “Videogpt+: Integrating image and video encoders for enhanced video understanding,” *arxiv*, 2024.

- [715] A. Miech, D. Zhukov, J.-B. Alayrac, M. Tapaswi, I. Laptev, and J. Sivic, “HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips,” in *ICCV*, 2019.
- [716] S. Venkataramanan, M. N. Rizve, J. Carreira, Y. M. Asano, and Y. Avrithis, “Is imagenet worth 1 video? learning strong image encoders from 1 long unlabelled video,” in *The Twelfth International Conference on Learning Representations*, 2024.
- [717] J. Fleiss, *Statistical methods for rates and proportions Rates and proportions*. Wiley, 1973.
- [718] R. Mokady, A. Hertz, and A. H. Bermano, “ClipCap: Clip prefix for image captioning,” *arXiv preprint arXiv:2111.09734*, 2021.
- [719] M. Cherti, R. Beaumont, R. Wightman, M. Wortsman, G. Ilharco, C. Gordon, C. Schuhmann, L. Schmidt, and J. Jitsev, “Reproducible scaling laws for contrastive language-image learning,” 2022.
- [720] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, “Sigmoid loss for language image pre-training,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11975–11986, October 2023.
- [721] J. Porter, “Chatgpt continues to be one of the fastest-growing services ever.” <https://www.theverge.com/2023/11/6/23948386/chatgpt-active-user-count-openai-developer-conference>, 2023. Accessed 12-Dec-2023.
- [722] B. C. Stahl and D. Eke, “The ethics of chatgpt – exploring the ethical issues of an emerging technology,” *International Journal of Information Management*, vol. 74, 2024.
- [723] J. Vincent, “Getty images is suing the creators of ai art tool stable diffusion for scraping its content.” <https://www.theverge.com/2023/1/17/23558516/ai-art-copyright-stable-diffusion-getty-images-lawsuit>, 2023. Accessed 12-Dec-2023.
- [724] J. McCormack, T. Gifford, and P. Hutchings, “Autonomy, authenticity, authorship and intention in computer generated art,” in *Proceedings of the international conference on Computational Intelligence in Music, Sound, Art and Design*, 2019.
- [725] A. Ridler, “Making sense of it all,” *DAMN Magazine*, 2019.
- [726] “Midjourney.” <https://www.midjourney.com/>. Accessed 12-Dec-2023.
- [727] “Nmkd stable diffusion gui.” <https://nmkd.itch.io/t2i-gui>. Accessed 12-Dec-2023.
- [728] “Stability.ai dreamstudio available at.” <https://dreamstudio.ai/>. Accessed 12-Dec-2023.
- [729] “Open ai api.” <https://platform.openai.com/>. Accessed 12-Dec-2023.
- [730] Pixlr, “Stable diffusion and pixlr: Unlocking the power of image editing.” <https://blog.pixlr.com/stable-diffusion-and-pixlr-unlocking-the-power-of-image-editing/>, 2022. Accessed 12-Dec-2023.
- [731] A. Liapis, G. N. Yannakakis, M. J. Nelson, M. Preuss, and R. Bidarra, “Orchestrating game generation,” *IEEE Transactions on Games*, vol. 11, no. 1, pp. 48–68, 2019.
- [732] A. Tam, “A gentle introduction to hallucinations in large language models.” <https://machinelearningmastery.com/a-gentle-introduction-to-hallucinations-in-large> 2023. Accessed 12-Dec-2023.

- [733] A. Liapis, “Artificial intelligence for designing games,” in *The Handbook of Artificial Intelligence and the Arts*, Springer, 2021.
- [734] J. Gwertzman and J. Soslow, “The generative ai revolution in games.” <https://a16z.com/the-generative-ai-revolution-in-games/>, 2022. Accessed 12-Dec-2023.
- [735] S. Sudhakaran, M. González-Duque, C. Glanois, M. Freiberger, E. Najarro, and S. Risi, “MarioGPT: Open-ended text2level generation through large language models,” *arXiv*, 2023.
- [736] T. Merino, R. Negri, D. Rajesh, M. Charity, and J. Togelius, “The five-dollar model: Generating game maps and sprites from sentence embeddings,” in *Proceedings of AIIDE*, 2023.
- [737] J. Zhu, A. Liapis, S. Risi, R. Bidarra, and G. M. Youngblood, “Explainable ai for designers: A human-centered perspective on mixed-initiative co-creation,” in *Proceedings of the IEEE Conference on Computational Intelligence and Games*, 2018.
- [738] E. Buonanno, “Functional error handling,” in *Functional programming in C# (Second edition)*, Manning Publications Co., 2022.
- [739] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Proceedings of the Neural Information Processing Systems International Conference*, 2017.
- [740] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language models are unsupervised multitask learners.” <https://openai.com/research/better-language-models>, 2019. Accessed 27 Feb 2024.
- [741] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, “Language models are few-shot learners,” *arXiv preprint arXiv:2005.14165*, 2020.
- [742] OpenAI, “GPT-4 technical report,” *arXiv preprint arXiv:2303.08774*, 2023.
- [743] A. Madotto, Z. Liu, Z. Lin, and P. Fung, “Language models as few-shot learner for task-oriented dialogue systems,” *arXiv preprint arXiv:2008.06239*, 2020.
- [744] S. Yu, J. Liu, J. Yang, C. Xiong, P. Bennett, J. Gao, and Z. Liu, “Few-shot generative conversational query rewriting,” in *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020.
- [745] H. Jiang, L. Ge, Y. Gao, J. Wang, and R. Song, “Large language model for causal decision making,” *arXiv preprint arXiv:2312.17122*, 2023.
- [746] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, “Chain-of-thought prompting elicits reasoning in large language models,” *arXiv preprint arXiv:2201.11903*, 2023.
- [747] M. Turpin, J. Michael, E. Perez, and S. R. Bowman, “Language models don’t always say what they think: Unfaithful explanations in chain-of-thought prompting,” *arXiv preprint arXiv:2305.04388*, 2023.

- [748] S. Frieder, L. Pinchetti, A. Chevalier, R.-R. Griffiths, T. Salvatori, T. Lukasiewicz, P. C. Petersen, and J. Berner, “Mathematical capabilities of ChatGPT,” *arXiv preprint arXiv:2301.13867*, 2023.
- [749] Z. Li, Z. Z. Chen, M. Ross, P. Huber, S. Moon, Z. Lin, X. L. Dong, A. Sagar, X. Yan, and P. A. Crook, “Large language models as zero-shot dialogue state tracker through function calling,” *arXiv preprint arXiv:2402.10466*, 2024.
- [750] K. Pelrine, M. Taufeeque, M. Zajac, E. McLean, and A. Gleave, “Exploiting novel GPT-4 apis,” *arXiv preprint arXiv:2312.14302*, 2023.
- [751] T. Cai, X. Wang, T. Ma, X. Chen, and D. Zhou, “Large language models as tool makers,” *arXiv preprint arXiv:2305.17126*, 2023.
- [752] OpenAI, “Function calling.” <https://platform.openai.com/docs/guides/function-calling>, 2023. Accessed 27 Feb 2024.
- [753] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, “GLUE: A multi-task benchmark and analysis platform for natural language understanding,” *arXiv preprint arXiv:1804.07461*, 2019.
- [754] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, “SQuAD: 100,000+ questions for machine comprehension of text,” *arXiv preprint arXiv:1606.05250*, 2016.
- [755] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, “A large annotated corpus for learning natural language inference,” in *Proceedings of the Empirical Methods in Natural Language Processing Conference*, 2015.
- [756] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao, “Large language models: A survey,” *arXiv preprint arXiv:2402.06196*, 2024.
- [757] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, “Deep unsupervised learning using nonequilibrium thermodynamics,” in *Proceedings of the International Conference on Machine Learning*, 2015.
- [758] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.
- [759] P. Dhariwal and A. Nichol, “Diffusion models beat gans on image synthesis,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 8780–8794, 2021.
- [760] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” *arXiv preprint arXiv:2112.10752*, 2022.
- [761] L. Zhang, A. Rao, and M. Agrawala, “Adding conditional control to text-to-image diffusion models,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2023.
- [762] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, “Lora: Low-rank adaptation of large language models,” *arXiv preprint arXiv:2106.09685*, 2021.
- [763] A. Liapis, G. N. Yannakakis, and J. Togelius, “Computational game creativity,” in *Proceedings of the Innovative Computing and Cloud Computing International Conference*, 2014.
- [764] E. J. Hastings, R. K. Guha, and K. O. Stanley, “Evolving content in the Galactic Arms Race video game,” in *Proceedings of the IEEE Symposium on Computational Intelligence and Games*, 2009.

- [765] A. Zook and M. O. Riedl, "Automatic game design via mechanic generation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 28, no. 1, 2019.
- [766] C. Browne and F. Maire, "Evolutionary game design," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 2, no. 1, 2010.
- [767] M. Guzdial and M. Riedl, "Automated game design via conceptual expansion," *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 14, no. 1, 2018.
- [768] M. J. Nelson and M. Mateas, "Towards automated game design," in *Proceedings of the AI*IA Conference on Artificial Intelligence and Human-Oriented Computing* (R. Basili and M. T. Pazienza, eds.), 2007.
- [769] M. Cook and S. Colton, "Ludus ex machina: Building a 3D game designer that competes alongside humans," in *Proceedings of the Innovative Computing and Cloud Computing International Conference*, 2014.
- [770] A. Liapis, G. Smith, and N. Shaker, *Mixed-initiative content creation*, ch. 11, pp. 195–214. Springer International Publishing, 2016.
- [771] E. Butler, A. M. Smith, Y.-E. Liu, and Z. Popovic, "A mixed-initiative tool for designing level progressions in games," in *Proceedings of the ACM Symposium on User Interface Software and Technology*, 2013.
- [772] A. Liapis, G. N. Yannakakis, and J. Togelius, "Sentient sketchbook: Computer-aided game level authoring," in *Proceedings of the Foundations of Digital Games Conference*, 2013.
- [773] G. Smith, J. Whitehead, and M. Mateas, "Tanagra: a mixed-initiative level design tool," in *Proceedings of the Foundations of Digital Games International Conference*, 2010.
- [774] A. Summerville and M. Mateas, "Mystical tutor: A magic: The gathering design assistant via denoising sequence-to-sequence learning," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2021.
- [775] R. van Rozen, "A pattern-based game mechanics design assistant," in *Proceedings of the Foundations of Digital Games International Conference*, 2015.
- [776] P. L. Lanzi and D. Loiacono, "ChatGPT and other large language models as evolutionary engines for online interactive collaborative game design," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2023.
- [777] Red Hook Studios, "Darkest Dungeon," 2016.
- [778] Pygame Community, "Pygame." <https://www.pygame.org>, 2000. Accessed 01 Jun 2024.
- [779] K. Brown, "Grammatical design," *IEEE Expert*, vol. 12, no. 2, 1997.
- [780] N. Chomsky, "Three models for the description of language," *IRE Transactions on Information Theory*, vol. 2, no. 3, 1956.
- [781] K. T. Taraldsen, "Generative grammar." "<https://oxfordre.com/linguistics/view/10.1093/acrefore/9780199384655.001.0001/acrefore-9780199384655-e-49>", 2016. Accessed 27 Feb 2024.



- [782] R. Gallota, K. Arulkumaran, and L. B. Soros, “Preference-learning emitters for mixed-initiative quality-diversity algorithms,” *IEEE Transactions on Games*, 2023.
- [783] D. Karavolos, A. Bouwer, and R. Bidarra, “Mixed-initiative design of game levels: Integrating mission and space into level generation,” in *Proceedings of the Foundations of Digital Games International Conference*, 2015.
- [784] G. Hermans, T. Winters, and L. D. Raedt, “Shape inference and grammar induction for example-based procedural generation,” *arXiv preprint arXiv:2109.10217*, 2021.
- [785] A. Madkour, S. Marsella, C. Hartevelde, M. S. El-Nasr, and J.-W. van de Meent, “Guiding generative graph grammars of dungeon mission graphs via examples,” in *Proceedings of the AIIDE Workshop on Experimental AI in Games*, 2021.
- [786] T. Smith, J. Padget, and A. Vidler, “Graph-based generation of action-adventure dungeon levels using answer set programming,” in *Proceedings of the Foundations of Digital Games International Conference*, 2018.
- [787] E. Buoanno, “Functional error handling,” in *Functional Programming in C#*, ch. 6, Manning Publications Co., 2017.
- [788] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.
- [789] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2018.
- [790] A. Mittal, A. K. Moorthy, and A. C. Bovik, “Blind/referenceless image spatial quality evaluator,” in *Proceedings of the Asilomar Conference on Signals, Systems and Computers*, 2011.
- [791] S. Kastyulin, J. Zakirov, D. Prokopenko, and D. V. Dylov, “Pytorch image quality: Metrics for image quality assessment,” *arXiv preprint arXiv:2208.14818*, 2022.
- [792] D. Hasler and S. E. Suesstrunk, “Measuring colorfulness in natural images,” in *Proceedings of the SPIE 5007*, 2003.
- [793] S. Xie and Z. Tu, “Holistically-nested edge detection,” in *Proceedings of IEEE International Conference on Computer Vision*, 2015.
- [794] F. Wilcoxon, “Individual comparisons by ranking methods,” *Biometrics Bulletin*, 1945.
- [795] T. Galanos, A. Liapis, and G. N. Yannakakis, “Affectgan: Affect-based generative art driven by semantics,” in *Proceedings of the ACII Workshop on What’s Next in Affect Modeling?*, 2021.
- [796] Z. Lin, U. Ehsan, R. Agarwal, S. Dani, V. Vashishth, and M. Riedl, “Beyond prompts: Exploring the design space of mixed-initiative co-creativity systems,” in *Proceedings of the International Conference on Computational Creativity*, 2023.
- [797] S. Snodgrass, O. Mohaddesi, J. Hart, G. R. Rodriguez, C. Holmgård, and C. Hartevelde, “Like PEAS in PoDS: the player, environment, agents, system framework for the personalization of digital systems,” in *Proceedings of the Foundations of Digital Games International Conference*, 2019.



- [798] M. Zohaib, “Dynamic difficulty adjustment (DDA) in computer games: A review,” *Advances in Human-Computer Interaction*, vol. 2018, 2018.
- [799] V. Vimpari, A. Kultima, P. Hämäläinen, and C. Guckelsberger, ““An adapt-or-die type of situation”: Perception, adoption, and use of text-to-image-generation ai by game industry professionals,” *Proceeding of the ACM Human-Computer Interaction*, vol. 7, no. CHI PLAY, 2023.

