

D5.1

Initial report on Multimedia Summarisation and Analysis

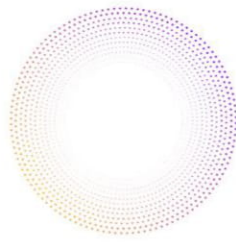
Project Title	AI4Media - A European Excellence Centre for Media, Society and Democracy
Contract No.	951911
Instrument	Research and Innovation Action
Thematic Priority	H2020-EU.2.1.1. - INDUSTRIAL LEADERSHIP - Leadership in enabling and industrial technologies - Information and Communication Technologies (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	Initial report on Multimedia Summarisation and Analysis
Deliverable number	D5.1
Deliverable version	1.0
Previous version(s)	N/A
Contractual date of delivery	August 31 2021
Actual date of delivery	August 24, 2021
Deliverable filename	AI4Media_D5.1-final.pdf
Nature of deliverable	Report
Dissemination level	Public
Number of pages	103
Work Package	WP5
Task(s)	T5.1, T5.3, T5.6
Partner responsible	AUTH
Author(s)	Michail Kaseris, Ioannis Pitas (AUTH), Alberto Messina, Maurizio Montagnuolo (RAI) Giuseppe Amato, Fabrizio Falchi, Lucia Vadicamo (CNR), Mihai Dogariu (UPB), Nicu Sebe, Kasim Sinan Yildirim (UNITN), Ioannis Patrassas, Ioannis Maniadias Metaxas (QMUL), Lucile Sassatelli, Frédéric Precioso, Fabien Gandon (3IA-UCA), Rémi Mignot, Lenny Renault (IRCAM), Vasileios Mezaris (CERTH), Jakob Abesser, Milica Gerhardt, Hanna Lukashevich, Sebastian RIBECKY, Patrick Aichroth (FHG-IDMT), Hannes Fassold (JR), Werner Bailer (JR), Michael Loidl (JR), Federico Pernici (UNIFI).
Editor	Ioannis Pitas (AUTH)
Officer	Evangelia Markidou

Abstract	This document reports on initial research performed on WP5 of AI4Media that concerns summarisation and analysis of multimedia content. Thus, the document sums up relevant research by the contributing partners in tasks T5.1, T5.3, and T5.6 for the period M1-M12 of the project. For each task, the relevant contributions are presented, along with relevant publications, links to software and plans for AI4EU integration (if available). The document concludes with a short presentation of future research directions.
Keywords	video analysis, video summarisation, key-frame extraction, information retrieval, symbolic reasoning, deep learning, learning from scarce data, data-efficient learning, few-shot learning, domain adaptation, semi-supervised learning, clustering, dictionary learning, music similarity analysis, music mixes generation, audio provenance analysis, audio phylogeny analysis



Copyright

© Copyright 2021 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
Patrick Aichroth	FHG-IDMT
Giuseppe Amato	CNR
Jakob Abesser	FHG-IDMT
Werner Bailer	JR
Mihai Dogariu	UPB
Fabrizio Falchi	CNR
Hannes Fassold	JR
Fabien Gandon	3IA-UCA
Milica Gerhardt	FHG-IDMT
Michail Kaseris	AUTH
Michael Loidl	JR
Hanna Lukashevich	FHG-IDMT
Ioannis Maniadis Metaxas	QMUL
Vasileios Mezaris	CERTH
Alberto Messina	RAI
Rémi Mignot	IRCAM
Maurizio Montagnuolo	RAI
Ioannis Patras	QMUL
Ioannis Pitas	AUTH
Frédéric Precioso	3IA-UCA
Lenny Renault	IRCAM
Sebastian Ribecky	FHG-IDMT
Lucile Sassatelli	3IA-UCA
Nicu Sebe	UNITN
Lucia Vadicamo	CNR
Kasim Sinan Yildirim	UNITN
Federico Pernici	UNIFI

Peer Reviews

NAME	ORGANIZATION
Antonios Liapis	UM
Thodoris Lymperopoulos, Thanos Kalligeris, Stratos Tzoannos	ATC





Revision History

Version	Date	Reviewer	Modifications
0.1	04/06/2021	Ioannis Pitas	First draft sent to partners for contributions.
0.2	23/06/2021	Ioannis Pitas	Updated version with updated structure after input from CERTH.
0.3	9/07/2021	Thodoris Lymperopoulos, Thanks Kalligeris, Stratos Tzoannos, Filareti Tsalakanid	Updated version with incorporated content from all participating partners, except JR, and input from AUTH for executive summary introduction and conclusions.
0.4	21/07/2021	Antonios Liapis	Updated version with incorporated content from JR.
0.5	20/08/2021	Filareti Tsakalanidou	Updated version based on internal review comments.
1.0	24/08/2021	Ioannis Pitas	Final version.

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
AI	Artificial Intelligence
ARE	Average Relative Error
AUC	Area Under the Curve
AVS	Ad-hoc Video Search
CDVS	Compact Descriptors for Visual Search
CNN	Convolutional Neural Network
CSN	Conditional Similarity Network
DCNN	Deep Convolutional Neural Network
DDLNC	Deep Dictionary Learning and Coding Network
DDSP	Differentiated Digital Signal Processing Synthesizer
DIR	Detection and Identification Rate
DNN	Deep Neural Network
DPP	Determinantal Point Process
FAR	False Alarm Rate
FCN	Fully Convolutional Network
FGVC	Fine-Grained Visual Categorization
FPN	Feature Pyramid Network
FSOD	Few-Shot Object Detection
GAN	Generative Adversarial Network
GRL	Gradient Reversal Layer
GPU	Graphics Processing Unit
GRU	Gated Recurrent Unit
HNSW	Hierarchical Navigable Small World
KIS	Know-Item Search
KNN	K-Nearest Neighbor
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MIDI	Musical Instrument Digital Interface
ML	Machine Learning
MPEG	Moving Picture Experts Group
MRR	Mean Reciprocal Rank
MSE	Mean Squared Error





Table of Abbreviations and Acronyms

Abbreviation	Meaning
PSA	positive Sample Augmentation
RDF	Resource Description Framework
RNN	Recurrent Neural Network
ROI	Region-of-Interest
SIFT	Scale-Invariant Feature Transform
SKOS	Simple Knowledge Organization System
SPARQL	SPARQL Protocol and RDF Query Language
SSDA	Semi-Supervised Domain Adaptation
SSL	Semi-Supervised Learning
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TFA	Two-stage fine-tuning approach
TFIDF	Term Frequency-Inverse Document Frequency
UDA	Unsupervised Domain Adaptation
VBS	Video Browsing Showdon





Contents

1	Executive Summary	14
2	Introduction	15
3	Media analysis and summarisation methods	17
3.1	Overview	17
3.2	A Survey on Deep Learning Methods for Video Summarisation	17
3.2.1	Overview of the landscape and future directions	17
3.2.2	Contributions to WP8 Use cases	20
3.2.3	Relevant Publications	20
3.3	Adversarial Reconstruction with Orthogonal Dictionaries for Deep Unsupervised Video Summarisation	20
3.3.1	Method Overview	21
3.3.2	The adversarial reconstruction framework	21
3.3.3	Proposed Method	22
3.3.4	Evaluation	23
3.3.5	WP8 Use cases Contributions	24
3.3.6	Relevant Publications	25
3.4	Exploiting Caption Diversity for Unsupervised Video Summarisation	25
3.4.1	Method Overview	25
3.4.2	DPP-caption Loss	25
3.4.3	Evaluation	26
3.4.4	WP8 Use cases Contributions	27
3.4.5	Relevant Publications	27
3.5	Joint optical flow and instance segmentation	27
3.5.1	Analysis of current state of the art for joint optical flow & instance segmentation	27
3.5.2	Contributions to WP8 Use cases	29
3.6	End-to-End Tools to Simplify the Creation, Curation and Usage of Data Sets for AI Applications	29
3.6.1	Method Overview	29
3.6.2	Face Management	29
3.6.3	Landmark/Work of Art Detection	30
3.6.4	Evaluation	30
3.6.5	Contributions to WP8 Use cases	32
3.6.6	Relevant External Resources	32
3.7	Learning and Reasoning for Cultural Metadata Quality	32
3.7.1	Method Overview	32
3.7.2	Learning and Reasoning for Cultural Metadata Quality	33
3.7.3	Contributions to WP8 Use cases	34
3.7.4	Relevant Publications	36
3.7.5	Relevant External Resources	36
4	Learning from scarce data	37
4.1	Few-shot object detection: facilitating training	37
4.1.1	Method Overview	37
4.1.2	Contributions to WP8 Use cases	39
4.1.3	Relevant External Resources	39





4.2	Domain Adaptation and Counting	39
4.2.1	Method Overview	40
4.2.2	Overall approach	40
4.2.3	Domain Adaptation Learning	41
4.2.4	Evaluation	42
4.2.5	Contributions to WP8 Use cases	42
4.2.6	Relevant Publications	44
4.2.7	Relevant External Resources	44
4.3	Video browsing and searching	44
4.3.1	Method Overview	44
4.3.2	Overall approach	45
4.3.3	System Architecture Overview	45
4.3.4	Evaluation results	47
4.3.5	Second Release of VISIONE (VBS 2021)	49
4.3.6	Contributions to WP8 Use cases	51
4.3.7	Relevant Publications	51
4.3.8	Relevant External Resources	52
4.4	Few-shot object detection: positive sample augmentation and ensembling	52
4.4.1	Method Overview	52
4.4.2	Contributions to WP8 Use cases	54
4.5	Adversarial Semi-supervised Learning for Fine Grained Visual Classification	55
4.5.1	Soft Pseudo-labeling Semi-Supervised Learning Applied to Fine-Grained Visual Classification	55
4.5.2	Fine-Grained Adversarial Semi-supervised Learning	57
4.5.3	Contributions to WP8 Use cases	62
4.5.4	Relevant Publications	63
4.6	DivClust - Learning Multiple Clusterings With a Diversity-Controlling Objective	63
4.6.1	Method Overview	63
4.6.2	DivClust	64
4.6.3	Clustering aggregation	64
4.6.4	Evaluation	65
4.6.5	Contributions to WP8 Use cases	67
4.7	Joint Deep Dictionary Learning and Coding Network	67
4.7.1	Method Overview	67
4.7.2	Contributions to WP8 Use cases	72
4.7.3	Relevant Publications	72
4.7.4	Relevant software and/or external resources	72
4.8	Curriculum Self-Paced Learning	72
4.8.1	Method Overview	72
4.8.2	Contributions to WP8 Use cases	74
4.8.3	Relevant Publications	74
5	Music Annotation and Audio Provenance Analysis	75
5.1	Overview	75
5.2	Disentanglement Representation Learning for Music Similarity	75
5.2.1	Method Overview	75
5.2.2	Contributions to WP8 Use cases	78
5.3	Realistic Music Mixes Generation	79





5.3.1	Method Overview	79
5.3.2	Piano synthesis for annotation of piano performances	80
5.3.3	Contributions to WP8 Use cases	81
5.4	Improving Audio Provenance Analysis with CNN	82
5.4.1	Audio Provenance: Method Overview	82
5.4.2	CNN-based Audio Phylogeny Analysis	83
5.4.3	Contributions to WP8 Use cases	84
6	Conclusions and Future Work	85



List of Tables

1	Comparative F-Score results of several DNN-based unsupervised video summarisation methods in two common benchmark datasets. The reported figures are from the original papers. The best results are highlighted in bold. The second best results are underlined.	24
2	Regularizer ablation study, using [1] as the main codebase in all cases. The proposed novel terms are \mathcal{L}_{dict} and \mathcal{L}_{ortho}	25
3	Comparative study against competitive unsupervised learning methods. The metric used here for evaluation is the F-score. Bold indicates the best results.	27
4	Experimental results obtained for the four considered domain shift. Three evaluation metrics were used: the Mean Absolute Error (MAE), the Mean Squared Error (MSE) and the Average Relative Error (ARE). Performance improvements was obtained in all the scenarios, considering all the three metrics.	44
5	MRR@k for eight combinations of the rankers (the four best, the four worst and the setting used at VBS2019) varying k. Statistically significant results with two-sided p value lower than 0.05 over the baseline $BM25-BM25-TF$ are marked with *.	49
6	Few-shot detection evaluation on PASCAL VOC 2012. The numbers in the split boxes represent the number of samples in the novel classes.	53
7	Few-shot detection evaluation on MS-COCO.	54
8	Results on the Semi-Supervised iNaturalist-Aves Dataset (FGVC7 challenge). Our method achieves a significant improvement by leveraging unsupervised data.	62
9	Results evaluating the effectiveness of DivClust in learning diverse clusterings on CIFAR10. When models were trained from scratch (SC), PICA was trained for 250 epochs and IIC for 1000. When trained on-top (OT), PICA and IIC models were trained for 200 and 1000 epochs, and additional clusterings were added subsequently in regular 25 and 50 epoch intervals respectively.	66
10	Results when applying DivClust on top of a pre-trained clustering model. The baseline model was trained with PICA and overclustering, using the training configuration proposed in [2]. DivClust was trained on top for the number of epochs noted in the table, with a total of 10 clusterings being added incrementally.	67
11	Classification accuracy (%) on Caltech 256.	71
12	Average Precision (AP) scores (in %) of several Faster R-CNN models trained using different state-of-the-art domain adaptation methods [3, 4, 5, 6, 7] versus a Faster R-CNN model trained using our domain adaptation approach based on curriculum self-paced learning. All domain adaptation methods include images without ground-truth labels from the target domain. Faster R-CNN baselines without adaptation (trained only on source) are also included to point out the absolute gain of each domain adaptation technique, with respect to the corresponding baseline. Faster R-CNN models trained on target domain images with ground-truth label are included as indicators of possible upper bounds of the AP scores. Results are reported for Sim10k→Cityscapes and KITTI→Cityscapes benchmarks. The best AP scores and the highest absolute gains are highlighted in bold. S+T indicates Source + Target.	74
13	Similarity Evaluation	79





List of Figures

1	A taxonomy of the existing deep-learning-based video summarisation methods. . .	18
2	The proposed Orthonormal Dictionary-based Summarisation training-stage architecture. AUTH contributions are encompassed in the light gray bounding box. . .	23
3	Automated creation of reference data sets for landmark/work-of-art recognition. . .	31
4	DIR vs. FAR curve describing the trade-off for rank one identification and false alarms for the face labelling task. The red dotted line represents a system that is no better than random guessing. The solid blue line represents the measured values. The AUC (Area Under the Curve) score is 0.97, denoting an excellent performance. The best balance between DIR and FAR is obtained for Cosine similarity equal to 0.4.	32
5	The data processing pipeline combining symbolic AI and machine learning to improve the quality of cultural metadata and information retrieval.	34
6	Examples of noise detection in the images that do not have a visually relevant term <i>cheval</i> (horse) with the prediction scores below 0.2.	35
7	Example of the artwork with the adjustment of prediction score of concept <i>mer</i> (sea) by the prediction score of concept <i>bateau</i> (boat).	35
8	Algorithm overview. Given $C \times H \times W$ images from source and target domains, they are processed by the density map estimation network to obtain output predictions. A density loss is computed for source predictions based on the ground truth. In order to improve target predictions, a discriminator is used to locally classify whether a density map belongs to the source or target domain. Then, an adversarial loss is computed on the target prediction and is back-propagated to the density map estimation and counting network.	41
9	Examples of the predicted density maps in the considered scenarios: (a) <i>Day2Nigh</i> Domain Shift using the <i>NDISPark</i> dataset; (b) and (c) <i>Camera2Camera</i> Domain Shift employing the <i>WebCamT</i> and <i>TRANCOS</i> datasets, respectively; (d) <i>Synthetic2Real</i> Domain Shift using the <i>GTA</i> dataset for the training phase and the <i>WebCamT</i> dataset for testing on real images. In the first row, the input images are reported. In the second row, the ground truth, while in the third, the predicted density maps obtained with our models.	43
15	Positive Sample Augmentation framework. Reinforcement branch on yellow background.	53
16	Illustration of the effect of second-order pooling in Semi-Supervised Fine-Grained Visual Categorization (SSL-FGVC). We show images from three different classes of Airbus aircraft models: A319 (<i>top</i>), A320 (<i>middle</i>) and A321 (<i>bottom</i>). They mainly differ by the number of doors and their position along the fuselage (circles). We propose to take advantage of the long-range attention based part-to-part relationships exploited by second-order pooling and back-propagate this information onto unlabeled data to perform unsupervised structure discovery.	58
17	An overview of the proposed model architecture. The inputs to the network are labeled and unlabeled examples. The model f_θ (light green) consists of the second-order pooling (iSQRT-COV) [8] feature extractor F (light red) and the classifier C having weight vectors \mathbf{w}_i (light blue). C is trained to maximize entropy on unlabeled target whereas F is trained to minimize it. To achieve the adversarial learning, the sign of gradients for entropy loss on unlabeled target examples is flipped by a gradient reversal layer (GRL) [9]. According to this, labeled and unlabeled back-propagation follows two distinct paths.	59





18	The $w \times h$ feature channels of dimension d of the last convolutional layer of the CNN architecture are used to compute the covariance matrix. The $\frac{d(d+1)}{2}$ -dimensional values of the upper triangular matrix constitute the internal feature representation vector that allows the model to determine the attention based long-range part-to-part relationships. The forward and backward propagation of the covariance in the adversarial optimization setting of Fig. 11 are computed according to the iSQRT-COV approximated method.	60
19	Adversarial Learning intuition. (a): High entropy between the classifier weight vectors \mathbf{w}_i (i.e., the prototypes) and the unlabeled data features forces the classifier weight vectors to “move” towards the unlabeled data features. (b): This force is counter to the standard cross entropy which instead tends to cluster labeled and unlabeled features around the estimated prototypes.	61
20	Illustration of the proposed framework assuming clusterings A and B with two clusters each. Given a set of data, a backbone network f , and projection heads h_k , each corresponding to a clustering k , DivClust restricts their similarity, enforcing that some samples belong to different clusters in each clustering.	63
21	The framework of the proposed Deep Dictionary Learning and Coding Network (DDLNCN).	68
22	Multi layers coding strategy. The first layer is mainly used to partition the space, while the main approximation power is achieved within the second layer, which embodies a ‘divide and conquer’ strategy.	70
23	Our curriculum self-paced learning approach for object detection. In the initial training stage (step 1.a), the object detector is trained on source images with ground-truth labels. In step 1.b, the object detector is further trained on source images translated by Cycle-GAN [10] to resemble images from the target domain. In steps 2, 3 and 4, the object detector is fine-tuned on real target images (different from those included in the test set), using the bounding boxes and the labels predicted by the current detector. In step 5, the model makes its predictions on the target test set for the final evaluation. Best viewed in color.	73
24	Specificity range of music similarity tasks.	75
25	Music similarity dimensions [11].	76
26	Similarity spaces created by combination of musical dimensions.	76
27	Deep metric learning [12].	76
28	Overview of the conditional similarity network and the embedding masking procedure [11].	77
29	Naive module architecture	77
30	Dimension reduction architecture	77
31	Implemented CSN-model architecture.	78
32	Architecture of the DDSP-based synthesizer of Piano tones.	80
33	Spectrograms of Piano notes.	81
34	Phylogeny analysis results for a set of near-duplicates, as visualized within the respective software tool: Nodes represent audio files, connections represent parent-child relations between nodes.	82
35	Partial audio matching without query.	83
36	Process of phylogeny analysis for one pair of audio files, using CNN for transformation prediction.	84
37	Score of different tree metrics for phylogeny tree reconstruction. Averaged over 40 different phylogeny trees	84





1. Executive Summary

Deliverable D5.1 “Initial report on Multimedia Summarisation and Analysis” is the first public 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 summarisation. D5.1 contains results of WP5 activities concerning Tasks T5.1 (“Media analysis and summarisation”), T5.3 (“Learning with scarce data”) and T5.6 (“Music Annotation and Audio Provenance Analysis”) performed during the period M1-M12 of the project. It presents the developed methods in their scientific context, the obtained evaluation results, as well as any relevant publications, public software or plans for AI4EU software integration.

The deliverable sums up all research activities of the AI4Media partners participating in these three Tasks up to M12, most of which have already led to several papers submitted or published to well-known, relevant scientific venues. Moreover, the majority of 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 on future research directions for the involved Tasks.

Task T5.1 focuses on AI-based analysis and summarisation of media data, such as images or video. Work performed up to now and presented in this deliverable mainly consists of: a) an exhaustive overview of the state-of-the-art in unsupervised video summarisation, along with identification of current limitations and promising directions for future research, b) two novel, complementary methods for unsupervised video summarisation, c) a literature survey on optical flow estimation, instance segmentation and their joint calculation, d) development of novel AI tools for creating, curating and managing media datasets based on archival data, and e) a novel method for information retrieval on cultural media datasets, relying on a synthesis of computational deep learning with symbolic semantic reasoning. The tools and methods developed in this Task contribute to project use cases 1D1, 1D2, 3C2, 3A1 and 3A3, which concern search and management of audiovisual items in archives, media content creation and adaptation, as well as intelligent exploitation of pre-existing archives and informative content.

Task T5.3 deals with learning from scarce data, focusing on training or adapting DNNs for scenarios marked by a lack of large-scale, domain-specific datasets and/or annotations. Work performed up to now and presented in this deliverable mainly consists of: a) two novel methods for few-shot visual object detection, b) an end-to-end CNN-based unsupervised domain adaptation algorithm for traffic density estimation and counting, c) a visual content-based retrieval system designed to support large-scale video search, d) a semi-supervised learning approach to fine-grained visual categorization, e) a novel clustering method relying on a diversity-controlling objective, f) a novel method for joint deep learning and dictionary-based representation learning for image recognition with limited data, and g) a new curriculum self-paced learning approach to domain adaptation for object detection. The tools and methods developed in this Task contribute to project use cases 3A3, 3C2, 2B1 and 4C2, which concern intelligent exploitation of pre-existing archives, media content creation and adaptation, automatic metadata tagging and video analysis.

Finally, Task T5.6 focuses on advanced audio analysis for automatic music annotation and audio partial matching/reuse detection, mainly relying on DNNs. Work performed up to now and presented in this deliverable mainly consists of: a) a novel method for music similarity analysis, b) development of AI-based tools for generating music mixes based on MIDI, and c) a new approach for audio phylogeny analysis with improved computational efficiency. The tools and methods developed in this Task contribute to project use cases 5B2, 1A3 and 4C3, which concern musical recording analysis, synthetic audio detection/verification and audio analysis.





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 that mostly rely on Deep Neural Networks (DNNs). It has the following objectives:

1. Addressing AI issues in content production and processing in textual, visual and audio media, multimedia production, enhancement, and summarisation.
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 summarisation, specifically addressing high resolution visual data and audio as special cases.

This document reports on activities concerning Tasks T5.1, T5.3 and T5.6, during Months M1-M12 of the project. T5.1 relates to Objectives 1 and 5, T5.3 relates to Objective 2 and T5.6 relates to Objectives 1 and 3.

Efficient media analysis and summarisation is 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 of AI4Media intends to advance the state-of-the-art in these areas, in a manner consistent with application needs. Relevant project research gives special emphasis to *unsupervised video summarisation*, i.e., the task of summing up a video into a set of temporally ordered key video frames, that jointly capture the original video content in a succinct manner. In this context, “unsupervised” refers to machine learning models trained without access to ground-truth, manually annotated summaries, since constructing such summaries is difficult and time-consuming. Furthermore, AI4Media aims to investigate different ways of analyzing video using machine learning approaches, particularly modern DNNs, for tasks such as *video captioning*, *face detection/recognition* or *human activity recognition*. Finally, T5.1 attempts to *combine knowledge representations with deep neurally-derived representations* to design new information retrieval engines, where symbolic reasoning and sub-symbolic learning are cooperating to increase media analysis potential. Progress in these areas is detailed in Section 3 of this document.

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. T5.3 of AI4Media aims 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 applicability of DNNs in real-world scenarios, with T5.3 also partially relating to WP3; notably to transfer learning and learning to count. Progress of T5.3 activities is detailed in Section 4 of this document.





Finally, AI-enabled music analysis is a topic of high industrial relevance that requires special attention. T5.6 of AI4Media deals 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. Progress of T5.6 activities is detailed in Section 5 of this document, while Section 6 draws conclusions from the presented effort.





3. Media analysis and summarisation methods

3.1. Overview

This Section reports work conducted on media analysis and summarisation. The first three subsections cover the topic of video summarisation: a literature survey that identifies promising future research directions in summarisation, performed by CERTH, is followed by two novel methods for unsupervised video key-frame extraction from AUTH. The next subsection presents an analysis conducted by JR on the current state of the art for joint optical flow & instance segmentation, taking into account combined methods as well as methods performing only one of these tasks. Following this, the problem of unavailability of training data for developing media analysis methods is addressed; relevant end-to-end AI-based tools are presented that simplify the creation and curation of data sets for training AI methods, developed by RAI. Finally, the last subsection concerns 3IA-UCA research on using the outputs of media analysis for information retrieval, exploiting the combination of symbolic knowledge representations with sub-symbolic/computational deep representations.

3.2. A Survey on Deep Learning Methods for Video Summarisation

Contributing partners: CERTH

3.2.1. Overview of the landscape and future directions

CERTH's work initially focused on studying the recent advances in the field of automatic video summarisation, with the aim to identify potential directions for future research. The outcome of this study was a comprehensive survey of the existing deep-learning-based methods for video summarisation, that represent the current state of the art. More than 40 different methods were discussed in the survey, and grouped according to the taxonomy in Fig. 1. This taxonomy divides the studied methods according to the utilized data modalities (first layer) and the adopted training strategy (second layer). Then, the penultimate layer of this arboreal illustration shows the different learning approaches that have been adopted in the bibliography. Finally, the leaves of each node of this layer show the utilized techniques for implementing each learning approach. This taxonomy was the basis for providing a systematic review of the relevant literature that showed the evolution of the deep-learning-based video summarisation technologies and led to a few suggestions for future developments.

Based on our study, we believe that future work should primarily target the development of deep learning methods that can be trained effectively without the use of ground-truth data. In this way, the research community will be able to tackle issues associated with the limited amount of annotated data, and to significantly diminish (or even completely eliminate) the need for laborious and time-demanding data annotation tasks. Towards this direction, the research community could invest efforts in designing and developing deep learning architectures that can be trained in a fully-unsupervised or in a semi-/weakly-supervised manner.

With respect to the development of unsupervised video summarisation methods, given the fact that most of the existing approaches try to increase the representativeness of the generated summary with the help of summary-to-video reconstruction mechanisms, future work could target the advancement of such methods by integrating mechanisms that force the outcome of the summarisation process to be aligned with additional criteria about the content of the generated summary, such as its visual diversity (that was considered in [13, 14, 15, 16]) and its uniformity (that was



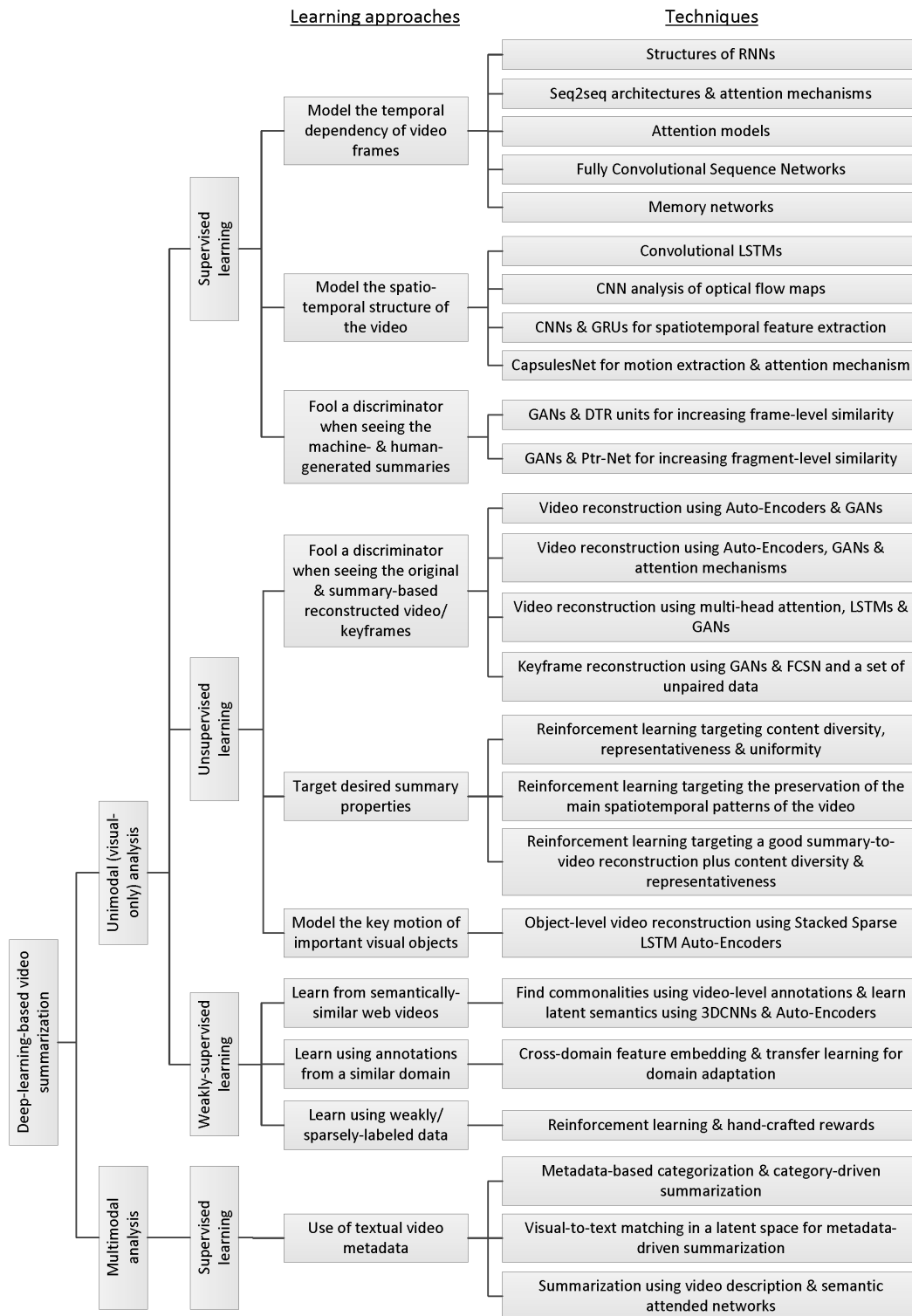


Figure 1. A taxonomy of the existing deep-learning-based video summarisation methods.



examined in [17]). On a similar basis, efforts could be put towards the extension of existing deep learning architectures that combine the merits of adversarial and reinforcement learning [18], by utilizing a Soft Actor-Critic [19] that is capable of further discovering the action space via automatically defining a suitable value for the entropy regularization factor, or by introducing additional rewards that are associated with the aforementioned summarisation criteria.

With regards to the development of semi- or weakly-supervised technologies, the goal would be to investigate ways to intervene in the summary production process in order to force the outcome (i.e., a video summary) to be aligned with user-specified rules. One approach in this direction, is the generation of a summary according to a set of textual queries that relate to the summary content (as in [20, 21, 22, 23, 24]). Another, more aspiring approach would be the use of an on-line interaction channel between the user/editor and the trainable summarizer, in combination with active learning algorithms that allow to incorporate the user's/editor's feedback with respect to the generated summary (as in [25]). Finally, the possibility of adapting Graph Signal Processing approaches [26], which have already been applied with success to data sampling [27] and image/video analysis tasks [28, 29], for introducing such external supervision could be examined. The development of effective semi- or weakly-supervised summarisation technologies will allow to better meet the needs of specific summarisation scenarios and application domains. For example, such developments are often important for the practical application of summarisation technologies in the News/Media Industry, where complete automation that diminishes editorial control over the generated summaries is not always preferred.

Concerning the training of unsupervised video summarisation methods, we show that most of these methods rely on the adversarial training of GANs. However, open questions with respect to the training of such architectures, such as sufficient convergence conditions and mode collapse, still remain. So, another promising research direction could be to investigate ways to improve the training process. For this, one strategy could be the use of augmented training data (that do not require human annotation) in combination with curriculum learning approaches. Such approaches have already been examined for improving the training of GANs (see [30, 31, 32]) in applications other than video summarisation. We argue that transferring the gained knowledge from these works to the video summarisation domain would contribute to advancing the effectiveness of unsupervised GAN-based summarisation approaches. Regarding the training of semi- or weakly-supervised video summarisation methods, besides the use of an on-line interaction channel between the user/editor and the trainable summarizer that was discussed in the previous paragraph, supervision could also relate to the collection of an adequately-large set of unpaired data (i.e., raw videos and video summaries with no correspondence between them) from a particular summarisation domain or application scenario. Taking inspiration from the method in [15], we believe that such a data-driven weak-supervision approach would eliminate the need for fine-grained supervision signals (i.e., human-generated ground-truth annotations for the collection of the raw videos) or hand-crafted functions that model the domain rules (which in most cases are really hard to obtain), and would allow a deep learning architecture to automatically learn a mapping function between the raw videos and the summaries of the targeted domain.

Another future research objective involves efforts to overcome the identified weaknesses of using RNNs for video summarisation that were discussed e.g., [33, 34, 14, 17] and mainly relate to the computationally-demanding and hard-to-parallelize training process, as well as to the limited memory capacity of these networks. For this, future work could examine the use of Independently Recurrent Neural Networks [35] that were shown to alleviate the drawbacks of LSTMs with respect to decaying, vanishing and exploding gradients [17], in combination with high-capacity memory networks, such as the ones used in [36, 37]. Alternatively, future work could build on existing approaches [33, 14, 38, 39] and develop more advanced attention mechanisms that encode the relative position of video frames and model their temporal dependencies according to different





granularities (e.g., considering the entire frame sequence, or also focusing on smaller parts of it). Such methods would be particularly suited for summarizing long videos (e.g., movies). Finally, with respect to video content representation, the above proposed research directions could also involve the use of network architectures that model the spatiotemporal structure of the video, such as 3D-CNNs and convolutional LSTMs.

Currently, the dominant approach with respect to the utilized data modality for learning summarisation, is to focus on the analysis of the visual content. Nevertheless, the audio modality of the video could be a rich source of information as well. For example, the audio content could help to automatically identify the most thrilling parts of a movie that should appear in a movie trailer. Moreover, the temporal segmentation of the video based also on the audio stream could allow the production of summaries that offer a more natural story narration compared to the generated summaries based on approaches that rely solely on the visual stream. We argue that deep-learning architectures that have been utilized to model frames' dependencies based on their visual content, could be examined also for analyzing the audio modality. Following, the extracted representations from these two modalities could be fused according to different strategies (e.g., after exploring the latent consistency between them, as in [40]), to better indicate the most suitable parts for inclusion in the video summary.

Finally, besides the aforementioned research directions that relate to the development and training of deep-learning-based architectures for video summarisation, we strongly believe that efforts should be put towards the definition of better evaluation protocols to allow accurate comparison of the developed methods in the future. The discussions in [41] and [42] showed that the existing protocols have some imperfections that affect the reliability of performance comparisons. To eliminate the impact of the choices made when evaluating a summarisation algorithm (that e.g., relate to the split of the utilized data or the number of different runs), the relevant community should consider all the different parameters of the evaluation pipeline and precisely define a protocol that leaves no questions about the experimental outcomes of a summarisation work. Then, the adoption of this protocol by the relevant community will enable fair and accurate performance comparisons.

3.2.2. Contributions to WP8 Use cases

Relevant WP8 Use Cases: 1D2 (Audiovisual Item Management in Verification Archives), 3A1 (Informative Content exploitation). This study contributes to use cases 1D2 and 3A1, by providing future directions of research towards addressing the need for algorithms that enable generating video summaries. These are needed by journalists and programme producers, for allowing them to quickly decide whether an original (typically, longer) video is the video that they need or are searching for, and to (semi-)automatically generate the highlights from a retrieved video so as to speed-up the editorial content creation process.

3.2.3. Relevant Publications

- E. Apostolidis, E. Adamantidou, A. Metsai, V. Mezaris, I. Patras, "Video Summarisation Using Deep Neural Networks: A Survey", arXiv:2101.06072, <https://arxiv.org/abs/2101.06072>.

3.3. Adversarial Reconstruction with Orthogonal Dictionaries for Deep Unsupervised Video Summarisation

Contributing partners: AUTH





3.3.1. Method Overview

AUTH worked on unsupervised video summarisation, using a state-of-the-art implementation [1] of the common Deep Neural Network (DNN)-based adversarial reconstruction framework [43] as a baseline. This framework is architecturally composed of a multi-branch LSTM/GAN combination that is fed individual video frame representations, extracted from a pretrained CNN, as input. Under this paradigm, the neural model learns to select key-frames that are jointly able to non-linearly reconstruct the full original video sequence, while remaining diverse in visual content. AUTH contributed a novel loss function that can be added to the set of adversarial reconstruction training objectives, so as to further push the DNN towards selecting key-frames that can linearly reconstruct the complete video sequence, using a learnt, orthonormal global visual dictionary. Thus, this Orthonormal Dictionary-based Summarisation method is composed of three interacting components:

- A *Dictionary Loss* term that penalizes the inability of the extracted summary to linearly reconstruct the original video,
- An auxiliary *Orthonormality Regularizer* that affects the Dictionary Loss computation, and
- A *Full-video Autoencoder LSTM*, i.e., a new, pretrained neural branch that is appended to the overall training-stage architecture and runs in parallel to the existing neural modules, so that the Dictionary Loss can be computed. It can be discarded at the inference stage.

3.3.2. The adversarial reconstruction framework

The original/full/complete input video sequence is represented by a matrix $\mathbf{X} \in \mathbb{R}^{M \times T}$, where M is the dimension of each input video frame representation and T is the total number of video frames in the source sequence. The temporally ordered video frame representations $\mathbf{x}_t \in \mathbb{R}^M, t \in [1, \dots, T]$ are typically extracted from a pretrained CNN. These representations are being sequentially fed to the LSTM-based Summarizer which unfolds into T time steps. The Summarizer comprises of three LSTM submodules: the *Selector*, the *Encoder* and the *Decoder*. The Selector outputs a real vector $\mathbf{s} \in [0, 1]^T$ which encodes the scalar importance of each input video frame, i.e., its suitability as a key-frame. Subsequently, the scalar products $s_t \mathbf{x}_t, t \in [1, \dots, T]$ are computed and fed to the Encoder, which in turn produces the internal fixed-length representation of the summarized video $\mathbf{e} \in \mathbb{R}^H$, where H is the vector length of the Encoder's hidden state. Then, \mathbf{e} is fed to the Decoder, whose output is the summary-based reconstructed video sequence $\hat{\mathbf{X}} \in \mathbb{R}^{M \times T}$. Finally, the columns of $\hat{\mathbf{X}}$ and \mathbf{X} are fed to the Discriminator LSTM, also unfolding into T time steps, which is tasked to discern which sample is an original video real one and which is a summary-based reconstruction. Thus, across all training iterations, the Discriminator is fed both original input videos and summary-based reconstructions, as real and fake samples, respectively, at an approximately 50:50 ratio. The overall architecture is depicted in Figure 2. Training is performed with error back-propagation and gradient descent, with module parameters being jointly learnt so that various loss functions are concurrently minimized. We define $\theta_s, \theta_e, \theta_d, \theta_c$ as the parameters of the Selector, the Encoder, the Decoder and the Discriminator, respectively. Additionally, let $\phi(\mathbf{X})/\phi(\hat{\mathbf{X}})$ be the final hidden state vector of the Discriminator when it is fed as input an original video/its summary-based reconstruction, respectively. Finally, let $C(\mathbf{X})/C(\hat{\mathbf{X}})$ be the output probability of the Discriminator when it is fed $\mathbf{X}/\hat{\mathbf{X}}$, respectively. The following loss functions are used [43, 1]:

- Reconstruction Loss $\mathcal{L}_{recon} = \|\phi(\mathbf{X}) - \phi(\hat{\mathbf{X}})\|_2^2$. The distance between the Discriminator's internal fixed-length representation of an original video and of its summary-based reconstruction (outputted by the Decoder) is used to update parameters θ_s, θ_e and θ_d . This is the main loss directing the training of the LSTM-based Autoencoder and the Selector.



- Originality Loss $\mathcal{L}_{original} = (1 - C(\mathbf{X}))^2$, represents the mean squared error between a ground-truth label of 1, denoting that the Discriminator’s input is actually an original/complete/full (“real”) video, and the Discriminator output. It is used to update θ_c .
- Summary Loss $\mathcal{L}_{sum} = (C(\hat{\mathbf{X}}))^2$, which penalizes the deviation between a ground-truth label of 0, denoting that the Discriminator’s input is actually a summary-based reconstruction (“fake”) video, and the Discriminator output. It is used to update θ_c .
- Generator Loss $\mathcal{L}_{gen} = (1 - C(\hat{\mathbf{X}}))^2$ represents the mean squared error between a label of 1 and the Discriminator output, when the Discriminator’s input is actually a summary-based reconstruction (“fake”) video. It is used to update θ_d , so that the Decoder learns to fool the Discriminator.
- Sparsity Loss $\mathcal{L}_{sparsity} = \|\frac{1}{T} \sum_{t=1}^T s_t - \sigma\|_2$, which pushes the Selector towards assigning high importance (i.e., key-frame status probability) to a specific percentage of the total number of original video frames, defined by a scalar hyperparameter $\sigma \in [0, 1]$. This penalty term updates θ_s .
- Determinantal Point Process (DPP) Loss, which is a regularizer pushing towards high global summary saliency (i.e., high visual content diversity in the selected key-frames). It is used to update θ_s and θ_e . First, we consider a matrix $\mathbf{L} \in \mathbb{R}^{T \times T}$ by computing the pairwise cosine similarity for time step t and t' that is, $\mathbf{L} = \mathbf{e}_t \mathbf{e}_{t'}$. Then, the DPP loss is given by $\mathcal{L}_{dpp} = \frac{\det(\mathbf{L}_y)}{\det(\mathbf{L} - \mathbf{I})}$ where \mathbf{L}_y is a minor submatrix of \mathbf{L} whose rows and columns are dictated by the indices of the selected key-frames, according to \mathbf{s} , and \mathbf{I} is the identity matrix.

After training is complete, the Selector LSTM is the only component needed for inference; the Autoencoder and the Discriminator can be readily discarded.

3.3.3. Proposed Method

The *Full-video Autoencoder* is composed of two LSTM submodules. Its encoder sequentially receives original video frame representations \mathbf{x}_t and its decoder reconstructs them across T time steps. The final hidden state of the encoder $\mathbf{h} \in \mathbb{R}^N$, obtained after the T -th video frame has been processed, constitutes an internal, fixed-length representation of the entire full/original input video. The Full-video Autoencoder is pretrained before being inserted into the general adversarial reconstruction framework, by minimizing a MSE-based reconstruction loss function. However, after it has been trained, its decoding LSTM is no longer required; only the encoding part is in fact needed for inference, while training the augmented adversarial reconstruction architecture.

The addition of the Full-video Autoencoder permits us to compute an “ideal” fixed-length representation \mathbf{h} of the original/complete video, which reflects original visual content more accurately than the Discriminator’s internal representation $\phi(\mathbf{X})$. This is because the latter one is optimized for discerning between “real” and “fake” examples and not for serving as a compressed, fixed-length representation of the original video.

The proposed *Dictionary Loss* term, which is added to the pool of employed loss terms described in 3.3.2, employs the video-specific representation \mathbf{h} and a learnable matrix $\mathbf{A} \in \mathbb{R}^{N \times H}$, shared across all training videos. It is defined as follows:

$$\mathcal{L}_{dict} = \|\mathbf{h} - \mathbf{A}\mathbf{e}\|_2, \quad (1)$$

where $\mathbf{e} \in \mathbb{R}^H$ is the final hidden state of the LSTM Encoder, i.e., a fixed-length representation of the summary video.

Each time \mathcal{L}_{dict} is computed, \mathbf{A} projects \mathbf{e} onto a vector space that is being learnt from the original data distribution, rendering \mathbf{A} a global visual dictionary. The benefit we reap from this



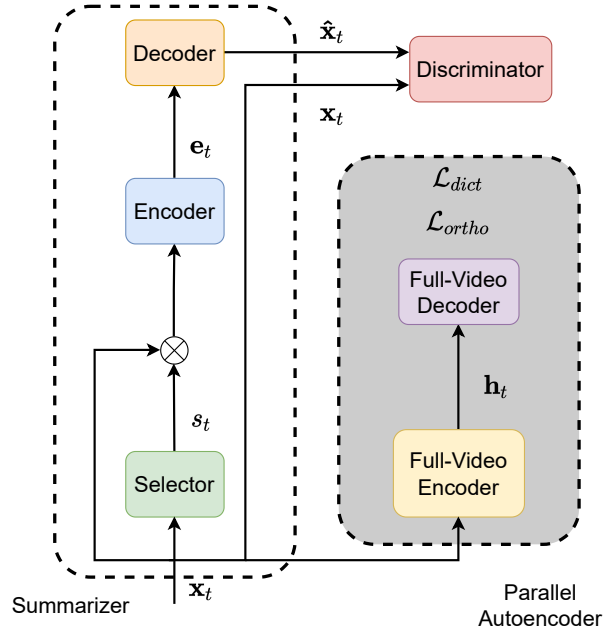


Figure 2. The proposed Orthonormal Dictionary-based Summarisation training-stage architecture. AUTH contributions are encompassed in the light gray bounding box.

is double-fold. First, each summarized video representation \mathbf{e} is forced to be a set of coefficients capable of linearly reconstructing the corresponding original input representation \mathbf{h} . This linear reconstruction constraint is complementary to the non-linear one enforced by the Decoder and \mathcal{L}_{recon} , thus pushing further towards selecting key-frames that are visually representative of the full original video. Secondly, \mathcal{L}_{dict} is only used to update parameters θ_s , θ_e and matrix \mathbf{A} , thus preventing an overfitting of θ_d to the training set to compensate for subpar key-frame selection by the Selector.

In order to push the atoms of the global visual dictionary \mathbf{A} towards being independent, thus minimizing redundancy, an additional *Orthonormality Regularizer* is also proposed that exhorts the columns of matrix \mathbf{A} to be orthonormal. Thus, they are strongly encouraged to be linearly independent basis vectors. This is accomplished by the following loss term:

$$\mathcal{L}_{ortho} = \|\mathbf{A}\mathbf{A}^T - \mathbf{I}\|_2^2, \quad (2)$$

where \mathbf{I} is the identity matrix. The gradient signal produced by this regularizer, penalizes the entries of matrix \mathbf{A} and pushes its rank to maximization. As a result, training tends to converge to a \mathbf{A} which is orthogonal (if $N = H$) or semi-orthogonal (if $N \neq H$).

\mathcal{L}_{dict} and \mathcal{L}_{ortho} are jointly employed, along with the pretrained encoding part of the Full-video Autoencoder, at the training stage only. Both matrix \mathbf{A} and the Full-video Autoencoder can be discarded afterwards, so that the runtime overhead of the proposed method during inference is zero.

3.3.4. Evaluation

Evaluation of unsupervised video key-frame extraction methods on a dataset is typically conducted using the popular F-Measure metric (F), also known as F-Score, or F1. Assuming that: a) a summary/key-frame set is represented as a binary vector $\mathbf{y} \in \mathbb{R}^T$, where the entry y_i , $1 \leq i \leq T$ is





Method	TVSum	SumMe
SUM-GAN-AAE [1]	58.3%	48.9%
SUM-GAN-AAE [1]+ \mathcal{L}_{dpp}	<u>61.0%</u>	<u>56.5%</u>
SUM-FCN _{unsupervised} [44]	52.7%	41.5%
DR-DSN [45]	57.6%	41.4%
EDSN [46]	57.6%	42.6%
Unpaired VSN [47]	55.6%	47.5%
PCDL [48]	58.4%	42.7%
SUM-GAN-sl [49]	58.4%	47.8%
Cycle-SUM [50]	57.6%	41.9%
ACGAN [51]	58.5%	46.0%
Proposed Method	65.0%	62.2%

Table 1. Comparative F-Score results of several DNN-based unsupervised video summarisation methods in two common benchmark datasets. The reported figures are from the original papers. The best results are highlighted in bold. The second best results are underlined.

1/0 if the i -th video frame is/is not a part of this summary, and b) a ground-truth summary exists for the test set, then:

$$F = 2 \times \frac{PR}{P + R} \times 100\%, \quad (3)$$

where P is the precision and the R the recall. Assuming that the DNN-generated summary/key-frame set is \mathbf{s} and the ground-truth one is \mathbf{g} , it holds that:

$$P = \frac{\mathbf{s} \cap \mathbf{g}}{|\mathbf{s}|}, \quad (4)$$

$$R = \frac{\mathbf{s} \cap \mathbf{g}}{|\mathbf{g}|}. \quad (5)$$

An implementation of the adversarial reconstruction framework augmented with the proposed method achieves state-of-the-art results in quantitative comparisons with competing unsupervised approaches, in two commonly employed and publicly available datasets (TVSum [52], SumMe [53]): it achieves F-Score gains of 4%/5.7%, in the TVSum/SumMe dataset, respectively, compared to baseline. Table 1 depicts F-Score results for several recent DNN-based unsupervised key-frame extraction methods, given a sparsity percentage of $\sigma = 15\%$ (meaning that the requested summary must have approximately 15% of the temporal length of the original video). To better analyze the performance of the proposed method, an ablation study was conducted on top of the [1] codebase.

This work has been submitted as a conference paper. Previously, a preliminary version (without the Orthonormality Regularizer and building upon a less advanced baseline) had also been accepted as a paper to the IEEE International Conference on Image Processing 2021 (ICIP), showcasing less impressive F-Score gains compared to baseline (1%/2.1%).

3.3.5. WP8 Use cases Contributions

Relevant WP8 Use Cases: 1D2 (Audiovisual Item Management in Verification Archives), 3A1 (Informative Content exploitation). This method contributes to use cases 1D2 and 3A1, by pro-





Regularizer combination	TVSum	SumMe
$\mathcal{L}_{sparsity}$ ([1])	58.3%	48.9%
$\mathcal{L}_{sparsity} + \mathcal{L}_{dpp}$	61.0%	56.5%
$\mathcal{L}_{sparsity} + \mathcal{L}_{dict}$	59.3%	51.0%
$\mathcal{L}_{sparsity} + \mathcal{L}_{dict} + \mathcal{L}_{ortho}$	64.9%	59.3%
$\mathcal{L}_{sparsity} + \mathcal{L}_{dict} + \mathcal{L}_{ortho} + \mathcal{L}_{dpp}$	65.0%	62.2%

Table 2. Regularizer ablation study, using [1] as the main codebase in all cases. The proposed novel terms are \mathcal{L}_{dict} and \mathcal{L}_{ortho} .

viding algorithms supporting easy video storage and browsing. These are needed by journalists and programme producers.

3.3.6. Relevant Publications

- M. Kaseris, I. Mademlis, I. Pitas, "Adversarial Unsupervised Video Summarisation Augmented With Dictionary Loss", IEEE International Conference on Image Processing (ICIP) 2021, accepted for presentation (Zenodo Record: <https://zenodo.org/record/4899284>).
- M. Kaseris, I. Mademlis, I. Pitas, "Learning Orthonormal Dictionaries for Deep Unsupervised Video Summarisation", technical report, submitted as conference paper

3.4. Exploiting Caption Diversity for Unsupervised Video Summarisation

Contributing partners: AUTH

3.4.1. Method Overview

Independently from Orthonormal Dictionary-based Summarisation, AUTH also investigated a way to increase key-frame extraction performance of the common Deep Neural Network (DNN)-based adversarial reconstruction framework [43] (as described in Subsection 3.3.2), using [1] as a baseline. To this end, a novel regularizer was devised that pushes towards increased diversity on the selected key-frames, with regard to their caption-related latent representations. These representations are LSTM hidden states, corresponding to each selected key-frame, that are being internally produced by a pretrained DNN-based image captioner. Each time the summarizer DNN processes a video frame, the latter one is also fed to the captioner DNN in order to compute its respective caption-related latent representation. Thus, the novel regularizer penalizes key-frame sets with low caption diversity. Similarly to Orthonormal Dictionary-based Summarisation, this method too imposes zero runtime overhead during inference.

3.4.2. DPP-caption Loss

Image captioning consists in generating a textual, natural-language description for a given RGB image. The primary challenge lies in two aspects: extracting adequate information from the visual content and generating grammatically correct, human-readable sentences. Several supervised DNN-based image captioning approaches exist, mostly involving architectures relying on CNNs and LSTMs.





The proposed method is a reformulation of the so-called *DPP loss*, which has been successfully applied as a regularizer for enforcing summary diversity in [43]. Determinantal Point Processes (DPPs) are elegant probabilistic models of repulsion that, in the video summarisation context, quantify the variance of selected key-frame representations. Since the employed convolutional representations per video frame encode object-centric semantic information, the original DPP loss pushes towards summaries composed of key-frames that depict different objects.

In the context of this work, a novel variant of the DPP loss is proposed, called *DPP-caption loss*, or \mathcal{L}_{dpp-c} , which relies on a pretrained DNN-based image captioner. At each iteration of the summarisation DNN training, \mathcal{L}_{dpp-c} pushes towards selecting key-frames that differ in their textual description according to the respective captioner output. This enforces additional diversity in the derived summary, based on a non-object-centric semantic modality. For instance, an image caption may focus on depicted activities or scene context, instead of the visible objects.

The proposed novel regularizer \mathcal{L}_{dpp-c} requires an LSTM-based image captioner, pretrained on a generic mass-scale annotated dataset, which we denote by P . During training an unsupervised summarisation DNN falling under the adversarial reconstruction framework, each video frame is forwarded to P (in inference mode), in parallel to feeding them to the encoder E . Thus, the final hidden state of P encodes features representing a semantic textual description of said image, including visible objects, activities and scene context.

Then, \mathcal{L}_{dpp-c} can be computed as a loss term similarly to the original DPP term, in the following manner:

$$\mathcal{L}_{dpp-c} = -\log \left(\frac{\det \mathbf{P}(\mathbf{s})}{\det \mathbf{P} + \mathbf{I}} \right), \quad (6)$$

where $\mathbf{P} \in \mathbb{R}^{N \times N}$ is a similarity matrix between every two final hidden states of the LSTM in P and $\mathbf{P}(\mathbf{s})$ is a smaller square matrix cut down from \mathbf{P} given \mathbf{s} . \mathcal{L}_{dpp-c} is also used to update θ_s .

Evidently, \mathcal{L}_{dpp-c} induces a different kind of semantically informed diversity into the computed summary, in comparison to original \mathcal{L}_{dpp} . The proposed method simply consists in adding \mathcal{L}_{dpp-c} to the pool of the employed loss terms while training the complete summarisation DNN model. After training is finished, P may be completely removed from the architecture; thus there is zero runtime overhead in inference mode.

3.4.3. Evaluation

In order to evaluate the proposed method, the implementation [1] of the adversarial reconstruction framework (SUM-GAN-AAE) was adopted as a baseline. The reason behind this choice was solely practical; *in principle, the proposed method can be used to augment any other variant of the general framework, as well.*

The employed image captioner was comprised of a typical Encoder-Decoder architecture. The Encoder was a ResNet-152 CNN [54], pretrained for whole-image classification on the generic ImageNet dataset [55]. The CNN produces a 2048-dimensional vector representation capturing the semantic, object-centric content of the input image. Subsequently, this is fed to the LSTM Decoder, in order to predict a textual, natural-language caption for the given image. The LSTM is temporally unfolded for K time instances, where K is the maximum caption length (in words).

Evaluation was conducted on two publicly available, commonly used datasets: TVSum [52] and SumMe [53]. Each one was partitioned into 5 random splits, using a 80%-to-20% ratio for training and testing, respectively. The typically used F-Score metric was employed for performance evaluation, as is common in the literature. Table 3 depicts F-Score results for several recent DNN-based unsupervised key-frame extraction methods, given a sparsity percentage of $\sigma = 15\%$. The reported final figure is the mean F-Score performance across the 5 validation set splits. Evidently,





Method	TVSum	SumMe
SUM-FCN _{unsupervised} [44]	52.7%	41.5%
DR-DSN [45]	57.6%	41.4%
EDSN [46]	57.6%	42.6%
Unpaired VSN [47]	55.6%	47.5%
PCDL [48]	58.4%	42.7%
SUM-GAN-sl [49]	58.4%	47.8%
Cycle-SUM [50]	57.6%	41.9%
ACGAN [51]	58.5%	46.0%
SUM-GAN-AAE [1]	58.3%	48.9%
Proposed Method ([1] + \mathcal{L}_{dpp-c})	62.6%	56.9%

Table 3. Comparative study against competitive unsupervised learning methods. The metric used here for evaluation is the F-score. Bold indicates the best results.

augmenting the baseline codebase of [1] with the proposed method during training, gives rise to significant F-score gains.

This work has been submitted as a conference paper.

3.4.4. WP8 Use cases Contributions

Relevant WP8 Use Cases: 1D2 (Audiovisual Item Management in Verification Archives), 3A1 (Informative Content exploitation). This method contributes to use cases 1D2 and 3A1, by providing algorithms supporting easy video storage and browsing. These are needed by journalists and programme producers.

3.4.5. Relevant Publications

- M. Kaseris, C. Aslanidou, I. Mademlis, I. Pitas, "Exploiting Caption Diversity for Unsupervised Video Summarisation", technical report, submitted as conference paper

3.5. Joint optical flow and instance segmentation

Contributing partners: JR

3.5.1. Analysis of current state of the art for joint optical flow & instance segmentation

JR did an analysis of the current state of the art for joint optical flow & instance segmentation, taking into account combined methods as well as methods performing only one task and available datasets for training. Based on this analysis, JR will research a first prototype for joint optical flow & segmentation.

The term Optical Flow describes the motion of each pixel in an image relative to another image and was introduced in 1981 by Horn and Schunck [56]. This method is based on energy minimization, and is thus computationally expensive. Early methods, like the Lucas-Kanade Registration algorithm [57] have been replaced by deep learning methods. Some of these architectures make use of CNNs, like FlowNet [58] and its improved version FlowNet 2.0 [59]. But as the FlowNet2 method has a quite large model, it has a quite high memory footprint and is more prone to overfitting. A





big improvement has been achieved with the PWC-Net [60]. It makes use of warping, pyramids and cost volume and achieves a higher performance and accuracy than both FlowNet variants by having a more compact model. Further improvements were made in the RAFT [61] paper, which introduces an encoder-decoder based transformer [62] model. The latest state-of-the-art method is presented in the GMA [63] paper. It is based on RAFT [61] and additionally introduces global motion aggregation. There, it is assumed that all points which belong to the same object are moving into the same direction. As a result, especially occlusion handling in the two-frame setting is significantly improved.

Semantic Segmentation defines the process of assigning each pixel in an image to a certain class with a certain label. Instance Segmentation is a special case of Semantic Segmentation, where a distinction between different in-class instances is made additionally. Well performing methods like Mask R-CNN [64] use a two stage approach, where candidate regions of interest (ROIs) are generated in the first and then classified and segmented in the second stage. This has the disadvantage that the computation time is quite high and far away from real time (≥ 30 fps). Another approach for Instance Segmentation has been proposed by Wang et al. [65]. There, the SOLO [66] algorithm, where center locations and object sizes are taken into account for performing the segmentation, was refined. This made it possible to solve the problem more efficiently and thus, finally doing Real-Time Instance Segmentation with the SOLO approach. YOLACT [67] proposed to split the instance segmentation problem into two parallel tasks, which are the generation of a set of prototype masks and a prediction of the mask coefficients per instance. This is the first real-time instance segmentation approach on the MS COCO [68] dataset, but with worse performance than the slower Mask R-CNN algorithm in terms of accuracy. The YOLOACT algorithm has been refined further in YOLACT++ [69], which came up with deformable convolutions in the backbone and a fast mask re-scoring branch. SipMask [70] has been released after the YOLACT [67] paper and states that the lack of accuracy in YOLACT has its reason in the loss of spatial information, and tackles this problem by preserving it per bounding box. Also, the prediction of a mask is split into multiple predictions of sub-masks. As a result, the delineation of spatially adjacent objects has been improved compared to YOLACT. This is currently the best performing approach for real-time Instance Segmentation.

Multi task learning [71] is a generalization improving approach, which uses the domain information contained in two related tasks during training of two or more related tasks as inductive bias. A shared representation is used for learning those tasks in parallel. In an optimal case, this leads to better results for all of tasks, compared to doing the training individually. Although there has been no research regarding joint learning of Optical Flow and Instance Segmentation in a real-time scenario published yet, there are some approaches which combine Optical Flow with related tasks like Semantic Segmentation [72] [73] [74], and both of them in combination with Disparity Estimation [75]. Also, there has been research in the direction of combining Disparity Estimation with Instance Segmentation [76]. Hur and Roth [72] use a Superpixel approach, published by Yao et al. [77] for estimating the Optical Flow, combined with a FCN [78] for performing Segmentation. SegFlow [73] uses a fully connected CNN-based architecture for the Semantic Segmentation branch and a FlowNet [58] based architecture for the Optical Flow branch. In the paper it is shown that joint learning of both Optical Flow and Semantic Segmentation clearly improves the results of both tasks compared to learning them separately. Ding et al. [74] also perform Occlusion Estimation and use an Encoder-Decoder based structure with a shared encoder and a separate decoder for each, Flow estimation and Segmentation. This methodology outperforms all tasks in related settings. SENSE [75] additionally performs Occlusion and Disparity estimation. It is also based on an Encoder-Decoder architecture, having a shared encoder and separate decoders for each of the tasks. Also, the decoders are designed in a plug and play architecture, where one or more tasks can simply be removed or added from learning or inference.





With regards to available datasets, one of the most often used training datasets for Optical Flow is the Sintel [79] dataset. It is an animated short film which has explicitly been created for Optical Flow evaluation and is known for being challenging for current estimation methods. Also, the KITTI [80] dataset is well known for being widely used in Optical Flow evaluation. Flying Chairs [58] is a dataset which has an Optical Flow ground truth and has been extended to Flying Chairs 2 [81] with additional modalities. For Real-Time Instance Segmentation, mainly the MS COCO [68] dataset is used for evaluation.

3.5.2. Contributions to WP8 Use cases

Relevant WP8 Use Cases: 3C2-6 (Video object recognition), 3C2-7 (Video object localisation). The instance segmentation can contribute to use-cases 3C2-6 and 3C2-7, by recognizing and localizing (with a bounding box) objects in an image.

3.6. End-to-End Tools to Simplify the Creation, Curation and Usage of Data Sets for AI Applications

Contributing partners: RAI

3.6.1. Method Overview

RAI worked on the study of end-to-end tools to simplify the creation, curation and usage of data sets for AI applications. The work is supported by the observation that data sets for real-life tasks are scarce, expensive to produce and (often) inaccurate. A common idea to solve this problem is to exploit broadcaster's archives as a source of ground truth and hence of data sets. However, this approach has many advantages as well as disadvantages [82]. On the one hand, archival metadata are curated and checked by professionals, thus ensuring high-quality and quantity annotations. On the other hand, some important issues must be carefully considered. First, archive's metadata are stratified over many decades and compliant to different description models evolved over time. Even in the case in which information schemes are shared, they may be used following distinct criteria by different teams. Next, documentalists may interpret and apply annotation criteria differently, making difficult to link content annotated by multiple people. Finally, long to mid-term variations of documentation budget can influence the detail and depth of annotations, resulting in heterogeneous metadata even for the same content genre. As an example, archival metadata tell us that, e.g., some people are framed within a certain video segment, but no information is provided about the exact positions in time and space where their faces appear.

To address the aforementioned issues, a hybrid approach was investigated, using archival annotations, archive content and external resources (e.g., Web knowledge bases) as key assets. As representative case studies, two application scenarios were considered, namely *Face Management*, and *Landmark/Work-of-Art Detection*, which are overviewed in the following subsections. This is an on-going work for which preliminary results are presented later in Subsec. 3.6.4. Publication in relevant conferences and journals in the area of computer applications will be considered.

3.6.2. Face Management

Face management deals with the problem of detecting, clustering and (possibly) identifying facial images. Various approaches for unconstrained face recognition in videos have already been proposed [83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94], but several further issues must be considered.





To our knowledge, only few benchmark data sets have been publicly released to address this task. Even if huge in size, collecting in some cases millions of images, they suffer from some intrinsic limitations that impact on their usability. First, almost all of them focus on very popular celebrities only. However, it is often desirable to be able to treat less familiar entities like e.g., minor league players, supporting performers or emerging people. Next, they have been designed in terms of either breadth (i.e., many people but few images for each person) or depth (i.e., few people but many images for each person) of data, when they should instead combine both [95]. Moreover, they are prone to technological (e.g., camera settings or lighting conditions) or demographic (e.g., ethnicity, gender, age) biases that negatively impact the ability of AI models to generalise across data sets [96]. Finally, existing data sets are affected by incorrect annotations, which dramatically increase along the data set size [97].

Broadcasters' archives can help moving forward in this research endeavour. The workflow under investigation runs as follows. First, given an input video (e.g., an episode of a fiction series), the list of people involved in the video (e.g., the cast) is extracted with the help of the archive documentation or other knowledge bases. Next, a set of images depicting those people is collected from the Internet and used as reference gallery. Then, face detection, embedding and clustering tasks are performed to group faces depicted in the input video. To this purpose, several state-of-the-art face analysis and data clustering algorithms [98, 99, 100, 101, 102] have been studied and experimented to find the most appropriate solution. Finally, clustered faces are used as a probe set to be matched against the reference gallery, with the help of a state-of-art fast approximate nearest neighbor search algorithm [103].

Exploiting the richness, breadth and diversity of the archives, the workflow presented above may be used to construct more balanced data sets from a variety of input videos.

3.6.3. Landmark/Work of Art Detection

A similar approach has been adopted to address the landmark/work-of-art detection task. Landmark/work of art detection is a key task in multimedia management since it allows to enrich content with features that are tightly linked to certain users' needs. This is particularly true for genres like art, culture and tourism, where the information about monuments, sculptures, paintings and human-made landmarks are important for filtering and retrieval and for recommendation. Also in this case, broadcasters' archives are an important source of exemplary material, together with publicly available data repositories and services, such as Wikidata [104]. Unlike other kind of tasks, that of landmark recognition is one for which is quite difficult to build generic reference data sets since the relevance of detection is highly dependent on the content genre, the purpose of publication and the user context. For this reason, in the reference period, works have been focused on reviewing and studying automatic/semiautomatic methods for reference database construction rather than on the retrieval/matching technology. A visual model of the workflow under investigation is shown in Fig. 3.

3.6.4. Evaluation

As stated earlier, the face management tool is built on a deep learning pipeline, whose tasks include state-of-the-art algorithms for face finding, representation, grouping and labelling.

Finding the presence of faces in a video employs RetinaFace [98], a Feature Pyramid Network (FPN) for accurate face detection and 2D face alignment. The authors of RetinaFace showed that their method is robust, fast and highly accurate in estimating the position of faces even under critical conditions such as pose variations, illumination changes and occlusions.



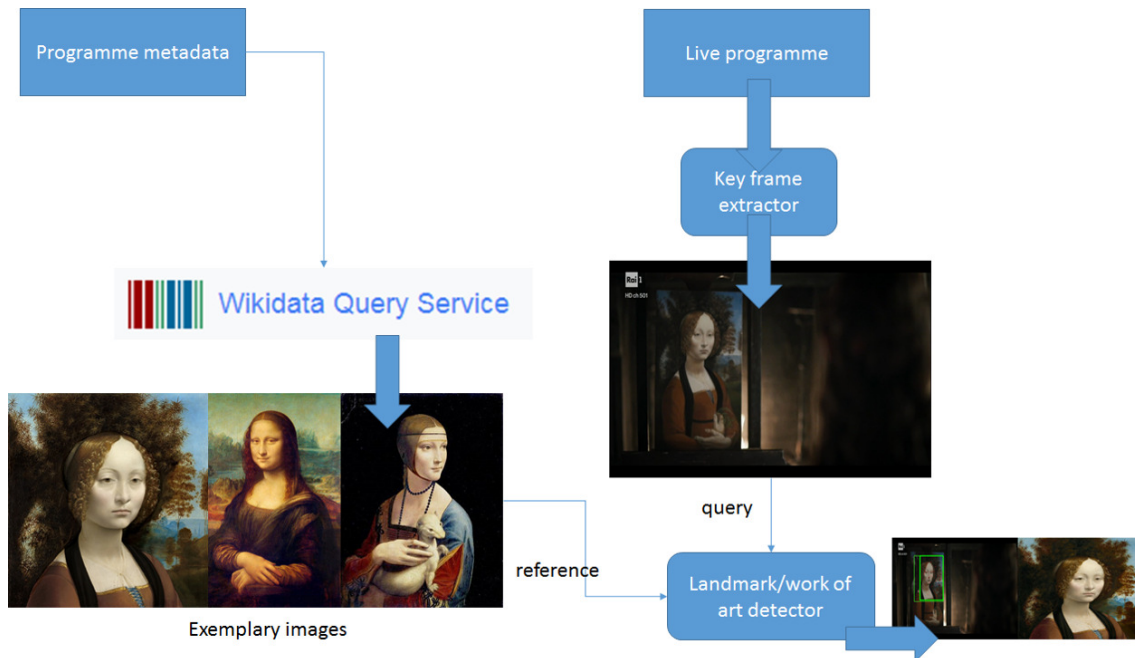


Figure 3. Automated creation of reference data sets for landmark/work-of-art recognition.

Face representation uses ArcFace [99], a Deep Convolutional Neural Network (DCNN) that implements an Additive Angular Margin Loss to obtain highly discriminative power of face feature embeddings. The authors of ArcFace demonstrated that their approach outperforms the best face recognition methods.

Face grouping first creates a graph of connected faces, where the nodes of the graph are the ArcFace embeddings, and the edges are the Cosine similarity between them. Then, it applies the Chinese Whispers graph clustering algorithm [102] to find groups of similar faces. The Chinese Whispers algorithm was chosen among others due to its ability to handle clusters of different sizes, densities and shapes in noisy high dimensional data, without the need of specifying any custom parameters. Initial qualitative studies show that this clustering approach is very promising, being able to group face images of the same person over different conditions like size, pose, illumination, make up and occlusions (e.g., glasses, caps, masks).

Lastly, face labelling applies a retrieval-based open-set face identification strategy to assign each cluster the identity of the corresponding person. In biometric applications, the objective of open-set identification is to correctly identify probe faces that are present in a gallery of reference faces, while rejecting probe faces that do not belong to the gallery. This is implemented through the Hierarchical Navigable Small World (HNSW) library [103], an efficient algorithm to perform approximate K-Nearest Neighbor (KNN) search. The capability of identifying people within the clusters was tested using a gallery of 66 RAI newsreaders, and a probe set of about 10,000 faces detected in RAI newscasts. The Cosine similarity was set as the distance metric. The performance was measured computing the Detection and Identification Rate (DIR) versus the False Alarm Rate (FAR) [105] for the rank K equal to one and Cosine similarity varying from zero to one (see Fig. 4).

Regarding the landmark/work of art detection task, initial experiments have been done using WikiData SPARQL endpoint to retrieve reference still pictures data sets and on a combination of state-of-the-art content analysis and management tools to build the detector, like MPEG CDVS

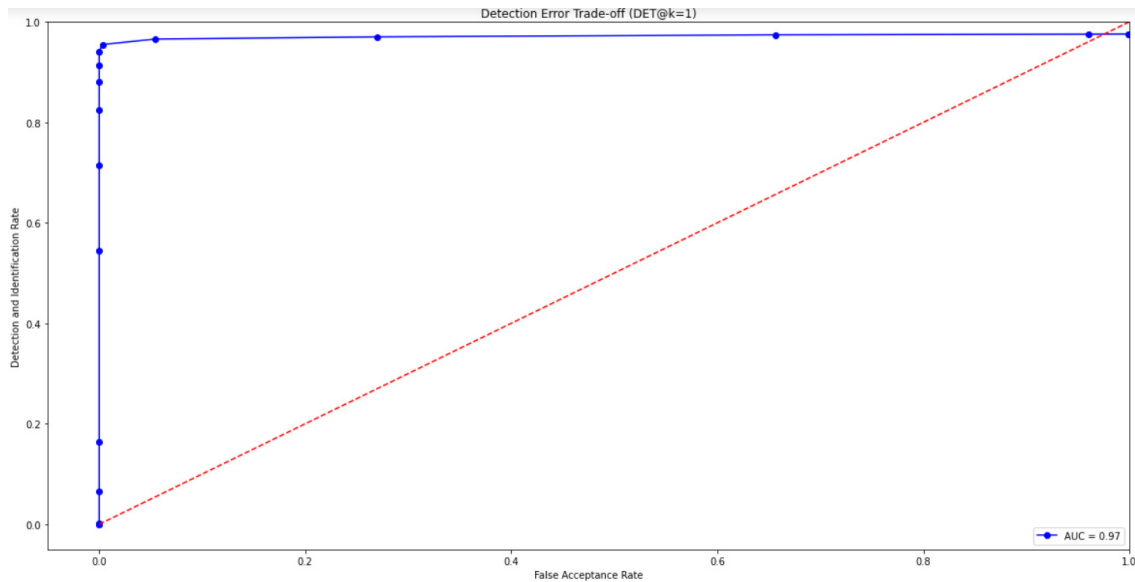


Figure 4. DIR vs. FAR curve describing the trade-off for rank one identification and false alarms for the face labelling task. The red dotted line represents a system that is no better than random guessing. The solid blue line represents the measured values. The AUC (Area Under the Curve) score is 0.97, denoting an excellent performance. The best balance between DIR and FAR is obtained for Cosine similarity equal to 0.4.

reference model [106], OpenCV and FFMpeg libraries.

3.6.5. Contributions to WP8 Use cases

Relevant WP8 Use Cases: 3A1 (Informative Content exploitation), 3A3 (Archive exploitation). The presented approach is directly related to use-cases 3A1 and 3A3, by providing a methodology for the detection and labelling of regions of interest (e.g., faces, paintings, monuments) within TV programme streams.

3.6.6. Relevant External Resources

- [Wikidata Query Service](#)
- [Wikidata SPARQL tutorial](#)

3.7. Learning and Reasoning for Cultural Metadata Quality

Contributing partners: 3IA-UCA

3.7.1. Method Overview

An important objective for 3IA-UCA in T5.1 is to combine knowledge representations with deep representation to design new symbolic and non-symbolic information retrieval engines. In their recent article [107], Anna Bobasheva, Fabien Gandon and Frédéric Precioso have shown how to couple symbolic AI and machine learning over a semantic Web knowledge graph to support museum curators in improving the quality of cultural metadata and information retrieval. The developed methods create a data pipeline above the data and metadata of the cultural collection, which





produces and reasons on its RDF [108] knowledge graph, trains a Convolutional Neural Network image classification model, makes prediction for the entire collection and expands the metadata to be the base for the SPARQL [109] search queries and curation techniques. They also developed methods to discover the new contextual relationships between the concepts in the metadata, and improved the model prediction scores based on the semantic relations. Their results show that cross-fertilization between symbolic AI and machine learning can indeed provide the tools to address the challenges of the museum curators describing the artwork pieces and searching for the relevant images.

Our overall architecture is designed to combine knowledge representation and reasoning methods (semantic reasoning and querying on RDF knowledge graphs) and machine learning methods (deep learning for images) in the management of a single visual art dataset documenting a large cultural collection. This is the keystone of this work as it enables to combine, enrich or contrast results from reasoning on the symbolic metadata, with Resource Description Framework (RDF) and Simple Knowledge Organization System (SKOS) [110] annotations of the collection, and learning on the symbolic data, with images of the collection.

3.7.2. Learning and Reasoning for Cultural Metadata Quality

This work has been carried out on the art Joconde dataset [111]. The original Joconde dataset metadata is stored in a type of database specific to RDF data called triplestore. Triplestores provide a mechanism for the storage and retrieval of RDF graphs through semantic queries (in the SPARQL language) and may support other types of intelligent processing including inferences and validation. This work's proposal is to extend this dataset with the results of image classification and the results of semantic reasoning by relying on the triplestore as an integration point in one unified knowledge graph. To do so, the data processing workflow shown in Fig. 5 was designed and evaluated. There is a two-pass dataflow for training and for scoring. For the training pass, the triplestore is queried using the SPARQL language to create the labeled image set for training, validation, and testing, benefiting from the results of the inferences that augmented the knowledge graph and, in particular, the available labels. The images and labels are selected based on criteria specific to the curators' needs. A CNN model is then fine-tuned on the training and validation sets, and the model performance is assessed on the test set.

For the scoring pass, we query again but this time with different constraints to create a dataset on which we run the fine-tuned classifier and obtain prediction scores for every class for every image. We create new triples associating the image with prediction scores. These results are represented in RDF and are stored back in the triplestore to be integrated and put in use with all the other metadata.

As a result, we created an extended knowledge graph that allows the ontology-based image search with quantified relevance of the search term. On top of this pipeline, we can then perform analytics queries leveraging all the annotations and their semantics, and design SPARQL queries to look for anomalies in the annotations.

By running the model on all the images of the Joconde dataset, we obtain the prediction scores for every image. We link these scores with the artwork records by saving the scores in the same RDF format as the initial metadata using a vocabulary we designed for this purpose. As a result, the RDF knowledge graph contains all the initial data plus all the classification results. The analysis of these results can therefore leverage semantic Web reasoning and querying capabilities in the formulation of analytic queries.

In Fig. 6, we present results of querying the extended metadata to detect the noise in the existing image annotations. In these examples, the query searches for the images that have the concept *cheval* (horse) or related concept *cavalier* (horseman) but have low prediction scores (\leq



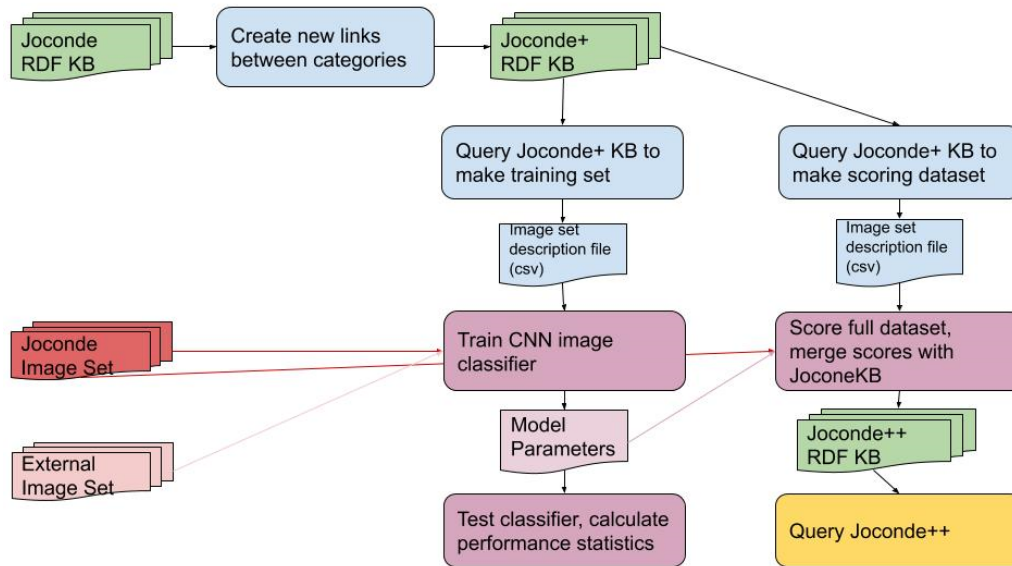


Figure 5. The data processing pipeline combining symbolic AI and machine learning to improve the quality of cultural metadata and information retrieval.

0.20). On the first row there is no visible horse on the sculpture, on the second row the horseman is barely visible on the background, and on the third row the horse is small and, although it cannot be ignored, it should have a low significance. All these examples are cases where a curator may want to revise and adjust the metadata.

The complete semantic annotation of an artwork in the collection provides a context that can help identify suspiciously present or missing concepts. We consider that the probability of appearance of a given concept in a context-similar pair should improve the probability score of the second concept in the same image. For example, a high classification probability score for a *bateau* (boat) should influence the score of the concept *mer* (sea). To achieve this, we used a logistic regression approach to build a pairwise regression predictor of appearance of a concept based on the presence of another concept in the same annotation of an art piece. More precisely, the regression estimates the log-odds of observing a concept *A* when a concept *B* is present, compared to situations when concept *B* is not present. An example of results is shown in Fig. 7.

To conclude, the pipeline eventually provides an environment to combine symbolic reasoning and sub-symbolic learning over a knowledge graph that integrates the inputs and outputs of all the methods to improve information retrieval and curation tools.

3.7.3. Contributions to WP8 Use cases

Relevant WP8 Use Cases: 1D1-2 (Audio/Video Search By Keyword (Content Based)). This work is only partly related to use-case 1D (Audiovisual Item Search in Verification Archives) because it has been carried out on an image database, not on video or audio data. However, its goal is to leverage Machine Learning to improve and validate datasets described with their semantics listed as keywords in a knowledge graph. In this sense, it is related to 1D1-2.





Image	Joconde Metadata	Prediction Scores
	<p><i>cheval</i> (horse)</p> <p><i>figure (saint Eloi de Noyon, évêque, en pied, bénédiction, vêtement liturgique, mitre, attribut, cheval, marteau, outil : ferronnerie)</i> (figure (Saint Eloi de Noyon, bishop, standing, blessing, liturgical vestment, mitre, attribute, horse, hammer, tool: ironwork))</p> <p>000SC022652</p>	<p><i>cheval</i> (horse): 0.006</p>
	<p><i>cheval</i> (horse) <i>cavalier</i> (horseman)</p> <p><i>figures bibliques (Vierge à l'Enfant, à mi-corps, assis, Enfant Jésus : nu, livre);fond de paysage (colline, cours d'eau, barque, cavalier)</i> (biblical figures (Virgin and Child, half-body, seated, Child Jesus: nude, book);landscape background (hill, river, boat, horseman))</p> <p>000PE027041</p>	<p><i>cheval</i> (horse): 0.009</p>
	<p><i>cheval</i> (horse)</p> <p><i>scène (satirique : Bismarck Otto von : Gargantua, repas, cheval, boisson : vin)</i> (scene (satire: Bismarck Otto von: Gargantua, meal, horse, drink: wine))</p> <p>5002E006121</p>	<p><i>cheval</i> (horse): 0.011</p>

Figure 6. Examples of noise detection in the images that do not have a visually relevant term *cheval* (horse) with the prediction scores below 0.2.


Image	Labels & Metadata	S(mer)	S(bateau)	S(mer) _{adj}
	<p><i>bateau</i> (boat)</p> <p><i>paysage (Le Havre, bateau à voiles, crépuscule, soleil)</i> (landscape (Le Havre, sailing boat, twilight, sun))</p>	0.8985	0.9981	0.9307

Figure 7. Example of the artwork with the adjustment of prediction score of concept *mer* (sea) by the prediction score of concept *bateau* (boat).





3.7.4. Relevant Publications

- A. Bobasheva, F. Gandon and F. Precioso. Learning and Reasoning for Cultural Metadata Quality. Submitted for publication in ACM Journal on Computing and Cultural Heritage, 2021.

3.7.5. Relevant External Resources

- [Official portal for the Joconde database](#)
- [Wikipedia page for the Joconde database](#)





4. Learning from scarce data

This Section presents work conducted in the context of T5.3, relating to learning in the face of data scarcity (insufficient data, different target domain etc.). There are different ways to approach this problem (like few-shot learning, domain adaption or semi-supervised learning), therefore the methods presented in this section differ considerably. JR and UPB worked on different aspects of few-shot object detection (see subsections 4.1 and 4.4). While JR focused in the first period on facilitating the training process by controlling it based on the available novel data and enable incremental training (updating the dataset and learned model multiple times with novel user-provided classes), UPB focused on positive sample augmentation during training and the use of ensembles of few-shot object detectors. Both groups use the framework [112] as a basis, which makes it easy to combine the results. Subsection 4.2 discusses the activities of CNR on unsupervised domain adaptation for traffic density estimation and counting, which can be directly linked to WP3 research, while this is followed by CNR work on the VISIONE system, i.e., a novel video browsing and search system relying on textual representations and a text retrieval engine. Subsection 4.5 continues with UNIFI research on a novel method for semi-supervised learning of Fine-Grained Visual Categorization (FGVC) using adversarial training. The following subsection 4.6 presents QMUL research concerning deep clustering with diversity-enforcing constraints, which is combined with a clustering aggregation approach. Clusters can be exploited as a source of pseudolabels, in case ground-truth annotations are not available. Subsection 4.7 presents a deep neural architecture by UNITN which combines deep representation learning and dictionary learning into a unified approach, by replacing the convolutional layers of a CNN with novel dictionary learning and coding neural layers. Finally, subsection 4.8 concludes with a novel UNITN method on self-paced curriculum learning for unsupervised domain adaptation in the object detection task, which relies on Generative Adversarial Networks (GANs).

4.1. Few-shot object detection: facilitating training

Contributing partners: JR

4.1.1. Method Overview

The work of JR in this task focuses on few-shot object detection serving use cases in annotating incoming material in media production or for archiving. We can observe three main aspects, where the setup of benchmarking problems (and thus the methods described in literature, as well as the existing implementations) deviate from the practical requirements of using few-shot object detection in media use cases:

- The typical setup of the problem is posed as n -way k -shot, i.e. a problem with n classes and k samples per shot. However, in practice the number of samples per class that are provided may differ.
- There is not fixed predefined dataset, but the set for base classes will contain a mixture of third party and maybe own data for some classes, while the novel classes are mined from own or third party media content (e.g., web sites). Thus the concept of a dataset is fluid, and the available data will evolve over time.
- Classes need to be added incrementally, which requires creating balanced training sets, but approaches should aim to keep the training effort low. This again means that there is no fixed notion of a dataset, but it needs to be updated on the fly.





Taking these aspects into consideration, we decided to focus on metric or contrastive than meta-learning type of approaches. In addition, we are interested to use a framework, which can be applied to object detection as well as segmentation. Methods of interest are thus [113], which uses FPN to create an object detection pipeline using metric learning. Classification is done different for pretrained classes, while few-shot learning is done with FPN (in the DCN variant) instead. [114] 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 reweighting of those features is proposed in [115]. A recent work applies fine-tuning only 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, using Faster R-CNN as a backbone [116]. It has been shown that it can outperform meta-learning approaches [112].

We are interested in a framework that can potentially also be used with single-stage detectors and extended to support segmentation. We are thus using [112] 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 trained 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 branch 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.

Facilitating training We aim to drive the few-shot training process by the available data in the process, so that few-shot training can be deployed as a service to be integrated in a media analysis toolchain. The expected input to such a service are a set of samples and corresponding annotations, as well as a small configuration file, which describes the base model to be used, the data locations and whether all or just some classes of the new data shall be used for training. The samples and annotations may be entirely user supplied (i.e., manually annotated), or may result from a semi-automatic process. Such a process may use weakly supervised object detection and tracking, where the user only coarsely identifies the object in one frame and provides a class label, but the bounding boxes (or masks) are determined automatically by segmenting the object throughout a sequence.

We have extended the framework of [112] with a tool that dynamically generates datasets and drives the training process. In particular, the tool covers the following steps:

- Determine the base and novel classes from the provided annotations. For both the base and novel classes only a subset may be actually used in the training. This provides more flexibility in the process, and is also required to support incremental training without splitting the source annotation files.
- Determine how many instances are available, and set up the k -shot n -way problem accordingly, with $k = \min(k_1, \dots, k_n)$.
- Prepare model structures for novel only training and fine-tuning of the combined base+novel model by adjusting the layer sizes to match the number of classes in the different sets. This includes scaling up the number of classes arbitrarily, which goes beyond the current functionality of the framework, that assumes a split of fixed number of classes.



- If the number of samples strongly varies, set up multiple training problems to make best use of the data. This is implemented by specifying a factor q , which causes a split if half of the classes have q times more samples than the one with the fewest samples, i.e., $\text{median}(k_1, \dots, k_n) \geq q \min(k_1, \dots, k_n)$.

The tool currently supports annotations in COCO format. However, this does not mean that COCO is required as a base model, as long as the annotations are provided in this format. The training tool has been tested with a pretrained based model containing 60 COCO classes, and adding a subset of 20 novel classes selected from the LVIS dataset. This example training setup is provided with the code.

Towards incremental training The proposed tool can also be used for incremental training of new classes. The training tool provides as additional output the dataset annotation files that are required to use the resulting model as a base model. Similar to the splitting of the training step in case of unequal number of samples, also different incremental training steps can use different values for k .

However, due to the two-stage fine-tuning (TFA) approach in the framework, the fine-tuning step is run for all classes, including the base classes and of novel classes from previous iterations. This step is typically computationally more expensive than the training process for novel classes, in particular, if the number of classes in an incremental training step is small. Thus the runtime saving in incremental training will be less than the fraction of the added classes to all novel classes. A recent paper proposes to avoid running this fine-tuning step after incremental training [117], but at the expense of training an instance feature embedding, and requiring access to these features at inference stage.

4.1.2. Contributions to WP8 Use cases

Relevant WP8 Use Cases: 3C2-6 (Video object recognition), 3C2-7 (Video object localisation). 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.

4.1.3. Relevant External Resources

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

4.2. Domain Adaptation and Counting

Contributing partners: CNR





4.2.1. Method Overview

Convolutional Neural Networks have produced state-of-the-art results for a multitude of computer vision tasks under supervised learning. However, the crux of these methods is the need for a massive amount of labeled data to guarantee that they generalize well to diverse testing scenarios. In many real-world applications, there is indeed a large domain shift between the distributions of the train source and test target domains, leading to a significant drop in performance at inference time. Unsupervised Domain Adaptation (UDA) is a class of techniques that aims to mitigate this drawback without the need for labeled data in the target domain. This makes it particularly useful for the tasks in which acquiring new labeled data is very expensive, such as for semantic and instance segmentation. An end-to-end CNN-based UDA algorithm for traffic density estimation and counting was developed, based on adversarial learning in the output space. The density estimation is one of those tasks requiring per-pixel annotated labels and, therefore, needs a lot of human effort. Experiments considering different types of domain shifts were executed, and two new datasets for the vehicle counting task were made publicly available, also used for our tests. One of them, the Grand Traffic Auto dataset, is a synthetic collection of images, obtained using the graphical engine of the Grand Theft Auto video game, automatically annotated with precise per-pixel labels, which represent a relevant solution to address applications with scarce data. Experiments show a significant improvement using our UDA algorithm compared to the model's performance without domain adaptation.

This activity is also related to T3.3 (Transfer Learning), and T3.7 (Learning to Count). This activity was developed in synergy with the AI4EU project and it has already been uploaded in the AI on Demand Platform. In AI4Media the learning with scarce data issue, which was addressed leveraging on solutions of domain adaptation, was particularly taken into consideration.

4.2.2. Overall approach

Our method relies on a CNN model trained end-to-end with adversarial learning in the output space (i.e., the density maps), which contains rich information such as scene layout and context. The peculiarity of our adversarial learning scheme is that it forces the predicted density maps in the target domain to have local similarities with the ones in the source domain.

Figure 8 depicts the proposed framework consisting of two modules: 1) a CNN that predicts traffic density maps, from which estimate the number of vehicles in the scene was estimated, and 2) a discriminator that identifies whether a density map (received by the density map estimator) was generated from an image of the source domain or the target domain.

In the training phase, the density map predictor learns to map images to densities based on annotated data from the source domain. At the same time, it learns to predict realistic density maps for the target domain by trying to fool the discriminator with an adversarial loss. The discriminator's output is a pixel-wise classification of a low-resolution map, as illustrated in Figure 8, where each pixel corresponds to a small region in the density map. Consequently, the output space is forced to be locally similar for both the source and target domains. In the inference phase, the discriminator is discarded, and only the density map predictor is used for the target images.

Density Estimation Network: The counting task is formulated as a density map estimation problem [118]. The density (intensity) of each pixel in the map depends on its proximity to a vehicle centroid and the size of the vehicle in the image so that each vehicle contributes with a total value of 1 to the map. Therefore, it provides statistical information about the vehicles' location and allows the counting to be estimated by summing of all density values.

This task is performed by a CNN-based model, whose goal is to automatically determine the vehicle density map associated with a given input image. Formally, the density map estimator,



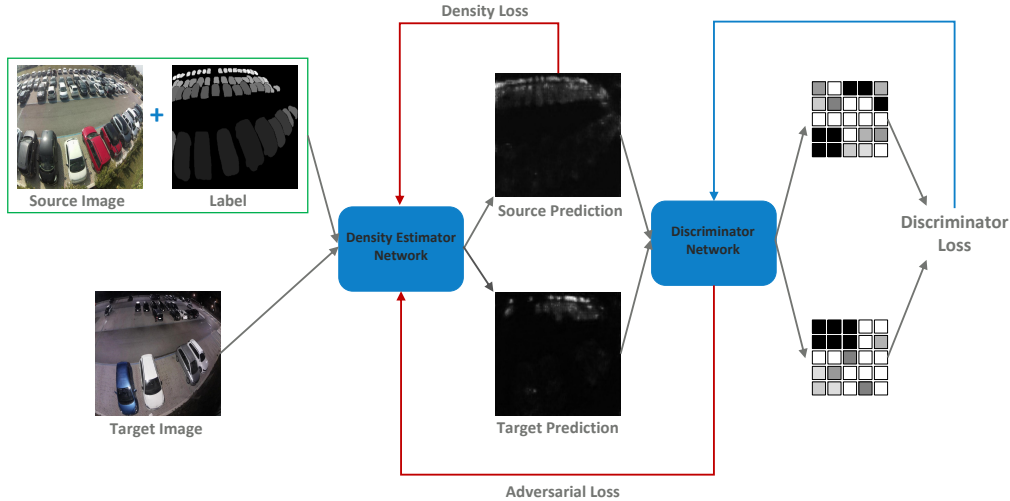


Figure 8. Algorithm overview. Given $C \times H \times W$ images from source and target domains, they are processed by the density map estimation network to obtain output predictions. A density loss is computed for source predictions based on the ground truth. In order to improve target predictions, a discriminator is used to locally classify whether a density map belongs to the source or target domain. Then, an adversarial loss is computed on the target prediction and is back-propagated to the density map estimation and counting network.

$\Psi : \mathcal{R}^{C \times \mathcal{H} \times \mathcal{W}} \mapsto \mathcal{R}^{\mathcal{H} \times \mathcal{W}}$, transforms a $\mathcal{W} \times \mathcal{H}$ input image \mathcal{I} with C channels, into a density map, $D = \Psi(\mathcal{I}) \in \mathcal{R}^{\mathcal{H} \times \mathcal{W}}$.

Discriminator Network: The discriminator network, denoted by Θ , also consists of a CNN model. It takes as input the density map, D , estimated by the network Ψ . Its output is a lower resolution probability map where each pixel represents the probability that the corresponding region (from the input density map) comes either from the source or the target domain. The goal of the discriminator is to learn to distinguish between density maps belonging to source or target domains. Through an adversarial loss, this discriminator will, in turn, force the density estimator to provide density maps with similar distributions in both domains. In other words, the target domain density maps have to look realistic, even though the network Ψ was not trained with an annotated training set from that domain.

4.2.3. Domain Adaptation Learning

The proposed framework is trained based on an alternate optimization of the density estimation network, Ψ , and the discriminator network, Θ . Regarding the former, the training process relies on two components: 1) density estimation using pairs of images and ground truth density maps, which is assumed to be only available in the source domain; and 2) adversarial training, which aims to make the discriminator fail to distinguish between the source and target domains. As for the latter, images from both domains are used to train the discriminator on correctly classifying each pixel of the probability map as either source or target.

To implement the above training procedure, two loss functions were used: one is employed in the first step of the algorithm to train network Ψ , and the other is used in the second step to train the discriminator Θ . These loss functions are detailed next.

Network Ψ Training: The loss function for Ψ is defined as the sum of two main components:





$$\mathcal{L}(\mathcal{I}^S, \mathcal{I}^T) = \mathcal{L}_{density}(\mathcal{I}^S) + \lambda_{adv} \mathcal{L}_{adv}(\mathcal{I}^T), \quad (7)$$

where $\mathcal{L}_{density}$ is the loss computed using ground truth annotations available in the source domain, while \mathcal{L}_{adv} is the adversarial loss that is responsible for making the distribution of the target and the source domain closer to each other. In particular, the density loss $\mathcal{L}_{density}$ is defined as the mean square error between the predicted and ground truth density maps, i.e. $\mathcal{L}_{density} = MSE(D^S, \hat{D}^{S-G^T})$.

To compute the adversarial loss \mathcal{L}_{adv} , the images belonging to the target domain are first forwarded through network Ψ , to generate the predicted density maps D^T . Then, D^T is forwarded through network Θ , to generate the probability map $P = \Theta(\Psi(\mathcal{I}^T)) \in [0, 1]^{H' \times W'}$, where $H' < H$ and $W' < W$. The adversarial loss is given by

$$\mathcal{L}_{adv}(\mathcal{I}^T) = - \sum_{h,w} \log(P_{h,w}), \quad (8)$$

where the subscript h, w denotes a pixel in P . This loss makes the distribution of D^T closer to D^S by forcing Ψ to fool the discriminator, through the maximization of the probability of D^T being locally classified as belonging to the source domain.

Network Θ Training: Given an image \mathcal{I} and the corresponding predicted density map D , D is fed as input to the fully-convolutional discriminator Θ to obtain the probability map P . The discriminator is trained by comparing P with the ground truth label map $Y \in \{0, 1\}^{H' \times W'}$ using a pixel-wise binary cross-entropy loss

$$\begin{aligned} \mathcal{L}_{disc}(\mathcal{I}) = & - \sum_{h,w} (1 - Y_{h,w}) \log(1 - P_{h,w}) + \\ & + Y_{h,w} \log(P_{h,w}), \end{aligned} \quad (9)$$

where $Y_{h,w} = 0 \forall h, w$ if \mathcal{I} is taken from the target domain and $Y_{h,w} = 1$ otherwise.

4.2.4. Evaluation

The proposed UDA method for density estimation and counting of traffic scenes was validated under different settings. First, the *NDISPark* dataset was used to test the *Day2Night* domain shift; then, the *WebCamT* and the *TRANCOS* datasets was used to take into account the *Camera2Camera* performance gap. Finally, the *GTA* dataset was used to consider the *Synthetic2Real* domain difference. For all the experiments, the evaluation of the models was carried out on three metrics widely used for the counting task: (i) Mean Absolute Error (MAE) that measures the absolute count error of each image; (ii) Mean Squared Error (MSE) that instead quantifies the squared count error for each image; (iii) Average Relative Error (ARE), which measures the absolute count error divided by the true count. Note that, as a result of the squaring of each error, the MSE effectively penalizes large errors more heavily than the small ones. Instead, the ARE is the only metric that considers the relation of the error and the total number of vehicles present for each image. Results are summarized in Table 4. Finally, some examples of the outputs obtained using our models were reported, showing their visual quality. In particular, Figure 9 shows the ground truth and the predicted density maps for some random samples of the considered scenarios.

4.2.5. Contributions to WP8 Use cases

Relevant WP8 Use Cases: 3A3 (Archive exploitation), 2B1 (Automatic metadata tagging), 3C2-6 (Video object recognition), 3C2-7 (Video object localisation), 4C2 (Moving Image (video))



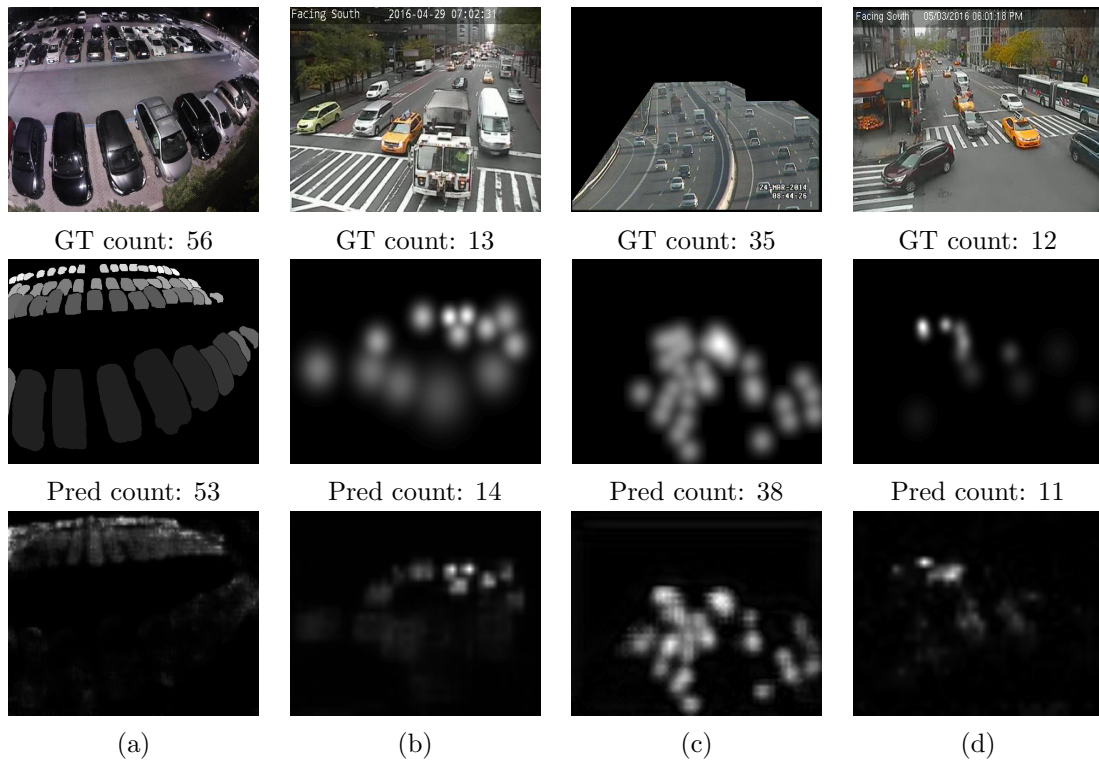


Figure 9. Examples of the predicted density maps in the considered scenarios: (a) Day2Nigh Domain Shift using the NDISPark dataset; (b) and (c) Camera2Camera Domain Shift employing the WebCamT and TRANCOS datasets, respectively; (d) Synthetic2Real Domain Shift using the GTA dataset for the training phase and the WebCamT dataset for testing on real images. In the first row, the input images are reported. In the second row, the ground truth, while in the third, the predicted density maps obtained with our models.





	MAE	MSE	ARE
<i>Day2Night Domain Shift - NDISPark Dataset</i>			
Baseline - CSRNet [119]	3.95	27.45	0.43
Our Approach	3.49	20.90	0.39
<i>Camera2Camera Domain Shift - WebCamT Dataset [120]</i>			
Baseline - CSRNet [119]	3.24	16.83	0.21
Our Approach	2.86	13.03	0.19
<i>Camera2Camera Domain Shift - TRANCOS Dataset [121]</i>			
Hydra-CNN [122]	10.99	68.70	0.71
FCN-MT [120]	5.31	-	0.85
LC-ResFCN [123]	3.32	-	-
Baseline - CSRNet [119]	3.56	30.64	0.10
Our Approach	3.30	23.60	0.08
<i>Synthetic2Real Domain Shift - GTA Dataset</i>			
Baseline - CSRNet [119]	4.10	25.83	0.28
Our Approach	3.88	23.80	0.27

Table 4. Experimental results obtained for the four considered domain shift. Three evaluation metrics were used: the Mean Absolute Error (MAE), the Mean Squared Error (MSE) and the Average Relative Error (ARE). Performance improvements was obtained in all the scenarios, considering all the three metrics.

analysis). Domain adaptation is applicable in any case where adapting an algorithm trained in one context, to perform properly in different contexts, is required. This is the case, for instance, of applications to automatic metadata generation and object recognition, where algorithms were trained in different domains than the target media where the tools are used.

4.2.6. Relevant Publications

- Domain adaptation for traffic density estimation, Ciampi, L., Santiago, C., Costeira, J.P., Gennaro, C., Amato, G., VISIGRAPP 2021 - Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, Volume 5, 2021, Pages 185-195, ISBN: 978-989758488-6, (Zenodo Record: <https://zenodo.org/record/5078270>)

4.2.7. Relevant External Resources

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

4.3. Video browsing and searching

Contributing partners: CNR

4.3.1. Method Overview

CNR has worked on the VISIONE system [124, 125], a video search system that allows users to search for videos using textual keywords, the occurrence of objects and their spatial relationships,





the occurrence of colors and their spatial relationships, and image similarity. These modalities can be combined together to express complex queries and meet users' needs. VISIONE preprocess videos (and video shots) to automatically generate all metadata needed to retrieve them. Annotations, features, cross-media information etc. is all automatically extracted, and no already existing metadata are needed. Metadata are generated at keyframe level, so video shot retrieval granularity is possible. An additional peculiarity of this approach is that we encode all information extracted from the keyframes, such as visual deep features, tags, color and object locations, using a convenient textual encoding that is indexed in a single text retrieval engine. This offers great flexibility when results corresponding to various parts of the query (visual, text and locations) need to be merged. The specially designed textual encodings for indexing and searching video content allows using the mature and scalable Apache Lucene full-text search engine [126], to index and retrieve non textual data.

CNR started working on the VISIONE system in 2019, when it participated in the Video Browsing Showdon (VBS) competition [127]. In the context of the AI4Media project, CNR has analyzed the VISIONE results and logs gathered at VBS 2019, in order to identify directions to improve the system. Therefore, it has investigated and integrated new features into the system, which in turn paved the way for the participation of VISIONE also in the 2021 edition of VBS (held in June 2021). The analysis of the VBS 2019 logs, briefly described here (Section 4.3.4), and fully discussed in [128], allowed us to grasp a lot of information on the system and to improve it without needing additional training data. The new functionalities integrated in the second release of VISIONE [125], which participated at VBS 2021, are briefly described in Section 4.3.5.

The approach used in VISIONE, leveraging on transfer learning properties, allows us indexing and searching a new unknown video dataset without prior training on the dataset itself. This activity is also related to T3.3 (Transfer Learning).

4.3.2. Overall approach

VISIONE is a visual content-based retrieval system designed to support large scale video search. It allows a user to search for a video describing the content of a scene by formulating textual or visual queries (see Figure 10).

VISIONE, in fact, integrates several search functionalities and exploits deep learning technologies to mitigate the semantic gap between text and image. Specifically it supports:

- *query by keywords*: the user can specify keywords including scenes, places or concepts (e.g., outdoor, building, sport) to search for video scenes;
- *query by object location*: the user can draw on a canvas some simple diagrams to specify the objects that appear in a target scene and their spatial locations;
- *query by color location*: the user can specify some colors present in a target scene and their spatial locations (similarly to object location above);
- *query by visual example*: an image can be used as a query to retrieve video scenes that are visually similar to it.

Moreover, the search results can be filtered by indicating whether the keyframes are in color or in b/w, or by specifying its aspect ratio.

4.3.3. System Architecture Overview

The general architecture of our system is illustrated in Figure 11. Each component of the system is described in detail in [128]; here we give an overview of how it works. To support the search functionalities introduced above, our system exploits deep learning technologies to understand and represent the visual content of the database videos. Specifically, it employs:



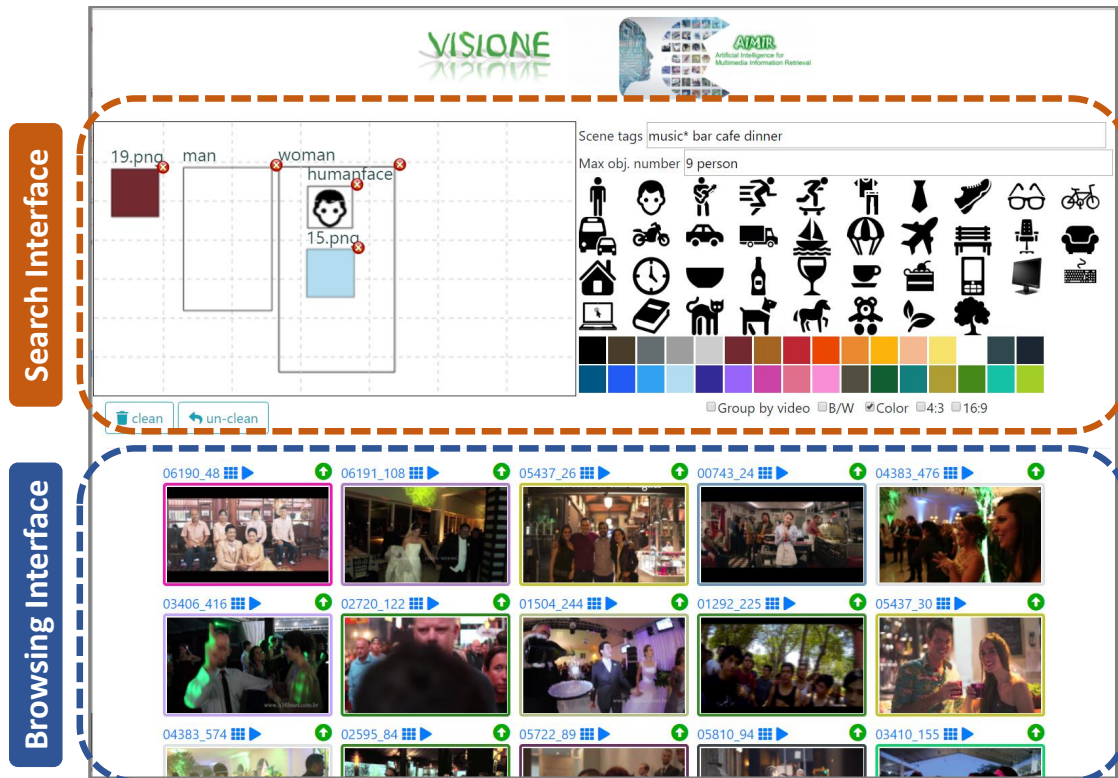


Figure 10. A screenshot of the VISIONE User Interface composed of two parts: the search and the browsing.

- an image annotation engine, to extract scene tags;
- state-of-the-art object detectors, like YOLO [129], to identify and localize objects in the video keyframes;
- spatial colors histograms, to identify dominant colors and their locations;
- the R-MAC [130] deep visual descriptors, to support the Similarity Search functionality.

The peculiarity of the approach used in VISIONE is to represent all the different types of descriptors extracted from the keyframes (visual features, scene tags, colors/object locations) with a textual encoding that is indexed in a single text search engine. This choice allows us to exploit mature and scalable full-text search technologies and platforms for indexing and searching video repository. In particular, VISIONE relies on the Apache Lucene full-text search engine [126].

Also the queries formulated by the user through the search interface (e.g., the keywords describing the target scene and/or the diagrams depicting objects and the colors locations) are transformed into textual encoding, in order to process them. We designed a specific textual encoding for each typology of data descriptor as well as for the user queries.

In the full-text search engine, the information extracted from every keyframe is composed of four textual fields, as shown in Figure 11:

- *Scene Tags*, containing automatically associated tags;
- *Object&Color BBoxes*, containing text encoding of colors and objects locations;
- *Object&Color Classes*, containing global information on objects and colors in the keyframe;
- *Visual Features*, containing text encoding of extracted visual features.

These four fields are used to serve the four main search operations of our system:

- *Annotation Search*, search the Scene Tags field for keyframes associated with specified anno-



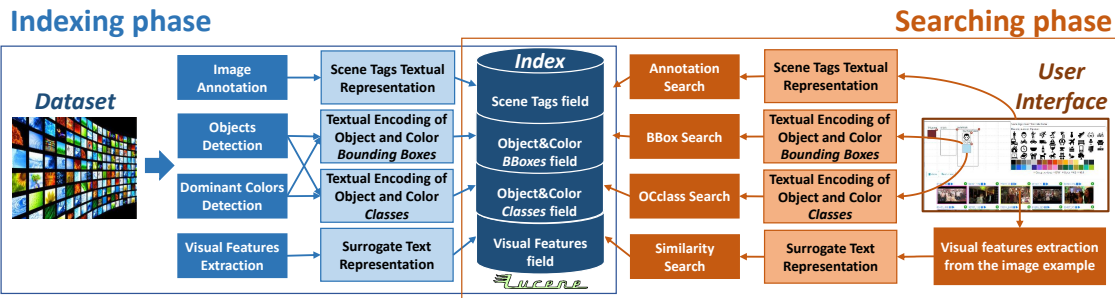


Figure 11. System Architecture: a general overview of the components of the two main phases of the system, the indexing and the browsing.

tations;

- *BBox Search*, search the Object&Color BBoxes field for keyframes having specific spatial relationships among objects/colors;
- *OCclass Search*, search the Object&Color Classes field for keyframes containing specified objects/colors;
- *Similarity Search*, search the Visual Features field for keyframes visually similar to a query image

The user query is broken down into three sub-queries (the first three search operations above), and a query rescorer (the Lucene QueryRescorer implementation in our case) is used to combine the search results of all the sub-queries. Note that the Similarity Search is the only search operation that is stand-alone in our system: it is a functionality used only on browsing phase. Since different text scoring functions could be employed by the four search operations introduced above, in the next section we report an analysis of the performance of VISIONE under different configurations to identify the most suitable text scoring functions to be used within the analyzed system.

4.3.4. Evaluation results

Since VISIONE is an interactive retrieval system, its performance cannot be easily tested outside of controlled contexts set up for this purpose, such as a user testing campaign or a competition like VBS. In fact, the user query is dynamically formulated and refined during a search process and the type of query employed (e.g. query by text, query by object location, etc.) depends on the user’s personal preferences and attitudes in formulating a search intent and the interaction with the system over time. For these reasons, CNR exploited the log of queries executed during the VBS 2019 competition to analyze the search functionalities of the VISIONE system. During the competition, both expert and novice users¹ interacted with VISIONE to solve several search tasks. Specifically, the VBS competition was divided in three content search *tasks*: *visual Know-Item Search* (KIS), *textual KIS* and *Ad-hoc Video Search* (AVS). For each task, a series of *runs* is executed. In each run, the users are requested to find one or more target videos. When the user believes that he/she has found the target video, he/she submits the result to the organization team that evaluates the submission.

We used the ground-truth segments and the log of the queries submitted to our system during VBS 2019 to evaluate the performance of VISIONE under different settings. We restricted the analysis only to the logs related to textual and visual KIS tasks since ground-truths for AVS tasks

¹Expert users are the developers of the in race retrieval system or people that already know and use the system before the competition. Novices are users who interact with the search system for the first time during the competition.





and that was chosen according to subjective feedback provided by the developers of the system, has a good performance, but it is not the best. In fact, we noticed that there exist some patterns in the combinations of the rankers used for the OCclass Search and the Annotation Search which are particularly effective and some which, instead, provide us with very poor results. For example, the combinations that use *TF* for the OCclass Search and *BM25* for the Annotation Search gave us the overall best results. While the combinations that use *BM25* for the OCclass Search and the *NormTF* for the Annotation Search have the worse performance. Specifically, we have a MRR of 0.023 for the best (NormTF-BM25-TF) and 0.004 for the worst (BM25-NormTF-BM25), which results in a relative improvement of the MRR of 475%. Moreover, the best combination has a relative improvement of 38% over the baseline used at the VBS2019. These results give us evidence that an appropriate choice of rankers is crucial for system performance.

Furthermore, to complete the analysis on the performance of the rankers, we analyze the MMR@ k , where k is the parameter that controls how many results are shown to the user in the results set. The results for some representative values of k are reported in Table 5. In order to facilitate the reading of results, we focused the analysis only on eight combinations: the four with the best MMR@ k , the four with the worst MMR@ k , and the configuration used at VBS2019. The latter is also used as baselines to evaluate the statistical significance of the results according to Fisher’s randomization test. Approaches for which the MMR@ k is significantly different from the MMR@ k of the baseline are marked with * in Table 5. We observed that the configuration *NormTF-BM25-TF* perform the best for all the tested k , however the improvement over the VBS2019 baseline is statistically significant only for $k \geq 10$, that is the case where the user inspects more than 10 results.

In conclusion, we identified the combination *NormTF-BM25-TF* as the best one, providing a relative improvement of 38% in *MRR* and 40% in *MRR@100* with respect to the setting previously used at the VBS competition.

Table 5. MRR@ k for eight combinations of the rankers (the four best, the four worst and the setting used at VBS2019) varying k . Statistically significant results with two-sided p value lower than 0.05 over the baseline *BM25-BM25-TF* are marked with *.

	$k = 1$	$k = 5$	$k = 10$	$k = 50$	$k = 100$	$k = 500$	$k = 1000$
NormTF-BM25-TF	0.015	0.017	0.019 *	0.022 *	0.022 *	0.023 *	0.023 *
TFIDF-BM25-TF	0.013	0.016	0.018 *	0.021 *	0.022 *	0.022 *	0.022 *
TF-BM25-TF	0.013	0.016	0.017	0.018 *	0.019 *	0.019 *	0.019 *
TF-BM25-BM25	0.013	0.015	0.016	0.017	0.017 *	0.018 *	0.018 *
TF-BM25-NormTF	0.013	0.015	0.016	0.017 *	0.017 *	0.018 *	0.018 *
BM25-BM25-TF (VBS 2019)	0.013	0.014	0.015	0.016	0.016	0.016	0.017
NormTF-TF-NormTF	0.000 *	0.001 *	0.003 *	0.004 *	0.004 *	0.005 *	0.005 *
NormTF-NormTF-BM25	0.000 *	0.001 *	0.002 *	0.004 *	0.004 *	0.005 *	0.005 *
BM25-NormTF-BM25	0.002 *	0.002 *	0.002 *	0.003 *	0.003 *	0.004 *	0.004 *
TFIDF-NormTF-BM25	0.000 *	0.001 *	0.001 *	0.003 *	0.004 *	0.004 *	0.004 *

4.3.5. Second Release of VISIONE (VBS 2021)

Besides using a new configuration for the search operations (i.e., NormTF-BM25-TF), as the result of the analysis reported in the previous section, the second release of VISIONE [125] integrated a novel retrieval module and several improvements as described in the following. An overview of the revised architecture of VISIONE is depicted in Figure 11.

One of the main limitations of the first version of VISIONE was the poor performance on textual KIS tasks, i.e. the task where only a textual description is provided. In fact, during the VBS 2019, the VISIONE team exactly solved only 2 out of 8 textual KIS tasks. To overcome



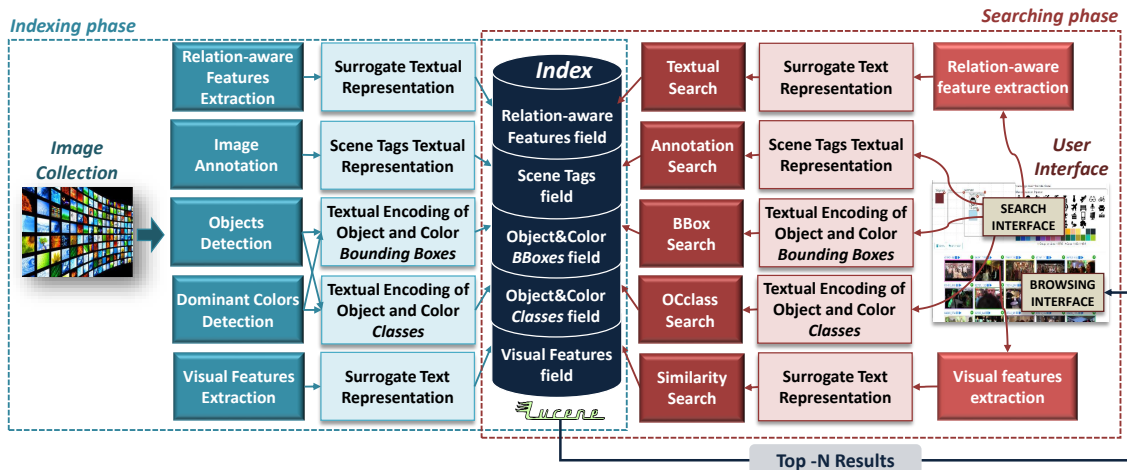


Figure 13. System Architecture of the second release of VISIONE.

this limitation, CNR developed a retrieval module that allows searching for a target scene using natural language queries, that is supporting *query by scene description*. Textual descriptions are full natural language sentences, usually between 5 to 50 words in length, describing a visual scene. For example, a valid textual description could be “A *tightly packed living room with a tv screen larger than the fireplace right beside it*”. These textual descriptions can include objects details, expressed using their physical or semantic attributes, and they can specify the spatial or abstract relationships linking objects together.

The search using natural language descriptions as a query is achieved by using a deep neural network architecture, called Transformer Encoder Reasoning Network (TERN), which was recently developed by CNR [134]. The TERN network is able to match images and sentences in a highly-semantic common space. The core of this architecture constitutes of deep relational modules called *transformer encoders* [135], which can spot out hidden intra-object relationships. In particular, in the visual pipeline, a stack of transformer encoders try to find links between image regions pre-extracted using a state-of-the-art object detector (Faster-RCNN); in the textual pipeline, using a pretrained BERT model plus another stack of transformer encoder layers, the model searches for relationships between sentence words. An overview of the architecture is shown in Figure 14.

The extracted cross-modal features are normalized and in principle very similar to visual descriptors like RMAC [130]. Hence, we indexed them using the same textual encoding that we already exploited to index the RMAC descriptors (see [128]). The textual encoding extracted from the cross-modal features are stored in a separate field of the index, named *Relational-aware features field* (see Figure 11). A *Textual Search* operator, acting on this field of the index, allows for searching keyframes associated to a given textual description. The TF text scoring function is employed by this search operation.

Moreover, the VISIONE system was revised in order to support temporal searches, where the user can describe two consecutive (or temporally close) keyframes of the same target video. To this scope, a second canvas and associated input text boxes were added to the user interface, allowing a user to simultaneously search for two keyframes that are temporally close in a video segment but that are different in the represented content. The search is executed by performing two queries to the index, each providing its own output results. The resulting keyframes, which belong to the same video and whose temporal distance is less than a given threshold, are then combined as pairs and shown in the browsing interface.

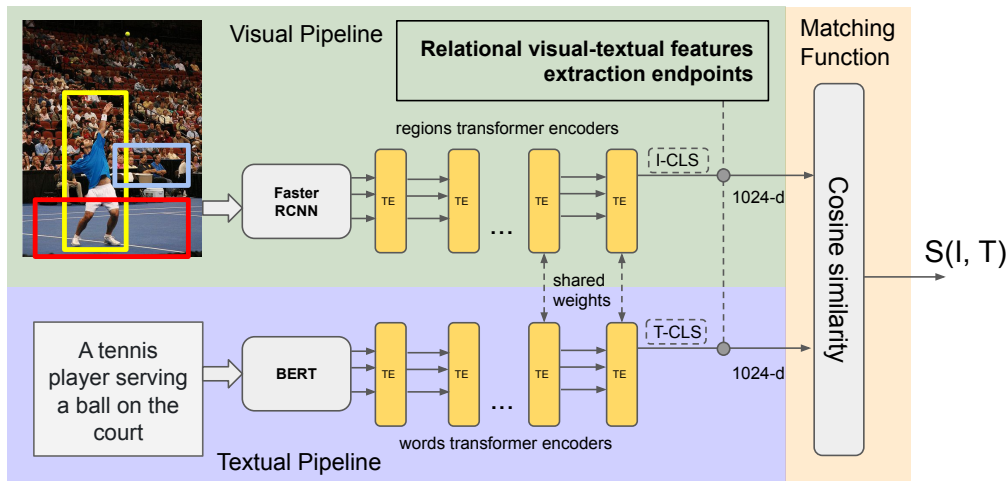


Figure 14. Overview of the TERN architecture. Orange boxes are Transformer Encoder (TE) layers. Final TE layers share their weights for better stability during the training phase.

Finally, several improvements were made to the user interface, including the possibility to search by similarity also using external images uploaded from a URL or file system, and the possibility of selecting multiple images to be submitted as a response to an AVS task.

During the VBS 2021 competition, the VISIONE team was able to exactly solve 17 out of a total of 21 Visual KIS tasks, and 3 out of a total of 6 textual KIS tasks (for one task no team was able to find the correct solution). Although the performance on textual KIS should be further improved, we would like to note that the new search module (i.e., *query by scene description*) was crucial in solving many visual KIS tasks other than the textual ones.

4.3.6. Contributions to WP8 Use cases

Relevant WP8 Use Cases: 3A3 (Archive exploitation), 2B1 (Automatic metadata tagging), 3C2-6 (Video object recognition), 3C2-7 (Video object localisation), 4C2 (Moving Image (video) analysis). VISIONE can be used for large scale video searching, to perform automatic metadata generation and object recognition.

4.3.7. Relevant Publications

- “VISIONE at Video Browser Showdown 2021”, Amato, G., Bolettieri, P., Falchi, F., ...Vadicamo, L., Vairo, C, 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, ISBN: 978-303067834-0, DOI:10.1007/978-3-030-67835-7_47, (Zenodo Record: <https://zenodo.org/record/5078245>)
- “The VISIONE video search system: exploiting off-the-shelf text search engines for large-scale video retrieval”, G. Amato, P. Bolettieri, F. Carrara, F. Debole, F. Falchi, C. Gennaro, Journal of Imaging, 2021 7 (5), 76, <https://doi.org/10.3390/jimaging705007>, (Zenodo Record: <https://zenodo.org/record/5078216>)
- “Transformer reasoning network for image-text matching and retrieval”, Messina, N., Falchi, F., Esuli, A., Amato, G., 25th International Conference on Pattern Recognition (ICPR) (pp. 5222-5229), 2021, <https://doi.org/10.1109/ICPR48806.2021.9413172>





4.3.8. Relevant External Resources

- <http://visione.isti.cnr.it/>

4.4. Few-shot object detection: positive sample augmentation and ensembling

Contributing partners: UPB

4.4.1. Method Overview

UPB worked on few-shot object detection (FSOD) algorithms applied on images with the aim of either improving the existing performances or reducing the current implementations' computational requirements. After a thorough study of the literature, UPB decided to take inspiration and start from the state-of-the-art implementation of a simple, yet effective few-shot object detector [112]. This method implies training a base model for object detection on a given dataset. Then, when new classes are presented to the model it is sufficient to fine-tune only the last two layers while freezing the rest of the model. This algorithm is successful and surprisingly simple to implement and understand. Based on this method, UPB studied the approaches described below.

Model Variation UPB first studied different model setups, similar to the ones mentioned in the original paper. Since the authors focused on a Faster R-CNN [116] model composed of ResNet-101 [54] and FPN [136] as backbone, UPB explored other variations as well. The model permits the adaptation of similar architectures such as ResNet-50, ResNet-151, ResNeXt variants [137] or VGG variants [138] in the backbone section. Another interesting architecture that can substitute these backbones is Mask R-CNN [139] where a small improvement is brought by the ROI pooling layer. On the same note, another idea that UPB investigated was the reduction of computation complexity by replacing the two-stage detectors with one stage detectors such as YOLO [140], SSD [141] and RetinaNet [142]. These have the advantage of being faster and using less parameters, but they usually suffer from a performance reduction. Since the fine-tuning approach acts only on the last layers of the network, the single-stage detectors are also viable solutions.

Positive Sample Augmentation Another interesting approach from the literature involves positive sample augmentation [143]. This algorithm is depicted in Figure 15 and it follows the next logic. There are two branches to the entire pipeline: a main one, consisting of the Few-Shot Object Detector and a reinforcement one, consisting of the positive sample augmentation (PSA), an FSOD and a hard samples reweighting module. The two FSODs form a Siamese network, therefore their weights are tied. The positive sample augmentation consists of several image enhancement mechanism that are applied to scarce samples from the dataset.

The reinforcement branch is used only in the training phase of the algorithm. Our work focused on adapting this setup to the previously mentioned fine-tuning approach. This means that after training the base model on the classes that have enough samples we proceed to freezing the Siamese nets except for their last two layers. These are then fine-tuned on the few-shot classes. This setup is also versatile since the FSOD module can be represented by any object detector. Multi-scale approaches are considered better for this algorithm since among the PSA methods there are also multi-scale replications of the hard samples. In the training phase, a combined loss is used:

$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{conf} + \mathcal{L}_{reg}, \quad (10)$$



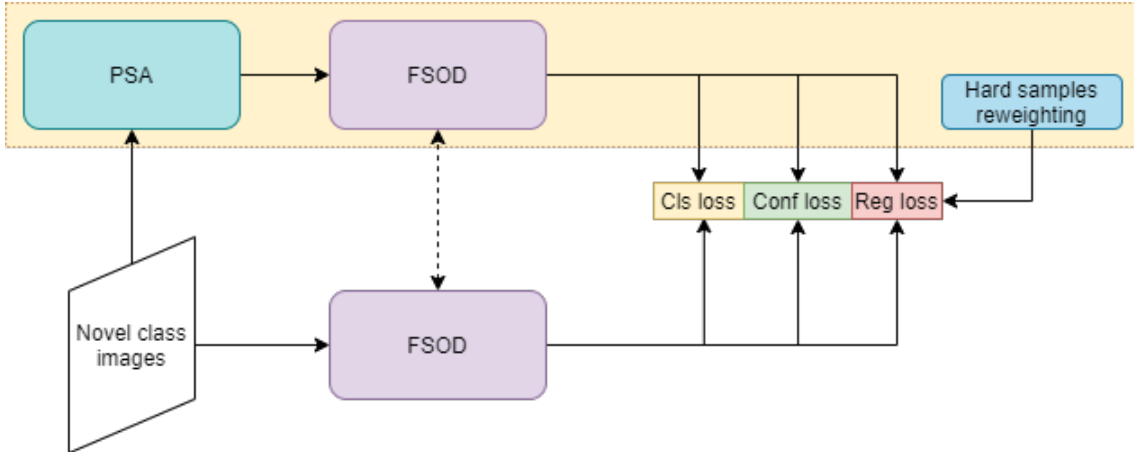


Figure 15. Positive Sample Augmentation framework. Reinforcement branch on yellow background.

Table 6. Few-shot detection evaluation on PASCAL VOC 2012. The numbers in the split boxes represent the number of samples in the novel classes.

Method	Split 1				Split 2				Split 3			
	1	3	5	10	1	3	5	10	1	3	5	10
FRCN+ft_full	34.0	34.5	37.4	37.5	30.1	32.5	32.6	32.2	36.5	36.5	38.1	37.3
YOLOv3+ft_full	14.2	30.6	40.3	44.2	14.5	28.9	33.7	40.4	16.1	32.4	40.1	43.1
YOLOv4+ft_full	15.5	32.8	45.4	51.3	15.6	30.6	36.3	44.8	17.5	35.9	45.8	49.6
TFA w/ fc	43.8	44.6	46.7	46.6	41.2	42.0	43.0	42.7	43.0	45.9	46.3	46.7
TFA w/ cos	44.0	45.0	47.2	47.3	41.1	42.2	43.1	43.3	42.3	45.7	46.3	47.0
FSSP	41.6	49.1	54.2	56.5	30.5	39.5	41.4	45.1	36.7	45.3	49.4	51.3

where L_{cls} is the classification loss, L_{reg} is the regression loss and L_{conf} is the confidence loss. L_{reg} and L_{conf} are the standard losses uses in object detection. L_{cls} is computed as the sum between a fine-tuning loss and a cosine similarity loss, meant to give positive samples high scores so that they can be detected and suppress hard-negative samples.

UPB ran the few-shot detection evaluation on two standard datasets that are largely used for this task, namely PASCAL VOC [144] and MS-COCO [68]. We present the best results that UPB obtained so far in Tables 6 and 7. For PASCAL VOC the following setup was used: for training, the VOC2007_train and VOC2012_train, and VOC2012_val sets were used and for validation the VOC2007_test set was used. The 20 classes are randomly divided into 15 base training classes and 5 novel classes. This random split is performed 3 times and the results on each of these splits are presented. For the COCO dataset the 2014 dataset was used and 5k images from the validation set were extracted for evaluation. The rest of the dataset is used for training. The same 20 classes that are used by PASCAL VOC are kept as novel classes and the rest are used as base classes for training. The FRCN+ft_full, YOLOv3+ft_full, and YOLOv4+ft_full models are obtained by finetuning the trained base model on the novel classes until full convergence is achieved. TFA w/ fc represents the model obtained by replacing the last 2 classification layers with fully connected layers, whereas TFA w/ cos is obtained by replacing the last 2 layers with a cosine similarity based classifier. Lastly, FSSP is the model obtained by performing the positive sample augmentation on the novel classes. The last 3 models use Faster RCNN with ResNet101 and FPN as backbone.





Table 7. Few-shot detection evaluation on MS-COCO.

Method	10-shot			30-shot		
	AP	AP50	AP75	AP	AP50	AP75
FRCN+ft_full	13.4	21.8	14.5	13.5	21.8	14.5
YOLOv3+ft_full	9.1	18.2	8.2	12.5	24.8	11.3
YOLOv4+ft_full	10.1	19.9	9.2	14.1	25.8	13.7
TFA w/ fc	26.3	41.8	28.6	28.4	44.4	31.2
TFA w/ cos	26.6	42.2	29.0	28.7	44.7	31.5
FSSP	9.9	20.4	9.6	14.2	25.0	13.9

Ensembling FSOD Considering that both the previous two mentioned methods can be customised for a large number of detection networks, UPB are following the approach proposed by Dvornik et al [145] regarding ensembling methods for FSOD. The authors propose 3 ensembling strategies that are currently under progress:

1. independent ensembling: several models are trained independently and frozen. Then, their last prediction layer is removed and, given a new class with few annotated samples, a mean centroid classifier is built for each network. The obtained probabilities are then averaged over networks, improving the accuracy.
2. diversity ensembling: this is done by introducing randomization in the model training through data augmentation or various initializations. This technique works best for a large number of networks.
3. cooperation ensembling: by encouraging conditional probabilities to be similar through symmetrized Kullback-Leibler divergence. This technique works best for a small number of networks, which is in contradiction with the previously mentioned method.

A fourth strategy is proposed as a compromise between the previous two ensembling methods, where the best trade-off between the high and low number of networks involved in the ensembling strategy is employed. They do so by randomly dropping a part of the networks at each training iteration, applying Dropout inside each network and applying different transformations on the input images.

Since there are many object detection architectures that have been validated during the past few years, UPB worked on creating several models with the PSA-fine-tuning hybrid approach and finally combine them using the aforementioned ensembling strategies. Preliminary results show that this method is worth investigating. This is currently an ongoing research. A conference paper is expected to be submitted on this topic by February 2022.

4.4.2. Contributions to WP8 Use cases

Relevant WP8 Use Cases: 3C2-6 (Video object recognition), 3C2-7 (Video object localisation). A few-shot object detector can contribute to use-cases 3C2-6 and 3C2-7, by recognizing and localizing (with a bounding box) objects in an image. A trivial extension to video sequences would be to apply this algorithm on a per-frame basis.





4.5. Adversarial Semi-supervised Learning for Fine Grained Visual Classification

Contributing partners: UNIFI

Novel methods to increase the amount of training data, for improving learning accuracy in Fine-Grained Visual Categorization (FGVC), were investigated. This problem has not been researched in the past, in spite of prohibitive annotation costs that FGVC requires. We are currently investigating a method to leverage unlabeled data with an adversarial optimization strategy in which the internal features representation is obtained with a second-order pooling model (Subsection 4.5.2). This combination allows to back-propagate the information of the parts, represented by second-order pooling, onto unlabeled data in an adversarial training setting. Preliminary evaluation and discussion on the basic mechanism of soft pseudo-labeling on which adversarial learning is based is discussed in Subsection 4.5.1.

4.5.1. Soft Pseudo-labeling Semi-Supervised Learning Applied to Fine-Grained Visual Classification

Introduction

Fine-Grained Visual Categorization (FGVC) aims to distinguish between image classes such as species of birds, dogs, flowers or even models of cars. This is much harder than general-purpose classification as only few subtle key features matter. A further issue of FGVC is that data annotation is very expensive and it requires domain experts. The data annotation problem can be partially alleviated using Semi-Supervised Learning (SSL) by leveraging large set of unlabeled data and few labeled ones [146]. Except for a very recent paper [147], SSL has not been investigated in FGVC. However, this topic is getting increasing support and attention to such an extent that a dataset for this specific problem has been released [148]. SSL has shown to be a suitable learning paradigm for leveraging unlabeled data to reduce the cost of large labeled datasets [149].

A common assumption in SSL is that the decision boundaries of the classifier should not pass through high-density regions of the marginal data distribution [146]. One way to impose this constraint is to force the output of the classifier to have low-entropy predictions on the unlabeled data [150]. This strategy is known as *entropy minimization* and is particularly interesting because *pseudo-label* SSL [151], the simplest algorithms in SSL, does entropy minimization implicitly by constructing hard labels from the most confident class predictions on unlabeled data. Class predictions are subsequently used as training targets in a standard supervised learning paradigm and optimized according to the cross-entropy loss. Entropy minimization can be considered a soft version of the pseudo-labeling method.

This work investigated the theoretical relationship between the two methods and evaluated SSL *entropy minimization*, on several FGVC datasets, including the recent Semi-Supervised iNaturalist-Aves [148] and compare the results with [147] in which a pseudo-label based learning method is used.

Experimental results show that although in some cases supervised learning may still have better performance than the semi-supervised methods, Semi Supervised Learning shows effective results. Specifically, we observed that entropy-minimization slightly outperforms a recent proposed method based on pseudo-labeling.





Related Work on Semi-Supervised FGVC

To the best of our knowledge, [147] is the only approach evaluating FGVC datasets in a SSL learning context and it proposes a pseudo-label based technique to leverage unlabeled data. The method, after each training, generates pseudo-labels on the unlabeled set to be added to the labeled training samples; it select the top- k most-confident label greater than a threshold value.

The Semi-Supervised iNaturalist-Aves dataset (FGVC7) has been recently released. It presents some of the challenges encountered in a realistic setting, such as fine-grained similarity between classes, significant class imbalance, and domain mismatch between the labeled and unlabeled data. As reported by the panel of the competition all participating teams applied the pseudo-label method [151] and the state-of-the-art method [152] provides similar performance but is computationally more expensive. Other recent state-of-the-art methods [149, 153, 154] are also exploited but do not improve the performance.

Problem Formulation

In this section we briefly review the formulation of SSL and the relationship between entropy minimization and pseudo-labeling. In Semi-Supervised Learning [146] we are provided with a dataset of K classes containing both labeled and unlabeled examples. The dataset \mathcal{D} is divided in two parts: a labeled subset $\mathcal{D}_l = \{(\mathbf{x}_i, y_i)\}_{i=1}^{|\mathcal{D}_l|}$ and an unlabeled subset $\mathcal{D}_u = \{(\mathbf{x}_j)\}_{j=1}^{|\mathcal{D}_u|}$, where $|\mathcal{D}_l|$ and $|\mathcal{D}_u|$ are respectively the number of examples of the labeled and unlabeled datasets. Semi-Supervised Learning (SSL) aims to improve model performance by incorporating a large amount of unlabeled data during training. Formally, the goal of SSL is to leverage the unlabeled data \mathcal{D}_u to produce a prediction function f^θ , with trainable parameters θ , that is more accurate than using the labeled data \mathcal{D}_l only.

Pseudo-label

Unlabeled samples are treated as labeled samples, and training proceeds with the standard supervised loss function:

$$\hat{y}_i^k = \begin{cases} 1 & \text{if } k = \operatorname{argmax}_k f_k^\theta(x_j) \\ 0 & \text{otherwise.} \end{cases}$$

In this way pseudo-labels of unlabeled samples are considered as if they were true labels. The Cross-Entropy loss calculated on pseudo-labeled samples is:

$$\mathcal{L}_{pl} = -\frac{1}{|\mathcal{D}_u|} \sum_{j=1}^{|\mathcal{D}_u|} \sum_{i=1}^K \hat{y}_i^j \log f_i^\theta(x_j).$$

So the overall objective function is:

$$\hat{\theta} = \operatorname{argmin}_{\theta} \left(-\frac{1}{|\mathcal{D}_l|} \sum_{j=1}^{|\mathcal{D}_l|} \sum_{i=1}^K y_i^j \log(f_i^\theta(x_j)) - \lambda \frac{1}{|\mathcal{D}_u|} \sum_{j=1}^{|\mathcal{D}_u|} \sum_{i=1}^K \hat{y}_i^j \log f_i^\theta(x_j) \right), \quad (11)$$

where the first term is the standard cross-entropy loss in which y_i^j is the label of the j -th sample of the class i and λ weights the contribution of the second term.





Entropy Minimization

One common assumption in many SSL methods is that decision function boundary should not pass through high-density regions of the marginal data distribution. One way to enforce this, is requiring the classifier to output low-entropy predictions on unlabeled data [150]. This encourages the network to make confident (i.e., low-entropy) predictions on unlabeled data regardless of the predicted class, discouraging the decision boundary from passing near data points where it would otherwise be forced to produce low-confidence predictions. This effect can be achieved by adding a simple loss term which minimizes the entropy of the prediction function $f^\theta(x)$:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}}(\mathcal{L}_{ce} + \lambda H).$$

In which the entropy H calculated on unlabeled data is:

$$H = -\frac{1}{|D_u|} \sum_{j=1}^{|D_u|} \sum_{i=1}^K f_i^\theta(x_j) \log f_i^\theta(x_j).$$

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \left(-\frac{1}{|D_l|} \sum_{j=1}^{|D_l|} \sum_{i=1}^K y_i^j \log(f_i^\theta(x_j)) - \lambda \frac{1}{|D_u|} \sum_{j=1}^{|D_u|} \sum_{i=1}^K f_i^\theta(x_j) \log f_i^\theta(x_j) \right). \quad (12)$$

As can be noticed Eq. 11 and Eq. 12, are equivalent: the hard pseudo-label \hat{y}_i^k is replaced by soft one in term of the network output $f_i^\theta(x_j)$. According to this, pseudo-labeling is closely related with entropy minimization.

4.5.2. Fine-Grained Adversarial Semi-supervised Learning

Introduction

Fine-Grained Visual Categorization (FGVC) lies in the continuum between categorization (i.e. object classification) and identification (i.e. instance recognition). FGVC is quite subtle and therefore difficult to address with general-purpose object classification methods based on DNNs [155]. FGVC is much more challenging than traditional classification tasks due to the inherently subtle intra-class object variability amongst sub-categories. Distinguishing between a cat and a giraffe is easy (i.e. large variability) while, in distinguishing fine-grained classes, typically only a few key features matter as in species of birds [156], dogs [157], flowers [158] or manufacturers and models of cars [159] and aircrafts [160].

The task becomes significantly more difficult in domains where data is not readily available (e.g., medical images) or domains for which training data is scarce [161]. It is likely that techniques used for representation learning like semi/self-supervised or unsupervised learning that are currently used for visual recognition are not sufficient to significantly improve FGVC. In addition to this, obtaining training data for fine-grained images is prohibitively expensive, as expert knowledge is typically required [162]. In view of these issues, we propose a learning method focusing on the FGVC problem in which labeled data is limited and unlabeled is available.

Recent top performing supervised learning methods have substantially shown that the most successful strategy to FGVC is obtained by identifying, either *explicitly* or *implicitly*, the object parts [163, 164, 165, 166]. The central underlying assumption is that fine-grained information resides within the parts. Many approaches particularly focus on explicitly localizing relevant regions in an image. This is typically achieved by leveraging the extra annotations of bounding box and part



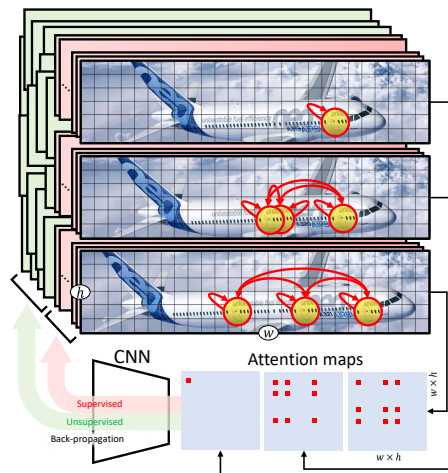


Figure 16. Illustration of the effect of second-order pooling in Semi-Supervised Fine-Grained Visual Categorization (SSL-FGVC). We show images from three different classes of Airbus aircraft models: A319 (top), A320 (middle) and A321 (bottom). They mainly differ by the number of doors and their position along the fuselage (circles). We propose to take advantage of the long-range attention based part-to-part relationships exploited by second-order pooling and back-propagate this information onto unlabeled data to perform unsupervised structure discovery.

annotations (some known datasets provide ground-truth part annotations [156, 167]) to localize regions that provide the most discriminative information. However, in addition to class labeling, the extra human annotations regions are not only difficult to obtain and prohibitively expensive, but can often be error-prone resulting in performance degradation [168]. Methods for unsupervised part detection and mining have been developed [169, 170, 171, 172], however, these methods pose various challenges, such as missing parts due to occlusions and parts not providing discriminative information. According to this, it remains controversial whether unsupervised detected parts are fully beneficial.

While extra parts annotation has been well studied, we are aware of only one published work in literature addressing FGVC in a Semi-Supervised Learning setting at the image label level [147]. Despite not being investigated, this topic is getting increasing attention and support. In confirmation of this, the Semi-Supervised iNaturalist-Aves Dataset dataset has been recently released for this specific problem ([148]), in the context of the challenge part of the FGVC7 workshop held in conjunction with CVPR2020. The dataset is intended to set out some of the difficulties faced in a practical environment. The competition panel reported that all teams applied the pseudo-label SSL method [151] and that the state-of-the-art Deep-SSL methods [152, 149, 153, 154] provide similar performance but are computationally more expensive.

According to this, we propose an approach that addresses the problem in a *complementary* way in both the SSL setting and the part-based assumption of FGVC. We adopt an adversarial optimization strategy that alternately maximizes the conditional entropy of unlabeled data with respect to the classifier and minimizes it with respect to a second-order feature encoder. This combination allows to *back-propagate* the information of the *parts* captured by the second-order pooling model onto *unlabeled data* in an adversarial training setting as illustrated in Fig. 16. The strategy extends the works in [9, 173], originally proposed for Domain Adaptation, to the specific Semi-Supervised Learning setting. To the best of our knowledge, this is the first approach to leverage adversarial optimization in the specific case of SSL, and we are not aware of previous works that combined SSL with specific strategies exploiting the information of *object parts*. Although the recent work [147] applies SSL to FGVC datasets, the information of the parts are not explicitly

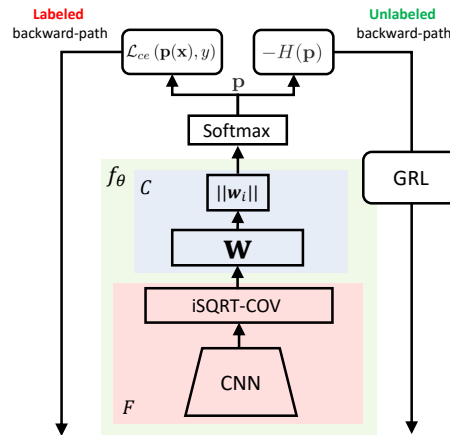


Figure 17. An overview of the proposed model architecture. The inputs to the network are labeled and unlabeled examples. The model f_θ (light green) consists of the second-order pooling (iSQRT-COV) [8] feature extractor F (light red) and the classifier C having weight vectors \mathbf{w}_i (light blue). C is trained to maximize entropy on unlabeled target whereas F is trained to minimize it. To achieve the adversarial learning, the sign of gradients for entropy loss on unlabeled target examples is flipped by a gradient reversal layer (GRL) [9]. According to this, labeled and unlabeled back-propagation follows two distinct paths.

taken into account during the semi-supervised learning process.

The main benefit of our adversarial optimization with respect to pseudo-label based methods [151, 147], is that the model can correct its own errors without incurring in wrong classifications that rapidly intensify resulting in confident but erroneous pseudo-labels on the unlabeled data.

We empirically demonstrate the superiority of our method over many baselines and show the method is safe [174].

Problem Formulation

In Semi-Supervised Learning, in addition to unlabeled data, the learning algorithm is provided with some supervision information, but not necessarily for all examples. In this case, the data is divided in two parts: the set for which labels are available $\mathcal{D}_l = \{(\mathbf{x}_i^l, y_i^l)\}_{i=1}^{n_l}$ and the set for which the labels are unknown $\mathcal{D}_u = \{(\mathbf{x}_i^u)\}_{i=1}^{n_u}$, where n_l and n_u are the number of examples of the labeled and unlabeled datasets, respectively. It is typically assumed that $\mathbf{X} \times Y$ is drawn from an unknown joint probability distribution $p(\mathbf{X}, Y)$ and that we observe it through the finite training sample \mathcal{D}_l . The main goal is to leverage the unlabeled data \mathcal{D}_u to learn a DNN model f_θ , with trainable parameters θ , that is more accurate than using the only \mathcal{D}_l . The data \mathcal{D}_u provide additional information about the structure of the data distribution $p(\mathbf{X})$ to better learn the internal feature representation of f_θ . The dependency between $p(\mathbf{X})$ and $p(Y|\mathbf{X})$ is typically established according to the *cluster assumption*, (i.e. data points in the same cluster of $p(\mathbf{X})$ have the same label Y); and *low-density separation*, (i.e. class boundaries of $p(Y|\mathbf{X})$ should lie in an area where $p(\mathbf{X})$ is small) [175]. Due to the low variability of classes in FGVC, these assumptions are quite hard to meet in practice.

Method Overview

Our base model architecture f_θ for FGVC consists of a special feature extractor F , based on second-order pooling, and a classifier C in which weights \mathbf{W} are normalized to exploit the approximated cosine distance criterion between the classifier prototypes and the features. According to this,





in order to classify examples correctly, the normalized direction of a weight vector has to be representative of the features of the corresponding class in term of an angular distance. In this respect, the weight vectors can be regarded as angular estimated prototypes for each class.

Angular classifier prototypes are learned taking into account unlabeled data by exploiting an adversarial entropy optimization. Unlabeled data follow a specific data path for back-propagation that allows to attract the prototypes towards them. As the feature representation takes in account the long-range part-to-part relationships, the information of the parts can be indirectly better back-propagated towards the unlabeled data. The architecture of our method is shown in Fig. 11 and will be detailed in the next Subsection.

Back-propagation of the Parts onto Unlabeled Data

The goal is to obtain representative classifier prototypes $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_K]$ where K is the number of classes for labeled and unlabeled data and to minimize the distance between the prototypes and the unlabeled examples. We approach the problem in mainly three different fronts: (1) reduce intra-class distance, (2) improve feature discriminativity exploiting part-to-part relationships (3) handling unlabeled data distance between prototypes and features.

For labeled data, the general purpose-classification linear classifier already minimizes the distance between features and classifier prototypes. However, as in many other instance recognition tasks (i.e. face recognition, re-identification), features in ideal FGVC are expected to have smaller maximal intra-class distance than minimal inter-class distance under a suitably chosen metric space. The vanilla linear classifier cannot effectively satisfy this criterion [176]. One simple method to enable convolutional neural network to produce more discriminative features is imposing discriminative constraints on a hypersphere manifold by normalizing the vectors of the classifier weights [177]:

$$\mathbf{w}'_i = \frac{\mathbf{w}_i}{\|\mathbf{w}_i\|} \quad i = 1, 2, \dots, K, \quad \mathbf{w}'_i \in \mathbb{R}^m \quad (13)$$

where, as detailed in the next subsection, $m = \frac{d(d+1)}{2}$ with d , is the channel dimension of the last convolutional layer of the architecture (Fig. 18).

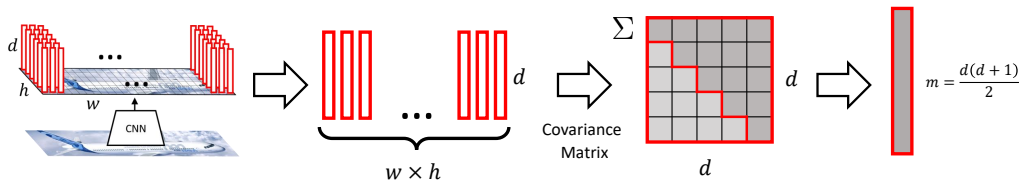


Figure 18. The $w \times h$ feature channels of dimension d of the last convolutional layer of the CNN architecture are used to compute the covariance matrix. The $\frac{d(d+1)}{2}$ -dimensional values of the upper triangular matrix constitute the internal feature representation vector that allows the model to determine the attention based long-range part-to-part relationships. The forward and backward propagation of the covariance in the adversarial optimization setting of Fig. 11 are computed according to the iSQRT-COV approximated method.

Handling Unlabeled Data with Adversarial Entropy Optimization

The obtained features and the relative prototypes provide improved discriminative power and reduced low-intra class variation, respectively. Yet, the overall goal remains how to include a strategy to minimize the distance between the prototypes and the unlabeled examples. As this





cannot be directly achieved with a linear classifier alone, we exploited the adversarial strategy originally proposed in [9, 173]. The adversarial part of our strategy clusters features computed from unlabeled data around the classifier learned prototypes. Therefore, we train the feature extractor F and the classifier C to classify labeled and utilize standard cross-entropy minimization objective to extract discriminative features for the labeled data:

$$\mathcal{L} = -\frac{1}{n_l} \sum_{j=1}^{n_l} \sum_{i=1}^K \hat{y}_j \log p(y_j = i | \mathbf{x}_j), \quad (14)$$

where \hat{y}_j is the true class label for x_j . The *unlabeled data* are used to *maximize* the entropy with respect to the classifier C and to *minimize* the entropy with respect the feature extractor F . The entropy is computed as follows:

$$H = -\frac{1}{n_u} \sum_{j=1}^{n_u} \sum_{i=1}^K p(y_j = i | \mathbf{x}_j) \log p(y_j = i | \mathbf{x}_j). \quad (15)$$

The intuition is that high entropy, namely the maximization of Eq. 15 between the classifier weight

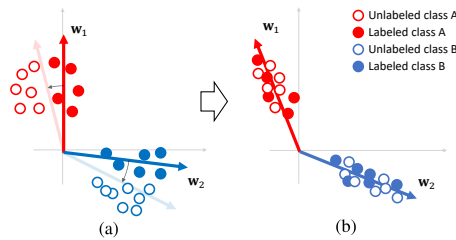


Figure 19. Adversarial Learning intuition. (a): High entropy between the classifier weight vectors w_i (i.e., the prototypes) and the unlabeled data features forces the classifier weight vectors to “move” towards the unlabeled data features. (b): This force is counter to the standard cross entropy which instead tends to cluster labeled and unlabeled features around the estimated prototypes.

vectors (i.e., the prototypes) and the unlabeled data features, forces the classifier weight vectors to “move” towards the unlabeled data features (Fig. 19(a)). This because high entropy tends to achieve a uniform distribution of the softmax output probabilities that consequently encourages each prototype w_i to be similar to all the unlabeled features. This strategy can be considered as an “adversarial move” of the classifier, whose intention is to “explore” the representation space driven by the unlabeled data (Fig. 19(a)). This force is counter to the force of the standard cross entropy (Eq.14) which instead tends to cluster unlabeled features around the estimated prototypes by learning the feature representation (Fig. 19(b)). Alternating these two opposite “forces” determines a sort of equilibrium in which discriminative features and a classifier may have better explored the representation space as driven by unlabeled data in a way that possible wrong pseudo labeling can be eventually recovered. This co-optimization can be formulated as an adversarial learning between C and F by weighted summing the two losses of Eq. 14 and Eq. 15 as follows:

$$\begin{aligned} \hat{\theta}_F &= \underset{\theta_F}{\operatorname{argmin}} \mathcal{L} + \lambda H \\ \hat{\theta}_C &= \underset{\theta_C}{\operatorname{argmin}} \mathcal{L} - \lambda H, \end{aligned} \quad (16)$$

where $\lambda > 0$ is the weighting factor. F and C are co-optimized in two steps: in the first step, both F and C are optimized by minimizing the cross-entropy loss on labeled data. In the second





step, F and C are optimized in opposite ways on unlabeled data, minimizing the entropy loss and maximizing the entropy, respectively (the signs of the entropy in the two equations of Eq. 16 are opposite). In the second step, input data follows the unlabeled path (Fig. 11), on which the classifier and the feature extractor are connected via a Gradient Reversal Layer (GRL) [9]. During forward propagation, GRL acts as an identity transform. During the back-propagation, GRL takes the gradient from the subsequent level, multiplies it by $-\lambda$ and passes it to the preceding layer (Fig. 11). By adding the gradient reversal layer, the training process described above can be achieved through normal model training.

Evaluation on the Semi-Supervised iNaturalist-Aves Dataset

We compare with Sup-Cov [178] and with the six different SSL methods evaluated in [179]: Pseudo-Labeling [151], Curriculum Pseudo-Labeling, [180], FixMatch [154], Self-Training [179], MoCo (Momentum Contrast) [181] and MoCo + Self-Training [182]. Specifically, the Self-training baseline initially trains a teacher model with only labeled data, then transfers the knowledge to a student model by distillation [183] using both labeled and unlabeled data. The MoCo + Self-Training performs a self-supervised pre-training with MoCo then removes the final MLP layers and adds a classification layer that is trained with labeled data. The results of the comparison are shown in Tab. 8. Our method shows state-of-the-art performance with respect to the baselines with an accuracy result of 69.85% and 65.4% with the ResNet101 and ResNet50 architecture, respectively. The gain in classification accuracy from FixMatch (i.e. the best performing algorithm) using the same CNN backbone is 8% (from 57.4% to 65.4%). As evidenced from the table the increase in classification accuracy is mostly due to the second order pooling layer and secondly by the adversarial strategy.

Table 8. Results on the Semi-Supervised iNaturalist-Aves Dataset (FGVC7 challenge). Our method achieves a significant improvement by leveraging unsupervised data.

Method	Accuracy	#Params
Pseudo-Label [151]	54.4	ResNet50 (25M)
Curriculum Pseudo-Label [180]	53.4	ResNet50 (25M)
FixMatch [154]	57.4	ResNet50 (25M)
Self-Training [179]	55.5	ResNet50 (25M)
MoCo [181]	55.5	ResNet50 (25M)
MoCo + Self-Training [182]	52.7	ResNet50 (25M)
Sup [179]	52.7	ResNet50 (25M)
Sup-Cov [178]	64.7	ResNet50 (25M)
Ours w/o Cov	50.5	ResNet50 (25M)
Ours	65.4	ResNet50 (25M)
Ours	69.85	ResNet101 (44M)

4.5.3. Contributions to WP8 Use cases

Relevant WP8 Use Cases: 3A3-12 (Visual concepts classification), 2B1 (Automatic metadata tagging). FGVC methods can directly be applied to 3A3-12 and 2B1.





4.5.4. Relevant Publications

- Mugnai, D., Pernici, F., Turchini, F., & Del Bimbo, A. (2021). Soft Pseudo-labeling Semi-Supervised Learning Applied to Fine-Grained Visual Classification. In Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part IV (pp. 102-110). Springer International Publishing.
- Mugnai, D., Pernici, F., Turchini, F., & Del Bimbo, A. (2021). Fine-Grained Adversarial Semi-supervised Learning, (submitted).

4.6. DivClust - Learning Multiple Clusterings With a Diversity-Controlling Objective

Contributing partners: QMUL

4.6.1. Method Overview

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 there is, in fact, no single inherently correct way to cluster a given set of data. QMUL has focused on this area, and developed a clustering loss component that can be used to train models to produce multiple clusterings of controlled diversity with each other, which explore different partitionings of a given dataset. The proposed objective can be combined with existing deep clustering approaches to learn diverse clusterings from scratch, or implemented on top of a trained clustering model to build from the clustering it has already established, in order to explore alternative solutions. Experiments were conducted with multiple datasets and clustering frameworks to demonstrate the effectiveness of the proposed approach, and show that DivClust can control clustering diversity without reducing the quality of the clusters. A clustering aggregation method was also proposed, that combines the diverse clusterings learned by the model to produce a single, aggregate one. Experimental results show that the resulting aggregate clusterings are consistently superior to the ones produced by single-clustering frameworks, with regard to their overlap with the ground truth labels of the corresponding datasets.

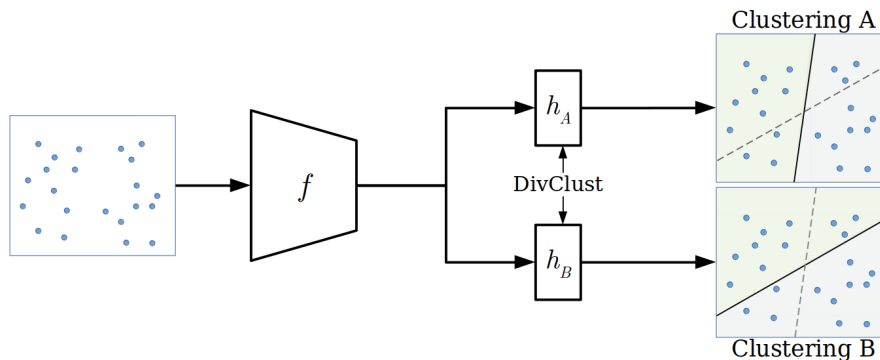


Figure 20. Illustration of the proposed framework assuming clusterings A and B with two clusters each. Given a set of data, a backbone network f , and projection heads h_k , each corresponding to a clustering k , DivClust restricts their similarity, enforcing that some samples belong to different clusters in each clustering.





4.6.2. DivClust

The architecture of QMUL’s method can be seen in Figure 20. It consists of a backbone network f , followed by K projection heads h_1, \dots, h_K , each corresponding to a clustering k . Assuming a set X of N unlabelled 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_k clusters. In this work, it is assumed that all clusterings have the same number of clusters C , so that $C_k = C \forall k$. Then, $p_k(x) = h_k(f(x)) \in \mathbb{R}^C$ represents the probability assignment vector mapping the sample $x \in X$ to C clusters in clustering k . The column $p_k(i) \in \mathbb{R}^C$, that is the probability assignment vector for the i -th sample, shows to which clusters that sample has been assigned. The row vector $q_k(j) \in \mathbb{R}^N$, that is the cluster membership vector for a cluster j , shows which samples are assigned to cluster j .

In order to assess the similarity between two clusterings A and B the inter-clustering similarity matrix $S_{AB} \in \mathbb{R}^{C \times C}$ is defined. Each entry $S_{AB}(i, j)$ represents the cosine similarity between the cluster membership vector $q_A(i)$ of the cluster i in A and the cluster membership vector $q_B(j)$ of the cluster j in B . That is,

$$S_{AB}(i, j) = \frac{q_A(i) \cdot q_B(j)}{\|q_A(i)\|_2 \|q_B(j)\|_2}. \quad (17)$$

This variable expresses the degree to which clusters i and j have been assigned the same samples, and is, therefore, a measure of their similarity. The similarity between any two clusters of two clusterings can then be controlled by applying constraints on the clustering similarity matrix S_{AB} . Toward that, two ways of controlling the inter-clustering similarity were proposed, with distinct loss functions: a) Cluster-wise diversity, where similarity constraints are applied to each pair of clusters, softly enforcing that no pair of clusters $i \in C_A$ and $j \in C_B$ have a cosine similarity $S_{AB}(i, j)$ greater than a desired similarity upper bound d . That is achieved via the loss presented in Eq. 18. b) Global diversity, where it is softly enforced that the two clusterings A and B do not have an *aggregate* similarity greater than a desired similarity upper bound d . Aggregate similarity is defined as the similarity of each cluster $i \in C_A$ with its most similar cluster $j \in C_B$, averaged over all clusters for each clustering. The loss is defined as in Eq. 19, where $[x]_+ = \max(x, 0)$.

$$\mathcal{L}_{CDiv}(S_{AB}) = \frac{1}{C_A} \sum_{i=1}^{C_A} \sum_{j=1}^{C_B} [S_{AB}(i, j) - d]_+, \quad (18)$$

$$\mathcal{L}_{GDiv}(S_{AB}) = \left[\frac{\sum_{i=1}^{C_A} \max_j(S_{AB}(i, j))}{C_A} - d \right]_+ \quad (19)$$

The total loss in the proposed framework, presented in Eq. 20, is the mean of the clustering loss \mathcal{L}_{main} at the individual heads, plus the mean of the diversity loss \mathcal{L}_{div} for each head. The former can be any of the losses proposed by deep clustering frameworks, for example that of PICA [2] or IIC [184]. The latter is either one of the proposed \mathcal{L}_{CDiv} and \mathcal{L}_{GDiv} losses.

$$\mathcal{L}_{total} = \frac{1}{K} \sum_{k=1}^K \mathcal{L}_{main}(P_k) + \frac{1}{K(K-1)} \sum_{k=1}^K \sum_{p \neq k}^K \mathcal{L}_{div}(S_{kp}) \quad (20)$$

4.6.3. Clustering aggregation

Complementary to DivClust, QMUL also proposed a clustering aggregation method, to utilise the diverse clusterings DivClust can produce in consensus clustering tasks, where a single clustering





solution is required from an ensemble of K non-identical clusterings. To solve this problem, the probability assignment tensor $P \in \mathbb{R}^{K \times N \times C}$ is assumed, for K clusterings, N samples and C clusters. The probability assignment matrix $P_A \in \mathbb{R}^{N \times C}$ of each clustering A is then projected to W_A as:

$$W_A = P_A \cdot P_A^T \in \mathbb{R}^{N \times N} \quad (21)$$

Each element of that matrix $W_A(i, j)$ is the inner product of probability assignment vectors $p_A(i), p_A(j) \in \mathbb{R}^C$, and reflects the confidence of the model that samples i and j are assigned to the same cluster. In this space, the results of multiple clusterings can be averaged to define the matrix W_{aggr} (Eq. 22). Each element $W_{aggr}(i, j)$ of this matrix indicates how frequently and with how much confidence two samples i and j were assigned to the same cluster, on average, over all clusterings K . It represents, therefore, an aggregation of those clusterings.

$$W_{aggr} = \frac{1}{K} \sum_{k=1}^K W_k \quad (22)$$

A mapping function is then required, that can combine the cluster assignment vectors $P_k(X)$ of each clustering k into a single assignment vector, that approximates W_{aggr} . Given $P_k \in \mathbb{R}^{N \times C}$, the joint cluster assignment vector is defined:

$$P_{joint} = [P_1, P_2, \dots, P_K] \in \mathbb{R}^{N \times CK} \quad (23)$$

The mapping function g is trained to project P_{joint} to the aggregate cluster assignment matrix $P_{aggr} = g(P_{joint}) \in \mathbb{R}^{N \times C}$, such that $\hat{W}_{aggr} = P_{aggr} \cdot P_{aggr}^T$ approximates W_{aggr} . This is achieved by minimising the MSE loss presented in Eq. 24. The training of g (a simple linear layer in QMUL's implementation) results in cluster assignments approximating the aggregate of the K diverse clusterings learned by the model.

$$\mathcal{L}_g = \text{mean}((P_{aggr} \cdot P_{aggr}^T - W_{aggr})^2) \quad (24)$$

4.6.4. Evaluation

DivClust is evaluated in several experimental settings to validate its effectiveness. The most significant of the conducted experiments are presented in Tables 9 and 10.

Datasets: The datasets used are CIFAR10, CIFAR100-20 [185] and STL-10 [186], standard datasets on which deep clustering frameworks are evaluated. CIFAR10 consists of 60,000 32X32 images separated among 10 classes. CIFAR100 consists of 60,000 images separated among 100 classes and 20 superclasses. Following previous deep clustering methods, the 20 superclasses are used as ground truth labels. Accordingly, this dataset is referred to as CIFAR100-20. Finally, STL-10 consists of 13,000 labelled images split between 10 classes, and 100,000 unlabelled images of size 96X96. For STL-10, again following the literature, models are trained on all samples and evaluated on the labelled part of the dataset.

Metrics: The objective of the proposed method is to generate diverse clusterings, without sacrificing quality, and which lead to an aggregate clustering with high overlap with the ground truth labels. To measure clustering diversity, the Normalised Mutual Information (NMI) metric is used. Specifically, the NMI between each pair of clusterings is calculated, and these values are averaged over all clusterings to measure how similar they are. Higher NMI values indicate more similar clusterings, therefore it is expected that the NMI should decrease as the similarity upper bound d





Method	Clusterings	Div. Loss	d	Tr.	CNF	NMI	Aggr. ACC	Max. ACC
PICA - Baseline	1	-	-	-	0.96	-	55.27	55.27
PICA+DivClust	10	Global	0.9	SC	0.944	0.824 / 0.827	61.9	67.81
				OT	0.9452	0.848	57.2	63.63
PICA+DivClust	10	Global	0.8	SC	0.942	0.711 / 0.75	62.62	62.27
				OT	0.934	0.743	55.51	63.07
PICA+DivClust	10	Cluster-wise	0.95	SC	0.936	81.5 / 76.07	59.75	62.11
				OT	0.944	0.734	56.64	60.6
PICA+DivClust	10	Cluster-wise	0.9	SC	0.942	0.708 / 0.667	61.07	61.34
				OT	0.921	0.6757	63.08	64.9
IIC - Baseline	1	-	-	-	99.7	-	44.43	44.43
IIC+DivClust	10	Global	0.9	SC	99.9	93.78 / 91.82	57.2	59.16
				OT	99.8	89.3	46.53	49.31
IIC+DivClust	10	Global	0.8	SC	99.8	87.9 / 86.72	56.8	57.5
				OT	99.8	82.06	54.42	57.86
IIC+DivClust	10	Cluster-wise	0.95	SC	99.8	90.3 / 89.6	54.9	56.2
				OT	99.8	89.66	46.2	46.82
IIC+DivClust	10	Cluster-wise	0.9	SC	99.8	80.00 / 82.21	58.9	60.08
				OT	99.8	89.78	47.78	48.27

Table 9. Results evaluating the effectiveness of DivClust in learning diverse clusterings on CIFAR10. When models were trained from scratch (SC), PICA was trained for 250 epochs and IIC for 1000. When trained on-top (OT), PICA and IIC models were trained for 200 and 1000 epochs, and additional clusterings were added subsequently in regular 25 and 50 epoch intervals respectively.

decreases. The avg. confidence of cluster assignments (**CNF**) is used to quantify clustering quality, as higher confidence in cluster assignments by the model is an indication that the clusters are well-defined. Finally, the accuracy metric (**ACC**) is used to measure the clusterings' overlap with the ground truth labels. The accuracy of the single aggregate clustering produced by DivClust and the proposed clustering aggregation method (**Aggr. ACC**), as well as the accuracy of the best performing clustering (**Max. ACC**), are provided. It should be noted, however, that there is no way to identify that best performing clustering without having access to the ground truth, hence the need for clustering aggregation.

Implementation: Regarding the training parameters of the models, they are trained as in the original papers. In the experiments provided, DivClust is trained with both the global (Eq. 19) and the cluster-wise (Eq. 18) diversity objectives, and it is combined with two deep clustering frameworks, PICA [2] and IIC [184]. Two distinct training scenarios are explored: a) Training from scratch (**SC**), where all clusterings are trained in parallel from the start. b) Training on-top (**OT**), where, initially, the model is trained with a single clustering as in its original framework, and additional clusterings are introduced incrementally in regular intervals.

In Table 9, DivClust is applied to CIFAR10. The objective is to evaluate whether it can, in fact, control clustering diversity with both PICA and IIC, under the various examined training configurations. The results demonstrate that: a) The proposed method is versatile, in that it can be combined with various deep learning frameworks and both proposed losses. b) It is effective, as it can achieve controlled diversity according to the similarity upper bound d without sacrificing clustering quality. c) It can be beneficial in consensus clustering tasks, as the resulting aggregate clustering is significantly and consistently better than the one achieved by the baseline, single clustering models.

In Table 10, the proposed method is applied *on-top* of PICA, having trained it with overclus-





Dataset	Epochs	Div. Loss	d	Baseline ACC.	Aggr. ACC (Mean±STD)	Max. ACC (Mean±STD)
CIFAR10	250	Global	0.8	68.16	72.61±3.67	69.74±0.44
CIFAR100-20	300	Cluster-wise	0.95	32.68	32.97±0.24	32.76±0.55
STL-10	400	Global	0.9	72.02	72.5±0.13	72.4±0.76

Table 10. Results when applying DivClust on top of a pre-trained clustering model. The baseline model was trained with PICA and overclustering, using the training configuration proposed in [2]. DivClust was trained on top for the number of epochs noted in the table, with a total of 10 clusterings being added incrementally.

tering (see [2] and [184] for details). Overclustering is dropped after additional clusterings are added. In the case of STL-10 the feature encoder is frozen, as dropping overclustering meant that the model was limited to the labelled section of the dataset, which would worsen the quality of the features. The results in Table 10 show that on all three datasets the application of DivClust and consensus clustering can improve clustering accuracy, compared to the initial baseline model which is trained to learn a single clustering.

A relevant paper is under preparation, to be submitted until the end of 2021.

4.6.5. Contributions to WP8 Use cases

Relevant WP8 Use Cases: 3A3-11 (Visual indexing and search), 3A3-12 (Visual concepts classification). DivClust is applicable to use-cases 3A3-11 and 3A3-12, since it can be used for visual indexing, searching and concept classification.

4.7. Joint Deep Dictionary Learning and Coding Network

Contributing partners: UNITN

4.7.1. Method Overview

The key step of classifying images is obtaining feature representations encoding relevant label information. In the last decade, the most popular representation learning methods were dictionary learning (or sparse representation) and deep learning. Dictionary learning is learning a set of atoms so that a given image can be well approximated by a sparse linear combination of these learned atoms, while deep learning methods aim at extracting deep semantic feature representations via a DNN. Scholars from various research fields have realized and promoted the progress of dictionary learning with great efforts, e.g., [187, 188] from the statistics and machine learning community, [189] and [190] from the signal processing community and [191, 192, 193] from the computer vision and image processing communities. However, what is a sparse representation and how can we benefit from it? These two questions represent the points we attempt to clarify among the fundamental philosophies of sparse representation.

In our research, we aim to improve the deep representation ability of dictionary learning. To this end, we propose a novel Deep Dictionary Learning and Coding Network (DDLDCN), which mainly consists of several layers, i.e., input, feature extraction, dictionary learning, feature coding, pooling, fully connected and output layer, as shown in Fig. 21. The design motivation of the proposed DDLDCN is derived from both Convolutional Neural Networks (CNNs) and dictionary learning approaches. However, the biggest difference is that the convolutional layers in CNNs are replaced by our proposed dictionary learning and coding layers. By doing so, the proposed DDLDCN can learn edge, line and corner representations from the shallow dictionary layers. Then additional sophisticated ‘hierarchical’ feature representations can be learned from deeper dictionary layers.



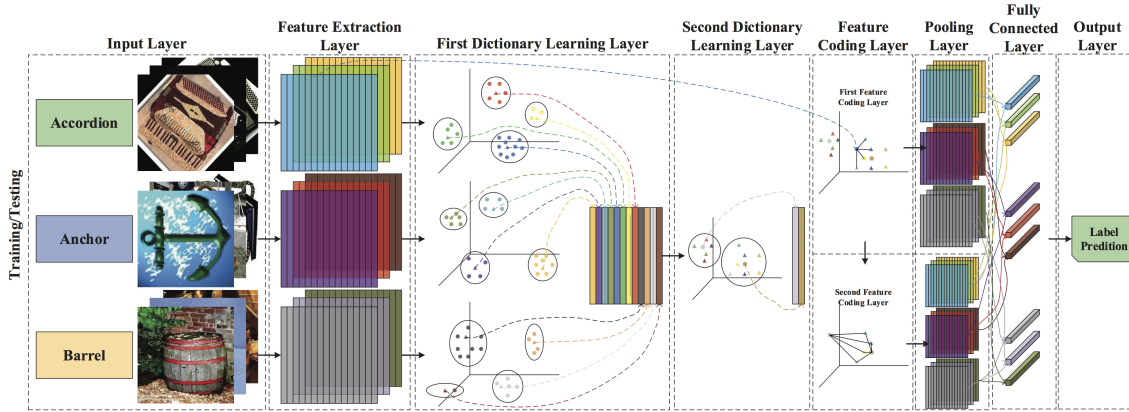


Figure 21. The framework of the proposed Deep Dictionary Learning and Coding Network (DDLCCN).

The proposed DDLCCN has a better approximation capability of the input since the introduction of the proposed dictionary learning and coding layer, which takes advantage of the manifold geometric structure to locally embed points from the underlying data manifold into a lower-dimensional deep structural space. Moreover, it also fully considers each fundamental basis vector adopted in the shallow layer coding, and incorporates additional gradient affects of nonlinear functions on it into the deeper local representation. Thus, the proposed DDLCCN can transfer a very difficult nonlinear learning problem into a simpler linear learning one. More importantly, the approximation power is higher than its single-layer counterpart.

Further, we sequentially introduce each layer of the proposed DDLCCN. Note that we only illustrate details of two-layer DDLCCN for simplicity. Extension of the proposed DDLCCN to multiple layers is straight forward.

Feature Extraction Layer. We first adopt a feature extractor F to extract a set of m -dimensional local descriptors $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_l] \in \mathbb{R}^{m \times l}$ from the input image I , where l is the total number of local descriptors. To highlight the effectiveness of the proposed method, we only use a single feature extractor in our experiment, i.e., Scale-Invariant Feature Transform (SIFT). The SIFT descriptor has been widely used in dictionary learning. However, one can always use multiple feature extractor to further improve performance. Specifically, for the input image I , we extract the SIFT feature \mathbf{y}_i by using the feature extractor F , this process can be formulated as $\mathbf{y}_i = F(I), i \in [1, \dots, l]$.

First Dictionary Learning Layer. Let r denote the total number of classes in the dataset. Then we randomly select p images in each class to train the dictionary of the corresponding class, and the number of the first layer dictionary per category is denoted as q . Thus, the number of the dictionary for the first layer can be calculated by $D_1 = r * q$. Next, we adopt the following dictionary learning algorithm,

$$\min_{\mathbf{V}_i} \left[\frac{1}{2} \|\mathbf{y}_i - \mathbf{V}_i \alpha_i\|_2^2 \right] \quad s.t. \quad \|\alpha_i\|_1 \leq \lambda \quad (25)$$

where $\mathbf{V}_i = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_q]$ is the dictionary for i^{th} class in the first-layer dictionary, which contains q atoms, i.e., \mathbf{v}_i . We then group all of them to form the first-layer dictionary \mathbf{V} after separately learning the dictionary of each class. Thus $\mathbf{V} = [\mathbf{V}_1, \mathbf{V}_2, \dots, \mathbf{V}_r] = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{D_1}] \in \mathbb{R}^{m \times D_1}$. α_i is a sparse coefficient introduced in [194]. By this way, the dictionary \mathbf{V}_i and the coefficients α_i can be learned jointly.

First Feature Coding Layer. After learning \mathbf{V} , each local feature is then encoded by \mathbf{V} through several nearest atoms for generating the first coding. By doing so, the first feature coding layer transfers each local descriptor \mathbf{y}_i into a D_1 dimensional code $\boldsymbol{\gamma}^1 = [\gamma_1^1, \gamma_2^1, \dots, \gamma_{D_1}^1] \in \mathbb{R}^{D_1 \times l}$.



Specifically, each code can be obtained using the following optimization,

$$\min_{\gamma_i^1} \left[\sum_{i=1}^l \frac{1}{2} \|\mathbf{y}_i - \mathbf{V}\gamma_i^1\|_2^2 + \beta \|\gamma_i^1 \odot \zeta_i^1\|_1 \right] \quad (26)$$

$$s.t. \quad \mathbf{1}^\top \gamma_i^1 = 1,$$

where $\zeta_i^1 \in \mathbb{R}^{D_1}$ is a distance vector to measure the distance between \mathbf{y}_i and \mathbf{v}_i . \odot denotes the element-wise multiplication. Typically, ζ_i^1 can be obtained by reducing a reconstruction loss in the corresponding layer. We note that [195] adopts a simple sparse coding model at the first layer, which overlooks the importance of quantity distributions of each item in the code γ_i^1 , thus it is prone to a rough approximation at the first layer. Therefore, the physical approximation of \mathbf{y} in the first layer can be expressed as,

$$\mathbf{y}' = \sum_{\mathbf{v} \in \mathcal{C}^1} \gamma^1(\mathbf{y}) \mathbf{v}, \quad (27)$$

where \mathcal{C}^1 is the set of anchor points to \mathbf{y} . An illustrative example is shown in Fig. 22.

Second Dictionary Learning Layer. Most existing dictionary learning frameworks only use a single layer, which significantly limits the discriminative ability of the feature coding. Meanwhile, we observe that better representation will be obtained by using deeper layers in most computer vision tasks. Thus, we borrow some idea from deep CNNs and present a new deeper dictionary learning and coding layer. Then the second layer dictionary $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{s_2}]$ can be learned from the first layer dictionary \mathbf{V} ,

$$\min_{\mathbf{U}} \left[\frac{1}{2} \|\mathbf{v}_i - \mathbf{U}\alpha_i\|_2^2 \right] \quad s.t. \quad \|\alpha_i\|_1 \leq \lambda \quad (28)$$

where $\mathbf{v}_i \in \mathbf{V}$ is one of the basis vectors in the first activated dictionary. At the second layer, we put more emphasis on the representation of each \mathbf{v}_i or each group of \mathbf{v}_i to further refine each basis \mathbf{v}_i . Specifically, after coding at the first layer, we try to map a nonlinear function f to a simplified local coordinate space with low intrinsic dimensionality. However, from the viewpoint of Lipschitz smoothness [195], this solo layer mapping only incorporates limited information about f with its derivative on \mathbf{y} , such that it is incapable of guaranteeing better approximation quality. That is why we would move deeper into the second layer to seek more information about f for further improving the approximation. By doing so, the first layer can capture the fine low-level structures from the input image, then the second coherently captures more complex structures from the first layer.

Second Feature Coding Layer. We can obtain the code of the second layer by using the following optimization,

$$\min_{\gamma_i^2} \left[\sum_{i=1}^{D_1} \frac{1}{2} \|\mathbf{v}_i - \mathbf{U}\gamma_i^2\|_2^2 + \beta \|\gamma_i^2 \odot \zeta_i^2\|_1 \right] \quad (29)$$

$$s.t. \quad \mathbf{1}^\top \gamma_i^2 = 1,$$

where $\gamma_i^2 = [\gamma_i^2(\mathbf{u}_1), \gamma_i^2(\mathbf{u}_2), \dots, \gamma_i^2(\mathbf{u}_{D_2})]^\top \in \mathbb{R}^{D_2}$ is the second coding and D_2 is the number dictionary of the second layer. $\zeta_i^2 \in \mathbb{R}^{D_2}$ is used to measure the distance between \mathbf{v}_i and each atom in \mathbf{U} . $\mathbf{v}_i \in \mathbf{V}$ is one of the basis vectors adopted in the representation of \mathbf{y}_i at the first layer.

By doing so, the activated atoms \mathbf{v}_i in the first layer can be further decomposed to obtain the second layer coding using \mathbf{U} . Thus, the approximation of \mathbf{y} in the second layer can be defined as,

$$\mathbf{y}'' = \sum_{\mathbf{v} \in \mathcal{C}^1} \left[\gamma^1(\mathbf{y}) \sum_{\mathbf{u} \in \mathcal{C}^{2,v}} \gamma^{2,v}(\mathbf{v}) \mathbf{u} \right], \quad (30)$$



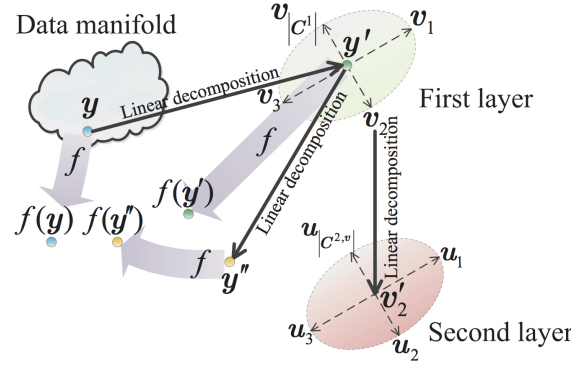


Figure 22. Multi layers coding strategy. The first layer is mainly used to partition the space, while the main approximation power is achieved within the second layer, which embodies a ‘divide and conquer’ strategy.

where $C^{2,v}$ is the set of anchor points to v . We also provide an illustrative example in Fig. 22 for better understanding. The core idea of the two-layer coordinate coding is that if both coordinate codings, i.e., y' and $v' = \sum_{u \in C^{2,v}} \gamma^{2,v}(v)u$, are sufficiently localized, then a point y lies on a manifold, which would be locally embedded into a lower-dimensional two-layer structure space. More importantly, not only the data point y is locally linearly represented, but also the function $f(y)$. This significant observation lays the foundation for our approach.

The n^{th} Dictionary Learning Layer. Similarly, we can learn the n^{th} dictionary $D^n = [d_1^n, d_2^n, \dots, d_{D_n}^n]$ from the previous layer dictionary D^{n-1} ,

$$\min_{D^n} \left[\frac{1}{2} \|d_i^{n-1} - D^n \alpha_i\|_2^2 \right] \quad s.t. \quad \|\alpha_i\|_1 \leq \lambda, \quad (31)$$

where $d_i^{n-1} \in D^{n-1}$ is one of the activated basis vectors in the previous $(n-1)^{th}$ dictionary layer.

The n^{th} Feature Coding Layer. Therefore, we can generalize the two-layer framework of DDLCN to a deeper one,

$$\min_{\gamma_i^n} \left[\frac{1}{2} \|d_i^{n-1} - D^n \gamma_i^n\|_2^2 + \beta \|\gamma_i^n \odot \zeta_i^n\|_1 \right] \quad (32)$$

$$s.t. \quad 1^\top \gamma_i^n = 1,$$

where γ_i^n is the n^{th} layer coding and ζ_i^n is employed to measure the distance between d_i^{n-1} and each atom in D^n . $d_i^{n-1} \in D^{n-1}$ is one of the basis vectors adopted in the feature representation of y_i at the $(n-1)^{th}$ coding layer. Through the proposed multi-layer learning and coding strategy, the proposed DDLCN can output a robust feature representation to accurately represent the input image. Moreover, DDLCN increases and boosts the separability of feature representations from different semantic classes. Lastly, DDLCN preserves the locality information of the input local features, avoiding very large values in the coding representation and reducing the error caused by over-fitting.

Pooling Layer. After the last dictionary learning and feature coding layer, we use a pooling layer for removing the fixed-size constraint of the input images [208]. Specifically, for each input image, we adopt 1×1 , 2×2 and 4×4 spatial pyramids with max-pooling.

Fully Connected Layer. The final feature representations of y_i can be obtained by integrating feature representation from each layer. Task two-layer framework for an example, each item (such as the j^{th} item) in the first layer’s codes γ_i^1 can be augmented into the form of $[\gamma_i^1(v_j), \gamma_i^1(v_j)[\gamma_j^2(u_1), \gamma_j^2(u_2), \dots, \gamma_j^2(u_{s_2})]]^\top$. Then we concatenate the first layer coding and



Table 11. Classification accuracy (%) on Caltech 256.

Num. of Train. Samp.	15	30	45	60
KC [196]	-	27.17 ± 0.46	-	-
LLC [197]	25.61	30.43	-	-
K-SVD [190]	25.33	30.62	-	-
D-KSVD [198]	27.79	32.67	-	-
LC-KSVD1 [199]	28.10	32.95	-	-
SRC [192]	27.86	33.33	-	-
Griffin [200]	28.30	34.10 ± 0.20	-	-
LC-KSVD2 [199]	28.90	34.32	-	-
ScSPM [201]	27.73 ± 0.51	34.02 ± 0.35	37.46 ± 0.55	40.14 ± 0.91
NDL [202]	29.30 ± 0.29	36.80 ± 0.45	-	-
SNDL [202]	31.10 ± 0.35	38.25 ± 0.43	-	-
MLCW [203]	34.10	39.90	42.40	45.60
LP- β [204]	-	45.8	-	-
M-HMP [205]	42.7	50.7	54.8	58.0
Convolutional Networks [206]	-	-	-	74.2 ± 0.3
VGG19 [207]	-	-	-	84.10
DDLDCN-2 (1-1)	26.30 ± 0.40	31.45 ± 0.21	34.69 ± 0.31	37.76 ± 0.25
DDLDCN-2 (15-15)	35.06 ± 0.26	41.26 ± 0.22	44.17 ± 0.35	47.48 ± 0.26
DDLDCN-2 (30-30)	45.25 ± 0.31	51.64 ± 0.51	55.11 ± 0.26	59.66 ± 0.45
DDLDCN-3 (30-30)	47.65 ± 0.22	54.28 ± 0.42	57.89 ± 0.32	62.42 ± 0.34

the second layer coding to form the final coding representation, which is a $D_1 \times (1 + D_2)$ dimensional vector.

Output Layer. We adopt the Support Vector Machine (SVM) as our classifier. Specifically, we employ LIBSVM to implement our multi-class SVM.

The classification of the input image is ultimately carried out by assembling deep dictionaries from different layers and assessing their contribution. Moreover, through jointly minimizing both the classification errors and the reconstruction errors of all different layers, the proposed DDLCN iteratively adapts the deep dictionaries to help to build better feature representations for image recognition tasks.

Evaluation. We evaluated the effectiveness of the proposed DDLCN on five widely-used datasets, i.e., Extended YaleB [209], AR Face [210], Caltech 101 [211], Caltech 256 [212] and MNIST [213], which are all standard datasets for dictionary learning evaluation. Note that we follow the same evaluation procedure with the previous works on each dataset for a fair comparison. For the sake of space we present here only the results for Caltech 256 dataset.

We conduct extensive experiments using 15, 30, 45 and 60 training images per class and compare with state-of-the-art methods. Table 11 shows the comparison results. We can see that the proposed DDLCN outperforms existing leading dictionary-based methods such as K-SVD, D-KSVD, LC-KSVD and LLC, which significantly validates the advantages of the proposed DDLCN.

Moreover, we observe that the proposed method achieves slightly worse results than both VGGNet [207] and convolutional network [206] when using 60 training samples. However, 1) [207] uses a very deep convolutional network, i.e., VGG19 [138], which consists of 16 convolutional layers and 3 fully connected layers. 2) Both [207] and [206] have limited practical applicability than our DDLCN since their reliance on careful hyper-parameter selection. 3) The proposed approach





has fewer hyper-parameters that need to be tuned. 4) Compared with [207] and [206], the feature learner and encoder of the proposed DDLCN are fixed after extracting features, and only the linear SVM classifier on top is needed to be updated during training. Thus, the training of DDLCN is offline and its testing is pretty fast. All these represent the big advantages of the proposed DDLCN.

4.7.2. Contributions to WP8 Use cases

Relevant WP8 Use Cases: 3A3-11 (Visual indexing and search), 3A3-12 (Visual concepts classification). DDLCN is a generic tool that allows combining the classical dictionary learning and deep learning and as such it can be directly used for visual indexing, searching and concept classification.

4.7.3. Relevant Publications

- H. Tang, H. Liu, W. Xiao, and N. Sebe, When Dictionary Learning Meets Deep Learning: Deep Dictionary Learning and Coding Network for Image Recognition with Limited Data, IEEE Transactions on Neural Networks and Learning Systems, 32(5):2129-2141, May 2021. Zenodo Record: <https://zenodo.org/record/5018256>.

4.7.4. Relevant software and/or external resources

- The code for our work "When Dictionary Learning Meets Deep Learning: Deep Dictionary Learning and Coding Network for Image Recognition with Limited Data" can be found in <https://github.com/Ha0Tang/DDLCN>.

4.8. Curriculum Self-Paced Learning

Contributing partners: UNITN

4.8.1. Method Overview

Training (source) domain bias affects state-of-the-art object detectors, such as Faster R-CNN [116], when applied to new (target) domains. To alleviate this problem, researchers proposed various domain adaptation methods to improve object detection results in the cross-domain setting, e.g. by translating images with ground-truth labels from the source domain to the target domain using Cycle-GAN [10]. On top of combining Cycle-GAN transformations and self-paced learning in a smart and efficient way, in our research, we propose a novel self-paced algorithm that learns from easy to hard. Our method is simple and effective, without any overhead during inference. It uses only pseudo-labels for samples taken from the target domain, i.e. the domain adaptation is unsupervised.

We propose a novel curriculum self-paced learning approach in order to adapt the object detector to the target domain. In self-paced learning, the model learns from its own predictions (pseudo-labels) in order to gain additional accuracy. Since we use image samples from the target domain during inference, the model has the opportunity to learn domain-specific features, thus adapting itself to the target domain. However, the main problem in self-paced learning is that the model can be negatively influenced by the noisy pseudo-labels, i.e. prediction errors. In order to alleviate this problem, we propose an effective combination of two approaches. In order to reduce the labeling noise level we apply a domain-adaptation approach that relies only on ground-truth labels before the self-paced learning stage. The approach consists in training a Cycle-consistent



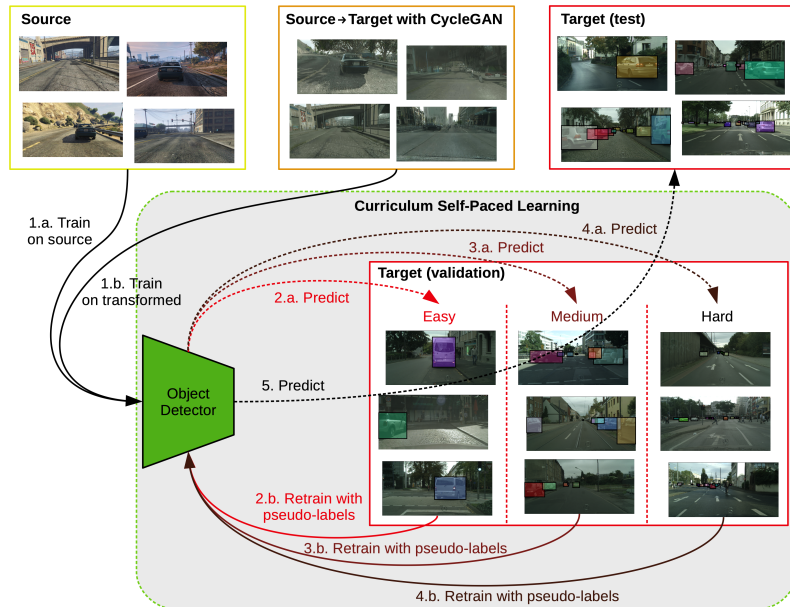


Figure 23. Our curriculum self-paced learning approach for object detection. In the initial training stage (step 1.a), the object detector is trained on source images with ground-truth labels. In step 1.b, the object detector is further trained on source images translated by Cycle-GAN [10] to resemble images from the target domain. In steps 2, 3 and 4, the object detector is fine-tuned on real target images (different from those included in the test set), using the bounding boxes and the labels predicted by the current detector. In step 5, the model makes its predictions on the target test set for the final evaluation. Best viewed in color.

Generative Adversarial Network (Cycle-GAN) [10] in order to learn how to transform images from the source domain to the target domain. The adaptation consists in fine-tuning the object detector on source images that are translated by Cycle-GAN to look like target images (see Figure 23 for some translated samples). In the experiments, we show that reducing the labeling noise before self-paced learning is indeed helpful, but still not satisfactory.

We hypothesize that the labeling noise inherently induced by the prediction errors is proportional to the difficulty of the images. Following this intuition, we perform self-paced learning starting with the easier images and then gradually adding more difficult image samples, inspired by the curriculum learning paradigm [214], as shown in Figure 23. Our hypothesis turns out to be supported by the empirical results, confirming the utility of our curriculum self-paced learning method. In order to estimate the difficulty of each image sample, we employ a score given by the number of detected objects divided by the average area of their bounding boxes. This is inspired by the previous work of Ionescu et al. [215], which found that image difficulty is directly proportional to the number of objects and inversely proportional to the average bounding box area.

We evaluate our curriculum self-paced learning approach on three cross-domain benchmarks, Sim10k→Cityscapes, KITTI→Cityscapes and PASCAL VOC 2007→Clipart1k, comparing it with recent state-of-the-art methods [3, 4, 5, 6, 7], whenever possible. The empirical results indicate that our approach provides the highest absolute gains (with respect to the baseline detector) and superior performance compared to all these methods [3, 4, 5, 6, 7]. Furthermore, we consider that our performance gains of +17.01% on Sim10k→Cityscapes, +12.34% on KITTI→Cityscapes and +8.91% on PASCAL VOC 2007→Clipart1k are significant (see the results in Table 12).





Model	Train Data	Sim10k→City	KITTI→City
Baseline Faster R-CNN [3]	Source	30.12	30.20
Baseline Faster R-CNN [4]	Source	31.08	31.10
Baseline Faster R-CNN [5]	Source	34.60	-
Baseline Faster R-CNN [6]	Source	30.10	30.20
Baseline Faster R-CNN [7]	Source	33.96	37.40
Baseline Faster R-CNN (ours)	Source	30.67	29.75
Domain-adapted Faster R-CNN [3]	S+T (no labels)	38.97 (+8.85)	38.50 (+8.30)
Domain-adapted Faster R-CNN [4]	S+T (no labels)	42.56 (+11.48)	42.98 (+11.88)
Domain-adapted Faster R-CNN [5]	S+T (no labels)	40.70 (+5.80)	-
Domain-adapted Faster R-CNN [6]	S+T (no labels)	39.60 (+9.50)	41.80 (+11.60)
Domain-adapted Faster R-CNN [7]	S+T (no labels)	43.02 (+9.06)	42.50 (+5.10)
Domain-adapted Faster R-CNN (ours)	S+T (no labels)	47.68 (+17.01)	42.93 (+13.18)
In-domain Faster R-CNN [4]	Target	68.10	68.10
In-domain Faster R-CNN [5]	Target	53.10	53.10
In-domain Faster R-CNN (ours)	Target	62.73	62.73

Table 12. Average Precision (AP) scores (in %) of several Faster R-CNN models trained using different state-of-the-art domain adaptation methods [3, 4, 5, 6, 7] versus a Faster R-CNN model trained using our domain adaptation approach based on curriculum self-paced learning. All domain adaptation methods include images without ground-truth labels from the target domain. Faster R-CNN baselines without adaptation (trained only on source) are also included to point out the absolute gain of each domain adaptation technique, with respect to the corresponding baseline. Faster R-CNN models trained on target domain images with ground-truth label are included as indicators of possible upper bounds of the AP scores. Results are reported for Sim10k→Cityscapes and KITTI→Cityscapes benchmarks. The best AP scores and the highest absolute gains are highlighted in bold. S+T indicates Source + Target.

4.8.2. Contributions to WP8 Use cases

Relevant WP8 Use Cases: 3A3-11 (Visual indexing and search), 3A3-12 (Visual concepts classification). Our approach is a generic tool that allows curriculum self-paced learning and as such it can be directly used for visual indexing, searching and concept classification.

4.8.3. Relevant Publications

- P. Soviany, R. Ionescu, P. Rota, and N. Sebe, Curriculum self-paced learning for cross-domain object detection, Computer Vision and Image Understanding, vol. 204, Article 103166, March 2021.
Zenodo Record: <https://zenodo.org/record/5142259>.





5. Music Annotation and Audio Provenance Analysis

5.1. Overview

T5.6 is about developing advanced audio analysis components for two domains: (a) automatic music annotation and music similarity analysis, and (b) audio partial matching/reuse detection and audio phylogeny analysis.

Regarding (a), this Section presents improvements regarding music similarity (by FHG-IDMT) and the generation of music mixes based on MIDI (by IRCAM). As for (b), a novel FHG-IDMT approach for audio phylogeny analysis with improved computational efficiency is then described.

5.2. Disentanglement Representation Learning for Music Similarity

Contributing partners: FHG-IDMT

5.2.1. Method Overview

Choosing the best suitable musical track can be hard, time consuming, and requires deep expert knowledge in the field. Traditionally, one could use metadata associated to the particular music track to speed up the retrieval process. Such metadata could be manually provided by music experts or extracted automatically. Another possibility is to use the similarity relations between musical pieces or parts of those. The latter method is particularly useful for the music replacement task, where an alternative song to a given query song shall be retrieved with the highest possible similarity.

Music similarity is not well-defined in multiple ways. First, the specificity spectrum of music similarity tasks is quite broad as shown in Fig. 24. In our current research, we focus on the medium

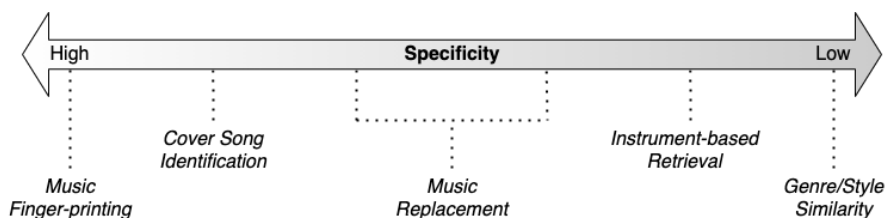


Figure 24. Specificity range of music similarity tasks.

specificity range that corresponds to the music replacement task. Second, music is inherently multi-dimensional and the definition of music similarity often depends on the targeted application scenario. Depending on each user's preferences, particular musical dimensions can have a stronger influence on the similarity perception between two songs than others. In the music replacement task, often additional similarity requirements exist that require an emphasis for instance on musical genre, mood, or instrumentation, among other high-level musical concepts. By adjusting those dimensions, configurable similarity spaces can be created to target certain pre-defined similarity tasks. As a consequence, a wide range of tasks of the specificity spectrum can be covered with a single framework (see Fig. 25 & Fig. 26).

Our research aims at developing a flexible music similarity algorithm which can be configured according to the six musical dimensions musical genre, mood, instrumentation, era, tempo, and key.



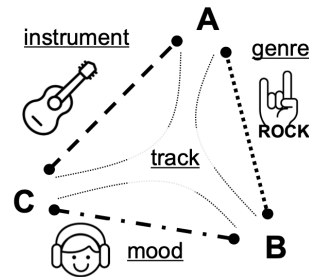


Figure 25. Music similarity dimensions [11].

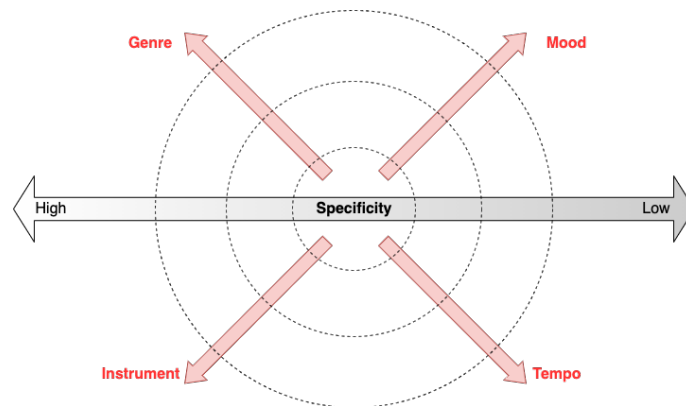


Figure 26. Similarity spaces created by combination of musical dimensions.

The proposed solution is based on metric learning for music similarity using a Conditional Similarity Network (CSN) [216, 11]. Here, the metric learning approach is combined with disentanglement during the training procedure.

The learnt latent space is tailored to the notion of similarity: similar samples are close in the latent space and dissimilar samples are far away from each other. Audio samples are mapped to the latent space using a deep embedding function as shown in Fig. 27.

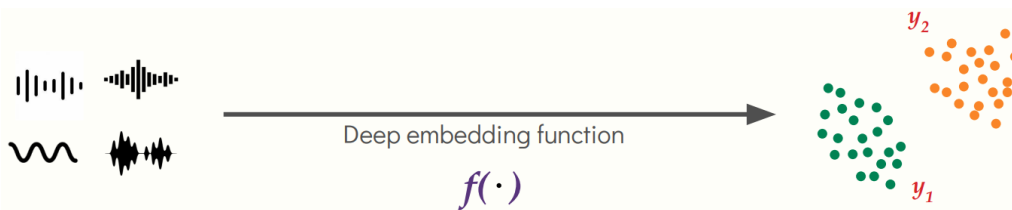


Figure 27. Deep metric learning [12].

In addition to metric learning, we aim to disentangle the aforementioned musical dimensions in the learnt embedding space. We use triplet learning within a CSN to combine both metric learning and disentanglement in one training procedure (see Fig. 28). Here, conditional triplets that include an *anchor*, a *positive*, and a *negative* example are selected according to existing annotations of musical genre, mood, instrumentation and era in the training data. Additionally, tempo and key information for each track are extracted using the state-of-the-art music signal processing algorithms contained within the Madmon python library [217]. At each training step, the parts of the embedding vectors that do not correspond to the current dimension-of-interest are





masked-out with zeros. In this way, non overlapping equal portions of the embedding space are assigned to each musical concept taken into account.

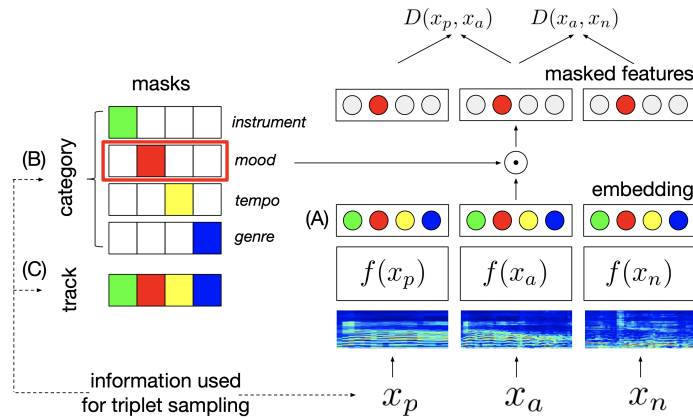


Figure 28. Overview of the conditional similarity network and the embedding masking procedure [11].

The CSN is trained with the triplet loss function $\mathcal{L}_T = \max\{0, D(f(x_a), f(x_p)) - D(f(x_a), f(x_n)) + \Delta\}$ with $f(\cdot)$ denoting the neural network mapping function applied to the input features x_a , x_p , and x_n of the anchor, positive, and negative, respectively, and Δ denoting a fixed margin value. The triplet sampling is done online in parallel for each musical dimension through a developed conditional sampling strategy based on the semi-hard negative mining approach [218], in order to speed-up convergence. During the training procedure we apply a track regularization to enforce consistency across multi-dimensional embedding spaces and combine both the triplet loss \mathcal{L}_T and the track regularization loss \mathcal{L}_{TR} as $\mathcal{L} = \mathcal{L}_T + \lambda \mathcal{L}_{TR}$. The weighting factor λ defines the trade-off between low and high-specificity, i.e., between semantic similarity and the self-similarity within the same musical track.

The backbone network $f(x)$ is an Inception network variant [11, 219] consisting of

- 1 x convolution layer
- 6 x inception blocks (1 naive + 1 dimension reduction module)
- 1 x dense layer (256 neurons)
- 1 x layer normalization

This architecture allows an increase in number of neurons per stage without excessive computational cost. Fig. 29 and Fig. 30 illustrate the architecture of the naive and dimension reduction modules.

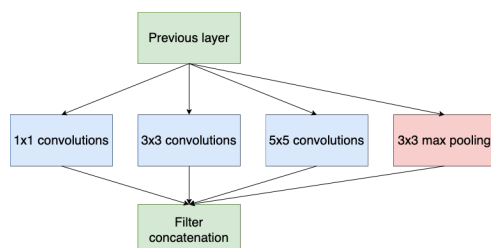


Figure 29. Naive module architecture

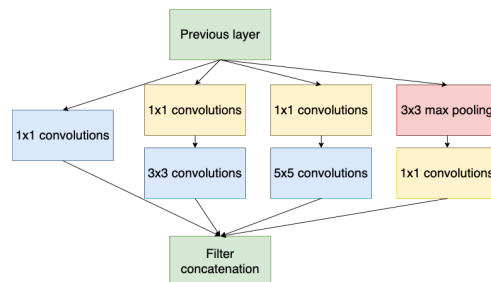


Figure 30. Dimension reduction architecture





For the experimental part we implement the CSN in Keras² using a data generator approach as shown in Fig. 31.

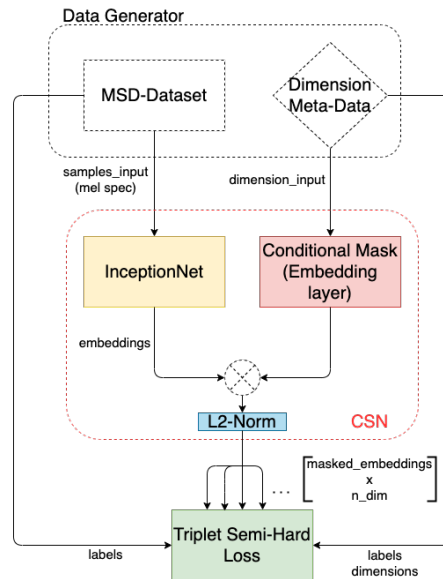


Figure 31. Implemented CSN-model architecture.

The training triplets for the genre, mood, instrument, and era dimensions are generated based on the annotations in the Million Song Dataset [220]. During training, the developed conditional semi-hard negative mining strategy selects negatives of the respective concept within a margin to the positive examples. This ensures the sampling of triplets with increasing difficulty, which favours a faster convergence.

We evaluate the trained representation using the DIM-SIM dataset [11]. This dataset includes around 4000 conditional triplets of 3s samples from the MDS test set. These triplets include similarity annotations by 5-12 people per triplet, leading to around 40K annotations (see [11] for details on the DIM-SIM dataset). We disregard the annotations with lower annotator agreement, and use around 450 high-agreement triplets with annotator agreement higher than 90%. We analyze influence of multi-dimensionality on the representation space and evaluate learned embeddings for the music similarity task.

As shown in Table 13, we compare our results against two baseline systems. The first system [221] similarly uses a CSN and triplet learning with 4 input dimensions based on mel spectrogram input. The second system is based on the openL3 embeddings [222]. The table shows that three out of four configurations of the proposed models outperform the baseline systems. The highest triplet score, i. e., the percentage of correctly assigned test set achors, of 0.838 was achieved for the proposed multi-input model with 6 dimensions. The results of this research will be submitted to the ICASSP 2022 conference.

5.2.2. Contributions to WP8 Use cases

Relevant WP8 Use Cases: 5B2 (Musical recording analysis). Learned representations can be used in the Epic 5B2 of the use-case UC5 (AI for Games), in order to find the most suitable

²<https://keras.io>





Table 13. Results of the similarity evaluation obtained for the implemented models without track regularization.

Model	Number of Dimensions	Multi-Input	Embedding Size	Triplet Score
<i>Baseline</i> [221]	4		256	0.8192
<i>OpenL3</i> [222]	-		512	0.7958
<i>Proposed Model</i>	4	x	256	0.8286
	6		258	0.7934
<i>Model</i>	6		384	0.8169
	6	x	384	0.8380

audio track based on the musical example and user preferences regarding the definition of music similarity.

5.3. Realistic Music Mixes Generation

Contributing partners: IRCAM

5.3.1. Method Overview

Among different topics related to music and sounds, IRCAM works on automatic audio analyses to retrieve music information from audio signals. With Machine Learning approaches and especially with DNNs, a central problem is the availability of properly annotated and sufficiently large training datasets. There exist many problems for which the annotation of a large quantity of annotated data is extremely difficult. Source separation for example requires the availability of the final music mixes together with the corresponding separated tracks. This situation has led to numerous approaches to use state of the art signal processing algorithms to synthetically produce annotated datasets for training DNNs. One of the most straight-forward approach is the use of MIDI synthesis and disklaiviers for the generation of large annotated datasets [223, 224]. Unfortunately, it was soon noted that the training with MIDI synthesized audio is not sufficiently realistic to be of practical use, and using only disklaiviers strongly limits the diversity of the instruments that are available for training. Another approach consists in applying high quality signal transformation algorithms (e.g. pitch shifting and time stretching) to augment the diversity of relatively small publicly available database [225, 226], or in creating annotated synthetic data by means of using high quality analysis/resynthesis algorithms [227, 228]. Nevertheless, in terms of variety of different songs, musical genres, and instrumentations, the diversity of the augmented dataset remains limited to the diversity of the initial dataset. Another successful approach consists in using a network that is trained on an annotated data to produce a predicted dataset from new data without annotation, see e.g. [229]. To be effective, this method requires the availability of special pre-conditions which are not possible for all training tasks.

To overcome the issue of annotating datasets, the present research aims to develop innovative algorithms to generate realistic music mixes based on a symbolic musical representation, that is the MIDI format. Initially developed in the context of Task 5.2, this approach makes possible the synthesis of instrument mixes of music pieces, for creative applications for example, but in the



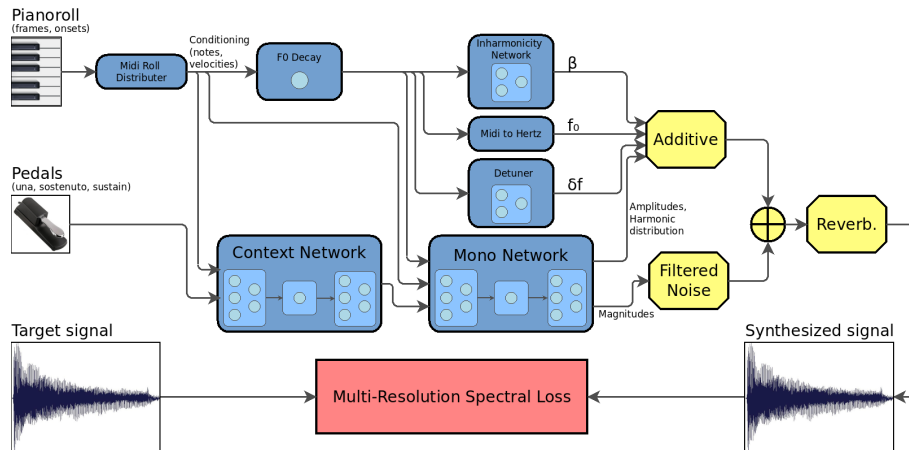


Figure 32. Synthesizer Architecture. The blue boxes represent the trained modules for the control of the synthesis. The synthesis modules from DDSP are represented by yellow boxes (Additive, Filtered Noise, and Reverberation). Finally, the Multi-Resolution Spectral Loss compares the input target signal (bottom left) and the output synthesized sound (bottom right).

context of Task 5.6 it also makes possible the generation of wide musical datasets based on symbolic musical representations. This music sound generator will be able to produce datasets of realistic audio with the associated digital scores. Consequently, we have access to many information directly contained in the MIDI score, for some tasks such as: recognition of key and mode, recognition of chord progression, automatic transcription, tempo estimation, instrument recognition, down-beat detection.

To achieve the objectives, this work exploits the trainable Differentiable Digital Signal Processing Synthesizer (DDSP) [230] and the Generative Adversarial Networks (GANs) [231]. The DDSP framework offers traditional sound synthesizers controllable by neural networks. The modularity of a DDSP-based architecture enables the injection of acoustic modeling knowledge into a DNN framework, which alleviates the need for a large quantity of training data. This is of particular interest for generating larger datasets of music annotations from a modest amount of initial data. The GAN architecture, in form of CycleGANs, made possible the domain translation of photos without making use of parallel datasets [232], for example by translating summer landscapes to winter landscapes. In this work, GAN type generators are used to create an audio augmentation network that allows enriching synthetic music with the variability and details that characterize real music. The research is inspired by [233] in that it aims to build artificial training data containing the relevant details that are present in real world.

5.3.2. Piano synthesis for annotation of piano performances

To facilitate the start of this work, the first and current step deals with the sound synthesis of piano notes only. The focus is on this instrument because some datasets exist with audio and MIDI scores [223, 224]. The developed method is based on a DDSP synthesizer which is composed of: an additive sinusoidal module for the production of harmonics or partials of the music tones, a filtered noise module for the residual signal and a reverberation module [230].

The new derived architecture is illustrated in Figure 32. It enhances to reproduce particular sound properties of the piano, such as partials inharmonicity, hammer and key noises and partials beating. All these acoustic properties are essential to build realistic training datasets: MIR models can be sensitive to some acoustic details, which non-realistic synthetic data fail to emulate. Piano

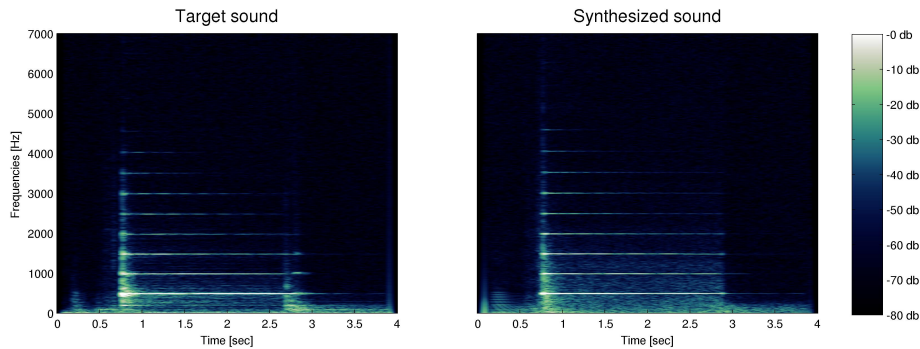


Figure 33. Spectrograms of Piano notes: target tone and synthesised sound.

notes are characterized by their partials being non-integer multiples of their “fundamental” frequencies. The global inharmonicity model from [234] is included and learned in our synthesizer. Furthermore, the majority of piano strings are doubled or tripled for each note, which leads to a recognizable double decay of the note amplitude, and partial beatings due to slight detunings from one string to another, see [235]. A monophonic recurrent network is used for capturing the partial amplitudes and decays, while a detuning factor is learned for re-creating the beating patterns. The neural network also controls a filtered noise module that simulates residual sounds, such as key strokes and pedal noises. The polyphony context is encoded with the pedal inputs and sent to the monophonic model to modify the partial amplitudes and the noise filter for emulating sympathetic resonances and raised damper noises.

An example of a piano note synthesised by the model is shown in Figure 33, compared with the ground-truth audio. The model was trained on a set of isolated notes of a real piano from [223], with various pitches and velocities. One can notice the frequency distribution and the amplitude decays of the note partials (horizontal lines) are well reproduced by the model. The noise filtered module complements the partials by adding residual noise during note onset and sustain, which resembles the target noise. Informal listening tests indicate that the added noise effectively improves realism. Note that this synthesised target tone has not been included in the training dataset.

In a next work, the method will be trained on full performances, instead of isolated notes, and compared to other synthesis approaches: a text-to-speech method adapted for piano synthesis [236], a WaveNet method [224], the open-source sample-based software *fluidsynth*³ and a commercial physical-based software *Modartt’s Pianoteq*⁴. A listening test will be conducted to evaluate the subjective quality of the synthesized piano sounds, to get a Mean Opinion Score for all tested methods.

This work is on-going and has not been published yet. A first submission in a conference is planned to present the method and its subjective evaluation.

5.3.3. Contributions to WP8 Use cases

Relevant WP8 Use Cases: 5B2 (Musical recording analysis). The developed approach is directly related to the Epic 5B2 of the use-case UC5 (AI for Games). The goal is to help Game Audio Designers when choosing suitable music tracks for a game.

³<https://www.fluidsynth.org/>

⁴<https://www.modartt.com/pianoteq>





← Phylogeny Analysis for Content De-Duplication

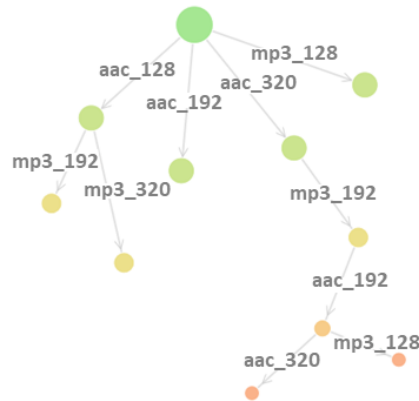


Figure 34. Phylogeny analysis results for a set of near-duplicates, as visualized within the respective software tool: Nodes represent audio files, connections represent parent-child relations between nodes.

5.4. Improving Audio Provenance Analysis with CNN

Contributing partners: FHG-IDMT

5.4.1. Audio Provenance: Method Overview

Retrieving information about the processing history and relationships among content items is key for multimedia asset management and disinformation detection, and *Audio phylogeny* aims at automatically detecting such relationships between audio objects. It provides a reconstructed phylogeny tree where nodes represent objects and edges represent causal relationship between them (e.g. operations/transformations that lead from one node to another), see Fig. 34. There are several well-performing methods for audio phylogeny, but all of them are only capable of detecting a very limited set of transformations [237, 238, 239], and extending this set, while being crucial for forensics and archival purposes, increases the complexity significantly.

As introduced in [240], *partial audio matching* aims at detection and localization of arbitrary partial matches, the existence and position of which is unknown (see Fig. 35). Partial audio matching has many applications in the fields of multimedia asset management and media forensics (see [241]). The combination of audio phylogeny and partial matching would allow us to create a system for advanced, integrated provenance analysis that can: analyze phylogeny forests (multiple phylogeny trees), introduce transformations as cut and paste (between trees/nodes) and execute phylogeny analysis on a segment level. Such a system would represent a unique approach in comparison to the state of the art, and provide a big improvement in the field of audio reuse detection.

As a part of the task 5.6 activities related to Audio Provenance Analysis, our first goal is to propose an audio phylogeny approach that is computationally efficient and can handle a large, easily extendable set of audio transformations. Based on that, our second goal is to combine the enhanced audio phylogeny and partial audio matching to achieve an improved, integrated audio provenance analysis.



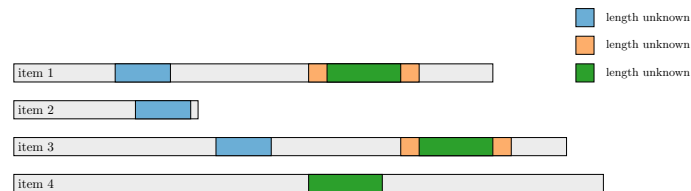


Figure 35. Partial audio matching without query.

5.4.2. CNN-based Audio Phylogeny Analysis

Up to now the main focus of our project work has been the development of an improved audio phylogeny approach that achieves high computational efficiency and can handle a large, easily extendable set of audio transformations. Audio phylogeny analysis uses a set of near duplicates files (same audio content with some audio transformations applied, e.g., encoding, fading or trimming) as input. The goal is to estimate a dissimilarity value for each pair of files and store it in an overall dissimilarity matrix. The calculation of the dissimilarity matrix is usually the most time-consuming part of phylogeny analysis. The reason for this is that the state-of-the-art approaches apply all possible transformations (or a pre-filtered set of them) to potential parent nodes in order to find the most likely transformation (and dissimilarity value). This extensive search for the best fitting audio transformation for every audio pair is what makes current approaches very inefficient. And for the same reason, it is difficult for state-of-the-art approaches to extend the set of considered audio transformations without further increasing the computational complexity.

In order to improve this, we propose using an intelligent transformation assessment between every pair of nodes by facilitating convolutional neural networks (CNNs). The idea is to train a CNN to use mel spectrograms (where frequencies are converted to the the mel scale, which is linear in low frequencies and logarithmic in high frequencies) of a parent and a child object as input and then estimate the most probable audio transformation between them. Therefore, this approach would improve scalability, by avoiding the exhaustive search for the most likely transformation. Moreover, extending the set of transformations would be done by re-training the network and would not affect the computation time of the actual phylogeny tree reconstruction.

Towards achieving our goal, we have established a new CNN-based approach for transformation detection in audio phylogeny that estimates the probability of applied transformations between two nodes. The network is based on the ResNet architecture. In our preliminary test we used a set of 11 different audio transformations. Based on the network's predictions, we were able to successfully reconstruct the phylogeny trees using the Oriented Kruskal algorithm.

Figure 36 shows the process of phylogeny analysis for one pair of audio files: potential parent a and potential child audio file b . The mel-spectrograms of these two audio files are given as input to the network that then outputs the predicted transformation that should be applied to a in order to get a version that is as close as possible to b . After applying the transformation on audio file a (in this case mp3 encoding with 128 kbit/s), the dissimilarity value is calculated between transformed $mp3_{128}(a)$ and b and saved in the dissimilarity matrix that holds 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 37 shows the results of a brief evaluation that we conducted to assess how well the current neural network detects transformations between two audio files. In our current phylogeny analysis set-up (as described above), this information is critical for successful tree reconstruction and influences all tree reconstruction metrics presented in Figure 37 (for detailed description of phylogeny tree reconstruction metrics please see [238]). In this evaluation we have reconstructed 40 phylogeny trees with 20 nodes each. From Figure 37 we can see that our CNN has correctly pre-



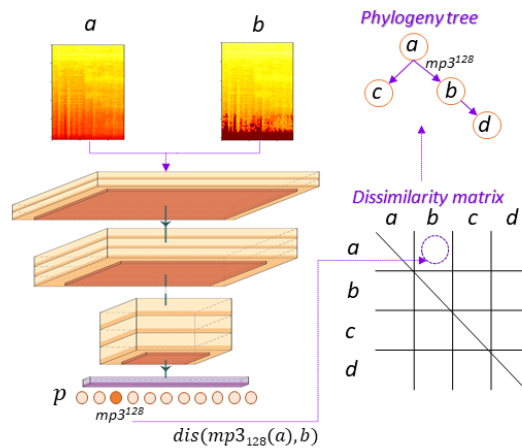


Figure 36. Process of phylogeny analysis for one pair of audio files, using CNN for transformation prediction.

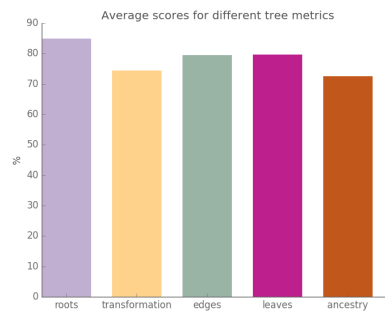


Figure 37. Score of different tree metrics for phylogeny tree reconstruction. Averaged over 40 different phylogeny trees

dicted transformations from parent to child node for 74.5% of edges in the reconstructed phylogeny trees (yellow bar). Hereby, we managed to deliver a proof of concept for our idea, successfully train the network for transformation detection, and use its prediction for phylogeny tree reconstruction.

This is still on-going work and a submission of a relevant paper is planned for 2022.

5.4.3. Contributions to WP8 Use cases

Relevant WP8 Use Cases: 1A3 (Synthetic Audio Detection/Verification), 4C3 (Audio analysis). Audio Provenance Analysis contributes to use-cases 1A3, by providing tools for verifying audio content, and 4C3, by providing tools for the comparison and tracking of content in archives.





6. Conclusions and Future Work

This document presented AI4Media research activities, concerning Tasks T5.1, T5.3, and T5.6, for the period M1-M12 of the project. This research involves media analysis and summarisation, machine learning in the face of data scarcity, as well as automated music analysis and annotation. Almost all of the presented methods rely on DNNs, with several of them having already been completed. Significantly, thanks to the clear focus of WP5, the majority of the discussed methods are directly linked to the use-cases of WP8 of the project. Overall, the work discussed in this deliverable is of very good quality and evidently aligned with WP5 objectives. Based upon research which is reported in this document, up to M12 of the project, 5 papers have been submitted and 8 papers have been already accepted to well-known, relevant scientific journals or conferences. 1 related method has been integrated into the AI4EU “AI on Demand” platform, while software implementations of 3 additional methods/systems that are presented here are available on-line.

Regarding the outcome of Task T5.1, this deliverable presents both newly developed AI-based algorithms/methods and relevant literature surveys that have been conducted. Their scope ranges from unsupervised video summarisation/key-frame extraction (CERTH, AUTH) to automatic media dataset creation, curation and management (RAI), and from information retrieval on cultural media datasets using symbolic/computational AI hybridization (3IA-UCA) to joint low-level and semantic video analysis, with a focus on simultaneous object instance segmentation and optical flow estimation (JR). The common theme is modern AI for image/video analysis and summarisation, with obvious applications in automated search, management, enrichment and update of media archives.

As far as Task T5.3 is concerned, this deliverable presented novel methodologies that have been developed or are under active development and focus on training or adapting DNNs for scenarios marked by a lack of large-scale, domain-specific datasets and/or annotations. The scope of the discussed methods covers few-shot object detection (JR, UPB), unsupervised domain adaptation for traffic density estimation/counting (CNR) or for visual object detection (UNITN), advanced video browsing and search (CNR), semi-supervised learning for fine-grained visual categorization (UNIFI), deep clustering for creating data pseudolabels (QMUL), as well as deep dictionary-based representation learning (UNITN). The common running theme is handling data and/or annotation scarcity when training or adapting DNNs, mostly for image/video analysis tasks, although certain algorithms are rather generic machine learning methods in nature.

Regarding Task T5.6, the work presented in this deliverable consisted of novel methodologies on advanced audio analysis for automatic music annotation and audio partial matching/reuse detection. The development of all discussed algorithms is still on-going, while their scope extends to automated music similarity analysis (FHG-IDM), music mixes generation based on MIDI (IRCAM), as well as to efficient audio phylogeny analysis (FHG-IDM).

Even though the activities reported in this document are only the outcomes of the first project period, future research plans have already been laid, with the intention to expand upon work that has been presented here. Thus, in Task T5.1, CERTH and AUTH intend to build on existing state-of-the-art unsupervised video summarisation methods, investigating neural architectural improvements and model selection criteria for choosing the optimally trained model. Furthermore, exploitation of complementary deep neural architectures designed for tasks such as activity recognition and/or image/video captioning will be attempted, in order to enhance unsupervised video summarisation performance, while deep dictionary learning-inspired methods will be adopted to extract more representative and/or more salient key-frames. JR will implement a first prototype for joint optical flow and segmentation, based on the analysis of the state of the art which has been performed. RAI will continue its current work by directly implementing end-to-end face verification and landmark detection pipelines based on archival content and metadata, able to adapt





dynamically to various contexts and genres. These will serve as a basis for augmenting or building reference data sets, in order to retrain/refine existing models in a fully automated way. Finally, 3IA-UCA will further investigate the combination of learning and reasoning to analyze media data, particularly to further media understanding with learning and first-order logic, identifying connected concepts and extracting relational properties from multimedia data.

In Task T5.3, CNR intends to adopt solutions for learning with scarce data in various application domains and using various techniques (e.g., domain adaptation, anomaly detection, studies on sample efficiency, applications to video browsing and searching), while UPB will focus on improving multi-scale feature learning in the FSOD framework, introducing attention at each level in the feature pyramid. Also, the possibility of ensembling two-stage and one-stage detectors will be investigated, in order to benefit from both types of models. JR will continue improving incremental training for few-shot detection and plans to investigate sampling training data from videos using tracking and instance search. This will be done by integration with the existing object detection and tracking framework [242]. The other aspect that will be addressed in future work is the extension to few-shot object segmentation. QMUL will focus on self-supervised representation learning, exploring the use of clustering as a source of pseudolabels and the development of computationally efficient frameworks, while 3IA-UCA will investigate adversarial active learning, where adversarial attacks help define and reach theoretical bounds on the minimum amount of data to train to a certain error a DNN. Finally, UNITN will concentrate on the problem of low-budget label query, which aims at maximizing the classification performance by selecting a convenient and small set of samples (i.e., low budget) to be manually labeled from an arbitrary big set of unlabeled data. An Unsupervised Domain Adaptation (UDA) method will be first considered, to better align source and target domains using consistency constraints. Then, using the previously trained model as reference, effective selection methods will be investigated for selecting the samples to be labeled.

Regarding Task T5.6, FHG-IDMT aims to use representation learning for music annotation, semi-supervised learning and domain adaptation, while future work on audio phylogeny and partial matching will involve experimentation with different network architectures, optimizing accuracy of transformation detection, as well as the integration of phylogeny analysis and partial matching. Finally, IRCAM plans to adapt its current architecture for realistic music mixes generation to be able to learn with: a) plated chords (rather than isolated notes), and b) real piano performances from the dataset of [224]. Thus, it will be possible to train automatic annotation tasks, such as automatic transcription, using the newly augmented and synthesized dataset. This approach will then be extended to other instruments.





References

- [1] E. Apostolidis, E. Adamantidou, A. I. Metsai, V. Mezaris, and I. Patras, “Unsupervised video summarization via attention-driven adversarial learning,” in *Proceedings of the International Conference on Multimedia Modeling (MMM)*, Springer, 2020.
- [2] J. Huang, S. Gong, and X. Zhu, “Deep semantic clustering by partition confidence maximisation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8849–8858, 2020.
- [3] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool, “Domain Adaptive Faster R-CNN for Object Detection in the Wild,” in *Proceedings of CVPR*, pp. 3339–3348, 2018.
- [4] M. Khodabandeh, A. Vahdat, M. Ranjbar, and W. G. Macready, “A Robust Learning Approach to Domain Adaptive Object Detection,” in *Proceedings of ICCV*, pp. 480–490, 2019.
- [5] K. Saito, Y. Ushiku, T. Harada, and K. Saenko, “Strong-Weak Distribution Alignment for Adaptive Object Detection,” in *Proceedings of CVPR*, pp. 6956–6965, 2019.
- [6] Y. Shan, W. F. Lu, and C. M. Chew, “Pixel and feature level based domain adaptation for object detection in autonomous driving,” *Neurocomputing*, vol. 367, pp. 31–38, 2019.
- [7] X. Zhu, J. Pang, C. Yang, J. Shi, and D. Lin, “Adapting object detectors via selective cross-domain alignment,” in *Proceedings of CVPR*, pp. 687–696, 2019.
- [8] 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*, pp. 1–1, 2020.
- [9] Y. Ganin and V. Lempitsky, “Unsupervised domain adaptation by backpropagation,” vol. 37 of *Proceedings of Machine Learning Research*, (Lille, France), pp. 1180–1189, PMLR, 07–09 Jul 2015.
- [10] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *Proceedings of ICCV*, pp. 2223–2232, 2017.
- [11] J. Lee, N. J. Bryan, J. Salamon, and Z. Jin, “Disentangled multidimensional metric learning for music similarity,” in *In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6–10, 2020.
- [12] J. Lee, N. J. Bryan, J. Salamon, Z. Jin, and J. Nam, “Metric learning vs classification for disentangled music representation learning,” *arXiv preprint arXiv:2008.03729*, 2020.
- [13] A. Kulesza and B. Taskar, *Determinantal Point Processes for Machine Learning*. Hanover, MA, USA: Now Publishers Inc., 2012.
- [14] 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.
- [15] M. Rochan and Y. Wang, “Video Summarization by Learning From Unpaired Data,” in *2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 7894–7903, June 2019.



- [16] K. Zhou and Y. Qiao, “Deep Reinforcement Learning for Unsupervised Video Summarization with Diversity-Representativeness Reward,” in *Proc. of the 2018 AAAI Conf. on Artificial Intelligence*, 2018.
- [17] G. Yaliniz and N. Ikizler-Cinbis, “Using independently recurrent networks for reinforcement learning based unsupervised video summarization,” *Multimedia Tools and Applications*, vol. 80, no. 12, pp. 17827–17847, 2021.
- [18] 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 Trans. on Circuits and Systems for Video Technology*, pp. 1–1, 2020.
- [19] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor,” in *Proc. of the 35th Int. Conf. on Machine Learning (ICML)*, 2018.
- [20] A. Sharghi, B. Gong, and M. Shah, “Query-Focused Extractive Video Summarization,” in *ECCV*, 2016.
- [21] A. B. Vasudevan, M. Gygli, A. Volokitin, and L. Van Gool, “Query-adaptive Video Summarization via Quality-aware Relevance Estimation,” in *Proc. of the 2017 ACM on Multimedia Conf. (MM '17)*, (New York, NY, USA), pp. 582–590, ACM, 2017.
- [22] Y. Zhang, M. C. Kampffmeyer, X. Liang, M. Tan, and E. Xing, “Query-Conditioned Three-Player Adversarial Network for Video Summarization,” in *British Machine Vision Conference 2018, BMVC 2018, Northumbria University, Newcastle, UK, September 3-6, 2018*, p. 288, 2018.
- [23] Y. Zhang, M. Kampffmeyer, X. Zhao, and M. Tan, “Deep Reinforcement Learning for Query-Conditioned Video Summarization,” *Applied Sciences*, vol. 9, no. 4, 2019.
- [24] J.-H. Huang and M. Worring, “Query-controllable video summarization,” in *Proc. of the 2020 Int. Conf. on Multimedia Retrieval, ICMR '20*, (New York, NY, USA), p. 242–250, Association for Computing Machinery, 2020.
- [25] A. G. del Molino, X. Boix, J. Lim, and A. Tan, “Active Video Summarization: Customized Summaries via On-line Interaction,” in *Proc. of the 2017 AAAI Conf. on Artificial Intelligence*, AAAI Press, 2017.
- [26] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, “Graph Signal Processing: Overview, Challenges, and Applications,” *Proc. of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.
- [27] Y. Tanaka, Y. C. Eldar, A. Ortega, and G. Cheung, “Sampling Signals on Graphs: From Theory to Applications,” *IEEE Signal Processing Magazine*, vol. 37, no. 6, pp. 14–30, 2020.
- [28] G. Cheung, E. Magli, Y. Tanaka, and M. K. Ng, “Graph Spectral Image Processing,” *Proc. of the IEEE*, vol. 106, no. 5, pp. 907–930, 2018.
- [29] J. H. Giraldo, S. Javed, and T. Bouwmans, “Graph Moving Object Segmentation,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.



- [30] T. Doan, J. Monteiro, I. Albuquerque, B. Mazoure, A. Durand, J. Pineau, and D. Hjelm, “On-line Adaptative Curriculum Learning for GANs,” in *Proc. of the 2019 AAAI Conf. on Artificial Intelligence*, March 2019.
- [31] K. Ghasedi, X. Wang, C. Deng, and H. Huang, “Balanced Self-Paced Learning for Generative Adversarial Clustering Network,” in *2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 4386–4395, 2019.
- [32] P. Soviany, C. Ardei, R. T. Ionescu, and M. Leordeanu, “Image Difficulty Curriculum for Generative Adversarial Networks (CuGAN),” in *2020 IEEE Winter Conf. on Applications of Computer Vision (WACV)*, pp. 3452–3461, 2020.
- [33] J. Fajtl, H. S. Sokeh, V. Argyriou, D. Monekosso, and P. Remagnino, “Summarizing Videos with Attention,” in *Asian Conf. on Computer Vision (ACCV) 2018 Workshops* (G. Carneiro and S. You, eds.), (Cham), pp. 39–54, Springer International Publishing, 2019.
- [34] B. Zhao, X. Li, and X. Lu, “TTH-RNN: Tensor-Train Hierarchical Recurrent Neural Network for Video Summarization,” *IEEE Trans. on Industrial Electronics*, vol. 68, no. 4, pp. 3629–3637, 2020.
- [35] S. Li, W. Li, C. Cook, C. Zhu, and Y. Gao, “Independently recurrent neural network (indrnn): Building a longer and deeper rnn,” *2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 5457–5466, 2018.
- [36] L. Feng, Z. Li, Z. Kuang, and W. Zhang, “Extractive Video Summarizer with Memory Augmented Neural Networks,” in *Proc. of the 26th ACM Int. Conf. on Multimedia (MM ’18)*, (New York, NY, USA), pp. 976–983, ACM, 2018.
- [37] J. Wang, W. Wang, Z. Wang, L. Wang, D. Feng, and T. Tan, “Stacked Memory Network for Video Summarization,” in *Proc. of the 27th ACM Int. Conf. on Multimedia (MM ’19)*, (New York, NY, USA), p. 836–844, ACM, 2019.
- [38] C. Huang and H. Wang, “A Novel Key-Frames Selection Framework for Comprehensive Video Summarization,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 30, no. 2, pp. 577–589, 2020.
- [39] Y. Jung, D. Cho, S. Woo, and I. S. Kweon, “Global-and-local relative position embedding for unsupervised video summarization,” in *Europ. Conf. on Computer Vision (ECCV) 2020* (A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, eds.), (Cham), pp. 167–183, Springer International Publishing, 2020.
- [40] B. Zhao, M. Gong, and X. Li, “Audiovisual video summarization,” *ArXiv*, vol. abs/2105.07667, 2021.
- [41] M. Otani, Y. Nakahima, E. Rahtu, and J. Heikkilä, “Rethinking the Evaluation of Video Summaries,” in *2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [42] 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.

- [43] 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.
- [44] M. Rochan, L. Ye, and Y. Wang, “Video summarization using Fully Convolutional Sequence Networks,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.
- [45] 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.
- [46] N. Gonuguntla, B. Mandal, and N. Puhan, “Enhanced deep video summarization network,” in *Proceedings of the British Machine Vision Conference (BMVC)*, 2019.
- [47] 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.
- [48] 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.
- [49] 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.
- [50] L. Yuan, F. Tay, P. Li, L. Zhou, and J. Feng, “Cycle-SUM: Cycle-consistent adversarial LSTM networks for unsupervised video Summarization,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.
- [51] 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.
- [52] 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.
- [53] 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.
- [54] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [55] 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.
- [56] B. Horn and B. Schunck, “Determining optical flow,” *Artificial Intelligence*, vol. 17, pp. 185–203, 08 1981.





- [57] B. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision (ijcai),” vol. 81, 04 1981.
- [58] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, “FlowNet: Learning optical flow with convolutional networks,” 04 2015.
- [59] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, “FlowNet 2.0: Evolution of optical flow estimation with deep networks,” pp. 1647–1655, 07 2017.
- [60] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, “Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume,” pp. 8934–8943, 06 2018.
- [61] Z. Teed and J. Deng, “Raft: Recurrent all-pairs field transforms for optical flow,” 2020.
- [62] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, pp. 5998–6008, 2017.
- [63] S. Jiang, D. Campbell, Y. Lu, H. Li, and R. Hartley, “Learning to estimate hidden motions with global motion aggregation,” 2021.
- [64] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017.
- [65] X. Wang, R. Zhang, T. Kong, L. Li, and C. Shen, “Solov2: Dynamic and fast instance segmentation,” in *Advances in Neural Information Processing Systems* (H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, eds.), vol. 33, pp. 17721–17732, Curran Associates, Inc., 2020.
- [66] X. Wang, T. Kong, C. Shen, Y. Jiang, and L. Li, “Solo: Segmenting objects by locations,” in *Computer Vision – ECCV 2020* (A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, eds.), (Cham), pp. 649–665, Springer International Publishing, 2020.
- [67] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, “Yolact: Real-time instance segmentation,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9156–9165, 2019.
- [68] 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 *Computer Vision – ECCV 2014* (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), (Cham), pp. 740–755, Springer International Publishing, 2014.
- [69] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, “Yolact++: Better real-time instance segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.
- [70] J. Cao, R. M. Anwer, H. Cholakkal, F. S. Khan, Y. Pang, and L. Shao, “Sipmask: Spatial information preservation for fast image and video instance segmentation,” *Proc. European Conference on Computer Vision*, 2020.
- [71] R. Caruana, *Multitask Learning*, pp. 95–133. Boston, MA: Springer US, 1998.
- [72] J. Hur and S. Roth, “Joint optical flow and temporally consistent semantic segmentation,” *ArXiv*, vol. abs/1607.07716, 2016.



- [73] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang, “Segflow: Joint learning for video object segmentation and optical flow,” in *IEEE International Conference on Computer Vision (ICCV)*, 2017.
- [74] M. Ding, Z. Wang, B. Zhou, J. Shi, Z. Lu, and P. Luo, “Every frame counts: Joint learning of video segmentation and optical flow,” in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 10713–10720, AAAI Press, 2020.
- [75] H. Jiang, D. Sun, V. Jampani, Z. Lv, E. Learned-Miller, and J. Kautz, “Sense: A shared encoder network for scene-flow estimation,” in *ICCV*, 2019.
- [76] R. Harb and P. Knöbelreiter, “Efficient multi-task learning of semantic segmentation and disparity estimation,” p. 147, May 2019.
- [77] J. Yao, M. Boben, S. Fidler, and R. Urtasun, “Real-time coarse-to-fine topologically preserving segmentation,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2947–2955, 2015.
- [78] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (Los Alamitos, CA, USA), pp. 3431–3440, IEEE Computer Society, jun 2015.
- [79] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, “A naturalistic open source movie for optical flow evaluation,” in *European Conf. on Computer Vision (ECCV)* (A. Fitzgibbon et al. (Eds.), ed.), Part IV, LNCS 7577, pp. 611–625, Springer-Verlag, Oct. 2012.
- [80] M. Menze, C. Heipke, and A. Geiger, “Joint 3d estimation of vehicles and scene flow,” in *ISPRS Workshop on Image Sequence Analysis (ISA)*, 2015.
- [81] E. Ilg, T. Saikia, M. Keuper, and T. Brox, “Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation,” in *European Conference on Computer Vision (ECCV)*, 2018.
- [82] A. Messina, “Dataset production as a new process in future ai-empowered media,” in *IBC 2020 Conference*, 2020.
- [83] M. Everingham, J. Sivic, and A. Zisserman, “Hello! my name is... buffy” – automatic naming of characters in tv video,” pp. 899–908, 01 2006.
- [84] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, “Labeled faces in the wild: A database for studying face recognition in unconstrained environments,” Tech. Rep. 07-49, University of Massachusetts, Amherst, October 2007.
- [85] L. Wolf, T. Hassner, and I. Maoz, “Face recognition in unconstrained videos with matched background similarity,” in *CVPR 2011*, pp. 529–534, 2011.
- [86] G. B. Huang and E. Learned-Miller, “Labeled faces in the wild: Updates and new reporting procedures,” Tech. Rep. UM-CS-2014-003, University of Massachusetts, Amherst, May 2014.



- [87] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, “Deepface: Closing the gap to human-level performance in face verification,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1701–1708, 2014.
- [88] D. Yi, Z. Lei, S. Liao, and S. Z. Li, “Learning face representation from scratch,” 2014.
- [89] Z. Liu, P. Luo, X. Wang, and X. Tang, “Deep learning face attributes in the wild,” in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.
- [90] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 815–823, 2015.
- [91] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, “Ms-celeb-1m: A dataset and benchmark for large-scale face recognition,” 2016.
- [92] S. Yang, P. Luo, C. C. Loy, and X. Tang, “Wider face: A face detection benchmark,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [93] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, “Vggface2: A dataset for recognising faces across pose and age,” 2018.
- [94] Y. Zhang, W. Deng, M. Wang, J. Hu, X. Li, D. Zhao, and D. Wen, “Global-local gcn: Large-scale label noise cleansing for face recognition,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7728–7737, 2020.
- [95] M. Wang and W. Deng, “Deep face recognition: A survey,” *Neurocomputing*, vol. 429, pp. 215–244, 2021.
- [96] M. Wang, W. Deng, J. Hu, X. Tao, and Y. Huang, “Racial faces in the wild: Reducing racial bias by information maximization adaptation network,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, (Los Alamitos, CA, USA), pp. 692–702, IEEE Computer Society, nov 2019.
- [97] F. Wang, L. Chen, C. Li, S. Huang, Y. Chen, C. Qian, and C. C. Loy, “The devil of face recognition is in the noise,” 2018.
- [98] J. Deng, J. Guo, E. Ververas, I. Kotsia, and S. Zafeiriou, “Retinaface: Single-shot multi-level face localisation in the wild,” in *CVPR*, 2020.
- [99] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, “Arcface: Additive angular margin loss for deep face recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4690–4699, 2019.
- [100] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD’96*, p. 226–231, AAAI Press, 1996.
- [101] L. McInnes, J. Healy, and S. Astels, “hdbscan: Hierarchical density based clustering,” *The Journal of Open Source Software*, vol. 2, no. 11, p. 205, 2017.



- [102] C. Biemann, “Chinese whispers: An efficient graph clustering algorithm and its application to natural language processing problems,” in *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing, TextGraphs-1*, (USA), p. 73–80, Association for Computational Linguistics, 2006.
- [103] Y. A. Malkov and D. A. Yashunin, “Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 4, pp. 824–836, 2020.
- [104] D. Vrandečić and M. Krötzsch, “Wikidata: A free collaborative knowledgebase,” *Commun. ACM*, vol. 57, p. 78–85, Sept. 2014.
- [105] P. J. Phillips, P. Grother, and R. Micheals, *Evaluation Methods in Face Recognition*, pp. 551–574. London: Springer London, 2011.
- [106] L.-Y. Duan, J. Lin, J. Chen, T. Huang, and W. Gao, “Compact descriptors for visual search,” *IEEE MultiMedia*, vol. 21, no. 3, pp. 30–40, 2014.
- [107] A. Bobasheva, F. Gandon, and F. Precioso, “Learning and reasoning for cultural metadata quality,” *Submitted for publication in ACM Journal on Computing and Cultural Heritage*, 2021.
- [108] D. Wood, R. Cyganiak, and M. Lanthaler, “RDF 1.1 concepts and abstract syntax,” W3C recommendation, W3C, Feb. 2014. <https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/>.
- [109] A. Seaborne and S. Harris, “SPARQL 1.1 query language,” W3C recommendation, W3C, Mar. 2013. <https://www.w3.org/TR/2013/REC-sparql11-query-20130321/>.
- [110] A. Miles and S. Bechhofer, “SKOS simple knowledge organization system reference,” W3C recommendation, W3C, Aug. 2009. <https://www.w3.org/TR/2009/REC-skos-reference-20090818/>.
- [111] M. Després-Lonnet, “L’écriture numérique du patrimoine, de l’inventaire à l’exposition: Les parcours de la base Joconde,” *Culture & Musées*, vol. 14, no. 1, pp. 19–38, 2009.
- [112] 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.
- [113] 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.
- [114] B. Singh, H. Li, A. Sharma, and L. S. Davis, “R-FCN-3000 at 30fps: Decoupling detection and classification,” in *Proc. CVPR*, 2018.
- [115] B. Kang, Z. Liu, X. Wang, F. Yu, J. Feng, and T. Darrell, “Few-shot object detection via feature reweighting,” in *Proc. ICCV*, 2019.
- [116] 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.

- [117] D. A. Ganea, B. Boom, and R. Poppe, “Incremental few-shot instance segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1185–1194, 2021.
- [118] V. Lempitsky and A. Zisserman, “Learning to count objects in images,” in *Advances in Neural Information Processing Systems 23* (J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, eds.), pp. 1324–1332, Curran Associates, Inc., 2010.
- [119] Y. Li, X. Zhang, and D. Chen, “Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1091–1100, 2018.
- [120] S. Zhang, G. Wu, J. P. Costeira, and J. M. Moura, “Understanding traffic density from large-scale web camera data,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5898–5907, 2017.
- [121] R. Guerrero-Gómez-Olmedo, B. Torre-Jiménez, R. López-Sastre, S. Maldonado-Bascón, and D. Oñoro-Rubio, “Extremely overlapping vehicle counting,” in *Pattern Recognition and Image Analysis* (R. Paredes, J. S. Cardoso, and X. M. Pardo, eds.), (Cham), pp. 423–431, Springer International Publishing, 2015.
- [122] D. Oñoro-Rubio and R. J. López-Sastre, “Towards perspective-free object counting with deep learning,” in *Computer Vision – ECCV 2016* (B. Leibe, J. Matas, N. Sebe, and M. Welling, eds.), (Cham), pp. 615–629, Springer International Publishing, 2016.
- [123] I. H. Laradji, N. Rostamzadeh, P. O. Pinheiro, D. Vazquez, and M. Schmidt, “Where are the blobs: Counting by localization with point supervision,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 547–562, 2018.
- [124] G. Amato, P. Bolettieri, F. Carrara, F. Debole, F. Falchi, C. Gennaro, L. Vadicamo, and C. Vairo, “Visione at vbs2019,” in *International Conference on Multimedia Modeling*, pp. 591–596, Springer, 2019.
- [125] G. Amato, P. Bolettieri, F. Falchi, C. Gennaro, N. Messina, L. Vadicamo, and C. Vairo, “Visione at video browser showdown 2021,” in *International Conference on Multimedia Modeling*, pp. 473–478, Springer, 2021.
- [126] Solr™, “Apache lucene™ project.” <https://lucene.apache.org/>. [Online; accessed 29-July-2021].
- [127] L. Rossetto, R. Gasser, J. Lokoc, W. Bailer, K. Schoeffmann, B. Muenzer, T. Soucek, P. A. Nguyen, P. Bolettieri, A. Leibetseder, and S. Vrochidis, “Interactive video retrieval in the age of deep learning - detailed evaluation of vbs 2019,” *IEEE Transactions on Multimedia*, pp. 1–1, 2020.
- [128] 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, 2021.
- [129] J. Redmon and A. Farhadi, “YOLOv3 on the Open Images dataset.” <https://pjreddie.com/darknet/yolo/>, 2018. [Online; accessed 22-April-2021].
- [130] G. Tolias, R. Sivic, and H. Jégou, “Particular object retrieval with integral max-pooling of CNN activations,” *CoRR*, vol. abs/1511.05879, 2015.



- [131] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford, “Okapi at TREC-3,” in *Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994*, vol. 500-225 of *NIST Special Publication*, pp. 109–126, National Institute of Standards and Technology (NIST), 1994.
- [132] K. Sparck Jones, “A statistical interpretation of term specificity and its application in retrieval,” *Journal of documentation*, vol. 28, no. 1, pp. 11–21, 1972.
- [133] M. D. Smucker, J. Allan, and B. Carterette, “A comparison of statistical significance tests for information retrieval evaluation,” in *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, CIKM '07, (New York, NY, USA)*, p. 623–632, Association for Computing Machinery, 2007.
- [134] N. Messina, F. Falchi, A. Esuli, and G. Amato, “Transformer reasoning network for image-text matching and retrieval,” in *International Conference on Pattern Recognition (ICPR) 2020 (Accepted)*, 2020.
- [135] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, pp. 5998–6008, 2017.
- [136] 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.
- [137] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.
- [138] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *ICLR*, 2015.
- [139] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, Oct 2017.
- [140] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.
- [141] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in *European conference on computer vision*, pp. 21–37, Springer, 2016.
- [142] 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.
- [143] H. Xu, X. Wang, F. Shao, B. Duan, and P. Zhang, “Few-shot object detection via sample processing,” *IEEE Access*, vol. 9, pp. 29207–29221, 2021.
- [144] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes challenge: A retrospective,” *International Journal of Computer Vision*, vol. 111, pp. 98–136, Jan. 2015.



- [145] 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.
- [146] J. E. Van Engelen and H. H. Hoos, “A survey on semi-supervised learning,” *Machine Learning*, vol. 109, no. 2, pp. 373–440, 2020.
- [147] O. T. Nartey, G. Yang, J. Wu, and S. K. Asare, “Semi-supervised learning for fine-grained classification with self-training,” *IEEE Access*, vol. 8, pp. 2109–2121, 2019.
- [148] J.-C. Su and S. Maji, “The semi-supervised iNaturalist-Aves challenge at FGVC7 Workshop,” *arXiv preprint arXiv:2103.06937*, 2021.
- [149] A. Tarvainen and H. Valpola, “Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results,” in *Advances in neural information processing systems*, pp. 1195–1204, 2017.
- [150] Y. Grandvalet and Y. Bengio, “Semi-supervised learning by entropy minimization,” in *Advances in neural information processing systems*, pp. 529–536, 2005.
- [151] D.-H. Lee, “Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks,” in *ICML 2013 Workshop : Challenges in Representation Learning (WREPL), Atlanta, Georgia, USA*, 2013.
- [152] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer, “S4l: Self-supervised semi-supervised learning,” in *Proceedings of the IEEE international conference on computer vision*, pp. 1476–1485, 2019.
- [153] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, “Mix-match: A holistic approach to semi-supervised learning,” in *Advances in Neural Information Processing Systems*, pp. 5049–5059, 2019.
- [154] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel, “Fixmatch: Simplifying semi-supervised learning with consistency and confidence,” *arXiv preprint arXiv:2001.07685*, 2020.
- [155] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *NIPS*, 2012.
- [156] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, “The caltech-ucsd birds-200-2011 dataset,” 2011.
- [157] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei, “Novel dataset for fine-grained image categorization,” in *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, (Colorado Springs, CO), June 2011.
- [158] M.-E. Nilsback and A. Zisserman, “Automated flower classification over a large number of classes,” in *Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.
- [159] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, “3d object representations for fine-grained categorization,” in *Proceedings of the IEEE international conference on computer vision workshops*, pp. 554–561, 2013.



- [160] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi, “Fine-grained visual classification of aircraft,” *arXiv preprint arXiv:1306.5151*, 2013.
- [161] X.-S. Wei, J. Wu, and Q. Cui, “Deep learning for fine-grained image analysis: A survey,” *arXiv preprint arXiv:1907.03069*, 2019.
- [162] J. Deng, J. Krause, and L. Fei-Fei, “Fine-grained crowdsourcing for fine-grained recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2013.
- [163] D. Korsch, P. Bodesheim, and J. Denzler, “End-to-end learning of a fisher vector encoding for part features in fine-grained recognition,” *arXiv preprint arXiv:2007.02080*, 2020.
- [164] F. Zhang, G. Zhai, M. Li, and Y. Liu, “Three-branch and mutil-scale learning for fine-grained image recognition (tbmsl-net),” *arXiv preprint arXiv:2003.09150*, 2020.
- [165] J. Ngiam, D. Peng, V. Vasudevan, S. Kornblith, Q. V. Le, and R. Pang, “Domain adaptive transfer learning with specialist models,” *arXiv preprint arXiv:1811.07056*, 2018.
- [166] 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,” 2020.
- [167] T. Berg, J. Liu, S. Woo Lee, M. L. Alexander, D. W. Jacobs, and P. N. Belhumeur, “Birdsnap: Large-scale fine-grained visual categorization of birds,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2011–2018, 2014.
- [168] H. Zheng, J. Fu, T. Mei, and J. Luo, “Learning multi-attention convolutional neural network for fine-grained image recognition,” in *Proceedings of the IEEE international conference on computer vision*, pp. 5209–5217, 2017.
- [169] W. Ge, X. Lin, and Y. Yu, “Weakly supervised complementary parts models for fine-grained image classification from the bottom up,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3034–3043, 2019.
- [170] D. Korsch, P. Bodesheim, and J. Denzler, “Classification-specific parts for improving fine-grained visual categorization,” in *German Conference on Pattern Recognition*, pp. 62–75, Springer, 2019.
- [171] L. Zhang, S. Huang, W. Liu, and D. Tao, “Learning a mixture of granularity-specific experts for fine-grained categorization,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 8331–8340, 2019.
- [172] J. Zhang, R. Zhang, Y. Huang, and Q. Zou, “Unsupervised part mining for fine-grained image classification,” *arXiv preprint arXiv:1902.09941*, 2019.
- [173] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, “Domain-adversarial training of neural networks,” *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.
- [174] Y. Wang and S. Chen, “Safety-aware semi-supervised classification,” *IEEE transactions on neural networks and learning systems*, vol. 24, no. 11, pp. 1763–1772, 2013.
- [175] O. Chapelle, B. Schölkopf, and A. Zien, “Semi-supervised learning,” 2010.



- [176] W. Liu, Y. Wen, Z. Yu, and M. Yang, “Large-margin softmax loss for convolutional neural networks,” in *Proceedings of the 33rd International Conference on International Conference on Machine Learning-Volume 48*, pp. 507–516, 2016.
- [177] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, “Sphereface: Deep hypersphere embedding for face recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 212–220, 2017.
- [178] P. Li, J. Xie, Q. Wang, and Z. Gao, “Towards faster training of global covariance pooling networks by iterative matrix square root normalization,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 947–955, 2018.
- [179] J.-C. Su, Z. Cheng, and S. Maji, “A realistic evaluation of semi-supervised learning for fine-grained classification,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12966–12975, 2021.
- [180] P. Cascante-Bonilla, F. Tan, Y. Qi, and V. Ordonez, “Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 6912–6920, May 2021.
- [181] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.
- [182] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. E. Hinton, “Big self-supervised models are strong semi-supervised learners,” in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.
- [183] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” *arXiv preprint arXiv:1503.02531*, 2015.
- [184] X. Ji, J. F. Henriques, and A. Vedaldi, “Invariant information clustering for unsupervised image classification and segmentation,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9865–9874, 2019.
- [185] A. Krizhevsky *et al.*, “Learning multiple layers of features from tiny images,” 2009.
- [186] A. Coates, A. Ng, and H. Lee, “An analysis of single-layer networks in unsupervised feature learning,” in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223, JMLR Workshop and Conference Proceedings, 2011.
- [187] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *JRSS*, vol. 58, no. 1, pp. 267–288, 1996.
- [188] D. L. Donoho, “Compressed sensing,” *IEEE TIT*, vol. 52, no. 4, pp. 1289–1306, 2006.
- [189] K. Engan, S. O. Aase, and J. Hakon Husoy, “Method of optimal directions for frame design,” in *ICASSP*, 1999.
- [190] M. Aharon, M. Elad, and A. Bruckstein, “K-svd: An algorithm for designing overcomplete dictionaries for sparse representation,” *IEEE TSP*, vol. 54, no. 11, pp. 4311–4322, 2006.



- [191] H. Liu, H. Tang, W. Xiao, Z. Guo, L. Tian, and Y. Gao, “Sequential bag-of-words model for human action classification,” *CAAI TIT*, vol. 1, no. 2, pp. 125–136, 2016.
- [192] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, “Robust face recognition via sparse representation,” *IEEE TPAMI*, vol. 31, no. 2, pp. 210–227, 2009.
- [193] H. Tang and H. Liu, “A novel feature matching strategy for large scale image retrieval.,” in *IJCAI*, 2016.
- [194] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, “Online dictionary learning for sparse coding,” in *ICML*, 2009.
- [195] Y. Lin, Z. Tong, S. Zhu, and K. Yu, “Deep coding network,” in *NeurIPS*, 2010.
- [196] J. C. van Gemert, J.-M. Geusebroek, C. J. Veenman, and A. W. Smeulders, “Kernel codebooks for scene categorization,” in *ECCV*, 2008.
- [197] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, “Locality-constrained linear coding for image classification,” in *CVPR*, 2010.
- [198] Q. Zhang and B. Li, “Discriminative k-svd for dictionary learning in face recognition,” in *CVPR*, 2010.
- [199] Z. Jiang, Z. Lin, and L. S. Davis, “Learning a discriminative dictionary for sparse coding via label consistent k-svd,” in *CVPR*, 2011.
- [200] G. Griffin, A. Holub, and P. Perona, “Caltech-256 object category dataset,” *California Institute of Technology*, 2007.
- [201] J. Yang, K. Yu, Y. Gong, and T. Huang, “Linear spatial pyramid matching using sparse coding for image classification,” in *CVPR*, 2009.
- [202] J. Hu and Y.-P. Tan, “Nonlinear dictionary learning with application to image classification,” *PR*, vol. 75, pp. 282–291, 2018.
- [203] S. R. Fanello, N. Noceti, C. Ciliberto, G. Metta, and F. Odone, “Ask the image: supervised pooling to preserve feature locality,” in *CVPR*, 2014.
- [204] P. Gehler and S. Nowozin, “On feature combination for multiclass object classification,” in *ICCV*, 2009.
- [205] L. Bo, X. Ren, and D. Fox, “Multipath sparse coding using hierarchical matching pursuit,” in *CVPR*, 2013.
- [206] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *ECCV*, 2014.
- [207] M. Simon and E. Rodner, “Neural activation constellations: Unsupervised part model discovery with convolutional networks,” in *ICCV*, 2015.
- [208] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” *IEEE TPAMI*, vol. 37, no. 9, pp. 1904–1916, 2015.
- [209] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, “From few to many: Illumination cone models for face recognition under variable lighting and pose,” *IEEE TPAMI*, vol. 23, no. 6, pp. 643–660, 2001.





- [210] A. M. Martinez, “The ar face database,” *CVC TR*, vol. 24, 1998.
- [211] L. Fei-Fei, R. Fergus, and P. Perona, “Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories,” *CVIU*, vol. 106, no. 1, pp. 59–70, 2007.
- [212] G. Griffin, A. Holub, and P. Perona, “Caltech-256 object category dataset,” *CIT Technical Report*, 2007.
- [213] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, pp. 2278–2324, Nov 1998.
- [214] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” in *Proceedings of ICML*, pp. 41–48, 2009.
- [215] R. Ionescu, B. Alexe, M. Leordeanu, M. Popescu, D. P. Papadopoulos, and V. Ferrari, “How hard can it be? estimating the difficulty of visual search in an image,” in *Proceedings of CVPR*, pp. 2157–2166, 2016.
- [216] A. Veit, S. Belongie, and T. Karalestos, “Conditional similarity networks,” in *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1781–1789, 2017.
- [217] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer, “Madmom: A new python audio and music signal processing library,” in *Proceedings of the 2016 ACM Multimedia Conference*, pp. 1174–1178, Association for Computing Machinery, Inc, oct 2016.
- [218] F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A unified embedding for face recognition and clustering,” in *In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 815–823, 2015.
- [219] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going Deeper with Convolutions,” in *In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 1–9, 2015.
- [220] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, “The million song dataset,” 2011.
- [221] J. Lee, N. J. Bryan, J. Salamon, Z. Jin, and J. Nam, “Disentangled multidimensional metric learning for music similarity,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020)*, pp. 6–10, IEEE.
- [222] J. Cramer, H.-h. Wu, J. Salamon, and J. P. Bello, “Look, Listen, and Learn More: Design Choices for Deep Audio Embeddings,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (Brighton, UK), pp. 3852–3856, 2019.
- [223] V. Emiya, R. Badeau, and B. David, “Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1643–1654, 2010.
- [224] 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,” *arXiv preprint arXiv:1810.12247*, 2018.



- [225] R. Mignot and G. Peeters, “An analysis of the effect of data augmentation methods: experiments for a musical genre classification task,” *Transactions of the International Society for Music Information Retrieval*, vol. 2, no. 1, 2019.
- [226] A. Cohen-Hadria, A. Roebel, and G. Peeters, “Improving singing voice separation using deep u-net and wave-u-net with data augmentation,” in *2019 27th European Signal Processing Conference (EUSIPCO)*, pp. 1–5, 2019.
- [227] L. Ardaillon and A. Roebel, “Fully-Convolutional Network for Pitch Estimation of Speech Signals,” in *Proc. Interspeech 2019*, pp. 2005–2009, 2019.
- [228] L. Ardaillon and A. Roebel, “Gci detection from raw speech using a fully-convolutional network,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6739–6743, 2020.
- [229] 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.
- [230] J. Engel, L. H. Hantrakul, C. Gu, and A. Roberts, “Ddsp: Differentiable digital signal processing,” in *International Conference on Learning Representations*, July 2020.
- [231] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” *Advances in neural information processing systems*, vol. 27, December 2014.
- [232] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2223–2232, October 2017.
- [233] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, “Learning from simulated and unsupervised images through adversarial training,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 2107–2116, June 2017.
- [234] 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.
- [235] G. Weinreich, “Coupled piano strings,” *The Journal of the Acoustical Society of America*, vol. 62, no. 6, pp. 1474–1484, 1977.
- [236] E. Cooper, X. Wang, and J. Yamagishi, “Text-to-speech synthesis techniques for midi-to-audio synthesis,” *arXiv preprint arXiv:2104.12292*, 2021.
- [237] M. Nucci, M. Tagliasacchi, and S. Tubaro, “A phylogenetic analysis of near-duplicate audio tracks.,” in *MMSP*, 2013.
- [238] M. Maksimovic, L. Cuccovillo, and P. Aichroth, “Phylogeny analysis for MP3 and AAC coding transformations,” in *ICME*, 2017.
- [239] S. Verde, S. Milani, P. Bestagini, and S. Tubaro, “Audio phylogenetic analysis using geometric transforms,” in *WIFS*, 2017.
- [240] M. Maksimovic, P. Aichroth, and L. Cuccovillo, “Detection and localization of partial audio matches,” in *CBMI*, 2018.





- [241] M. Maksimovic, P. Aichroth, and L. Cuccovillo, “Detection and localization of partial audio matches in various application scenarios,” *Multimedia Tools and Applications*, 2021.
- [242] H. Fassold and R. Ghermi, “Omnitrack: Real-time detection and tracking of objects, text and logos in video,” in *2019 IEEE International Symposium on Multimedia (ISM)*, pp. 245–2451, IEEE, 2019.