AI application upgrade, deployment, and operation plan

iMagine Deliverable D3.3

07/06/2024

Abstract

iMagine is an EU-funded project with the mission to deploy, operate, validate, and promote a dedicated iMagine AI framework and platform, connected to EOSC and AI4EU, giving researchers in aquatic sciences open access to a diverse portfolio of AI based image analysis services and image repositories from multiple RIs, working on and of relevance to the overarching theme of 'Healthy oceans, seas, coastal and inland waters. To effectively achieve the objectives of the project, currently eleven use cases in different areas of aquatic science collaboratively engage with the iMagine AI Platform providers. Of those eleven, there are five so-called 'mature use cases' which are destined to become operational services, offering their services to external users. This document details the planned upgrading of those 5 AI application use cases towards service deployment and operations.

Funded by
the European Union

iMagine receives funding from the European Union's Horizon Europe research and innovation programme under grant agreement No. 101058625.

Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union, which cannot be held responsible for them.

Document Description

D3.3 AI application upgrade, deployment, and operation plan			
Work Package number 3			
Due date	31/04/2024	Actual delivery date:	07/06/2024
Nature of document	Report	Version	1
Dissemination level	Public		
Lead Partner	MARIS		
Authors	Valentin Kozlov (KIT), Gergely Sipos (EGI), Dick Schaap (MARIS)		
Reviewers	Amanda Calatrava (UPV)		
Public link	https://doi.org/10.5281/zenodo.11520846		
Keywords	Production delivery, operational service, upgrading		

Revision History

Issue	Item	Comments	Author/Reviewer
V 0.1	Draft version	Drafting a skeleton for the deliverable, including different models, and questions for use cases	Valentin Kozlov, Gergely Sipos
V 0.2	Revised version	Refining and adding text about training data sets management	Dick Schaap
V 0.3	Revised version	Description of deployment and operation plans for each mature use case	Carolin Leluschko, Jean-Olivier Irisson, Enoc Martinez, Antoine Lebeaud, Catherine Borrenabs-Slegers, Damian Smyth, Igor Atake, Donatello Elia, Marco Mariano De Carlo, Sandro Luigi Fiore, Muhammad Arabi Tayyab, Rune Lagaisse, Wout Decrop

V 0.4	Revised version	Consolidating input for finalising	Dick Schaap
V 0.5	Revised version	Review	Amanda Calatrava
V 0.6	Revised version	Processing review remarks / suggestions	Dick Schaap, Gergely Sipos, Valentin Kozlov
V 0.7	Revised version	Adding summary sections about the use cases and next steps	Gergely Sipos
V 1.0	**FINAL**		

Copyright and license info

Table of content

Figures

Tables

Acronyms

For acronyms used, you are referred to:
https://confluence.egi.eu/display/IMPAIP/Glossary

Executive Summary

This is the third deliverable of a series covering technical aspects of the mature use cases of the iMagine project. While D3.1 and D3.2 addressed the AI model development and training phases, this report details the final stage of the iMagine project, focusing on delivering functional AI image analysis services derived from the project's mature use cases.

After 18 months, these use cases have developed accurate AI models for image classification and object detection in aquatic environments.

This document outlines the process for transitioning these models into production-ready AI applications. Four potential delivery methodologies are presented, allowing each use case to choose the best approach.

Additionally, the report also details a Zenodo-based approach for publishing the training datasets that were used to train these AI models. Data publishing can enable wider scientific research and development and increase the trust in the AI models.

Each mature use case justifies the chosen application delivery model and specifies the training datasets they will publish for broader use within the scientific community. At the end the document summarises next steps to open the services for external users.

Introduction

iMagine, an EU-funded project contributes to the overarching mission of the EU of 'Healthy Oceans, Seas, Coastal and Inland Waters'. It does so by working towards the following project objectives:

- Establish a robust IT infrastructure for image analysis: create and maintain a scalable iMagine AI platform using the AI4OS[1] software stack. Also, improve accessibility through federation to ensure a seamless environment for aquatic researchers (WP4).
- Further, develop existing image analysis services: The aim is to improve research performance and provide virtual access to external researchers (WP3).
- Prototyping of new image analysis services: Development and testing of prototypes on the iMagine AI Platform with a focus on training image labelling, model migration, and validation. The goal is to accelerate progress towards healthier oceans, seas, and coastal and inland waters (WP3).
- Capture and disseminate best practices: Systematically capture best practices from iMagine AI Platform providers (WP4) and use case developers (WP3). Promote the AI platform in the European Open Science Cloud[2] and AI4EU initiatives.
- Provide a portfolio of scientific image and image analysis services aimed at researchers in the marine and aquatic sciences: Develop trained models and FAIR images to create a diverse portfolio of scientific image and analysis services (WP5).

To effectively achieve the objectives of the project, currently eleven use cases (WP3) in different areas of aquatic science collaboratively engage with the iMagine AI Platform providers (WP4). Of those eleven, there are five so-called ‘mature use cases’ which are destined to become operational services, offering their services to external users. These five mature use cases are:

- UC1 Marine litter assessment
- UC2 Zooscan - EcoTaxa pipeline
- UC3 Marine ecosystem monitoring at EMSO sites (OBSEA, Azores, SmartBay)
- UC4 Oil spill detection
- UC5 Flowcam plankton identification

This document serves as a comprehensive report that provides a detailed description of the plans of each of the five mature use cases (UC1 - UC5) for upgrading and deploying their AI applications into production and for making its services operationally available to external users.

[1] https://ai4os.eu/

[2] https://open-science-cloud.ec.europa.eu/

Delivery methodology

Data delivery

Each mature use case, and also the prototype use cases, have generated and used labelled datasets to train their applications. One objective of iMagine is to make these image training data sets available for external users in order to:

- Show the images that were used to train the AI models, increasing users' trust and understanding of those models.
- Enable the reuse of the images for additional use cases, including retraining the models from iMagine, or training additional AI models outside the consortium.

For that purpose, it has been decided that the labelled images used for training all the iMagine use cases (UC1-8) are to be catalogued for download through the Zenodo system. In practice, Zenodo will serve as physical storage for some of these datasets, while for others, Zenodo might only be used as a metadata catalogue gateway for files that are physically stored in institutional repositories. This cataloguing of training data sets will be elaborated in the section on technical implementations.

Next to the labelled image datasets as initially used for the training of the AI models, there will also be images used as input during the operation of the use case services, and these runs will give data results like classified images, recognised types of fishes, plankton, number of counted fishes etc. Users of the use case services should be able to manage these input and output data sets as part of the application service delivery, implicating import, export, storage and possible sharing of both input and output, as to be decided by the users.

Each mature use case has labelled datasets, the so-called 'training image data sets' and these are to be made available for external users in order to:

- Show the images that were used to train the AI models, increasing users' trust and understanding of those models.
- Enable the reuse of the images for additional use cases, including retraining the models from iMagine, or training additional AI models outside the consortium.

Note: These 'training image data sets' are not only to be made publicly available by the mature UC1 - UC5 use cases but also over time by the additional prototype use cases which are developed and tested as part of iMagine.

Europe already has developed an impressive capability for aquatic environmental observation, data-handling and sharing, modelling and forecasting, second to none in the world. This builds upon national environmental observation and monitoring networks and programs, complemented with EU initiatives such as the Copernicus programme (CMEMS) and EMODnet, and European Research Infrastructures (RIs), like SeaDataNet, EurOBIS, EBI-ENA, and others. However, none of these marine data management infrastructures is (yet) dealing with images for providing long-term storage and stewardship. For that reason, as an alternative it has been decided to store and make available the iMagine training image data sets through Zenodo and then in particular under the iMagine community in Zenodo: **https://zenodo.org/communities/imagine-project**.

This choice can be motivated as follows:

- It provides a long-term perspective as Zenodo is guaranteed by the EU as their major service for storage and distribution of EU RTD results
- A DOI is assigned to every publication
- It provides a clear versioning of publications (including datasets)
- There is 50GB is available by default for every publication, and more on-demand
- It tracks and gives Download and View statistics

A drawback is that Zenodo serves as a general repository for a wide range of EU results spanning various domains. Its publication template is primarily tailored to accommodate reports and papers, thus making it less conducive for describing datasets that are substantiated by domain-specific vocabularies. However, iMagine has discussed and agreed with Zenodo - as part of its EU-funded Zenodo-ZEN project to work on a more domain-specific approach. This implies that iMagine will formulate a dedicated template for iMagine training image datasets as a DCAT profile, supported by aquatic vocabularies. While Zenodo will undertake efforts to implement the proposed iMagine template, once agreed, as a special by configuring a dedicated Content Management System (CMS) and later on also an API to facilitate a machine-to-machine exchange of the iMagine Zenodo entries. Establishing this planned solution will take effort from iMagine partners and the Zenodo team and also will have a time factor of circa 6 to 9 months.

Therefore, in the short term, the iMagine use cases will continue to make use of the common existing Zenodo template and editing facility for documenting and sharing their training image data sets. In that case, the use cases are encouraged to include a number of relevant aspects in the free text Zenodo description as well. To describe the training image datasets in the most complete way using the available fields that Zenodo offers, below is the information that should be included:

1. Via the fields that are available in the Zenodo input form.
2. Via the description field, including the additional information that is relevant to training image datasets but that is not covered by the Zenodo input fields.

The following fields (from top to bottom), which are available via the Zenodo input form, should be filled in for each training images dataset:

- "*Select a community*": The iMagine project should be included here.
- "*Upload files*": The training images dataset should be uploaded here as a compressed data package.
- "*Digital Object Identifier*": If the training images dataset has not been uploaded elsewhere, a DOI should be created. If it has been uploaded elsewhere, the DOI of that external location should be mentioned here. In that case, the dataset should not be uploaded again here, but a reference can be made in the description to the external location where the dataset is stored (e.g. SEANOE).
- "*Resource type*": Choose 'dataset'.
- "*Title*": Include a relevant title.

- "*Publication date*": In case your upload was already published elsewhere, use the date of the first publication.
- "*Creator*": Can add multiple persons and organisations. Currently, organisations can be included via ROR, ISNI or GND, and manually as free text. In our next conversations with Zenodo, we can opt to include the EDMO option here as well. EDMO is a directory of marine organisations used in many leading European marine data repositories.
- "*Description*": The additional information that cannot be covered by the Zenodo input fields goes here. For the proposed content, see the proposed listing further below the standard Zenodo fields.
- "*Licences*": Include here the licences for the dataset (it is possible to add a custom licence as well); CC-BY-4.0 is recommended.
- "*Contributors*": Can include persons or organisations (same options as for creators).
- "*Keywords*": Choose relevant keywords.
- "*Languages*": Choose language English.
- "*Dates*": Multiple dates can be added for different uses (e.g. collected, created), in addition to a description.
- "*Version*": Include version of the training dataset.
- "*Funding*": Include funding via the iMagine project using grant agreement ID - 101058625.
- "*Software*": Can include links to software applied on the training dataset.

After analysing the available input fields on Zenodo and comparing them to the general occurring metadata fields in the environmental sciences and machine-learning training dataset repositories, the following iMagine recommendations are given for information to be included in the description field of the Zenodo form. This allows to list the 'missing' metadata information relevant to training images datasets for AI in the aquatic domain using the following structure:

- *"Training dataset (header 1)":* Include a brief explanation of the dataset including its purpose, contents (amount of images) and other relevant information, i.e. in what context the training data set is used.
- *"Technical details (header 1) "*
- *"Data preprocessing (header 2) ":* Details about any preprocessing steps applied to the data, such as for example augmentation.
- *"Data splitting (header 2) ":* Explain how the training dataset is split into training, validation and prediction.
- *"Classes, labels and annotations (header 2) ":* Describe the classes and labels used. If images are annotated, describe the type (bounding boxes, segmentation masks).
- *"Parameters (header 2) ":* Include here the parameters from the training dataset. It is recommended here to include links to vocabularies (e.g. **NVSP01** to express parameters from the BODC Parameter Usage Vocabulary as part of the NERC

Vocabulary Server (NVS). An easy look-up of P01 and possibly other relevant vocabularies is provided through SeaDataNet: https://vocab.seadatanet.org/search

- *"Data sources (header 2) "*: Instrument/gear, sensors, website, database, conditions under which images were captured.
- *"Data quality (header 2) "*: Information about the quality and reliability of the data, including any known limitations or sources of error.
- *"Image resolution (header 2) "*: Dimensions of images in pixels (width x height).
- *"Spatial coverage (header 2) "*: geographic extent covered by the data.
- "*Temporal coverage (header 2)*": time periods as relevant for the data.
- *"Contact information (header 2) "*: Who to contact for questions about the training dataset or the application related to it.

As an example of how a published training dataset might look after including the proposed fields and extra information as listed below, an example is already available: **https://zenodo.org/records/10777441.**

Currently, there are already a number of iMagine training images datasets included in Zenodo in the iMagine community:

https://zenodo.org/communities/imagine-project/records?q=&f=resource_type%3Adataset&l=list&p=1&s=10&sort=newest

However, these will need to be amended and enriched following the instructions as provided above.

Application service delivery

iMagine development activities focused so far on the training of AI models to make them accurate in object classification and image segmentation. Every use case has created a trained AI-model, and their accuracy has been validated. These models are either based on open source third-party models, or on in-house developed models. Both types of models are integrated into the iMagine AI Platform[3] and leverage the DEEPaaS API[4] (REST API for AI/ML/DL). The trained and validated models can be offered and made accessible for external users through the following complementary approaches:

1. **Marketplace inference service delivery**: The models are made accessible through the iMagine Marketplace[5] component of the Platform, which offers the possibility for users to choose and run the trained AI models for inference on the connected back-end cloud resources of the iMagine infrastructure.
2. **Marketplace download service delivery**: The models are made accessible for download through the iMagine Marketplace component of the Platform. This allows

[3] https://www.imagine-ai.eu/services/imagine-ai-platform/

[4] https://docs.ai4os.eu/projects/deepaas/

[5] https://dashboard.cloud.imagine-ai.eu/marketplace

users to download the AI modules as Docker images and run them on in-house/third-party external compute resources.
For the 1st case, the operational responsibility is on the iMagine project, while for the 2nd case, it is on the consumer user/organisation. However, both cases are offered when a model is published on the Marketplace. In addition, the following two approaches could be considered:

3. **Inference service delivery**: The trained AI models are deployed by the project partner on computational resources from the iMagine consortium and are made available for inference execution via APIs that can be invoked from third-party user interfaces. Use cases can choose this option to design their own service presentation layer and model execution scheme (e.g. with Apache OpenWhisk) or leverage the OSCAR[6] open-source platform behind custom Web GUIs for the event-driven serverless execution of the trained model on the iMagine cloud resources. End-to-end operational responsibility in this case is with the iMagine project members but shared among multiple WPs (WP5 for the presentation layer, WP4/WP5 for the execution layer, WP4 for the cloud infrastructure layer).
4. **Retraining service delivery**: Users may request retraining of the already trained AI models with their own data to make the model more precise for the specific classification/prediction cases they are facing. The project partners can offer support for such retraining. Once the retraining is performed by the project partner, a new model is deployed for inference with option 1 - 2 described above (or even 3 if no GUI update is needed). Operational responsibility is with the iMagine project member or the consumer organisation depending on where the retrained module is deployed.

Each option has pros and cons and in dialogue with the use cases the following have been gathered.

Table 1 - Application service delivery options pros and cons

1. Marketplace inference delivery	+Pros: the API interface is simple, standard, and discoverable +Pros: integrated interface with links to specific datasets, publication, source code, user-friendly workflow for users with less coding experience -Cons: user will need an account if they also need a training option next to inference -Cons: Computing resources provided for inference are limited
2. Marketplace download delivery	+Pros: this approach not only allows inference but also retraining the models -Cons: users will need to arrange and manage their own computing resources

[6] https://docs.oscar.grycap.net/

3. Inference service	+Pros: faster to instantiate the model execution for a specific user (then from the Marketplace) -Cons: need for another deployment, separate from the Marketplace. This can rely on the centrally operated OSCAR of iMagine or in-house OSCAR or other serverless platform deployments
4. Retraining service	+Pros: use of EGI GPUs to train +Pros: retraining will allow a wider applicability beyond the local study areas -Cons: requires interfacing the training code also, which is not as easy as the inference part -Cons: iMagine platform may run out of GPU capacity if this is a frequently requested delivery mode

Use case-specific roadmaps

Summary

This section captures which of the use cases will adopt which of the 4 delivery models for application delivery, and their approach for data delivery. The section begins with a summary table that pulls together the key aspects of the delivery across all the use cases.

Table 2 - key aspects of the delivery across all the use cases

Use case		Data delivery approach	Application delivery approach
UC1 Marine litter assessment		● First version is stored in Zenodo ● Updated version expected in Q2 2024	Option 2 and 3 (relying on the project OSCAR instance)
UC2 ZooScan - EcoTaxa pipeline		Dataset is stored in the SEANOE (SEA scieNtific Open data Edition) repository and catalogued in Zenodo.	Option 3 (relying on the project OSCAR instance)
UC3 Marine ecosystem monitoring	UC3o EMSO OBSEA	● Training data to be published on Zenodo ● Live stream to be on Youtube	Option 3 (relying on the project OSCAR instance)

	UC3a EMSO Azores	To be published in SeaDataNet SEANOE (SEA scieNtific Open data Edition)	Initially option 1 and 2, later may also rely on option 3.
		To be published in Zenodo (Underwater Marine species dataset and Nephrops Burrow dataset)	Option 2, running offline on research vessels. Option 1 for initial validations.
UC4 Oil spill detection		Stored in an Elasticsearch DB at the University of Trento, with public access and metadata already in Zenodo.	Option 3 (relying on the project OSCAR instance), with a graphical Web front-end in Streamlit
UC5 Flowcam plankton identification		Training dataset with over 300,000 images is already in Zenodo. To grow this up to 2.2 million (!) during the project.	Aim to use a combination of methods: Option 1, Option 3 (relying on project OSCAR), Option 4

UC1 Marine litter assessment

Description

Use Case 1 aims to develop an operational environment at the iMagine platform for the detection and quantification of plastic litter, floating on the water surface. The service allows users to ingest their drone footage of their area of interest to get back an analysis of the presence and the count of litter items. This information is relevant for multiple stakeholder groups, for instance for non-governmental organisations (NGOs) or monitoring agencies to investigate whether or not set policies show an effect of reduced count of certain litter categories.

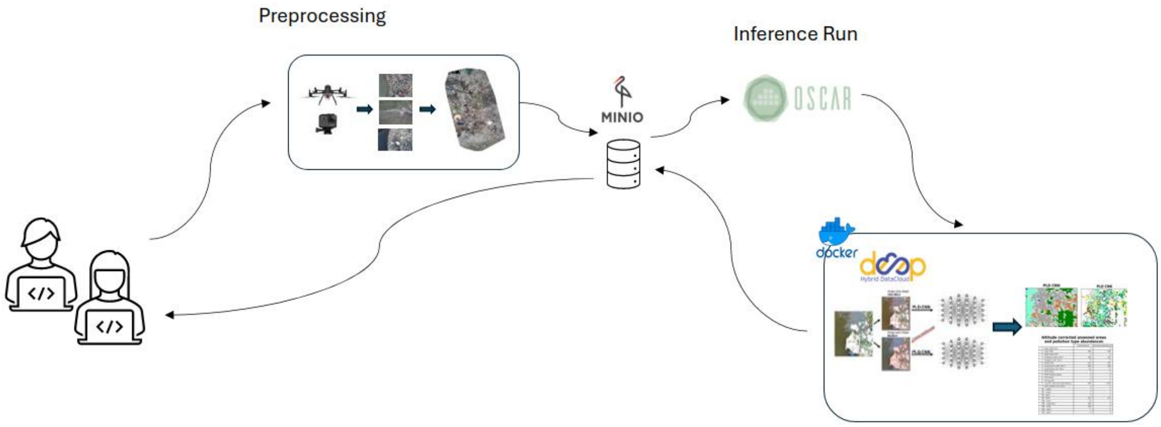

Figure 1 - UC1.1 - High-level architecture of the UC1 image service.

Data delivery

Training data

The initial dataset used for training the model has been made publicly available on Zenodo (**https://zenodo.org/records/4552389**). An upgraded version of the dataset is planned to be published on Zenodo in the near future. The upgrade will consist of an updated contingent of litter categories, as well as a refined tile size for both the detection and the quantification of plastic litter, floating on the water surface. It is also planned to provide the results of a feasibility test for a mapping of the initial litter categories to the Joint List of Litter Categories for Marine Macrolitter Monitoring (JLIST). This test compares the categories resulting from the AI analysis with manual annotations of the same image data.

Service Delivery

In the following, we cover the most important aspects of service delivery and operation:

1. *Which service deployment and serving methodology(ies) do you follow, and why are you choosing this/these methodology(ies)?*
   This service will provide its AI model and the processing methodology on the iMagine Marketplace, so that users can download the docker image and adjust it to their specific setting if needed. This follows the **Marketplace download delivery approach**. In addition to that, the service will be delivered utilising the OSCAR instance operated by iMagine as **inference service to invoke the model from an external application portal**. This way, the end-users can utilise the service with minimal skills required which should minimise the barriers for potential users.
2. *How many end-users per day are expected to use your service?*
   Cannot answer this as end-users per day, as this service is not connected to a continuous data stream. Images are collected and then manually fed into the service.
3. *Do you expect your end-users to go through authentication?*
   No for the Marketplace download option (because the Marketplace redirects to DockerHub for download and does not demand authentication before that.)
   Yes for the OSCAR delivery option because OSCAR needs to distinguish users and offer I/O storage space for each.
4. *Do you have to use a particular AAI (authentication and authorisation interface) (e.g. EGI Check-In) for the end-users?*
   For the OSCAR instance of iMagine, users interacting with the service will need to access through EGI Check-in.
5. *Do you expect anonymous usage of the service?*
   Users can clone the git repository to download the processing methodology and run the inference on their own machines. Git does not track usage statistics relevant for this service. So yes, we do expect anonymous usage of the service.

6. *How much computing power the inference service would require?* The service does not require much computing power. One CPU should be sufficient to run the inference.
7. *How much storage will the inference service require?* The storage capacities for this service highly depends on the amount of images uploaded by the end user. The service itself stores the uploaded images as well as the results during one inference run. In OSCAR, users can define which storage system they want to use (from the supported ones (minIO, S3, ONEDATA, WebDAV-based) both for input and output and the bucket/folder. The service itself does not need to worry about the management of the files. So far, UC1 tested with miniO.
8. *Do you expect a rather flat usage or there can be high-demand periods?* Rather flat stream with potential peaks when a litter recognition campaign collects and uploads all of its data. We don't expect a continuous stream of data.
9. *How do you plan to trigger the inference of AI models (e.g. uploading files, programmatically via an API, etc.?)* Inference is triggered when an image is manually uploaded to the storage system, as we do not expect a continuous stream of images.
10. *What is the expected rate of inference requests (e.g. 20 per hour)?* As this service does not expect a continuous stream of input images, we cannot provide an expected rate of inference requests. Even during the time of use, the user can upload different numbers of images.
11. *What support channel do you want to offer to the user? (Helpdesk, email?)* An email address will be provided to the end users.
12. *What service usage statistics do you need/must collect?*
    a. *Number of unique users of the AI image processing service*
    b. *Number of images processed per year*
    c. *Number of images ingested*
    d. *Number of countries of users*
    e. *Names of countries reached*
13. *How will you deal with the user input and output data? Will it be open to other users or only accessible to the actual user?*
    The resulting analysis of the inserted drone images will be provided for the end user in two formats. To get a visual representation of the classification result, the user can request a plot which shows the different classification categories distributed over the original input image. To get a more detailed and quantitative understanding of the total number of items per litter category, the user will additionally have the option to request a .csv file, containing the total counts per classification category.

UC2 ZooScan – EcoTaxa pipeline

Description

This service accelerates the processing of images taken with the ZooScan instrument until their upload on the EcoTaxa platform. The ZooScan is a waterproof scanner dedicated to zooplankton samples, invented by the Laboratoire d'Océanographie de Villefranche (LOV) and commercialised by a French SME; over 300 units have been sold to research laboratories and environment assessment companies worldwide. EcoTaxa is a web application that stores images and metadata associated with one individual imaged object and uses AI to accelerate their labelling; it currently hosts over 400M images from over 700 organisations worldwide.

The service is a complete rewrite of the existing ZooProcess software, using modern libraries. It adds an AI component to sort and then separate objects touching each other on images. Currently, this is done manually to ensure that only individual objects make their way into EcoTaxa and this is a time-consuming process (1.5 hours per day and per operator; there probably are 50 to 100 persons operating a Zooscan every business day).

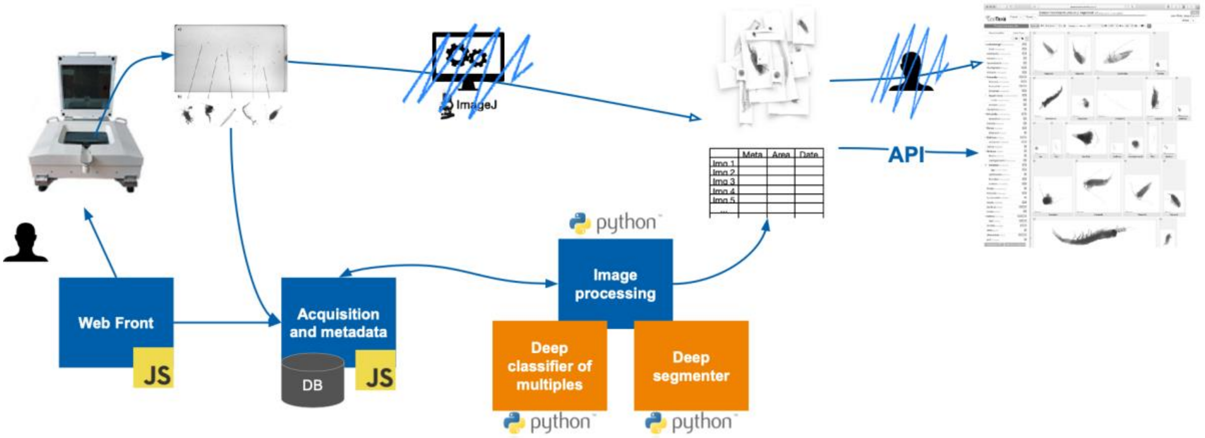

Figure 2 - UC2.1 - High-level architecture of the UC2 image service. The old software based on ImageJ is replaced by a modular web application, with a user-facing frontend and two backends (blue boxes): one for metadata handling (written in JavaScript) and one for image processing (written in Python). The image processing back end interfaces with the two AI modules (orange boxes). The final import into EcoTaxa, which used to be done manually, is done programmatically, through the API.

Data delivery

Training data

The training data for the classifier are images of individual planktonic objects versus multiple planktonic objects. This allows training a binary classifier that separates multiple from single plankters. Then, the images labelled as multiple are sent to the segmenter. For this, the training data are images of multiple plankton with hand-drawn separation lines to turn them into single objects. The goal of the segmenter is to reproduce this manual separation action. Its output is a binary mask with lines separating the original objects. The training data is delivered through SeaNoe: https://www.seanoe.org/data/00885/99663/

With a DOI: 10.17882/99663

It contains the information required in the template described above. A link to it is made from Zenodo so that the dataset is included in the iMagine community: https://zenodo.org/records/11108274.

Service Delivery

Below we clarify the service delivery and details of the service operation:

1. *Which service deployment and serving methodology(ies) do you follow, and why are you choosing this/these methodology(ies)?*
   We will be mostly using the **inference service** because it provides fast and scalable inference capabilities, and this is what we need for this use case.
2. *How many end-users per day are expected to use your service?*
   Initially, <10. Eventually 50 to 100 if all users go through the iMagine service rather than a local computing node. This will depend on the possibility of sustaining the service after the end of the project.
3. *Do you expect your end-users to go through authentication?*
   No, users will authenticate within our software pipeline and then the pipeline itself will send the data through the services. We would like to avoid forcing the users to register on a service authenticated with another set of credentials; this would be confusing.
4. *Do you have to use a particular AAI (authentication and authorisation interface) (e.g. EGI Check-In) for the end-users?*
   No.
5. *Do you expect anonymous usage of the service?*
   Yes, from the point of view of the users. We could still track the usage stats by authenticating with a given set of credentials from our software, if need be.
6. *How much computing power the inference service would require?*
   A few minutes of GPU time, up to 30 minutes on CPU for a few hundred segmented images originating from a single image acquisition.
7. *How much storage will the inference service require?*
   Almost none: images are streamed in, and results are streamed out. The only storage will be for the container and model themselves.
8. *Do you expect a rather flat usage or there can be high-demand periods?*
   Probably homogenous usage through time with some periods more active than others. But it takes ~20 minutes to take the image which is then processed on the service in ~2 minutes (on GPU) so it will never generate extreme peaks of activity.
9. *How do you plan to trigger the inference of AI models (e.g. uploading files, programmatically via an API, etc.?)*
   Via the same API as the DEEPaaS.
10. *What is the expected rate of inference requests (e.g. 20 per hour)?*

3-4 acquisitions per working hour inducing a few hundred to thousands of individual images per working hour; that would amount to minutes of GPU time per hour.

11. *What support channel do you want to offer to the user? (Helpdesk, email?)*
    Email.
12. *What service usage statistics do you need/must collect?*
    Number of users, number of countries, number of acquisitions processed.
13. *How will you deal with the user input and output data? Will it be open to other users or only accessible to the actual user?*
    The user input and output data are his sole property and will not be accessible to other users. But we will likely increase the size and diversity of the training set in the future, and this will be made public as a new version of the existing training dataset.

UC3o Marine ecosystem monitoring at EMSO OBSEA

Description

Within UC3o two services are envisioned: a) fish abundance estimation and b) real-time fish detections. The target users of the first service are scientists and biologists aiming to perform biodiversity analysis and ecosystem studies. The second results annotated live stream videos and targets marine scientists as well as the general public for dissemination and outreach purposes.

The fish abundance estimation service aims to produce datasets covering all of the OBSEA underwater observatory video. Furthermore, these datasets will be updated in real-time as more and more data comes in. The real-time data will be visible at EMSO-OBSEA cyber-infrastructure, and the datasets will be periodically updated on the Zenodo repository. The datasets will comprise the original picture as well as time-series of fish detections classified by species. Thus, the produced datasets will include underwater pictures (jpeg or png formats) and time series with fish detections by taxa (csv or NetCDF formats). It is envisioned to produce one picture per minute for every deployed camera (usually up to 3) during daylight hours.

The fish detection streaming service will be used mainly as a dissemination tool to raise awareness about marine biodiversity among the general public. The fish detections will be embedded in the video stream and published on YouTube to simplify access to the end users.

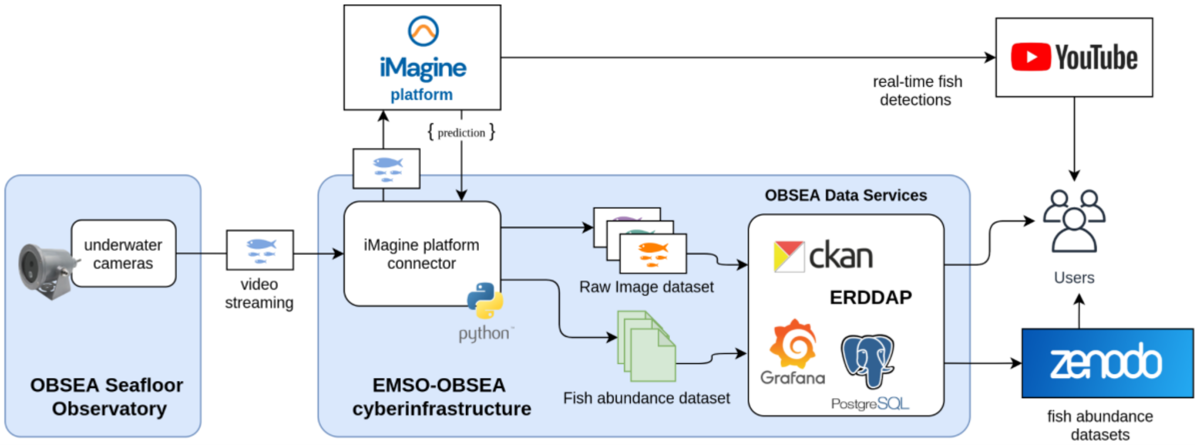

Figure 3 - UC3o.1 - High-level architecture of the UC3o image services. Underwater cameras are streaming video in real-time to the EMSO-OBSEA cyber-infrastructure, where the stream is separated into real-time video and periodic pictures (one per minute). Both images and pictures are sent to the iMagine platform for inference

Data delivery

Training Dataset

The training dataset as well as the generated scientific datasets (both fish detections and pictures) will be made publicly available at Zenodo under CC-BY licence. The source code will also be published at Github repository under MIT licence.

Service Delivery

Below we cover the details of service delivery and operations:

1. *Which service deployment and serving methodology(ies) do you follow, and why are you choosing this/these methodology(ies)?*

   For picture inference we are programmatically doing inference using an API. The API proves easy to integrate and is adequate for the workload of these services.

   However, we also need the project to add a new feature into OSCAR to enable reanalysis of historical data with improved/alternative models. The feature would allow the re-execution of an analysis

   For the video inference we are relying on a Kafka streaming service to deliver a high volume of real-time video data. Kafka is a robust and mature technology widely used for streaming large volumes of data.
2. *How many end-users per day are expected to use your service?*
   For the fish abundance datasets, we expect tens of users per year. For the real-time fish detections, we expect hundreds of unique views.
3. *Do you expect your end-users to go through authentication?*
   No, for usage we will rely on Zenodo and YouTube usage statistics.

4. *Do you have to use a particular AAI (authentication and authorisation interface) (e.g. EGI Check-In) for the end-users?*
   The AI model will be run for inference for a pre-defined EMSO station which acts as its single user. Therefore, we don't need sophisticated AAI. The secondary users will be researchers who use the fish statistics produced from the inference (from Zenodo), and citizens who watch the annotated live feed (from Youtube).
5. *Do you expect anonymous usage of the service?*
   See previous point.
6. *How much computing power the inference service would require?*
   For picture inferences we expect to have a maximum of 3 or 4 inferences every minute with 4k images. A single GPU should be more than enough for it. Regarding the video inference we require another GPU for video inference in real-time.
7. *How much storage will the inference service require?*
   A few GBytes to handle input images. Occasionally the whole picture archive may be re-analyzed with new AI models. In these rare cases with 100-200 GB should be enough.
8. *Do you expect a rather flat usage or there can be high-demand periods?*
   Generally, we expect flat usage for real-time video/picture inference. In very rare occasions (new model deployments) the reanalysis of all the image archive may produce a high-demand period.
9. *How do you plan to trigger the inference of AI models (e.g. uploading files, programmatically via an API, etc.?)*
   For the picture inference, we run the inferences programmatically with an API.
   For the video inference we expect the stream to be running continuously. The stream will be delivered by a Kafka streaming service and sent to YouTube.
10. *What is the expected rate of inference requests (e.g. 20 per hour)?*
    For picture inference up to 4 per minute (240/hour). For video, we expect to run a lightweight AI model on a real-time video stream at 24 frames per second.
11. *What support channel do you want to offer to the user? (Helpdesk, email?)*
    Direct contact via email.
12. *What service usage statistics do you need/must collect?*
    Number of images generated/processed, dataset downloads, real-time video views and number of countries reached.
13. *How will you deal with the user input and output data? Will it be open to other users or only accessible to the actual user?*
    The generated scientific datasets (both fish detections and pictures) will be made publicly available at Zenodo under a CC-BY licence.

UC3a Marine ecosystem monitoring at EMSO Azores

Description

The "Ecosystem monitoring at EMSO sites by video imagery" use case aims to establish an operational and integrated service for the automatic processing of video imagery, collected by cameras at EMSO underwater sites, identifying and further analysing interesting images for purposes of ecosystem monitoring. In this context, citizen's annotations from Deep Sea Spy were prepared to train a CNN-based object detection algorithm to automatically identify species present on the images. This will result ultimately in a user-friendly pipeline including the cleaning of citizen science data and Yolov8 network training or inference.

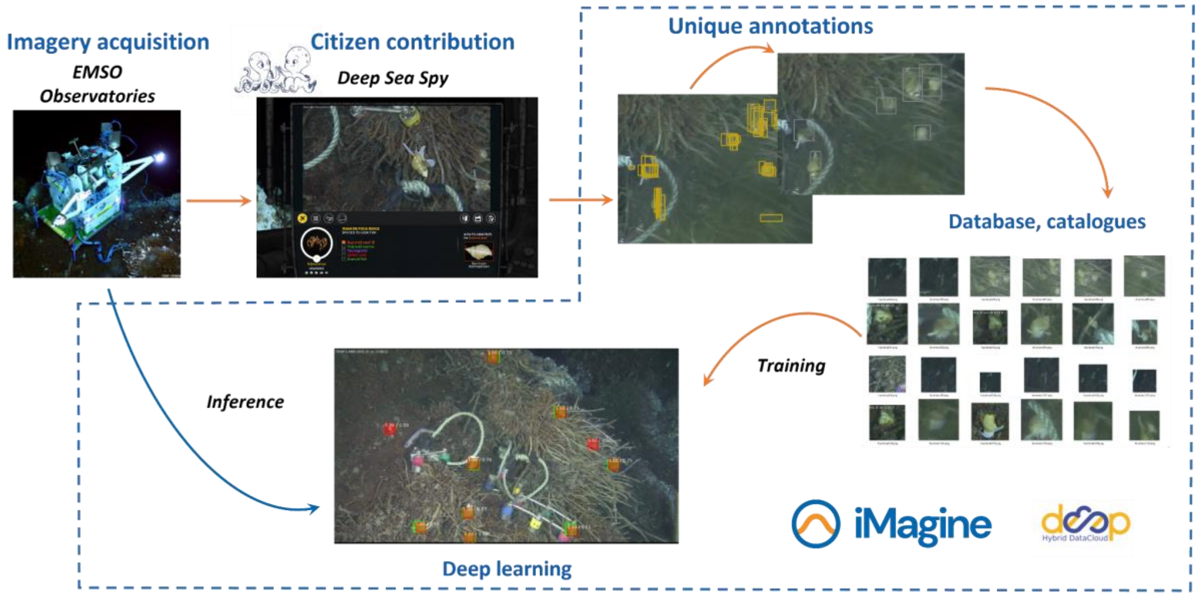

Figure 4 - UC3a.1 - High-level architecture of the UC3a image service.

Data delivery

Training dataset

The dataset includes 3,156 images distributed across 15 classes, with a total of 235,323 annotations. The training dataset is a cleaned version on the buccinid class since it's the species that is the most accurately labelled (3,156 images, 17,128 annotations). More classes will be integrated as a second step. The entire raw dataset and the training dataset will be made available on the SeaDataNet SEANOE (SEA scieNtific Open data Edition) repository with a DOI assigned. A link will be made from Zenodo so that the dataset will be included in the iMagine community.

Service Delivery

In the following, we will detail the specifics of service delivery and operational processes:

1. *Which service deployment and serving methodology(ies) do you follow, and why are you choosing this/these methodology(ies)?*

We will firstly use the marketplace as a service provider, for our data preparation pipeline (jupyter notebook) and a pre-trained YOLOv8 model on our dataset (Zenodo/seanoe) (**Marketplace inference delivery** and **Marketplace download delivery).** This way, users can freely use which service they need from our work, locally or on the iMagine resources. We will consider the **inference service delivery** for the deployment of our model. We don't plan to include a retraining service.

2. *How many end-users per day are expected to use your service?*
   Flat usage, only certain periods of activity is expected, not daily.
3. *Do you expect your end-users to go through authentication?*
   We deemed it not necessary for our use case.
4. *Do you have to use a particular AAI (authentication and authorisation interface) (e.g. EGI Check-In) for the end-users?*
   No, not specifically.
5. *Do you expect anonymous usage of the service?*
   Yes.
6. *How much computing power the inference service would require?*
   It depends. If only the data filtering and cleaning are needed, then we only need a CPU (relatively good one depending on the size of the raw dataset) and RAM. For inference, one GPU should be enough (1h30 for a full training on our dataset).
7. *How much storage will the inference service require?*
   It depends on the dataset size from the user and the service's usage. If the user wants to visualise and generate thumbnails, the used storage can rise rapidly. Our dataset is roughly 1GB. We think, as the pipeline and the inference can generate images, 5GB of data might be the minimum for allowing different tests with our service. So, the space needed might be 5x the size of the user's dataset.
8. *Do you expect a rather flat usage or there can be high-demand periods?*
   Flat usage.
9. *How do you plan to trigger the inference of AI models (e.g. uploading files, programmatically via an API, etc.?*
   The DEEPaas API should be used for inference of AI models. The data preparation pipeline doesn't need AI to run.
10. *What is the expected rate of inference requests (e.g. 20 per hour)?*
    Not easy to assess at this stage, likely low inference requests
11. *What support channel do you want to offer to the user? (Helpdesk, email?)*
    Email.
12. *What service usage statistics do you need/must collect?*
    Mainly the number of visits, unique users, list of countries, and number of processed images.
13. *How will you deal with the user input and output data? Will it be open to other users or only accessible to the actual user?*

The user's data is considered as a private property, and thus won't be available to other users.

UC3s Marine ecosystem monitoring at EMSO SmartBay

Description

The EMSO-SmartBay underwater platform, when operational, streams real-time video footage and records video and imagery to an Archive along with standard oceanographic chemical and physical parameters. The EMSO-SmartBay site has 4 use cases looking at Machine Learning approaches to Video Quality Assessment in real-time and Archive video footage, to identify periods of poor and good quality video footage, video Marine Underwater Species Detection from video and imagery using YOLOv8.

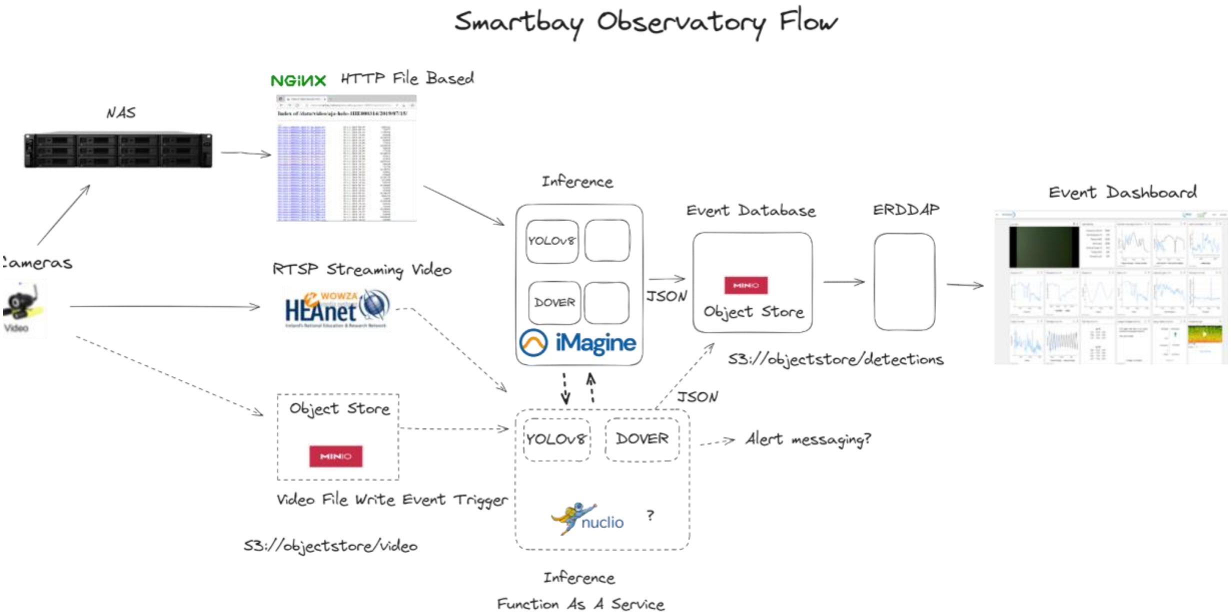

Figure 5 - UC3s.1 - High Level Architecture for EMSO-SmartBay Video Quality Analysis and Species detection UC3s.E1.US1, UC3s.E1.US2, UC3s.E1.US3.

EMSO-SmartBay is also looking at the usage of Machine learning to assist Nephrops (Prawn) Burrow detection and enumeration in fisheries surveys carried out by the Marine Institute in Irish waters.

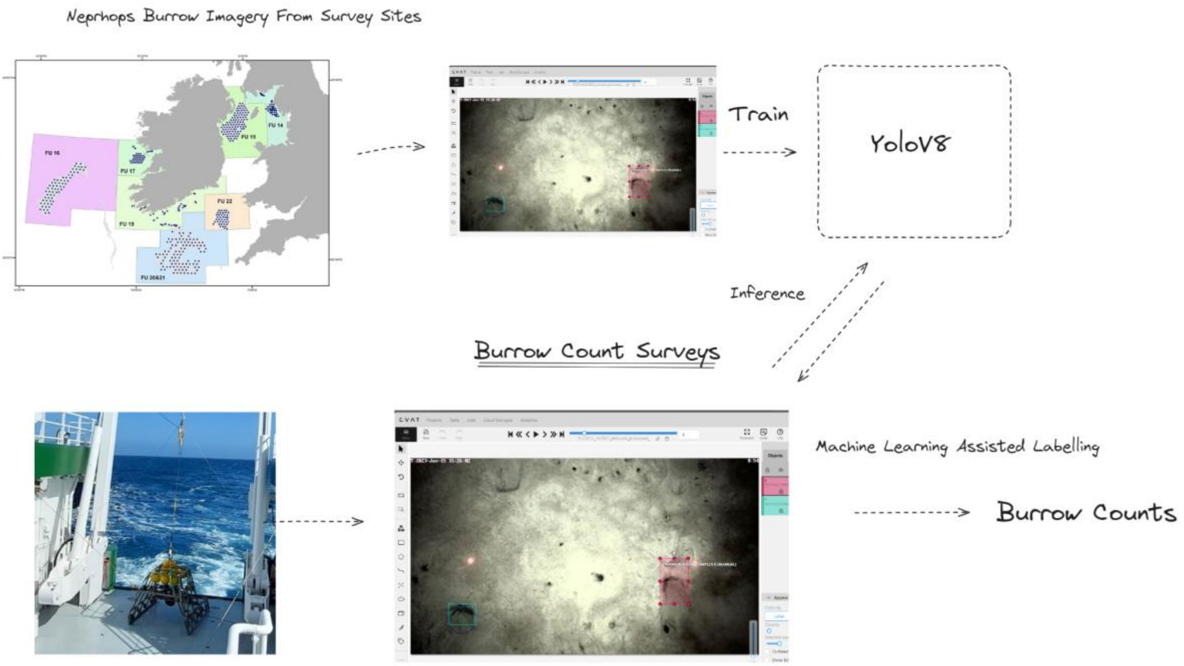
Neprhops Burrow Imagery From Survey Sites
Train
YoloV8
Inference
Burrow Count Surveys
Machine Learning Assisted Labelling
Burrow Counts

Figure 6 - UC3s.2 - High-level architecture of the UC3s.E1.US4 EMSO-Smartbay Nephrop burrow count survey image service - Locally deployed CVAT + Nuclio Yolov8 Model assisted burrow counting object detection and tracking

Data delivery

Training data set

The EMSO-SmartBay Observatory labelled Underwater Marine species training dataset will be made available on Zenodo.

The EMSO-SmartBay Nephrops Burrow Image training dataset will be made available on Zenodo.

The EMSO-SmartBay Video Quality Assessment with the DOVER VQA model is currently not expected to produce a training dataset; However, if it requires further tuning, any resulting training dataset would be uploaded to Zenodo.

Service Delivery

Up next, we outline the essential elements of service delivery and operation:

1. *Which service deployment and serving methodology(ies) do you follow, and why are you choosing this/these methodology(ies)?*
   The "Download image and run in house" from the iMagine Market methodology would be the best fit for the EMSO-SmartBay use cases (**Marketplace download delivery)**. The Models will be run in-house as part of a processing pipeline. However, we will also investigate running the models online with the iMagine platform acting as the inference platform (**Marketplace inference delivery)**.
   The Nephrops burrow model will most likely be run "Locally" on a Research Vessel whilst at sea, as internet connectivity may not be consistent enough.

2. *How many end-users per day are expected to use your service?*
   It is intended that the EMSO-Smartbay VQA and Species detection models would be used daily, by 2 separate processes.
   The EMSO-SmartBay Nephrops Burrow detection model would be used during survey and post survey. Usage would be sporadic and associated with Survey times.
3. *Do you expect your end-users to go through authentication?*
   We would be expecting a couple of processes to use the models, it may make sense if inference is carried out on the iMagine platform to implement authentication and access control, given the limited GPU resources available.
4. *Do you have to use a particular AAI (authentication and authorisation interface) (e.g. EGI Check-In) for the end-users?*
   It is expected that most of the processing pipeline will run locally as a service with authentication or application identities or keys in place. If we deploy inference on the iMagine platform, it could use EGI Check-In or an Application ID and a key associated with a user if necessary, to facilitate calls from a pipeline.
5. *Do you expect anonymous usage of the service?*
   We would expect interest in the trained models, and the underlying training datasets, The data may be of quite local interest to Western Europe.
   There may be interest in anonymous access depending on the accessibility.
6. *How much computing power the inference service would require?*
   Possibly minimal, CPU or small single GPU maybe sufficient for Inference on the iMagine platform.
7. *How much storage will the inference service require?*
   Assuming minimal storage required, possibly < 30GB if it is purely inference with a low number of users. It would be expected that any uploaded imagery or video would no longer be needed after Inference has been carried out and the output results are returned to the user.
8. *Do you expect a rather flat usage or there can be high-demand periods?*
   ESMO-Smartbay would expect rather flat usage in the scenario where we are using just "real-time" video and images from our observatory camera, if our pipeline results in a single process submitting imagery or video and the results are stored and made available elsewhere, e.g. on-premises database, then this would naturally limit the creation of spikes, as most of the work would be done by a single process as the imagery becomes available.
   A more intense usage scenario may arise if multiple parallel processes examine a historical archive, but this may only be practical for local inference.
9. *How do you plan to trigger the inference of AI models (e.g. uploading files, programmatically via an API, etc.?)*
   It is anticipated that file write events to an object store to trigger the submission of the file to a ML function, or a message queue or job list submitted to a function or the analysis of a real time RTSP video feed. In the Nephrop Burrow counting use

case inference may be triggered from the CVAT annotation environment on loading an image.

10. *What is the expected rate of inference requests (e.g. 20 per hour)?*
    At least 30 videos (2-minute videos) per hour for smartbay species detection, maybe also 30 x 2-minute videos per hour for Video Quality Assessment.
    Nephrop burrow counting would be sporadic but could be > 600 images per hour local inference.
11. *What support channel do you want to offer to the user? (Helpdesk, email?)*
    Helpdesk or Email whichever is deemed the most appropriate medium.
12. *What service usage statistics do you need/must collect?*
    It would be interesting to see standard metrics number of downloads of Model image, per country etc. If inference is made publicly available, no. of users, countries etc. volumes of user submitted imagery (uploaded data for inference etc.).
13. *How will you deal with the user input and output data? Will it be open to other users or only accessible to the actual user?*
    It is anticipated that any image or video data submitted to any of our inference services running on the imagine platform will not be kept on the iMagine platform, the resulting output will not be captured and stored on the platform itself either. The resulting Json or image output predictions will simply be returned to and only available to the actual user's calling process. Separately where our "on-premises" workflows call out to either an inference service running "on-premises" or on the iMagine Platform, the resulting Json and/or image data prediction output will be captured and stored locally in an object store or database or in the case of CVAT using a model for inference and semi-automatic annotation stored in its annotations database.

UC4 Oil spill detection

Description

UC4 aims to deploy an operational oil spill forecasting system with pre-trained data on past observed events, as well as the possibility to navigate through satellite-based oil spill observations, through ElasticSearch. Users will be able to access past oil spill events and observe their results in the platform and also be able to perform simulations on their own, provided they can upload georeferenced oil spill area files. The system will be accessible through a simple user interface, that currently we aim for it to be Streamlit, so a less experienced user can perform simulations on their own without the need for extensive computing skills.

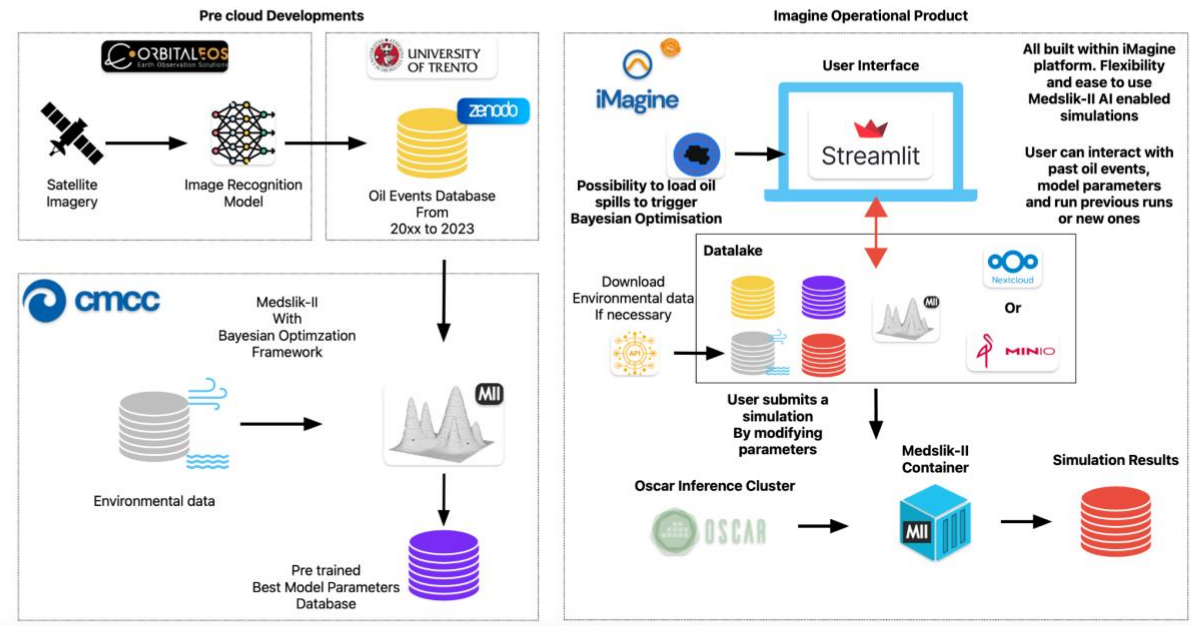
Pre cloud Developments
ORBITALEOS
Satellite Imagery
Image Recognition Model
UNIVERSITY OF TRENTO
zenodo
Oil Events Database From 20xx to 2023
cmcc
Medslik-II With Bayesian Optimzation Framework
Environmental data
MII
Pre trained Best Model Parameters Database
Imagine Operational Product
iMagine
User Interface
Streamlit
All built within iMagine platform. Flexibility and ease to use Medslik-II AI enabled simulations
User can interact with past oil events, model parameters and run previous runs or new ones
Possibility to load oil spills to trigger Bayesian Optimisation
Datalake
Download Environmental data If necessary
Nextcloud
Or
MINIO
User submits a simulation By modifying parameters
Medslik-II Container
Simulation Results
Oscar Inference Cluster
OSCAR

Figure 7 - UC4.1 describes the whole procedure, starting from the image acquisition performed by Orbital EOS, by the database preparation provided by University of Trento and, finally, the Bayesian Optimization Framework and the optimised oil spill simulations performed by CMCC. UC4.1 - High-level architecture of the UC4 oil spill inference system. The plan is to provide a simple user interface in which simulations can be launched either from the historical database provided or let the user provide the system with oil spill shapefiles of his own and then proceed to use the Bayesian Optimization Workflow. Results can be observed or downloaded from the platform. We aim to use the OSCAR cluster as our distributing job to execute the inferences. The whole system only relies on CPU's.

Data delivery

Training dataset

The data catalogue consists of 324 oil spill events that took place from 2019 to 2023 over 32 countries worldwide. The event data comprises information like date, time, name of satellite from where the event was observed, spill class (i.e. mineral oil, most likely mineral oil, less likely mineral oil), spill typology (oil platform, unknown, ship, oil seep, pipeline, shipwreck), total area and average area covered etc.
An Elasticsearch database has been used to index the observational data and provide search and discovery capabilities. It is deployed on a server hosted at the University of Trento (imagine.disi.unitn.it). The dataset has been made available via Zenodo by OrbitalEOS (DOI: 10.5281/zenodo.11354663).

The database consists of the following indices:

1. Catalog_statistics

2. Heatmap_points

3. Model_statistics

4. Shapefile

The catalog_statistics index has 6 fields containing general information about the catalogue namely spill_number, total_area, average_area, sensor_mapping, event_class_mapping, source_type_mapping. The last three fields contain the associations between specific codes and the labels. The heatmap_points index contains the data used to generate the heatmap, in particular, a lot of points with the following fields: lon (longitude), lat (latitude) and spills. The spills contain all the general data about the observation contained in the point defined by the latitude and longitude. Similarly to the catalogue statistics, some general data about the simulation model. The shapefile index is a complex one. It contains all the shapefile data about the observations. Further description of the indices with the relevant fields can be found in tables 3 – table 6.

Table 3 - Catalog_statistics

Field Name	Data type	Description
Spill_number	integer	Number of spills
Total_area	float	Total area covered
Average_area	float	Average spills area
Sensor_mapping	nested object	Represent satellite label name and number of spills detected by that satellite
Event_class_mapping	nested object	Represent event class label name and no. of spills detected by that event class. It describes the origin of the spill
Source_type_mapping	nested object	Represent source type label name and number of spills detected by that source type. It describes the origin of the spill

Table 4 - Heatmap_points

Field Name	Data type	Description
point	geo_point	Represent the latitude and longitude i.e. Point (lat, long)
spills	object	It contains the count of spills in that point

Table 5 - Model Statistics

Field Name	Data type	Description
Simulations_number	integer	Number of generated simulations
Average_skill_score	float	Average skill score of the simulations
Spills_with_skills_score	integer	Spills with skill score

Table 6 - Shapefile

Field Name	Data type	Description
Identifier	keyword	Represent the spill identifier
Author	keyword	Represent the author of the spill
ImageUrl	keyword	Represent logo of the author
Done	boolean	Orbital OES convention data
value	object	Represent type of spill (Polygon or MultiPolygon), event class of the spill, satellite that detected the spill, minimum volume, maximum volume, source type, sheen area, acquisition date and time of the spill, centre of the spill and area of the spill
geometry	geo_shape	It contains all the geometry structure that defines the border of the spill.

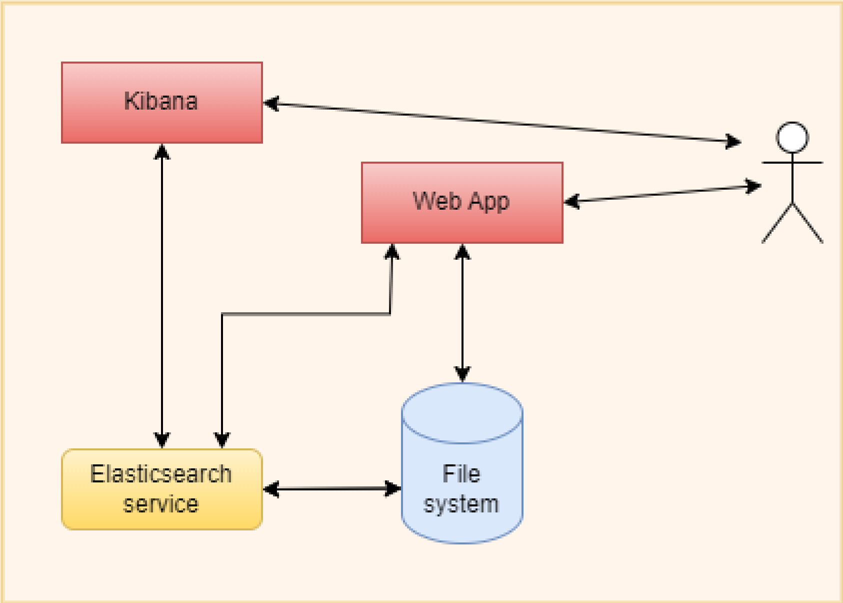

Figure 8 - UC4.2: High level architecture diagram of system, describing the interaction between the system and end user.

The above diagram visually represents the components of the system deployed at the University of Trento server. The system includes web application, file system, Elasticsearch, Kibana and their interactions. In this architecture, the web application provides an interface for user interaction and allows users to request, analyse, and visualise data. Kibana provides an analytics platform and interacts with Elasticsearch for data retrieval. Elasticsearch database stores oil spill data in indices, interacts with the file catalogue and serves data to both web application and Kibana. The file catalogue stores data files (i.e. shape files), accessible by Elasticsearch for data ingestion. Users can access and interact with the system through both the web application and Kibana, depending on their needs and preferences. Overall, this architecture enables users to request, analyse, and visualise oil spill data efficiently using a combination of web applications, Elasticsearch database, and Kibana.

Service Delivery

In the following, we clarify the most important aspects of service delivery and operation:

1. *Which service deployment and serving methodology(ies) do you follow, and why are you choosing this/these methodology(ies)?*
   We plan to use the OSCAR inference cluster to execute the requests asked by end users (**Inference service with OSCAR)**. The inference usually does not require an intensive computation and rare cases will need the Bayesian optimization workflow, given the difficult nature of providing oil spill imagery into the system.
2. *How many end-users per day are expected to use your service?*
   It should vary from time to time. Currently, we have fewer users than other periods, and we expect around 2 users per day as a medium value.

3. *Do you expect your end-users to go through authentication?*
   Not necessary, we have many users using our system nowadays without the need to subscribe.
4. *Do you have to use a particular AAI (authentication and authorisation interface) (e.g. EGI Check-In) for the end-users?*
   No.
5. *Do you expect anonymous usage of the service?*
   Yes.
6. *How much computing power the inference service would require?*
   One CPU node per request. In the case of Bayesian Optimization workflow, it would be good to have more CPU available. The whole process nowadays takes approximately 2 hours on CMCC HPC.
7. *How much storage will the inference service require?*
   No more than 1 GB including both input/outputs.
8. *Do you expect a rather flat usage or there can be high-demand periods?*
   Demand is usually flat; however, in periods in which impacting oil spill occurs, we expect more users to be curious about it and access the platform.
9. *How do you plan to trigger the inference of AI models (e.g. uploading files, programmatically via an API, etc.?)*
   Accessing through the interface as reported in Figure UC4.1.
10. *What is the expected rate of inference requests (e.g. 20 per hour)?*
    1 per hour (depending on users' requests).
11. *What support channel do you want to offer to the user? (Helpdesk, email?)*
    We will provide the service email.
12. *What service usage statistics do you need/must collect?*
    a. Number of unique users of the AI image processing service
    b. Number of countries of users
    c. Names of countries reached
13. *How will you deal with the user input and output data? Will it be open to other users or only accessible to the actual user?*
    Each user should have its own files and should not interact with other users' files. Users can add shapefiles into the interface to perform on demand simulations; however, only their submission should be available in the system for visualisation, other than the pre-trained dataset provided by Orbital EOS. The platform will provide access to output (forecast) of the Medslik model related to simulations performed considering as input the set of oil spills provided by OrbitalEOS and published on Zenodo.

UC5 Flowcam plankton identification

Description

The aim is to establish an iMagine platform service for processing FlowCam[7] (Sieracki et al., 1998) images to determine the taxonomic composition of phytoplankton samples. FlowCam is a high throughput imaging device distributed by FluidImaging with several dozen instruments currently in use in Europe. It allows fast and repeatable acquisition of suspended cells and has been used in the monthly monitoring of phytoplankton in the past 6 years in the Belgian Part of the North Sea. The objectives of this service include setting up an operational environment for users to reuse pre-trained models, refining the AI tools for taxonomic identification, and improving the FAIRness of the full image library data as well as sampled training sets. So far the pipeline of the FlowCam module is up and running via the iMagine platform, and a pre-trained model and annotated training dataset are published.

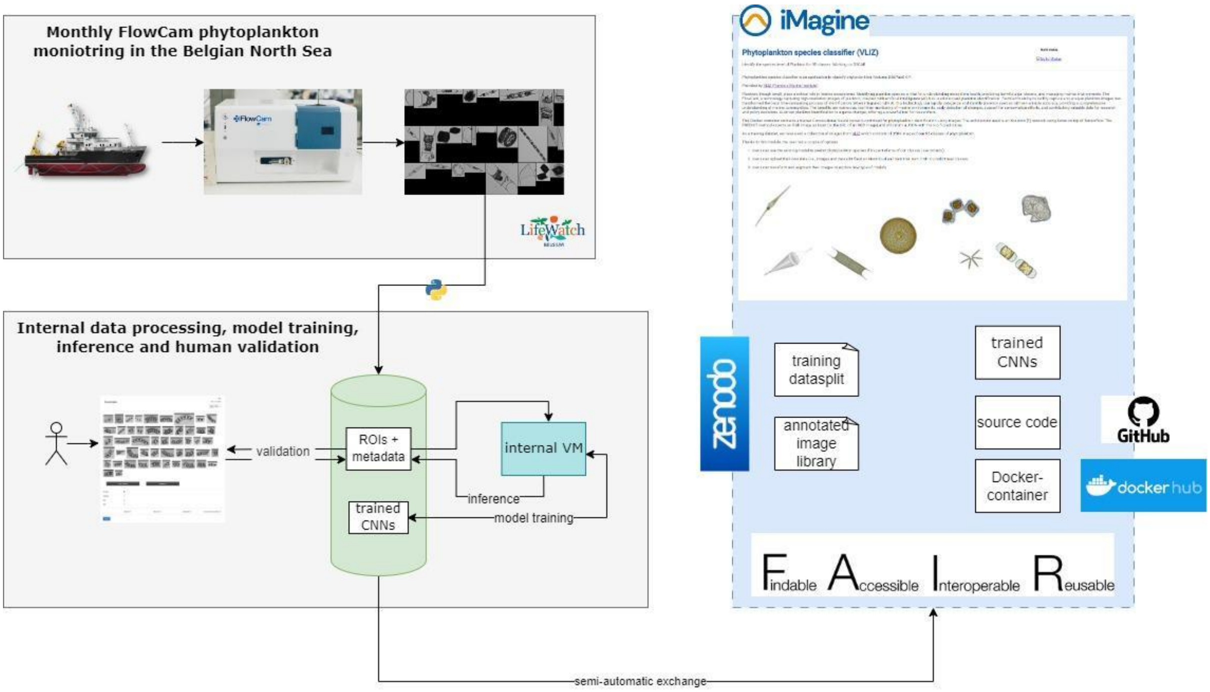
Monthly FlowCam phytoplankton moniotring in the Belgian North Sea
LifeWatch
Internal data processing, model training, inference and human validation
validation
ROIs + metadata
trained CNNs
internal VM
inference
model training
iMagine
Phytoplankton species classifier (VLIZ)
zenodo
training datasplit
annotated image library
trained CNNs
source code
Docker-container
GitHub
docker hub
Findable Accessible Interoperable Reusable
semi-automatic exchange

Figure 9 - UC5.1 - High-level architecture of the UC5 phytoplankton identification service.

Data delivery

Training dataset

This training dataset comprises a data split of 337,613 images distributed across 95 classes, with each class containing a minimum of 100 and a maximum of 10,000 images. The goal of this dataset is to be able to facilitate model training, here we have organised the data into a standard split, with 80% allocated for training, 10% for validation, and another 10% for testing purposes. The full annotated image library of over 2.2 million

[7] https://www.fluidimaging.com/

images will be made available later this year in a separate repo and will also include additional metadata regarding sampling location and datetime.

The training data has been uploaded on Zenodo and can be accessed through the following link: **https://doi.org/10.5281/zenodo.10554845.**

Service Delivery

Below, we provide details of service delivery and operational aspects:

1. *Which service deployment and serving methodology(ies) do you follow, and why are you choosing this/these methodology(ies)?*

   We aim to use a combination of the methods, each for different purposes:
   - 1. Use the **Marketplace Inference Service** to perform inference.
   - 2. OSCAR **Inference Service**: an easy-to-use system to perform inference and which will allow us in the future to track the necessary usage metrics.
   - 3. Train their own CNN:
     - Big project: call in to use Imagine Marketplace?
     - Small project: train on their own server based on docker or Github code and can ask help from us.
2. *How many end-users per day are expected to use your service?*
   - Flat usage, only a certain period of activity is expected, not daily.
3. *Do you expect your end-users to go through authentication?*
   - Through EGI authentication service.
4. *Do you have to use a particular AAI (authentication and authorisation interface) (e.g. EGI Check-In) for the end-users?*
   - EGI authentication service.
5. *Do you expect anonymous usage of the service?*
   - Implementing source code published on GitHub to train on third-party infrastructure.
6. *How much computing power the inference service would require?*
   - 1 CPU should be sufficient.
7. *How much storage will the inference service require?*
   - 1-5GB
   - Other databases are accessible through OSCAR.
8. *Do you expect a rather flat usage or there can be high-demand periods?*
   - Flat usage.
9. *How do you plan to trigger the inference of AI models (e.g. uploading files, programmatically via an API, etc.?)*
   - Via OSCAR, asynchronous.
10. *What is the expected rate of inference requests (e.g. 20 per hour)?*
    - In batch periods, when a certain user wants to predict their data.
11. *What support channel do you want to offer to the user? (Helpdesk, email?)*
    - Contact email is provided.
12. *What service usage statistics do you need/must collect?*

- Number of unique users of the AI image processing service.
- Number of countries of users.
- Names of countries reached.

13. *How will you deal with the user input and output data? Will it be open to other users or only accessible to the actual user?*
    - Through OSCAR buckets of users should be private. Phytoplankton data predictions (input and output) will be shared. If the user wants their information to be confidential they can run the model from their own server/pc.

Next steps and open questions

Based on the use case specific roadmaps the project identified the following steps to reach delivery under virtual access:

1. **Going public with the services:** Even if the use cases choose different delivery method for their model inference, they all developed and validated their models in the iMagine Platform and as a consequence they can be all delivered to users via the Option 1 and 2 (Inference from Marketplace; Download from Marketplace). In the next months we will make these models properly described on the Marketplace, we will make the training datasets available via Zenodo, and we will setup 1 service page on the iMagine project website for each of the 5 mature use cases. These service specific pages will serve both as promotional pages, and access channels to external users to reach the model inference services and the labelled training datasets. WP2 is working with the use cases to reach this stage in August-September.
2. **Completion of application portals:** UC1, UC2, UC3o, UC4 will use the inference based delivery as preferred application delivery method, and UC3a and UC5 as optional delivery mode. The OSCAR based inference requires the use cases to develop and operate custom made interfaces through which the users will interact with the inference service and with the trained AI model. This requires additional investment from the use cases and will require a more complex sustainability approach for the post-project period. WP5 and WP2 (Sustainability planning) are working on this.
3. **Tracking usage for the application delivery mode option 2:** Several of the WP5 VA metrics[8] are difficult or even impossible to track in the 'git download' delivery option (delivery mode 2) because git download sessions are unanimous. We can track 'Number of pulls' from the DockerHub image, but this is not necessarily unique users and does not allow us to identify the user and follow up to see e.g.

[8] Number of unique users of the AI image processing service, Number of images processed per year, Number of images ingested, Number of countries of users, Names of countries reached

number of images processed. Because of this we must encourage the other delivery options (option 1, 3, 4) across all the WP5 services.

4. **Monitoring actual GPU consumption:** The iMagine AI platform is currently measuring the allocation of GPUs to individual users, and reports this as GPU usage statistics for the VA report. We know that allocation of a GPU does not mean actual usage, therefore the real usage is lower than the reporting. Although the mature use cases move to inference delivery and therefore will require CPUs instead of GPUs, their retraining (if demanded) and the training of the 3 prototype and the onboarded 3 additional use cases will require GPUs. According to the reporting the project used so far 80% of the GPU capacity. Alternative ways for actual GPU usage, or stricter rules for GPU allocation and release may be needed in the project to avoid running out of GPU capacity. WP4 will work on this.
5. **Usage statistics for UC3:** The AI model of the UC3o and UC3s are serving 1-1 specific EMSO observatory station - so their user base in this respect is determined and we need to consider in WP5 how to measure usage and impact for the Virtual Access reporting:
   a. UC3o will produce species statistics and a live annotated video stream which will be consumed by researchers and citizens. These stakeholders should be considered as users for the Virtual Access reporting.
   b. UC3s will be run on research vessels thus the number of vessels, research staff on those can be a usable metric.
   c. Both services may be replicated at other EMSO sites, and this should be facilitated by EMSO ERIC.