D4.2

Prototype platform for AI dataset benchmarking

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	Prototype platform for AI dataset benchmarking
Deliverable number	D4.2
Deliverable version	1.0
Previous version(s)	-
Contractual date of delivery	28 February 2022
Actual date of delivery	28 February 2022
Deliverable filename	AI4Media_D4.2.pdf
Nature of deliverable	Demonstrator
Dissemination level	Public
Number of pages	29
Work Package	WP4
Task(s)	T4.6
Parner responsible	UPB
Author(s)	Mihai Gabriel Constantin (UPB), Liviu-Daniel Ștefan(UPB), Mihai Dogariu (UPB), Bogdan Ionescu (UPB), Henning Müller (HES-SO)
Editor	Mihai Gabriel Constantin (UPB)
Officer	Evangelia Markidou

Abstract	This deliverable presents the initial results achieved for task T4.6: Benchmarking of AI Systems, which is part of WP4: Explainability, Robustness and Privacy in AI. The document describes the prototype platform for AI dataset benchmarking, showing the current state-of-the-art for Evaluation-as-a-Service platforms, the work that has been done so far in developing our prototype, and listing the high-level functions that will be available in the final version of the platform.
Keywords	AI benchmarking, Evaluation-as-a-Service, datasets, benchmarking systems, Algorithm-to-Data, EaaS

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

Contributors

NAME	ORGANIZATION
Mihai Gabriel Constantin	UPB
Liviu-Daniel Ștefan	UPB
Mihai Dogariu	UPB
Bogdan Ionescu	UPB
Henning Müller	HES-SO

Peer Reviews

NAME	ORGANIZATION
Ioannis Patras	QMUL
Mara Graziani	HES-SO

Revision History

Version	Date	Reviewer	Modifications
0.1	15/02/2022	Mihai Gabriel Constantin (UPB)	Prefinal version, ready for internal review
0.2	25/02/2022	Mihai Gabriel Constantin (UPB)	Updated version, with review from Ioannis Patras and Maria Graziani
1.0	28/02/2022	Mihai Gabriel Constantin (UPB)	Final version

Table of Abbreviations and Acronyms

Abbreviation	Meaning
ABC	Abstract Base Class
AI	Artificial Intelligence
AMI	AWS virtual Machine Image
API	Application Programming Interface
AWS	Amazon Web Services
BSD	Berkeley Software Distribution
CLI	Command Line Interface
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CSV	Comma-Separated values
CV	Curriculum Vitae
DNN	Deep neural Networks
EaaS	Evaluation as a Service
GDPR	General Data Protection Regulation
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HTML	HyperText Markup Language
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
KPI	Key Performance Indicator
QA	Question Answering
VM	Virtual Machine
WP	Work Package
YAML	Yet Another Markup Language

Contents

List of Tables

List of Figures

1 Executive Summary

This document shows the current status of task T4.6: Benchmarking of AI Systems, a task included in WP4: Explainability, Robustness and Privacy in AI. The goal of the task is the creation of a novel Evaluation-as-a-Service AI benchmarking platform. In developing the AI4Media platform we follow a list of goals, that will be analyzed and presented throughout this document: (i) creating an European-based Evaluation-as-a-Service platform that would help AI benchmarking task organizers in creating and managing their competitions, (ii) encourage the development of reproducible and computationally efficient AI methods, through the high-level functions and options offered to the platform's users, (iii) provide approaches that give freedom to competition organizers with regards to how the competition is set up and to the integration of third party virtual machines, containers and code sharing concepts.

As we show throughout the document, the platform is now in a prototype phase, currently hosting a competition that helps us showcase some of the most important use cases for such an architecture. The source code for the AI4Media benchmarking platform is publicly available on a GitHub repository[1]. In the context of AI4Media, the creation of such a platform is of importance for other work packages as well, that deal with the creation of AI models for different tasks, like WP3, WP4, WP5, WP6, and can even be applied for directly testing the use case demonstrators in WP8.

The document presents a feature-based study of the current state-of-the-art of AI benchmarking platforms, that helps select an open-source codebase for starting the development of the AI4Media platform. Based on this study, we identify and propose new features for benchmarking platforms that would improve the current state-of-the-art and encourage AI benchmarking organizers to use the AI4Media platform.

Firstly, to the best of our knowledge, no current platforms propose a set of metrics for the measurement of the computational complexity of AI methods. As part of our goals, we will develop the necessary software environment as well as a series of metrics that can be easily integrated into any competition by the organizers. While we will also provide the organizers with freedom in developing their own complexity metrics, we believe that providing a starting set of metrics would ease the adoption of such metrics, as well as encourage the organizers to think about this novel dimension for measuring AI algorithm performance. The development and adoption of such metrics would allow interested parties to better understand the environmental footprint that AI algorithms have.

Secondly, while current benchmarking platforms may provide some options for reproducibility, such as API integration or the use of containers for submitting methods, few of them provide both options. We therefore propose to integrate both these options and furthermore improve the current state with regards to software development by fully utilizing the power software design patterns like Abstract and Concrete Factories, programming to Interfaces, loose coupling and Chain-of-Responsibility [1, 2]. These patterns would allow a better debugging and maintenance for the AI4Media platform, allow interested parties to change and adapt the resulting code much easier, as well as ease the integration of other options in the future.

Finally, it is necessary to underline the importance of having our own, EU-based AI benchmarking platform. This would not only allow us to concentrate on features considered top-priority by EU lawmakers and media agencies, such as explainable and trustworthy AI, but also give competition organizers more control over data storage, thus allowing better data privacy when needed.

While the current status of the platform is established through a series of use cases, we also analyze where this work is heading by formulating a development plan. Our plan consists of 13

[1] https://github.com/AIMultimediaLab/AI4Media-EaaS-prototype-Py2-public

high-level main attributes and user-centric functions that we plan to showcase in the final version of the platform, which is due in M40 (D4.6-"Final platform for AI dataset benchmarking"). This deliverable presents the work done in the first 18 months of the project. For the remainder of the project, the work will be continued with the focus on developing the high-level functions we present here, and integrating more competitions that target AI methods developed or created for other work packages in our platform.

2 Introduction

This deliverable presents the results achieved in task T4.6: Benchmarking of AI Systems, which is part of in WP4: Explainability, Robustness and Privacy in AI. As presented in the description of the work package, WP4 focuses on concepts like AI robustness, explainability, privacy and fairness as well as on legal and ethical frameworks associated with AI development. More to the point, T4.6 is described as having the following goals in the *Description of Action*:

> This task aims to achieve the following 5 goals: (i) provide annotated data to support the training of the proposed AI systems, (ii) build a community around benchmarking activities to stimulate the innovation and share of resources for better AI, (iii) encourage the development of computationally efficient and effective systems to reduce the power footprint via introducing dedicated metrics for complexity, (iv) foster reproducible systems via re-running the submitted systems in the evaluation phase, (v) building a common repository for sharing the data and to develop approaches for distributed benchmarking with container submission on possibly confidential data (sometimes called EaaS - Evaluation-as-a- Service).

Deliverable 4.2 targets the last three goals, by presenting the first steps into building a dedicated platform for AI dataset benchmarking tasks. This deliverable will be followed by D4.6 in M40 of the project, that will present the final version of the AI dataset benchmarking platform.

Current trends in the development of AI systems show an increased interest in developing benchmarking initiatives. These initiatives are conferences, events or special sessions at conferences, and development tracks that provide participants with a set of common conditions that ensure fair comparison between their AI methods and ideas. These common conditions are most importantly represented by common concept and task definitions, data samples, data splits, metrics, annotations, pre-processed features and descriptors, as well as interesting use case scenarios that fit the chosen domain. One of the most well known and successful examples of such initiatives is the ImageNet Large Scale Visual Recognition Challenge[2], dedicated to the evaluation of methods for object detection and image classification at large scale. Given that an impressive number of popular CNN-based approaches have been proposed and validated using ILSVRC datasets [3], we can safely say that this initiative influenced the current state of computer vision and deep learning algorithms.

The multimedia domain is no stranger to this trend, and a number of popular benchmarking initiatives, like CLEF[3], ImageCLEF[4] and MediaEval[5], have a long history of proposing interesting multimedia-centered tasks for the competition participants. It is also interesting to note that such initiatives foster cooperation between academia and industry. For example, tasks like media interestingness [4] and the detection of violent content [5] have their definition, use cases, and data created by industrial partners, according to their internal needs and to external market demands, thus ensuring that academia partners take these aspects into account.

Recently, the state of AI benchmarking initiatives has been heavily influenced by the development and adoption of Evaluation-as-a-Service (EaaS) platforms [6]. EaaS is a cloud computing-based software architecture that provides a series of tools, scripts, test instances and virtual environments that help make the assessment process more reliable by ensuring a common environment for all participants and for task organizers.

[2]`https://www.image-net.org/challenges/LSVRC/`
[3]`https://clef2022.clef-initiative.eu/`
[4]`https://www.imageclef.org/`
[5]`https://multimediaeval.github.io/`

This deliverable will present the AI4Media proposed prototype for the development of an EaaS dataset benchmarking platform, starting from a popular open-source solution and presenting a use case that applies to one of the AI tasks targeted in the context of AI4Media.

2.1 Structure of the document

This deliverable presents an analysis of the current state of EaaS systems in Section 3, reviewing some interesting main principles in Section 3.1, gathering a list of features and requirements for the AI dataset benchmarking platform in Section 3.2, reviewing a list of current EaaS platforms in Section 3.3 and comparing them in Section 3.4. Section 4 describes the current status of the AI dataset benchmarking prototype, containing a short description of the platform in Section 4.1, our development plan in Section 4.2 and a set of use case scenarios in Section 4.3. The deliverable concludes and is summarized in Section 5.

3 State of the art on Evaluation as a Service systems

Conducting a fair assessment of artificial intelligence models, their performance and capabilities is of utmost importance in understanding their reliability and drawing correct and insightful conclusions with regards to the way certain methods, parameters, model variations or data pre- and post-processing impact the final results of AI models. This section presents a state-of-the-art study on EaaS systems, comparing them according to a set of features and requirements targeted for the multimedia domain in general and for AI4Media in particular. Based on this analysis, a popular open source platform is chosen that represents the starting point for the development of the AI4Media platform model.

3.1 Main principles of Evaluation as a Service systems

Benchmarking initiatives have a long history in the AI community. The current literature [7] lists two main paradigms with regards to how participants test their systems: *Data-to-Algorithms* and *Algorithms-to-Data*. The Data-to-Algorithm approach involves participants using their methods on a published and downloaded testing set, and submitting the results of their prediction methods on the testing set. Results are then compared, according to the official competition metrics, against the corresponding ground truth data. Opposite to this, the Algorithms-to-Data paradigm involves a centralized cloud-based approach [8], where participants submit their methods instead of their results, either by integrating them via an Application Programming Interface (API) or by providing virtual environments or machines (such as Docker[6] containers) that contain their algorithms and are called in a competition-specific manner. In this approach, the results are communicated to the participants once their systems are processed in the centralized system. Currently, popular EaaS platforms implement one of these two paradigms, or even both of them, giving organizers two choices for running their competition.

While Algorithms-to-Data approaches may generate more complexity with regards to organizing the competition, by forcing the organizers to create APIs for integrating participant methods or virtual environments that allow containerization, there are several clear advantages [7] brought by the Algorithms-to-Data approach:

- *Data privacy.* In some cases, there may be a need to ensure privacy for (at least) part of the data. By making the testing set invisible for participants and making them submit their systems directly and not the prediction results, the testing data can be hidden.
- *Computational efficiency.* By running the algorithm on a centralized common architecture, certain dedicated metrics that measure complexity can be used and integrated into the competition.
- *Reproducibility and reliability.* The methods created by task participants can be tested and checked by task organizers, thus ensuring their legitimacy, but also, in case the task allows it, by any other researchers in the targeted field of study.
- *Data updates.* The testing data can be constantly updated with new samples, or new social and political trends in the targeted data, giving a real-time image of what methods and approaches give the best results.
- *Open source.* The use of Algorithms-to-Data approaches may also encourage participants to publish their entire code either on the platform itself, or on other popular code sharing platforms.

[6]https://www.docker.com/

- *Continuity.* Creating automated processes via the Algorithms-to-Data paradigm may allow for a competition to be continued even after it is officially closed, as the need for organizer oversight may be minimized.

- *Ease of integration.* The use of a common API or virtual environment setup may help with integration in real-world applications where needed.

To the best of our knowledge, the creation and use of computational efficiency and complexity-based measures has not been explored in depth by current EaaS platforms. This would represent an interesting new avenue for research, as the centralized nature of EaaS-driven benchmarking initiatives may allow re-running all the participant systems on a common architecture. Using such metrics may also involve running the systems on a larger set of computing architectures and will prove a complex endeavour, as differences may arise based on hardware acceleration availability, DNN libraries utilized by the benchmark participants, programming language efficiency and software-hardware interactions and optimizations, however, we believe the payoff would be substantial, adding a novel and useful dimension to EaaS platforms. This represents an important step towards fostering the development of Green AI. Recent studies [9] show the impact of training just one AI model on the environment. Small NLP Transformer models [10] can produce approximately 10 kilograms of CO_2 just in their training phase, while more complicated architectures, that involve processes like Neural Architecture Search approaches [11] are shown to produce as much as five cars during their entire lifetime. While these represent only the figures for the training phase of the neural network models, data published by NVIDIA[7] show that inference represents between 80% and 90% of the total energy consumption costs of average neural network models [12], a figure that can be influenced by the lifetime of the model, its size and number of parameters and the optimization its creators take into account. This last aspect is crucial, given that an EaaS-based approach would be able to measure several types of inference-based metrics, due to the centralized nature of Algotihms-to-Data approaches. Participants would submit their systems, either as code implementing a common API or as containers or virtual machines, while organizers would be able to check their real-world computational complexity by using these metrics on one or more hardware configurations.

While this is still at a very early stage of development for us, we propose creating and providing metrics that measure inference time and processing unit and memory occupancy, while paring these with important information regarding the way tests are performed, targeted hardware platforms and software libraries involved in the computational process.

3.2 Targeted features and requirements

Starting from the three goals stated in the description of this task that can benefit from the creation of an EaaS platform, namely:

- encourage the development of computationally efficient and effective systems to reduce the power footprint via introducing dedicated metrics for complexity;

- foster reproducible systems via re-running the submitted systems in the evaluation phase;

- building a common repository for sharing the data and developing approaches for distributed benchmarking with container submission on possibly confidential data;

[7] `https://www.youtube.com/watch?v=ET2KVe2du3Y`

and the current trends in the development of EaaS systems, we compile a list of important features and requirements for choosing the most appropriate open-source platform as the development starting point for our prototype. Thus, we choose the following features:

- **Open-source**. The chosen platform must be open-source, allowing us to download and locally replicate the server backbone in order to organize competitions using our own infrastructure and servers. Also, an open-source platform allows for changes to the source code, thus helping with correcting bugs, optimizing, as well as adding functionality by changing the source code. In fact, we consider this one of the most important requirements for such a platform, as any missing features can be later added on as needed.
- **Web-based**. The EaaS platform must provide a web-based interface that allows organizers to host, define and manage the benchmarking competitions and resources, participants to submit their systems or runs and view the results, and other interested parties to view the competitions.
- **Maintained**. Is the open-source project still maintained to this day? While development on this platform would involve changing the source code more or less, starting from a maintained project may give us the advantage of starting from an up-to-date codebase, having access to regular updates, as well as the possibility of receiving help from the developers of the platform.
- **Multi-phase**. The platform must allow for the creation of multiple phases for each competition, that may correspond, for example, to a regular training - validation - testing cycle of algorithm development.
- **Sub-tasks**. The platform must provide options to create sub-tasks or sub-competitions, that allow different aspects of the competition to be explored by participants. This may include, depending on the case, different types of media data (e.g., image versus video processing sub-tasks), different goals for the submitted methods (e.g., simple prediction versus generalization), or analysis of different aspects (e.g., prediction metrics versus computational complexity metrics).
- **Scheduling**. Competition organizers should be able to create schedules, allowing certain components of the competition to be locked or unlocked based on a predefined schedule. This may include hiding the leaderboards until a certain time, locking and unlocking phases or sub-tasks according to the organizers' needs, etc.
- **Dissemination**. The platform should be able to allow organizers to send and receive notification, emails regarding competition news and participant status, as well as any other messages are considered important.
- **Forums**. Each competition can have an attached forum, where organizers, competitors and other interested parties can exchange views and ideas, offer help or discuss the future of the competition and analyze the results.
- **Parser**. Parsers are pieces of code or scripts that are executed when certain events happen. For example, they can ingest the submission results from competitors and check whether the format of the submitted runs is correct. On the other hand, in cases where submission is done via an Algorithm-to-Data approach, the parser may check whether the API model or Docker requirements are met by the submitted algorithms.

- **Scoring**. The systems allows organizers to create and upload scoring programs that compare participant submissions with the reference or ground-truth data. Each competition must accept several metrics either as the primary (main) metrics or as secondary ones.
- **Leaderboards**. The performance of the submitted systems must be displayed in an easy to understand and manipulate graphical manner.
- **Auditing**. Enables participants and organizers to see the histories for each successive submission. For a given competition, participants can view the history for their own submissions, while organizers can see the history for all submissions.
- **Reproducibility**. Participants can submit shared code or containers that will allow organizers and interested parties to run for the individual submissions in order to test result reproducibility. Organizers must have the ability to provide API or container models and guidelines that will help participants ensure their algorithms are compatible and will run without problems when tested.
- **Containerization**. Containers can be used for defining the common environment of a run bundle, as well as creating a set of pre-determined run environments that organizers can share and use for different purposes.
- **Shared computing**. Allows the assignment and setup of a server to which reproducible submissions can be send and executed.

3.3 Current Evaluation as a Service platforms

This section presents some of the most popular and important EaaS platforms currently available, with a focus on their most important and unique features, providing links to their main website as well as to their source code, in case we are dealing with open-source platforms.

Codalab[8] is an EaaS platform that is focused on hosting and running reproducible research competitions, licensed under the Apach License 2.0. It is designed with flexibility in mind, allowing connections to popular cloud-computing solutions and using private worker machines. At this moment, the platform provides a "Worksheets" segment, for running reproducible experiments and creating executable papers, and a "Competitions" segment for creating and participating in competitions.

EvalAI[9] is a centralized and open-source platform under BSD License for hosting, participating and collaborating in AI challenges that allow custom evaluation protocols and phases, remote evaluation, evaluation inside environments in the form of docker images evaluated against test environments on the evaluation server, and CLI support. It is easy to install and configure. It currently provides a docker image that can be built on any machine that runs a Linux environment. Creating a benchmark is easy without involving completing forms or interacting with the platform developers to get started. Furthermore, the platform and documentation is complete and constantly updated.

AIcrowd[10] is an open platform for open data challenges. Its source code is available on GitHub under the GNU Affero General Public License v3.0 (Permissions of this strongest copyleft license are conditioned on making available complete source code of licensed works and modifications, which include larger works using a licensed work, under the same license. Copyright and license

[8] `https://codalab.org/`, `https://github.com/codalab/codalab-competitions`
[9] `https://eval.ai/`, `https://github.com/Cloud-CV/EvalAI`
[10] `https://www.aicrowd.com/`, `https://github.com/AIcrowd/AIcrowd`

notices must be preserved. Contributors provide an express grant of patent rights. When a modified version is used to provide a service over a network, the complete source code of the modified version must be made available.). The platform allows organizers to launch benchmarks, define the evaluation rules and provide data and other information to competitors. Furthermore, it has support for GDPR, fine-grained email preferences. One drawback is that the documentation is scarce and incomplete, being difficult to deploy on arbitrary servers. Hosting challenges on AIcrowd is not independent of the platforms managing team. It is required to make a request for each instance.

Dynabench[11]is a framework for dynamic data collection, hosting benchmarking and performing holistic model comparison. Facebook develops the platform, currently under development. The platform is not yet released. It promises techniques for circumventing issues of accessibility and reproducibility, allowing models to be submitted directly to be evaluated in the cloud.

Tira[12] is a general web service based platform, aimed at general computer science tasks. Organizers can receive submissions in the form of executable software, providing this function without the integration of containerized approaches. The platform is currently in use for several years, being employed in the context of the PAN evaluation tasks and the CoNLL conference.

VISCERAL Registration System[13] is an open source platform that was used for the VISCERAL benchmarks and is publicly released on github. The system allows both participant and administrator access. Its main purpose is participants registration, which includes a user agreement download and upload for the document signed by the participants. Registered users are assigned a virtual machine (VM). Several benchmarks can be created at once for different tasks. These also have a leaderboard assigned to them, which is visible to participants during the competition. Participants are allowed multiple submissions inside the VMs and they can choose which results to make publicly available and which ones they should keep private. The system's code is written in Java EE and it closely interacts with the Azure cloud.

OpenML[14] is an open framework, under the BSD 3-Clause, for sharing and organizing data, machine learning algorithms and experiments. It is designed to create a frictionless, networked ecosystem, that can be readily integrate into existing processes/code/environments, allowing collaboration irrespective of the tools and infrastructure.

TREC[15] supports competition and research in the information retrieval community. It is a well known platform and it has a long history being used during the TREC evaluation campaigns, while also providing popular tools for performance evaluation in the information retrieval domain.

Optil IO[16] is a website that allows competitions on optimization functions. Each problem requires to find a maximal/minimal value of an objective function and users compete for the best values. It consists of a computation engine, which evaluates each submitted solution and establishes a leaderboard. Each run is assigned its own share of computational resources (CPU, memory) and it must undergo several restrictions (runtime, output limit, memory etc,). If a task can be expressed as an optimization problem, then it can most likely be adapted for this tool.

CBIBOP[17] is an EaaS platform that is forked from the popular Codalab codebase, that features some custom made changes to the codebase, targeting both containerization and VM integration, and medical imaging viewers and data handlers.

HOBBIT[18] is an online benchmarking platform which allows administrators to create their

[11]`https://www.dynabench.org/`, `https://github.com/facebookresearch/dynalab`

[12]`https://www.tira.io/https://github.com/tira-io/`

[13]`https://github.com/Visceral-Project/registration-system`

[14]`https://www.openml.org/`, `https://github.com/openml/OpenML`

[15]`https://trec.nist.gov/`,`https://github.com/usnistgov/trec_eval`

[16]`https://www.optil.io/optilion/`

[17]`http://cbibop.github.io/`,`https://github.com/cbibop`

[18]`https://hobbit-project.github.io/`, `https://github.com/hobbit-project`

own benchmarks (systems may be written in different languages) and deploy them in a highly modularized and virtualized manner, running on Docker containers. If the user is logged in as challenge organizer, the page might vary. The platform is written in Java and its communication is based on RabbitMQ. This platform serves as a deployment environment for custom systems that each benchmark may want to design. It also allows for different subtasks for a given competition, multiple submissions and customised KPIs. Several reports are generated at the end of the competition that allow for a good understanding of the results.

Analytics Vidhya[19] is a community of Analytics and Data Science building the next-gen data science ecosystem. The aim of the platform is to become a complete portal serving all knowledge and career needs of Data Science Professionals. This platform frequently posts data hacks, skill tests, etc., but lacks many features and control for independent task organizers.

Unearthed[20] is a closed source general AI platform that implements EaaS functions and solutions aimed at real-world data from geology to general computing processing problems extracted and defined by companies.

Kaggle[21] is a popular platform that hosts a bundle of resources for the machine learning community (open challenges, datasets, source codes, courses, etc.). Setting up a research challenge on Kaggle needs to follow some specific regulations such as offering a cash prize and conditioning it on the winning solution source code to be openly available. On this platform, the organizer decides the competition's format, i.e. if the submissions are under the form of results file, Jupyter notebooks, etc. Several metrics are imposed and two leaderboards are available: a public one available to all users during the competition and a private one available only after the competition ends. The private leaderboard is usually compiled after running the users' systems on a subset of the test data. This test data may be visible or not to the participants beforehand. Users can register independently or as part of a team and can submit an unlimited number of runs. The platform also gives access to processing resources such as CPU and GPU, provided that the code is uploaded under a Jupyter notebook format. In order to participate to challenges, users must comply to the challenge's specific rules, which cover standard license agreements. Advantages of this platform include its high popularity, thus attracting a large number of competitors, and the ability to store and share large datasets.

Drivendata[22] hosts online data science competitions where the data and problem are posed by a socially-minded organization. One can either join a competition or host one on the platform. It provides accounts, content management, rules agreement, team formation, submission scoring, live leaderboard, audit trail, platform security. Moreover, the datasets listed in Drivendata are related to Non-Profits ranging from wildlife preservation to public health. Hosting challenges on Drivendata is not independent of the platform's managing team. It is required to make a request for each instance.

Datasource[23] is a platform similar to Kaggle as it is able to host machine learning competitions in the same manner. Administrators can propose a challenge where they provide the dataset, the evaluation metric and sometimes several other rules. Competitors are requested to upload a results file under a given format. Users can participate only individually to these challenges and winners are obliged to upload their machine learning model in a Jupyter Notebook at the end of the competition so that it is accessible by the party that proposed the challenge. Since these parties are usually smaller businesses, the awarded prizes are also a bit smaller than those proposed on Kaggle, but the competitions also span a shorter time interval. In addition, this

[19]`https://datahack.analyticsvidhya.com/`
[20]`https://unearthed.solutions/`
[21]`https://www.kaggle.com/`
[22]`https://www.drivendata.org/competitions/`
[23]`https://www.datasource.ai/en`

platform also features a tournament where users compete against each other on different stages, until a final winner is selected. Moreover, Datasource also acts as a headhunting platform since the companies that propose challenges can also access the competition's winner's CV. At the end of the competition, the uploaded models are used for validation against a private test dataset.

BioASQ[24] organizes challenges on biomedical semantic indexing and question answering (QA). The challenges include tasks relevant to hierarchical text classification, machine learning, information retrieval, QA from texts and structured data, multi-document summarization and many other areas. Hosting challenges on BioASQ is different from the other all-purpose platforms, meaning that the challenges must be accepted by the BioASQ organizers on which they setup workshops to the CLEF international conference.

Codeforces[25] is a platform aimed at creating general computer programming competitions. Going one step further, Codeforces also proposes to create a social network where professional software developers, and computer science students can interact and exchange ideas. Furthermore, the participant's skills are reflected by their rating achieved in competitions they previously participated to.

PAN[26] is a platform for a series of scientific events and shared tasks on digital text forensics and stylometry, which are part of the CLEF, FIRE and PAN Workshops competitions. Users are presented with the competition's description, dataset, a format that their results file must respect, the validation code that verifies the format of the output files, a baseline approach and a leaderboard to check their progress live. There are multiple tasks that are run each year, each with several subtasks and an unlimited number of submissions is allowed for each independent user. The evaluation of results is done on TIRA after a VM request and the code from the best submissions are uploaded and stored on the project's github page. In some cases (e.g. the Hyperpartisan News Detection Task), cash prizes were also awarded to the best team.

3.4 Comparing the current platforms

In order to choose the most appropriate open-source platform as a starting point for our development process, we compare the platforms listed in Section 3.3 using the features and requirements listed in Section 3.2. The results are presented in Table 1.

The most evident conclusion of the table is that Codalab satisfies all the requirements, and is the single EasS platform to do so. There is also a great amount of interest around this platform, with its repositories[27] being updated on a regular schedule. The different branches of the Codalab repository are well documented, either through a special documentation webpage[28] or through wiki pages on GitHub[29]. *Given these aspects, we conclude that Codalab represents the most appropriate starting point for our platform.*

It is also encouraging to point out that the great majority of these platforms are still maintained, with only two exceptions identified by us. Also, some of the listed projects are intertwined, either by being developed as extensions of other projects (e.g., the PAN platform uses the TIRA environment for evaluation, by using virtual machine requests), or by being forked from a base project (e.g, the CBIBOP project is an extension of Codalab). Furthermore, new platforms are also being currently under development, such as the Dynabench project. Also, while not all the listed platforms are dedicated to tasks and competitions that use multimedia data (e.g., Codeforces is dedicated to

[24] `http://bioasq.org/`, `https://github.com/BioASQ`
[25] `https://codeforces.com/`
[26] `https://pan.webis.de/`, `https://github.com/pan-webis-de`
[27] `https://github.com/orgs/codalab/repositories?type=all`
[28] `https://codalab-worksheets.readthedocs.io/en/latest/`
[29] `https://github.com/codalab/codalab-competitions/wiki`

Platform	os	wb	mt	mp	sb	se	di	fo	pa	sc	lb	au	rp	co	sh
Codalab	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EvalAI	✓	✓	✓	✓	✓	✓	no	no	✓	✓	✓	✓	✓	✓	✓
AIcrowd	✓	✓	✓	no	✓	✓	✓	✓	no	✓	✓	✓	no	no	no
Dynabench	no	✓	✓	✓	✓	✓	no	no	no	✓	✓	✓	✓	✓	✓
Tira	✓	✓	✓	✓	✓	✓	no	no	✓	✓	✓	✓	no	no	no
VISCERAL	✓	✓	no	no	✓	✓	no	no	no	✓	✓	✓	✓	✓	no
OpenML	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	no	no
TREC	✓	✓	✓	✓	✓	✓	no	no	no	✓	✓	✓	no	no	no
Optil IO	no	✓	✓	no	✓	no	no	✓	no	✓	✓	✓	✓	✓	no
CBIBOP	✓	✓	no	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	no
HOBBIT	✓	✓	✓	✓	✓	✓	no	no	no	✓	✓	✓	✓	✓	no
A. Vidhya	no	✓	✓	✓	no	✓	no	✓	no	✓	✓	no	✓	no	no
Unearthed	no	✓	✓	✓	no	no	no	✓	✓	✓	✓	✓	no	no	no
Kaggle	no	✓	✓	✓	no	✓	✓	✓	no	✓	✓	✓	✓	no	✓
Drivendata	no	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Crowdanalytix	no	✓	✓	✓	✓	✓	✓	✓	no	✓	✓	✓	✓	no	no
Datasource	no	✓	✓	no	no	✓	no	✓	no	✓	✓	no	✓	no	no
BioASQ	no	no	✓	✓	✓	no	no	no	no	✓	✓	no	no	no	no
Codeforces	no	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	no	no
PAN	✓	✓	✓	no	✓	✓	no	no	no	✓	✓	✓	no	no	no

Table 1. Comparing the EaaS platforms listed in Section 3.3 by using the features and requirements listed in Section 3.2. The following abbreviations are used for the the features and requirements: open-source (os), web-based (wb), maintained (mt), multi-phase (mp), sub-tasks (sb), scheduling (se), dissemination (di), forums (fo), parser (pa), scoring (sc), leaderboards (lb), auditing (au), reproducibility (rp), containerization (co), shared computing (sh).

general computer programming tasks and problems, TREC is traditionally involved with text processing), we still believe their inclusion in this work to be of importance, as they create a larger overview of this domain. Considering that the chosen features are not necessarily targeted for multimedia data, but rather at functions and programming paradigms that can be exploited by any type of EaaS platform, we consider the inclusion of non-multimedia platforms in this study as an useful addition.

4 Prototype platform for AI dataset benchmarking

This Section presents the prototype of the AI4media AI dataset benchmarking platform. Starting from the Codalab baseline, chosen in Section 3, we present some aspects with regards to the development of the prototype, including a general description of the available code, a development plan, and a use case scenario. We provide the source code for the AI4Media platform in the form of a GitHub repository[30], containing the source code for the platform as well as the use case needed for testing it.

4.1 Platform description

Codalab is an open-source framework for running research competitions licensed under the Apache License 2.0. It can be deployed front-end and back-end independently on an arbitrary server or using their limited hosting capabilities. For the former, it allows full customization, control and privacy with pre-made and re-usable AWS virtual machine image (called AMI) or Google Cloud Image, both with Docker-Codalab-Ubuntu already pre-configured, hence easier to install or to configure Codalab from scratch working on Azure, Google Cloud Compute, etc., (any space that can run an Ubuntu/Linux box). For the latest, it allows using their hosts with limited workers capabilities. An important aspect is that it allows adding more private workers (your workstations) for enhancing the computational power for a faster evaluation process of the submitted runs. It provides a powerful dashboard for the organizers allowing email notification, self-hosting storage, scheduled backups or restoring, setting groups and permissions, code checker, executing jobs, scoring programs, configure leaderboards, etc. One important aspect is that Codalab competition bundles allow an arbitrary number of phases and metrics and customized ingestion and scoring programs. Furthermore, it is well documented and maintained with an active community and developers, with its source publicly available on a well-known repository (GitHub).

4.2 Development plan

We present several of the high-level main attributes and user-centric functions that are planned for the platform in its final form, and indicate the current state of development, according to the following descriptions: (i) development not started yet, (ii) development started, no tests performed, (iii) prototype development and testing finished, (iv) final development finished, (v) final development and testing finished. We present these attributes and functions below, while also specifying some visual examples for the features already developed and tested for this prototype, in the context of the use case presented in Section 4.3. The functions and their current development and testing status are summarized in Table 2.

- **Creating a competition via a competition bundle.** Current status: prototype development and testing finished. Organizers can create a competition bundle, respecting a basic given format, and define all the aspects of the competition as listed in Section 3.2. A visual example of this function is presented in Figures 2 and 3.
- **Creating a competition via GUI.** Current status: development not started yet. Organizers can create and define all the aspects of a competition as listed in Section 3.2 in a Graphical User Interface (GUI).

[30]`https://github.com/AIMultimediaLab/AI4Media-EaaS-prototype-Py2-public`

- **Editing a competition via GUI.** Current status: prototype development and testing finished. Organizers can edit all the aspects of a competition in a GUI. A visual example of this function is presented in Figure 4.
- **Participants have access to all the data via GUI.** Current status: prototype development and testing finished. Participants can register, download data and submit their runs via the web-based GUI. A visual example of this function is presented in Figure 5.
- **Participant submissions are logged and errors are displayed accordingly.** Current status: prototype development and testing finished. Participants can see the output of competition scoring and parsing scripts and detailed error logs are shown in case problems or bugs are encountered. A visual example of this function is presented in Figure 7.
- **Complete leaderboards can be displayed and browsed by interested parties.** Current status: prototype development and testing finished. Leaderboards containing the entire history of submission can be accessed by interested parties, according to the visibility level set by the organizers. A visual example of this function is presented in Figure 6.
- **Organizers, participants and other interested parties can interact.** Current status: development started, no tests performed. Organizers, participants and other interested parties have automated means of communication, like forums, announcement boards, and mailing lists.
- **Computational efficiency measured via complexity metrics.** Current status: development not started yet. A set of complexity metrics are developed and will be made available for organizers, that are able to measure the computational efficiency of participant systems.
- **Testing system reproducibility via API integration.** Current status: prototype development and testing finished. APIs can be designed and made available by organizers, that would foster system reproducibility by re-running submitted systems. A visual example of this function is presented in Figure 8.
- **Testing system reproducibility via virtual machines or containers.** Current status: development started, no tests performed. Virtual machines or containers can be submitted by participants, allowing organizers to test system reproducibility by re-running the submitted containers.
- **Ensuring data privacy by hiding the testing data and allowing only API or container-based submission.** Current status: development started, no tests performed. The testing part of the dataset can be hidden from participants, by making them submit trained systems and running them on separate cloud-based instances.
- **Third party cloud integration.** Current status: development not started yet. Popular third party cloud-based instances (e.g., Azure, AWS) can be attached as processing instances to any competition.
- **Auditing tools for platform hosts and maintainers.** Current status: development started, no tests performed. A complete set of auditing tools is available for platform hosts and maintainers, that would provide important data for helping task organizers getting their tasks online.

Funtion	Development status	Testing status
Creating a competition via a competition bundle	proto finished	proto testing
Creating a competition via GUI	not started	not started
Editing a competition via GUI	proto finished	proto testing
Participants have access to all the data via GUI	proto finished	proto testing
Submissions logged and errors displayed	proto finished	proto testing
Complete leaderboards displayed	proto finished	proto testing
Organizers and participants can interact	started	not started
Computational efficiency measured	not started	not started
Testing system reproducibility via API integration	proto finished	proto finished
Testing system reproducibility via VMs or containers	started	not started
Ensuring data privacy by hiding the testing data	started	not started
Third party cloud integration	not started	not started
Auditing tools for platform hosts and maintainers	started	not started

Table 2. This table presents the main high-level attributes and user-centric functions that are planned for the AI4Media platform, while indicating their current development and testing status.

4.3 Use case scenario

To test the proposed AI4Media benchmarking platform, we select a relevant use case scenario. The selected use case scenario is based on the 2022 ImageCLEFfusion competition[31], that is currently under development and will be finalized and published towards the end of July 2022, according to the official schedule. We plan to continue this initiative, after the official rounds hosted by ImageCLEF are over, by hosting it on our prototype platform. The aim of this competition is the development of late fusion mechanisms, that solve a set of regression and retrieval problems, focusing on the prediction of media interestingness and diverse search content retrieval.

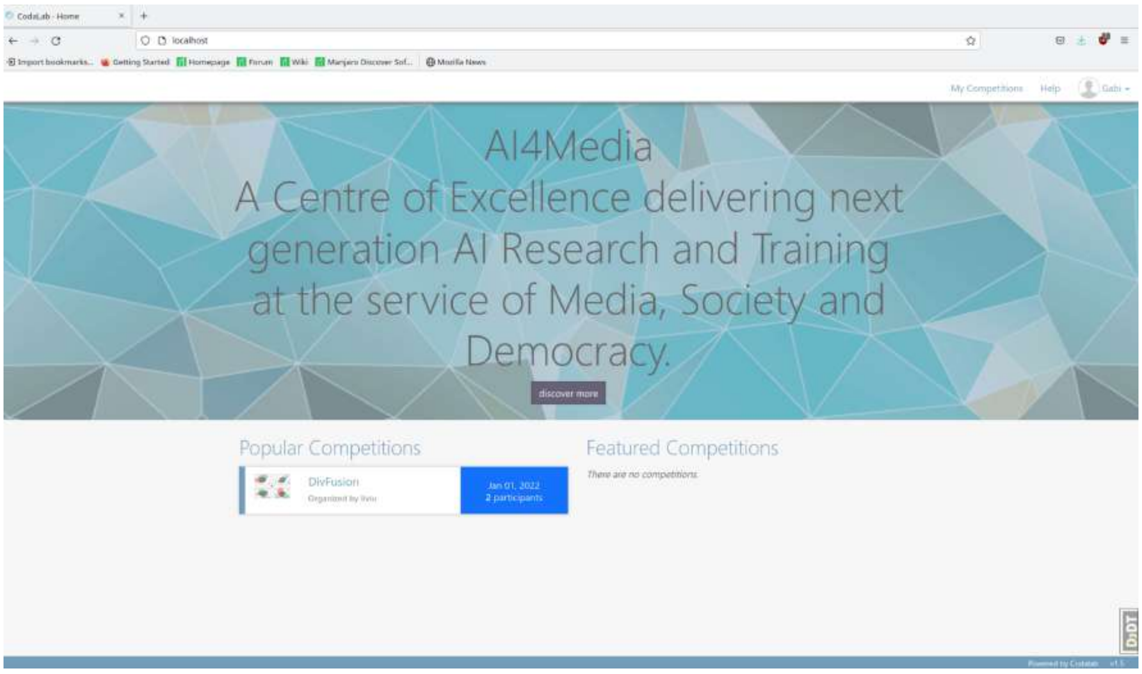
Starting from the default homepage of the platform, shown in Figure 1, organizers and users can create and manage competitions, submit runs, view results and communicate. We will present some of these aspects in the following sections, looking at the options organizers and participants have for browsing the data on the platform, and analyze some aspects related to reproducibility.

4.3.1 Organizer use cases

First of all, organizers can **create a new competition**, by defining and populating a "Competition Bundle" (see Figure 2). A Competition Bundle is a zip bundle that contains the competition's YAML file. Other assets are included in the zip archive, but they will only be used if they are mentioned in the competition.yaml file. The logo, the data, the documentation (in html format), and the scoring applications and scripts are all included in the package. Given our use case scenario, the contents of the bundle file can be defined by the following general structure:

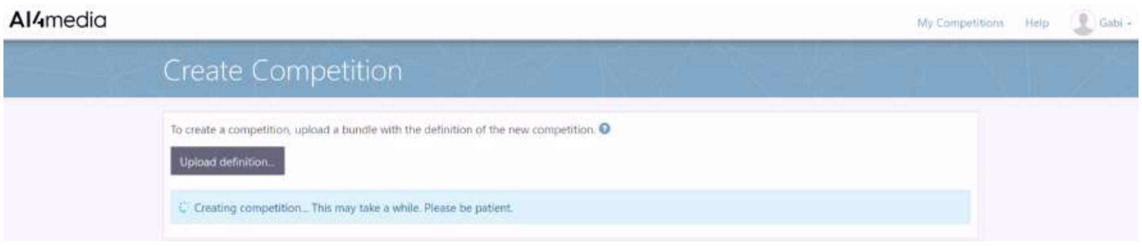
- logo.jpg - a logo for the competition;
- overview.html - a webpage that presents the overview of the competition, presenting its scope, definitions and general considerations;

[31] `https://www.imageclef.org/2022/fusion`



AI4Media
A Centre of Excellence delivering next generation AI Research and Training at the service of Media, Society and Democracy.

Figure 1. The main page of the prototype platform, including links to the most popular competitions currently opened.



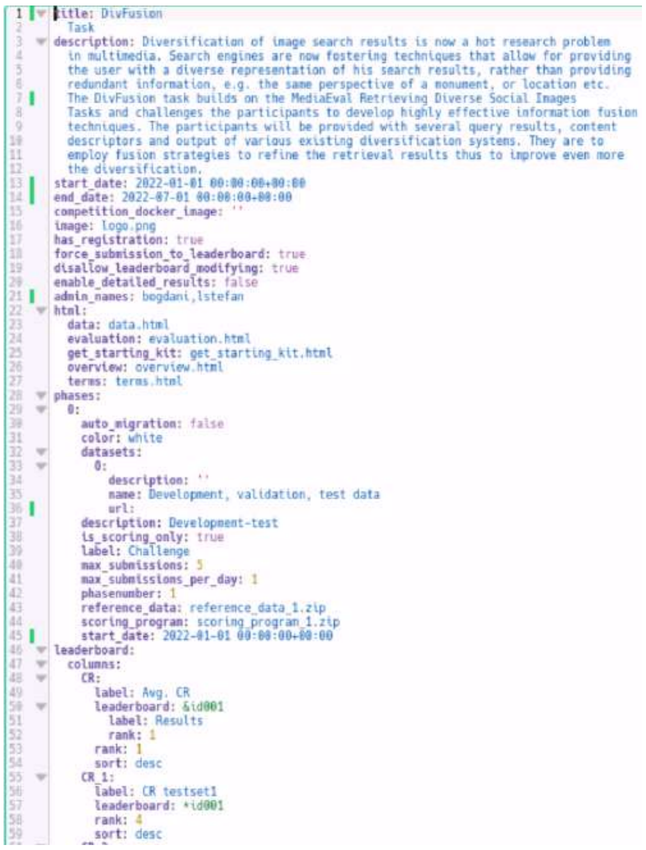
To create a competition, upload a bundle with the definition of the new competition.

Creating competition... This may take a while. Please be patient.

Figure 2. Uploading the competition bundle creates a new competition, using the information stored in the archive.

- data.html - a webpage that contains information about the data used in the competition;
- evaluation.html - a webpage that describes the evaluation process;
- terms_and_conditions.html - a webpage that defines the terms that participants have to acknowledge in order to participate;
- reference.zip - contains the ground truth data associated with the competition
- program.zip - contains the parser for the files participants submit to the competition;
- competition.yaml - defines the parameters of the competition and links all the other files included in the bundle in order to create the final competition.

The parser program archive can be used in several scenarios, checking participant submissions and deciding how to process it (e.g., the organizers may allow the participants to submit either

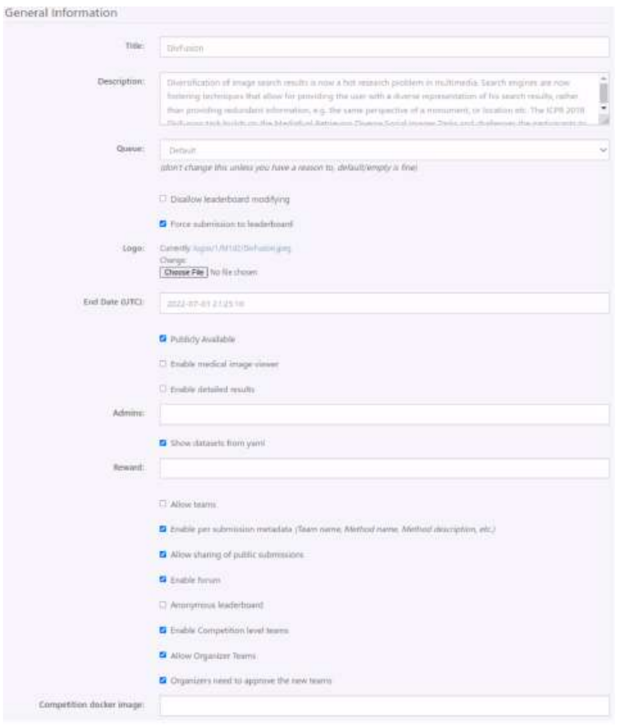


```yaml
title: DivFusion
  Task
description: Diversification of image search results is now a hot research problem
  in multimedia. Search engines are now fostering techniques that allow for providing
  the user with a diverse representation of his search results, rather than providing
  redundant information, e.g. the same perspective of a monument, or location etc.
  The DivFusion task builds on the MediaEval Retrieving Diverse Social Images
  Tasks and challenges the participants to develop highly effective information fusion
  techniques. The participants will be provided with several query results, content
  descriptors and output of various existing diversification systems. They are to
  employ fusion strategies to refine the retrieval results thus to improve even more
  the diversification.
start_date: 2022-01-01 00:00:00+00:00
end_date: 2022-07-01 00:00:00+00:00
competition_docker_image: ''
image: logo.png
has_registration: true
force_submission_to_leaderboard: true
disallow_leaderboard_modifying: true
enable_detailed_results: false
admin_names: bogdani,lstefan
html:
  data: data.html
  evaluation: evaluation.html
  get_starting_kit: get_starting_kit.html
  overview: overview.html
  terms: terms.html
phases:
  0:
    auto_migration: false
    color: white
    datasets:
      0:
        description: ''
        name: Development, validation, test data
        url:
    description: Development-test
    is_scoring_only: true
    label: Challenge
    max_submissions: 5
    max_submissions_per_day: 1
    phasenumber: 1
    reference_data: reference_data_1.zip
    scoring_program: scoring_program_1.zip
    start_date: 2022-01-01 00:00:00+00:00
leaderboard:
  columns:
    CR:
      label: Avg. CR
      leaderboard: &id001
        label: Results
        rank: 1
      rank: 1
      sort: desc
    CR_1:
      label: CR testset1
      leaderboard: *id001
      rank: 4
      sort: desc
```

Figure 3. A basic example of the competition.yaml file.

results, code, or data), call some organizer provided API functions that take testing data input and generate predictions, or simply calculate the metrics associated with each participant run. The competition.yaml file is the most important file in this setup, as it defines how the competition is setup and how all the other files and scripts interact. A basic example is presented in Figure 3, where some vital details are given with regards to the competition. As shown in the example, organizers can define a logo for the competition, provide a short description, manage the schedule and the phases associated with the competition (development, validation, training), provide a link to the docker images attached to the competition, provide a general outline for the webpage containing information of the competition, setup the scoring scripts, ground truth data and submission details, setup the leaderboards according to the desired metrics.

Furthermore, **during the competition**, organizers will continually manage the users and their submissions, having access to all of them. After the initial bundle is uploaded and the competition is defined, changes can be made by the organizers from the webpage directly, in an easy-to-use graphical interface. Such an example is presented in Figure 4, where organizers can change the basic details of the competition, but this can be applied to all the aspects of competitions. In future developments, we plan to extend the use of the graphical interface and even allow the creation of competitions through it, thus easing the organizer's jobs.



General Information

Title: DivFusion

Description: Diversification of image search results is now a hot research problem in multimedia. Search engines are now fostering techniques that allow for providing the user with a diverse representation of his search results, rather than providing redundant information, e.g. the same perspective of a monument, or location etc. The ICPR 2018 [illegible]

Queue: Default
(don't change this unless you have a reason to, default/empty is fine)

☐ Disallow leaderboard modifying

☑ Force submission to leaderboard

Logo: Currently: logos/1/M182/DivFusion.jpeg
Change:
Choose File No file chosen

End Date (UTC): 2022-07-01 21:25:18

☑ Publicly Available

☐ Enable medical image viewer

☐ Enable detailed results

Admins:

☑ Show datasets from yaml

Reward:

☐ Allow teams

☑ Enable per submission metadata (Team name, Method name, Method description, etc.)

☑ Allow sharing of public submissions

☑ Enable forum

☐ Anonymous leaderboard

☑ Enable Competition level teams

☑ Allow Organizer Teams

☑ Organizers need to approve the new teams

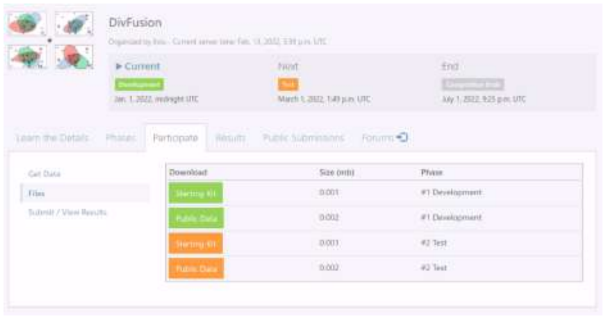
Competition docker image:

Figure 4. Editing the details of a competition.

4.3.2 Participant use cases

From a participant standpoint, the first steps in participating to a competition are represented by **registering** and having the registration accepted by the task organizers. Following these initial steps, the data associated with the opened phases of the competition can be **downloaded**, as presented in Figure 5. Furthermore, **participant submissions** must respect a format, as defined by the organizers in the description of the task. The platform allows the organizers to check the format, therefore participants can be warned in case their submissions do not respect the desired input / output restrictions. Currently, two types of submissions are accepted: code or results. While results submissions involve participants uploading files that contain their method's predictions on the testing data, code submissions must respect the organizer provided API and must also provide an entry point for their method, represented in our use case by a metadata file that specifies what command must be executed in order for the method to run.

Finally, participants can **view their results and the leaderboads**, as presented in Figure 6, for each phase and each subtask, can arrange the submissions according to their desired metrics or parameters and can even download the leaderboard data in csv format. Any errors that may show up during the evaluation process are also available for participants, in order to help them debug their systems or correct the format, by viewing the errors logs and standard ouput logs associated with each submission, as shown in Figure 7. Many types of logs are stored and associated with each run, giving both participants and organizers a complete picture of what errors, warnings and additional information each run produces, including but not limited to scoring, prediction, and ingestion logs.



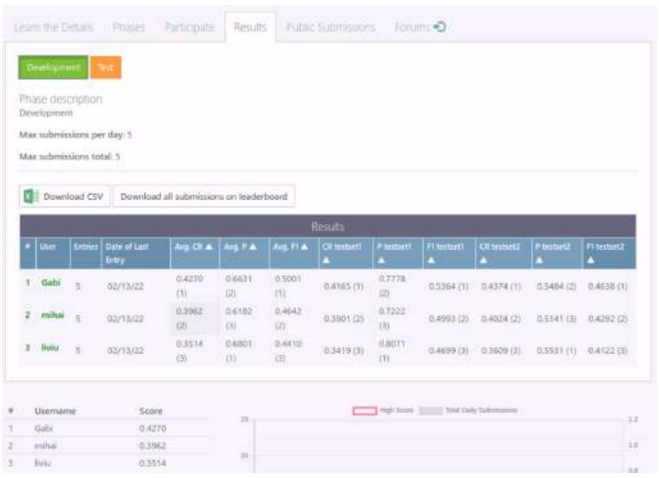
DivFusion

Organized by liviu - Current server time: Feb. 13, 2022, 3:38 p.m. UTC

▶ Current	Next	End
Development	Test	[illegible]
Jan. 1, 2022, midnight UTC	March 1, 2022, 1:49 p.m. UTC	July 1, 2022, 9:25 p.m. UTC

Download	Size (mb)	Phase
Starting Kit	0.001	#1 Development
Public Data	0.002	#1 Development
Starting Kit	0.001	#2 Test
Public Data	0.002	#2 Test

Figure 5. Participant view of the phases of the competition and data download options.



Phase description
Development

Max submissions per day: 5

Max submissions total: 5

Results

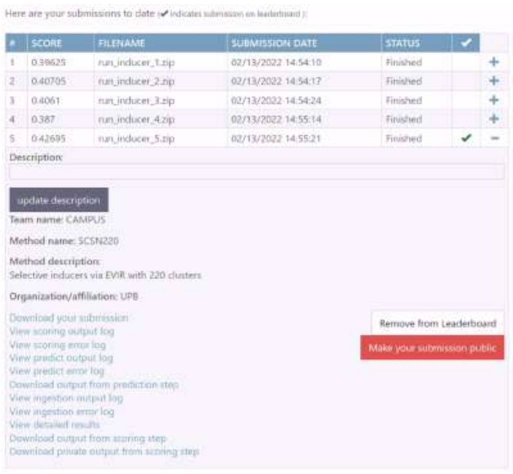
#	User	Entries	Date of Last Entry	Avg. CR ▲	Avg. P ▲	Avg. F1 ▲	CR testset1 ▲	P testset1 ▲	F1 testset1 ▲	CR testset2 ▲	P testset2 ▲	F1 testset2 ▲
1	Gabi	5	02/13/22	0.4270 (1)	0.6631 (2)	0.5001 (1)	0.4165 (1)	0.7778 (2)	0.5364 (1)	0.4374 (1)	0.5484 (2)	0.4638 (1)
2	mihai	5	02/13/22	0.3962 (2)	0.6182 (3)	0.4642 (2)	0.3901 (2)	0.7222 (3)	0.4993 (2)	0.4024 (2)	0.5141 (3)	0.4292 (2)
3	liviu	5	02/13/22	0.3514 (3)	0.6801 (1)	0.4410 (3)	0.3419 (3)	0.8011 (1)	0.4699 (3)	0.3609 (3)	0.5531 (1)	0.4122 (3)

#	Username	Score
1	Gabi	0.4270
2	mihai	0.3962
3	liviu	0.3514

Figure 6. Participant view of results and of the current leaderboard.

4.3.3 Algorithm-to-Data API and Reproducibility

Finally, it is important to show some capabilities of the prototype platform that relate to reproducibility, as this in one of the most important aspects and advantages of EaaS systems. For this use case we have chosen an API-based reproducibility model, where participants are provided with a general code model or Abstract Base Class (ABC) associated with the code submission that initializes a "model" ABC, and can ingest input data according to a tensor size, outputting predictions. The outline of such a class, as defined by the task organizers, is presented in Figure 8. This submission format can be checked with the help of the parser function in order to ensure that the format is respected before having to run the system itself. Furthermore, all the participant files associated with the official submissions can be downloaded, including the AI models themselves, thus allowing for future interested parties to test, check and reproduce the official results.


Here are your submissions to date (✔ indicates submission on leaderboard):

#	SCORE	FILENAME	SUBMISSION DATE	STATUS	✔	
1	0.39625	run_inducer_1.zip	02/13/2022 14:54:10	Finished		+
2	0.40705	run_inducer_2.zip	02/13/2022 14:54:17	Finished		+
3	0.4061	run_inducer_3.zip	02/13/2022 14:54:24	Finished		+
4	0.387	run_inducer_4.zip	02/13/2022 14:55:14	Finished		+
5	0.42695	run_inducer_5.zip	02/13/2022 14:55:21	Finished	✔	−

Description:

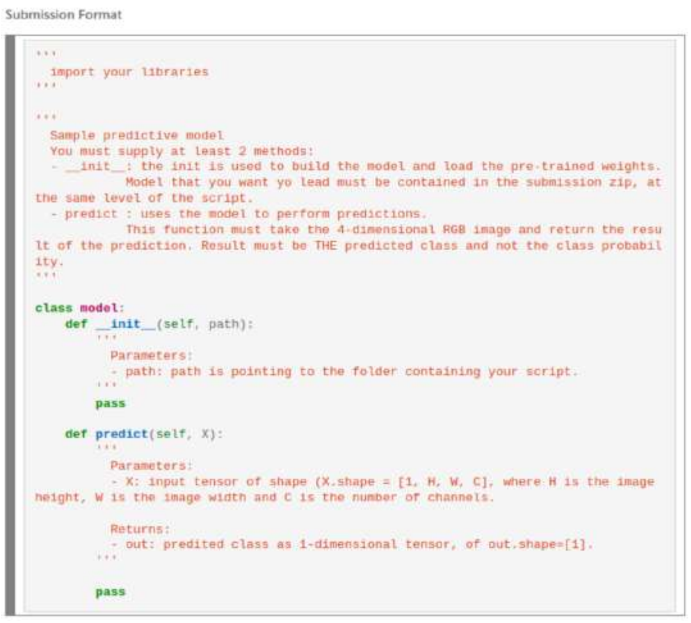
Team name: CAMPUS

Method name: SCSN220

Method description:
Selective inducers via EVIR with 220 clusters

Organization/affiliation: UPB

Figure 7. User submission panel, containing information regarding individual submissions, logs and errors associated with each individual run.


Submission Format

```
'''
  import your libraries
'''

'''
  Sample predictive model
  You must supply at least 2 methods:
  - __init__: the init is used to build the model and load the pre-trained weights.
            Model that you want yo lead must be contained in the submission zip, at
the same level of the script.
  - predict : uses the model to perform predictions.
            This function must take the 4-dimensional RGB image and return the resu
lt of the prediction. Result must be THE predicted class and not the class probabil
ity.
'''

class model:
    def __init__(self, path):
        '''
          Parameters:
          - path: path is pointing to the folder containing your script.
        '''
        pass

    def predict(self, X):
        '''
          Parameters:
          - X: input tensor of shape (X.shape = [1, H, W, C], where H is the image
height, W is the image width and C is the number of channels.

          Returns:
          - out: predited class as 1-dimensional tensor, of out.shape=[1].
        '''

        pass
```

Figure 8. An overview of an API description that uses an Abstract Base Class called “model”.

5 Summary and conclusions

This deliverable provides an overview of the research and work conducted in task T4.6: Benchmarking of AI Systems, which is part of WP4: Explainability, Robustness and Privacy in AI. We showed our work towards developing a prototype platform for AI dataset benchmarking, based on an Evaluation-as-a-Service paradigm. To this end, we proposed choosing an appropriate open-source platform as a starting point in the development of our own EaaS prototype.

We started by analyzing a set of main principles of EaaS systems, following this by defining a collection of features and requirements we consider as “must-haves” for a fully functional EaaS platform. We listed some of the most popular and studied platforms in this domain, with the goal of comparing them according to the collection of features we defined. Following this analysis, we drew the conclusion that Codalab represented the most appropriate open-source baseline solution for developing the AI4media platform.

Next, we presented the prototype we developed at this point. We started by listing a collection of high-level attributes and user-centric functions that the final version of the EaaS platform must have, presenting their development status according to five stages, namely: i) development not started yet, (ii) development started, no tests performed, (iii) prototype development and testing finished, (iv) final development finished, (v) final development and testing finished. Finally, we analyzed a use case scenario, based on the upcoming 2022 ImageCLEFfusion benchmarking competition. This represents a direct applicability for the AI4Media project, as the development of state-of-the-art fusion systems and the study of subjective affective multimedia aspects are part of WP6: Human- and Society-centred AI, being part of T6.6 – Measuring and Predicting User Perception of Social Media. We studied several scenarios that highlight some of the main actions that organizers and participants can take, but also showing how reproducibility can be achieved at this stage of the platform, by implementing an API-based Algorithm-to-Data paradigm.

The next update to this work is deliverable D4.6 ”Final platform for AI dataset benchmarking”, which will be published in M40 of the project, and will present the final version of the Evaluation-as-a-Service platform, featuring the working high-level functions defined in this deliverable.

References

[1] E. Gamma, R. Helm, R. Johnson, R. E. Johnson, J. Vlissides, *et al.*, *Design patterns: elements of reusable object-oriented software.* Pearson Deutschland GmbH, 1995.

[2] D. Phillips, *Python 3 object-oriented programming: Build robust and maintainable software with object-oriented design patterns in Python 3.8.* Packt Publishing Ltd, 2018.

[3] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[4] M. G. Constantin, L.-D. Ştefan, B. Ionescu, N. Q. Duong, C.-H. Demarty, and M. Sjöberg, "Visual interestingness prediction: A benchmark framework and literature review," *International Journal of Computer Vision*, vol. 129, no. 5, pp. 1526–1550, 2021.

[5] M. G. Constantin, L. D. Stefan, B. Ionescu, C.-H. Demarty, M. Sjoberg, M. Schedl, and G. Gravier, "Affect in multimedia: Benchmarking violent scenes detection," *IEEE Transactions on Affective Computing*, 2020.

[6] F. Hopfgartner, A. Hanbury, H. Müller, I. Eggel, K. Balog, T. Brodt, G. V. Cormack, J. Lin, J. Kalpathy-Cramer, N. Kando, *et al.*, "Evaluation-as-a-service for the computational sciences: overview and outlook," *Journal of Data and Information Quality (JDIQ)*, vol. 10, no. 4, pp. 1–32, 2018.

[7] A. Hanbury, H. Müller, K. Balog, T. Brodt, G. V. Cormack, I. Eggel, T. Gollub, F. Hopfgartner, J. Kalpathy-Cramer, N. Kando, *et al.*, "Evaluation-as-a-service: overview and outlook," *arXiv preprint arXiv:1512.07454*, 2015.

[8] A. Hanbury, H. Müller, G. Langs, M. A. Weber, B. H. Menze, and T. S. Fernandez, "Bringing the algorithms to the data: cloud–based benchmarking for medical image analysis," in *International Conference of the Cross-Language Evaluation Forum for European Languages*, pp. 24–29, Springer, 2012.

[9] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in nlp," *arXiv preprint arXiv:1906.02243*, 2019.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[11] T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: A survey," *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 1997–2017, 2019.

[12] R. Desislavov, F. Martínez-Plumed, and J. Hernández-Orallo, "Compute and energy consumption trends in deep learning inference," *arXiv preprint arXiv:2109.05472*, 2021.