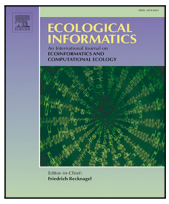
Ecological Informatics 91 (2025) 103306

Contents lists available at ScienceDirect

Ecological Informatics

journal homepage: www.elsevier.com/locate/ecolinf



Best practices for AI-based image analysis applications in aquatic sciences: The iMagine case study☆

Elnaz Azmi [a,*], Khadijeh Alibabaei [a], Valentin Kozlov [a], Tjerk Krijger [b], Gabriele Accarino [c,d], Sakina-Dorothée Ayata [e,f], Amanda Calatrava [g], Marco Mariano De Carlo [c], Wout Decrop [h], Donatello Elia [c], Sandro Luigi Fiore [i], Marco Francescangeli [j], Jean-Olivier Irisson [k], Rune Lagaisse [h], Martin Laviale [l], Antoine Lebeaud [m], Carolin Leluschko [n], Enoc Martínez [j], Germán Moltó [g], Igor Ruiz Atake [c], Antonio Augusto Sepp Neves [o], Damian Smyth [p], Jesús Soriano-González [q], Muhammad Arabi Tayyab [i], Vanessa Tosello [m], Álvaro López García [r], Dick Schaap [b], Gergely Sipos [s]

[a] *Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany*
[b] *Mariene Informatie Service BV (MARIS), Nootdorp, Netherlands*
[c] *CMCC Foundation - Euro-Mediterranean Center on Climate Change, Lecce, Italy*
[d] *Department of Earth and Environmental Engineering, Columbia University, New York, USA*
[e] *Sorbonne Université, MNHN, CNRS, IRD, Laboratoire d'Océanographie et du Climat : Expérimentations et Approches Numériques, LOCEAN, Paris, France*
[f] *Institut Universitaire de France, Paris, France*
[g] *Instituto de Instrumentación para Imagen Molecular (I3M), Centro mixto CSIC - Universitat Politècnica de València, Valencia, Spain*
[h] *Vlaams Instituut voor de Zee (VLIZ), Ostend, Belgium*
[i] *University of Trento, Trento, Italy*
[j] *Electronics Department, Polytechnic University of Catalonia (UPC), Barcelona, Spain*
[k] *Sorbonne Université, CNRS, Laboratoire d'Océanographie de Villefranche, LOV, Villefranche-sur-Mer, France*
[l] *Université de Lorraine, CNRS, LIEC, Metz, F-57000, France*
[m] *Institut Français de Recherche pour l'Exploitation de la Mer (Ifremer), Brest, France*
[n] *German Research Center for Artificial Intelligence GmbH (DFKI), Oldenburg, Germany*
[o] *Orbital EOS - Earth Observation Solutions, Valencia, Spain*
[p] *Marine Institute Rinville, Galway, Ireland*
[q] *Balearic Islands Coastal Observing and Forecasting System (SOCIB), Palma, Mallorca, Spain*
[r] *Instituto de Fisica de Cantabria (IFCA), CSIC-UC, Santander, Spain*
[s] *EGI Foundation (EGI), Amsterdam, Netherlands*

ARTICLE INFO

Dataset link: https://github.com/ai4os, https://github.com/ai4os-hub, https://dashboard.cloud.imagine-ai.eu/marketplace, https://zenodo.org/communities/imagine-project/records?q=&f=resource_type%3Adataset

Keywords:
Machine learning
Deep learning
Computer vision
Image processing
FAIR data
Open science
Aquatic sciences
Ocean and marine sciences

ABSTRACT

The iMagine project is an EU-funded initiative led by the EGI Foundation. One of the objectives of this project is to provide an AI platform that leverages AI-powered tools to improve the processing and analysis of imaging data from marine and freshwater ecosystems, contributing to the study of the health of oceans, seas, coasts, and inland waters. Connected to the European Open Science Cloud (EOSC), iMagine supports the development, training, and deployment of AI models by collaborating with twelve use cases across diverse aquatic science fields. This collaboration fosters valuable insights and accelerates scientific progress by refining existing solutions in data acquisition, preprocessing, and model deployment. The platform offers trained models as a service, integrating AI tools for image annotation, ensuring the creation of high-quality datasets that comply with FAIR principles. Through these methodologies, iMagine enhances consistency, enabling researchers to efficiently publish and share data in repositories.

Beyond its AI tools, iMagine places a strong emphasis on deep learning models, such as convolutional neural networks, for tasks like image classification, object detection, and segmentation, tailored to the unique requirements of aquatic sciences. It also provides robust performance evaluation tools, including experiment

☆ This article is part of a Special issue entitled: 'Ocean Learning' published in Ecological Informatics.

* Corresponding author.
E-mail address: elnaz.azmi@kit.edu (E. Azmi).

https://doi.org/10.1016/j.ecoinf.2025.103306
Received 14 February 2025; Received in revised form 23 June 2025; Accepted 24 June 2025
Available online 7 July 2025

Automated species classification

tracking, while tackling challenges such as AI model drift and data biases to ensure research reproducibility and transparency. The platform enables users to develop, train, share, and deploy AI models within a flexible environment that integrates with federated cloud and high-performance computing infrastructures, using Docker containers for smooth execution. Additionally, iMagine fosters collaboration with projects like AI4EOSC and Blue-Cloud, and Research Infrastructures such as EMSO and SeaDataNet, expanding its impact on the scientific community.

This paper summarizes the key lessons and best practices learned in the iMagine project through the full process of AI-based aquatic image analysis, from data preparation and annotation to model deployment and evaluation. The paper therefore helps aquatic scientists advance their AI-driven image analysis approaches.

1. Introduction

Machine learning, particularly Deep Learning (DL), has recently revolutionized the field of image analysis. Advances in neural network architectures, such as Convolutional Neural Networks (CNNs) and Generative Adversarial Networks (GANs), have enabled computers to achieve high accuracy in tasks like object recognition, image segmentation, and image generation. These developments have transformed environmental monitoring by providing powerful tools for extracting meaningful information from image data (Rubbens et al., 2023).

The iMagine project is funded by the European Commission to contribute to the overarching mission of the EU for healthy oceans, seas, coasts, and inland waters. It does so by working toward key objectives, such as enhancing aquatic research through Artificial Intelligence (AI) applications, leveraging EOSC for developing, training, and deploying AI models. It enables online data stream analysis in distributed environments, facilitating collaboration among research infrastructures to share images and AI applications. Additionally, the project develops best practices about how to create and deliver AI-powered image processing services.

The general objective of iMagine is to deploy, operate, validate and promote a dedicated iMagine AI platform, connected to EOSC and AI4EU and to apply the platform for the development and delivery of thematic AI-powered image analysis services in aquatic sciences. This provides open access to a diverse portfolio of AI-based image analysis services and image repositories from multiple research infrastructures, working on and being relevant to the overarching theme of 'Healthy oceans, seas, coastal and inland waters'. The AI-based image analysis services emerged from the twelve scientific use cases that the project supports: five mature use cases (aimed at delivering AI-services), three prototype use cases (aiming to reach validated AI applications) and four additional use cases (selected for support via the iMagine open call). These use cases gained practical experience in the overall AI workflow, consisting of developing AI models, collecting images, and preparing training datasets to train those models, validating the models, deploying them for inference and supporting third-party external users.

Although utilizing AI methods in image analysis is well-documented, there remains a lack of a cohesive framework tailored to the unique challenges and requirements of the aquatic science domain. This paper addresses this gap by providing insight from multiple subdisciplines, establishing a structured set of domain-specific best practices and guidelines for AI-based image analysis in aquatic sciences.

The objective of this paper is to summarize the lessons learned and experiences gained by the iMagine Competence Center, the distributed support team that was responsible for implementing the AI use cases. The experiences cover the areas of training data preparation, image annotation, preprocessing techniques, AI model selection, FAIR data and model evaluation, dataset publication, and serving trained AI models for inference. The results emerged from the close collaboration between iMagine's use cases and the developers/providers of the iMagine AI platform, and are relevant for producers and providers of image sets, and developers, evaluators and providers of AI-based image analysis applications.

The remainder of this paper is structured as follows: We begin with an overview of related works (Section 2) on the application of AI-based image analysis in aquatic sciences. We continue with an introduction to the iMagine Competence Center (Section 3) as the support entity that captured the best practices. Next, in Section 4 we review various neural networks relevant to image and video analysis, with a particular emphasis on those used in iMagine's use cases. In the following section, we discuss annotation tools, a crucial step in preparing training datasets for AI model development (Section 5). This section evaluates the annotation tools that have proven useful in aquatic science applications. Section 6 focuses on data repositories and open-source datasets for aquatic applications, describing how training datasets can be shared and published once they are ready for third-party use. We then cover preprocessing techniques (Section 7), detailing methods to improve data quality before feeding it into AI models. Performance metrics and evaluation methods are explored in Section 8. Tools for monitoring model performance during training are addressed in Section 9, emphasizing experiment-tracking tools to ensure reliability. The document then delves into data biases in aquatic sciences models (Section 10), examining how these biases arise and how they can affect the accuracy of models and analyses. Next, we discuss model delivery (Section 11), explaining how trained and evaluated AI models can be shared with the aquatic community and deployed for inference. In Section 12, on AI model drift tools, we explore tools designed to detect, analyze, and mitigate model drift in AI systems deployed in production. (Section 13), offers insights into the AI techniques used, and the lessons learned from the iMagine use cases. The results and discussion of our main findings are presented in Section 14. Finally, the paper concludes with Section 15.

2. Related work

The rapid growth of ocean observation technologies has led to the accumulation of vast amounts of "blue data" driving ocean science toward a data-driven approach. The field of Data-Driven Oceanography, deep blue AI (DBAI), combines new technologies, methods, and tools to transform data into valuable knowledge, heavily relying on ocean expert knowledge and data features (Chen et al., 2022). DBAI is discussed in three main areas: AI feature engineering, AI detection frameworks, and AI time-series prediction frameworks, with applications ranging from sea surface studies to deep-sea exploration (Bonino et al., 2024). These methods have already been used for detecting fish, as well as for seagrass, plankton, and coral classification (Moniruzzaman et al., 2017) from images and videos. Despite the potential of AI in oceanography, challenges exist, particularly in understanding and labeling ocean data, which often relies on expert knowledge. The development of unsupervised DL methods could help to address these limitations. Chen et al. (2022) outline the evolution of AI from crafted knowledge to statistical learning and contextual adaptation, emphasizing the importance of AI's integration with ocean science to extract meaningful insights. However, challenges remain, particularly in overcoming the "black box" nature of AI models, requiring collaboration between ocean scientists and computer scientists. Ultimately, the goal of DBAI is to extract valuable knowledge from ocean data, with independent validation from ocean experts essential to demonstrate AI's utility in the field. The paper highlights the importance of fostering a scientific discourse around blue data and ocean data science to ensure that AI advances responsibly, aiming to uncover the physical mechanisms behind oceanic phenomena.

Web references

AI4Compose	https://github.com/ai4os/ai4-compose
AI4EOSC	https://ai4eosc.eu
AI4OS	https://ai4os.eu
BIIGLE	https://biigle.de
Blue-Cloud	https://blue-cloud.org
ClearML	https://clear.ml
CMCC	https://www.cmcc.it
CVAT	https://www.cvat.ai
DEAL	https://pml.ac.uk/projects/deal-decentralised-learning-for-automated-image
DOVER VQA	https://github.com/VQAssessment/DOVER
EGI Notebooks	https://notebooks.egi.eu
Elyra	https://github.com/elyra-ai/elyra
EOSC	https://eosc.eu
FlowFuse	https://flowfuse.com
Hasty	https://hasty.cloudfactory.com
iMagine	https://www.imagine-ai.eu
Label Studio	https://labelstud.io
Labelbox	https://labelbox.com
Marketplace	https://dashboard.cloud.imagine-ai.eu/marketplace
MinIO	https://min.io
MLflow	https://mlflow.org
MLflow tracking server	https://mlflow.cloud.imagine-ai.eu
Node-RED	https://nodered.org
Orbital EOS	https://www.orbitaleos.com
OSCAR Batch	https://github.com/grycap/oscar-batch
OSCAR exposed services	https://docs.oscar.grycap.net/exposed-services
OSCAR Inference	https://inference.cloud.imagine-ai.eu
OSCAR Inference-walton	https://inference-walton.cloud.imagine-ai.eu
patchify	https://pypi.org/project/patchify
Roboflow	https://roboflow.com
SEANOE	https://www.seanoe.org
Supervisely	https://supervisely.com
TensorBoard	https://www.tensorflow.org/tensorboard
VIAME	https://www.viametoolkit.org
Watershed	https://scikit-image.org/docs/stable/auto_examples/segmentation/plot_watershed.html
Weights and Biases	https://wandb.ai
Zenodo	https://zenodo.org/communities/imagine-project

DL is revolutionizing underwater video analysis by addressing challenges like poor water conditions, species similarity, and background clutter, which have previously hindered the quality of fish visual sampling (Xu et al., 2023; Marrable et al., 2022; Qin et al., 2015). DL, alongside advancements in monitoring hardware and underwater communication, enables comprehensive fish sampling across a range of environments, from shallow waters to the deep ocean, and provides a comparative understanding of marine and aquatic ecosystems. It also solves the problem of managing vast amounts of data generated by underwater video, turning what was once an expensive and cumbersome task into a simple computer-processing issue. This capability allows for more efficient, spatially and temporally replicated underwater fish surveys, facilitating significant advances in marine sciences. DL and related techniques are also valuable for improving data classification and feature extraction, as well as surveying fish habitats and tracking movement dynamics. However, to fully realize these benefits, concentrated efforts in data collection, model development, and transparent, reproducible research are essential for maximizing the potential of DL in marine habitat monitoring (Saleh et al., 2022). The authors survey computer vision and DL studies on fish classification in underwater habitats conducted between 2003 and 2021. It provides an overview of key DL concepts, analyzes and synthesizes relevant studies, and discusses the challenges faced in developing DL techniques such as model generalization, data set limitation, and image quality for underwater image processing. They also propose strategies to address these challenges and offer insights into the future of DL in marine habitat monitoring.

Gaur et al. (2023) discuss the issue of organic effluent enrichment in water, which can stimulate algal growth, leading to pollution and threats to aquatic ecosystems. Recent incidents of harmful algal blooms (HABs) have underscored the need for early detection technologies. Although previous research emphasized the role of DL in the identification of algal genera, selecting the optimal CNN model for effective monitoring remains a challenge. The study evaluates the performance of four DL models (MobileNet V-2, VGG-16, AlexNet, and ResNeXt-50) in classifying 15 types of bloom-forming algae. Using optimizers like Adam and RMSprop and activation functions such as softmax and ReLU, the models achieved classification accuracies of 40%, 96%, 98%, and 99%, respectively. The study concludes that ResNeXt-50 is the most accurate and versatile model for real-time algae identification, offering promise for AI-driven advances in phycological research and sustainable technologies.

Gunda et al. (2019) describe the development of an AI-based mobile application platform designed for automated color intensity identification from sensor images, aimed at cost-effective, field-deployable water monitoring using smartphones. AI, particularly deep CNNs, is used to process images where conventional detection methods might struggle, significantly outperforming traditional approaches. The application was tested for monitoring bacterial contamination in water, classifying sensor images based on the visual presence or absence of bacteria. The system achieved an impressive detection accuracy of approximately 99.99%, a substantial improvement over manual visual inspection methods. This enhanced accuracy is attributed to the elimination of subjective human decision-making, which can introduce inconsistencies in traditional image analysis.

Nagpal et al. (2024) examine the growing application of AI in wastewater treatment, focusing on its objectives, benefits, and major findings in three key areas: predicting the removal efficiency of organic and inorganic pollutants, real-time monitoring of water quality parameters, and detecting faults in treatment processes and equipment. AI technologies have demonstrated varying prediction accuracy for pollutant removal, with R^2 (coefficient of determination) values ranging from 0.616 to 0.997. The review also discusses the cost-effectiveness of AI systems in wastewater treatment and highlights ongoing pilot projects and demonstrations in various countries, showcasing the global adaptability and success of AI in improving wastewater treatment. Additionally, ethical considerations and potential future directions for AI in wastewater treatment are explored.

More than 20 years ago, the ADIAC project developed seminal approaches for digital imaging and automatic diatom identification (du Buf and Bayer, 2002). Since then, several teams have been working on automatic detection, segmentation, and classification of benthic diatoms in images (see for instance references in Venkataramanan et al. (2024)). Yet, the upscaling of the already existing deep learning models to real-world situations (in terms of number of species, image quality) remains strongly limited by the small number of large and open training datasets of benthic diatom images. Hopefully, the recent publication of

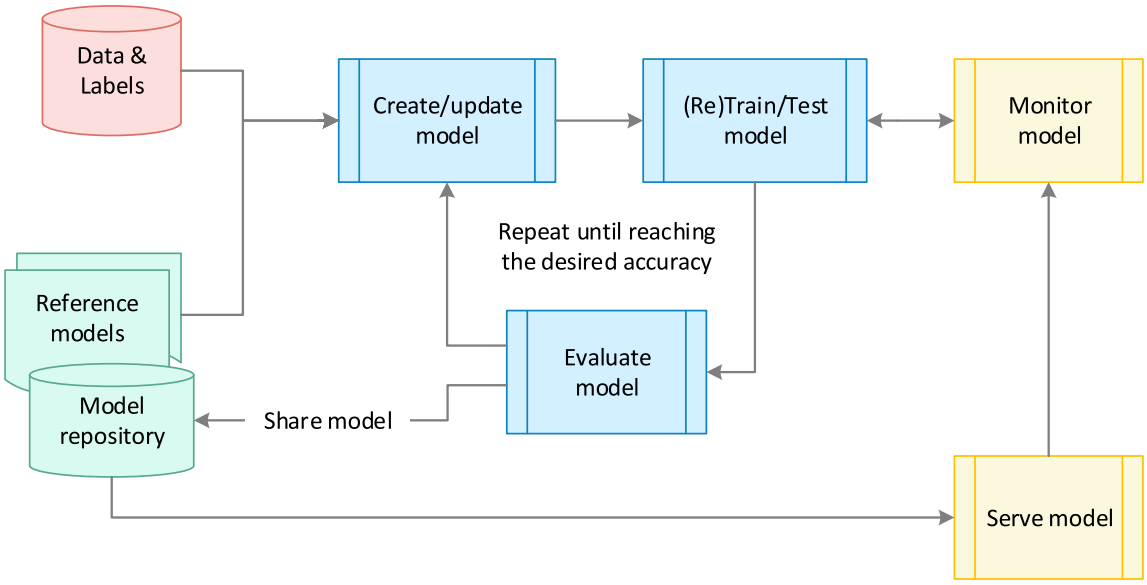


Fig. 1. Typical AI development process followed by the iMagine use cases, concluding in their AI model being available on the iMagine Marketplace.

a large image dataset of freshwater diatoms, containing 83,570 images of 611 diatom taxa (101 of which represented by at least 100 examples and 144 by at least 50 examples each) (Venkataramanan et al., 2024)) should contribute to overcome this limitation.

In Ferrari et al. (2025), the authors present an object detection pipeline for marine sciences that provides an end-to-end solution, leveraging the compute continuum paradigm. The pipeline processes underwater images using edge computing devices, with results transmitted to cloud infrastructure. Key design goals include minimizing execution time, response time, and power consumption, especially considering the power constraints in underwater environments. The paper highlights the challenges of using DL methods in such scenarios and notes that existing datasets, like Fish4Knowledge, are not suitable for the Mediterranean Sea's unique conditions. The authors demonstrate the performance of the YOLOv3 object detector and its Tiny version on a low-power Nvidia Jetson Nano, showing that power consumption trade-offs are important for long-duration autonomous underwater missions. They also explore how to configure edge computing in underwater observatories, evaluating the impact of different approaches on execution times. The study reveals that setup times, such as initializing Python and CUDA environments, affect performance and suggests tuning analysis frequency based on the need for immediate event response or data transfer optimization. The findings have broader applicability in environmental monitoring scenarios with limited power and bandwidth, where object detection models, with appropriate training, can be adapted to various domains.

As AI-based assistant tools are increasingly utilized in marine ecological research, particularly for analyzing image data from field biodiversity surveys, there is a growing need to establish standards to improve the robustness of related research. Reproducibility and comparability are essential for scientific integrity and international collaboration. To enhance these aspects of AI-based automated image analysis, recommendations include documenting biodiversity data sources, sharing all codes, algorithms, and software packages, and developing AI tools for automated analysis across multiple taxonomic levels. Given the advancements in AI and image databases, adaptive strategies can be applied to standardize practices, facilitating the integration of cutting-edge technologies to optimize automated image analysis, including improvements in accuracy for fine-grained image analysis (Zhou et al., 2024).

Our work extends the above studies by consolidating experiences of real-world applications in aquatics sciences (Table 1) and covers the entire AI/ML life cycle (Fig. 1).

3. Competence center

The iMagine Competence Center comprises a diverse group of experts spanning multiple disciplines, including AI, information technology (IT), and domain-specific fields. This multidisciplinary team encompasses marine and freshwater application experts who are crucial in tailoring solutions to these specific domains. Additionally, the center is supported by AI framework specialists and IT experts who play a key role in overseeing the technical integration of AI solutions into various use cases. These experts are responsible for ensuring that AI models are effectively integrated and function seamlessly within the broader technical infrastructure.

To facilitate the adoption and use of AI models, the iMagine AI platform (Heredia and Kozlov, 2025) offers various application delivery approaches, such as the iMagine Marketplace and OSCAR (Risco et al., 2021) inference system. These platform components provide a streamlined and user-friendly environment for discovering and implementing AI tools, making it easier for domain experts to integrate advanced AI technologies into their work.

The iMagine AI platform itself is powered by the AI4OS software stack, which is developed by the AI4EOSC consortium. This software stack delivers the necessary tools and frameworks for implementing and scaling AI-driven solutions, ensuring that the iMagine platform can meet the diverse needs of its users. By combining expertise in AI, IT, and domain-specific knowledge, the Competence Center offers comprehensive support to use cases, helping them to leverage cutting-edge AI technologies effectively.

As part of the iMagine Competence Center, we are actively involved in monitoring and supporting use cases through regular meetings. These meetings allow us to stay closely connected with the progress of each use case, ensuring that any challenges or needs are addressed promptly. In addition to these meetings, we offer a range of training sessions as part of our annual Competence Center workshops and webinars. These sessions are designed to equip participants with the knowledge and skills required to fully leverage the resources and expertise available through the center.

Furthermore, we carefully document the developments, achievements, challenges, best practices, and expert recommendations through deliverables. These documents serve as comprehensive guidelines for the standardization of processes and for enhancing the quality of datasets and AI models. By maintaining a clear and organized record

Table 1
List of real-world use cases. Mature (advanced in AI-workflow), Prototype (in validation), and Additional (via open call).

Name	Type	Description
Marine Litter Assessment	Mature	Aquatic litter monitoring system using drones
ZooScan - EcoTaxa Pipeline	Mature	Taxonomic identification of zooplankton from images of marine water samples taken by the ZooScan instrument
Marine Ecosystem Monitoring at EMSO OBSEA	Mature	Extraction of biological content from image data collected by an underwater camera observing various fish species
Marine Ecosystem Monitoring at EMSO Azores	Mature	Automatically annotate and validate submarine images
Marine Ecosystem Monitoring at EMSO SmartBay	Mature	Inspection of the video archive, quick detection and enumeration of prawns and prawn burrows, efficient flagging and referencing of interesting footage
Oil Spill Detection	Mature	Oil spill detection from satellite images
FlowCam Plankton Identification	Mature	Taxonomic identification of phytoplankton using FlowCam images
Underwater Noise Identification	Prototype	Underwater noise identification from acoustic recordings using spectrograms
Beach Monitoring	Prototype	Beach seagrass wracks identification, shoreline extraction, and rip-currents detection
Freshwater Diatoms Identification	Prototype	Identification of freshwater diatoms using microscopic images
Cold Water Coral Reefs	Additional	Improving knowledge about cold water coral reefs
Satellite Derived Bathymetry	Additional	Automated system for high-resolution bathymetry mapping using satellite imagery
Fish Otoliths	Additional	AI for image-based age reading from fish otoliths
EyeOnWater	Additional	Observing water color and transparency by citizens globally

of these contributions, we ensure that our work supports the continuous improvement of both the iMagine infrastructure and the broader community it serves.

The support provided by the xCompetence Center begins at the very start of the AI development cycle, starting with the labeling of raw data collected by the use cases. Our involvement continues throughout the entire AI development process, ensuring that each stage is carefully managed, from data preprocessing and model training to the final deployment of the AI model as a service (Fig. 1). This comprehensive support allows us to address the specific needs and challenges of each use case, providing tailored assistance that maximizes the potential of AI technologies.

As part of our commitment to transparency and collaboration, the labeled training datasets provided by use cases, which are crucial for the development of high-quality AI models, are published on public repositories such as Zenodo, along with corresponding metadata. This ensures that the datasets are accessible to the broader community, promoting reuse and further innovation (see Section 6 for more details).

From the lessons learned by the use cases throughout the entire AI development cycle, a comprehensive set of best practices has been established. This document consolidates these insights and recommendations gathered from real-world applications acting in different fields of aquatic sciences (Table 1) and processing data from various geographical locations and ecosystems (e.g., marine or freshwater). The use cases also helped to cross-validate the findings. The document offers valuable guidance to both current and future users on how to optimize AI model development, improve data quality, and ensure successful implementation.

4. Deep learning approaches

DL, a branch of AI, utilizes multi-layered neural networks to model and interpret complex patterns in data. It has transformed image processing by enabling machines to learn and identify features directly from raw images, eliminating the need for manual feature extraction. Models such as CNNs have driven significant advancements in tasks like image classification, object detection, and image generation. This progress is crucial for developing sophisticated applications across various fields, including aquatic sciences. Over time, various CNN architectures have been developed, each introducing unique structures designed to enhance performance, efficiency, and accuracy. These networks are highly effective at automatically detecting and classifying patterns in images, making them efficient for tasks such as monitoring marine biodiversity, mapping underwater habitats, and tracking changes in water quality or coral reef health. By automating these processes, CNNs improve the accuracy and efficiency of environmental assessments, facilitate more informed decision-making for conservation efforts, and enable more effective monitoring of aquatic ecosystems in response to climate change and human impacts.

4.1. Convolutional neural networks

CNNs are a class of deep neural networks that are most commonly used to analyze visual images, and have proven to be extremely successful in various computer vision tasks. We briefly review the most well-known CNNs for: 1. Image classification, which involves classifying images into predefined categories, 2. Object detection, which focuses on locating and identifying objects in an image, 3. Image segmentation, which involves segmenting images into different classes.

4.1.1. Classification methods

Classification is a fundamental task in the field of computer vision, which aims to classify input data into predefined classes or categories. In this field, CNNs have redefined image classification by automatically learning hierarchical features from raw pixel data. Over the years, various CNN architectures have emerged, all characterized by different features and performance metrics. For instance, AlexNet (Krizhevsky et al., 2017), the winner of the ImageNet Large Scale Visual Recognition Challenge in 2012, was a pioneer in the use of deep convolutional networks. VGG (Simonyan and Zisserman, 2014) increased the network depth up to 19 layers, while ResNet (He et al., 2016) introduced skip connections to address the vanishing gradient problem. MobileNet (Howard et al., 2017), on the other hand, represents a class of lightweight CNNs designed for optimized inference on mobile and embedded devices with limited computational resources. Table 2 shows some of the best-known models that have been developed for classification tasks.

4.1.2. Object detection and localization methods

These methods are aimed at detecting and localizing objects within an image or video, as well as identifying their class labels. For the classification task, a labeled dataset consists of images, each assigned a single class label. However, for object detection and localization, the dataset requires labeled images in which each object of interest is assigned a bounding box and the corresponding class label.

Current object detection algorithms can be divided into two main categories: 1. two-stage detectors, such as Region-based Convolutional Neural Network (R-CNN) (Girshick et al., 2014), Fast R-CNN (Girshick, 2015), and Faster R-CNN (Ren et al., 2016), and 2. one-stage detectors, such as Single Shot MultiBox Detector (SSD) (Liu et al., 2016) and You Only Look Once (YOLO) (Redmon et al., 2016), see Table 3.

Table 2
List of the best-known CNNs for classification.

Model name	Description
AlexNet	One of the first deep convolutional neural networks, introduced techniques such as ReLU and dropout (Krizhevsky et al., 2017).
VGG	Very deep convolutional networks, known for their uniform architecture with small (3 × 3) filters (Simonyan and Zisserman, 2014).
ResNet	Deep residual networks, known for their ability to train very deep networks effectively (He et al., 2016).
Mobilenet	Lightweight convolutional neural networks designed for mobile and embedded vision applications (Howard et al., 2017).
InceptionNet	Designed to improve computational efficiency and accuracy through the use of inception modules (Szegedy et al., 2016).
Xception	It is based on an improvement to the Inception model architecture, specifically utilizing depthwise separable convolutions to make the network more efficient in terms of computational cost and parameter count (Chollet, 2017).
DenseNet	Densely connected convolutional networks where each layer receives feature maps from all preceding layers (Huang et al., 2017).
ResNeXt	Extension of ResNet architecture with a cardinality parameter to improve representational power (Xie et al., 2017).
EfficientNet	Compound scaling method that uniformly scales dimensions of depth/width/resolution (Tan and Le, 2019).
CLAP	The CLAP model is a DL framework designed to align audio signals and textual descriptions in a joint embedding space (Wu et al., 2023b).

Table 3
List of the best-known CNNs for object detection.

Model name	Description
R-CNN	Region-based Convolutional Neural Network for object detection. It uses selective search to propose regions and applies a convolutional network to classify each region (Girshick et al., 2014).
Fast R-CNN	Evolution of R-CNN that improves speed and accuracy by introducing the Region Of Interest (ROI) pooling layer to efficiently extract region features (Girshick, 2015).
Faster R-CNN	Introduces Region Proposal Network (RPN) to generate region proposals instead of using external methods like selective search, making it faster and more accurate. Integrates RPN with Fast R-CNN architecture (Ren et al., 2016).
SSD	Single Shot MultiBox Detector designed for real-time object detection. Utilizes a single neural network to predict bounding boxes and class probabilities for multiple object instances simultaneously at different scales (Liu et al., 2016).
YOLO	You Only Look Once models for real-time object detection. Divided into multiple versions (v1 to v10) each with improvements in accuracy and speed, focusing on predicting bounding boxes and class probabilities directly from full images in a single evaluation (Redmon et al., 2016).
EfficientDet	A family of object detection models that achieve better accuracy and efficiency by using a compound scaling method to balance the depth, width, and resolution of the network (Tan et al., 2020).

Various software programs have been developed for these algorithms. For example, fasterrcnn pytorch training pipeline (Rath, 2024), provides a pipeline for training PyTorch Faster R-CNN models on custom datasets. With this pipeline, users can choose between official PyTorch models trained on the COCO dataset (Lin et al., 2014), use any backbone from Torchvision classification models, or even define their own custom backbones. The trained models can then be used for object detection tasks on specific datasets. We have integrated the DEEPaaS API (López García, 2019) into this existing codebase, and this model is accessible as a general-purpose model on the iMagine Marketplace.

Recently, the YOLO model has evolved considerably with the development of new versions designed to improve both the accuracy and efficiency of the system. YOLOv8 (Jocher et al., 2023) has substantially enhanced both the speed and accuracy of object detection compared to the previous versions. This model introduces various variants, including Nano (n), Small (s), Medium (m), Large (l), and Extra-Large (x), each with a different number of parameters that can be trained depending on the model variant.

Ultralytics (Jocher et al., 2023) is an active contributor to the open-source community, providing accessible and cutting-edge solutions for various artificial intelligence tasks such as detection, segmentation, classification, tracking, and pose estimation. YOLOv8 is provided by the Ultralytics organization using the PyTorch framework. This model can be used flexibly for classification, detection, object-oriented detection, and segmentation tasks. We have integrated a DEEPaaS API with Ultralytics YOLO and made it available as a general-purpose model on the iMagine Marketplace. Depending on the application, users can select one of the model variants and train it. The Docker images are provided along with these models. The users have the option of either starting a deployment on the iMagine platform from this Docker image and training the models on the custom data without further effort. Alternatively, they can run these Docker containers on both Clouds and High-Performance Computing (HPC) infrastructures.

4.1.3. Segmentation methods

The segmentation task in computer vision refers to the process of dividing an image into multiple segments or regions to simplify its representation and facilitate the analysis of its content. The goal is to group pixels that belong to the same object or share similar visual characteristics, such as color, texture, or intensity. A dataset used for segmentation tasks, such as semantic segmentation or instance segmentation, provides pixel-level annotations for the entire image. Unlike in object detection tasks, where bounding boxes are used, each pixel in the image is labeled with a class indicating the category of the object or region to which it belongs. Segmentation tasks can be roughly divided into the following types:

- **Semantic segmentation**: In semantic segmentation, each pixel in the image is assigned a class label that represents the category of the object or region to which it belongs. The main objective is to classify each pixel into predefined categories without distinguishing between the individual object instances (Guo et al., 2018).
- **Instance segmentation**: Instance segmentation is an extension of semantic segmentation, where the goal is not only to classify each pixel into categories, but also to distinguish between different instances of the same object class. This means that each individual object instance in the image is assigned a unique identifier (Sharma et al., 2022).
- **Panoptic segmentation**: Panoptic segmentation combines both semantic and instance segmentation in a unified framework. It aims to segment and detect all objects in the scene, both things (e.g., background elements such as sky, road, grass) and objects (e.g., cars, people, animals), while also distinguishing between individual instances of objects (Kirillov et al., 2019).

Fully Convolutional Networks (FCNs) represent an innovative approach in the field of semantic segmentation, transforming traditional CNNs into fully convolutional architectures. This transformation enables training and prediction for pixel-wise segmentation tasks without relying on fully connected layers. A key feature of FCNs is their ability to perform dense predictions for image segmentation by replacing the fully connected layers typically used for classification with convolutional layers. This structural modification allows the network to

Table 4
List of the best-known CNNs for segmentation.

Model name	Description
Fully Convolutional Networks (FCNs)	FCNs are designed for semantic segmentation and replace the fully connected layers of traditional CNNs with convolutional layers to produce dense predictions. The model is effective for generating pixel-wise segmentation maps (Long et al., 2015).
Segment Anything Model (SAM)	SAM is a versatile foundation model for image segmentation that excels in zero-shot learning, allowing it to segment unseen objects and perform edge detection, object proposal generation and instance segmentation. It improves speed and accuracy in annotation tasks and is available under a permissive open license (Kirillov et al., 2023).
Mask2Former	Features are extracted from the image (at several scale) by a CNN. A pixel decoder scales those features back up to the original image size. A transformer decoder receives object queries and information from the pixel decoder to predict a mask and a class (Cheng et al., 2022).
U-Net	U-Net is a convolutional neural network architecture designed specifically for biomedical image segmentation. It has a unique architecture that consists of a contracting path to capture context and a symmetric expanding path that enables precise localization (Ronneberger et al., 2015).

maintain spatial hierarchies and directly output segmentation maps that align with the dimensions of the input image. FCNs facilitate end-to-end learning and prediction, meaning the network can be trained and inferred in a single step, directly mapping input images to segmented outputs. By utilizing only convolutional layers, FCNs preserve spatial information throughout the network, which is crucial for accurate pixel-wise segmentation. FCNs have been extensively applied across various domains, such as medical imaging, autonomous driving, and scene understanding. For example, in biomedical image segmentation, FCNs have demonstrated strong performance in tasks like cell segmentation and lesion detection.

The Segment Anything Model (SAM) is a promptable foundation model for image segmentation that significantly enhances both the speed and quality of computer vision annotations. One of SAM's most compelling features is its capability as a promotable segmentation system, demonstrating Zero-Shot generalization to unfamiliar objects and images, thereby eliminating the need for additional training. Zero-Shot learning is a machine learning paradigm in which a model is trained to recognize classes or objects that it has never encountered during training. This is achieved by incorporating additional information or knowledge about these unseen classes, enabling the model to generalize and make predictions for unfamiliar instances. SAM leverages Zero-Shot learning to segment unseen objects during inference. SAM can be prompted to perform edge detection and segment anything, including object proposal generation. It can also segment detected objects, effectively performing instance segmentation. As a proof of concept, it can segment objects based on free-form text. SAM is available under a permissive open license (Apache 2.0).

Mask2Former is a deep learning model that combines various segmentation tasks, including instance, semantic, and panoptic segmentation, within a single architecture. Mask2Former leverages transformers and a novel masked attention mechanism to efficiently process image data by focusing on specific regions, thereby generating accurate segmentation masks. The model captures detailed relationships between pixels, leading to improved segmentation quality and computational efficiency.

U-Net's distinctive U-shaped structure consists of two main components: an encoder and a decoder, connected by skip connections. The encoder, or contracting path, follows the typical architecture of a convolutional network. As the input image progresses through the contracting path, its spatial dimensions are progressively reduced, while the number of feature channels is increased with each step. Each downsampling step doubles the number of feature channels. The decoder, or expansive path, aims to recover spatial information and generate a segmentation map. Each step in the decoder involves upsampling the feature map.

A summary of these best-known models for segmentation tasks can be found in Table 4. Furthermore, the detailed application of the aforementioned CNN models to aquatic sciences use cases within iMagine is described in Section 13.

4.2. Foundation models: The next-generation AI

Foundation models are large neural networks trained on large amounts of diverse data, making them a powerful base for fine-tuning across various downstream tasks. They serve as a foundational base upon which more specific models can be built. These models represent a significant advancement in AI due to their ability to generalize across tasks, transfer knowledge, and enhance the efficiency of AI development.

One of the most notable features of foundation models is their capacity for generalization and transfer learning. These models can transfer knowledge from one domain to another, enabling them to adapt to new challenges with minimal retraining. For instance, a model trained on large text data can be fine-tuned to understand scientific texts, legal documents, or even new languages. Examples of such models include GPT (Generative Pre-trained Transformer) (Brown et al., 2020) and BERT (Bidirectional Encoder Representations from Transformers) (Devlin, 2018). GPT, developed by OpenAI, is trained on massive text corpora and excels at generating human-like text, answering questions, translating languages, and more. However, BERT, developed by Google, is optimized for understanding the context of words within a sentence, making it particularly effective for tasks such as question answering and natural language understanding. Another important aspect of foundation models is their multimodal capabilities. Models like CLIP (Contrastive Language-Image Pretraining) (Radford et al., 2021) integrate multiple data modalities, such as text and images, allowing them to perform tasks that require understanding across different types of data. Finally, foundation models streamline AI development by reducing the need to train new models for every specific task from scratch. This efficiency accelerates the development and deployment of AI systems, making it easier to build solutions for a wide range of applications.

Leveraging foundation models can be considered the next step beyond CNNs for AI-based aquatic services, wherever applicable. However, iMagine use cases have not yet required their application.

5. Annotation tools

The rapid advancement of deep learning techniques, particularly CNNs, has revolutionized image analysis. However, the effectiveness of these models is highly dependent on the quality and quantity of annotated training data. Annotation tools enable users to label images with semantic tags, bounding boxes, polygons, and key points, facilitating the creation of annotated datasets tailored to specific research objectives and model requirements.

This section summarizes our experience with various annotation tools used in aquatic sciences and image processing applications within the iMagine project. We evaluate the features, functionality, and suitability of each tool for annotating images and videos, considering factors such as annotation complexity, scalability, and integration with AI frameworks.

5.1. BIIGLE

BIIGLE (Langenkämper et al., 2017) is a web-based, open-source tool for fast and effective labeling of images and videos. It was originally developed for monitoring and researching the marine environment, but it can also be used to annotate various image and videos. The video annotation tool allows automatic object tracking. BIIGLE features a label tree, which allows labels to be organized either in a simple flat structure or a more complex hierarchical format, making the labeling process more intuitive. BIIGLE is also highly scalable, capable of managing vast collections containing thousands or even tens of thousands of images or videos.

Users can download all images and videos into a folder, with files being easily loaded from a public web server, cloud storage, or through direct volume uploads. The tool also enables annotations to be viewed in a structured grid format, ensuring a clear and organized overview of labeled data. Additionally, BIIGLE allows users to request annotation assistance from others, making collaboration more seamless. Another notable feature is its Machine Learning Assisted Image Annotation (MAIA) method, which enhances the efficiency of the labeling process. Moreover, all labeling information can be downloaded for free in user-friendly formats, such as CSV.

Despite its many advantages, the file upload process can be slow and the automatic object tracking is only available for point and circle tools.

5.2. Roboflow

Roboflow is a platform designed to help developers and businesses efficiently manage and deploy computer vision models. It offers a comprehensive set of tools for data annotation, model training, and deployment, making it easier to develop custom computer vision applications. Key features include data augmentation, model versioning, and seamless integration with popular cloud services.

One of Roboflow's advantages is that it is primarily a cloud-based service, allowing users to access its tools from anywhere. It offers a free tier, though usage is subject to Roboflow's terms of service and agreements. When annotating videos, users can regenerate labels from previous frames to track objects effectively. Additionally, the platform allows data to be partitioned into training, validation, and test datasets, ensuring a structured approach to model training. Users can also do versioning of datasets, making it easy to manage updates and track changes over time. For video annotation, Roboflow provides the flexibility to label frames individually and choose the frequency of frames to be annotated. Annotations can be exported in various formats, including JSON, XML, and TXT, making integration with other tools seamless. Moreover, the platform supports model training such as YOLO, which streamlines the development process for object detection models.

Despite its many strengths, Roboflow has some limitations. It is a proprietary service built on closed-source software. On the free plan, all uploaded images are public, which may not be suitable for users dealing with sensitive data. Additionally, the paid plans can be costly, and many advanced features are only available with a subscription. Since Roboflow is cloud-based, a stable internet connection is required, and there is no official on-premises installation option. For organizations handling proprietary or sensitive data, uploading information to the cloud may pose privacy and security concerns.

5.3. Label studio

Label Studio is an open-source data labeling tool designed for a wide range of data types, including text, images, audio, video and time series. It is highly customizable, supports various annotation tasks and can be integrated into machine learning pipelines.

One of Label Studio's key advantages is its flexibility. The Community Edition is fully open-source under the Apache License 2.0, allowing users to install and run it on their infrastructure. For those seeking additional features and managed services, an Enterprise Cloud version is available under a commercial license. The tool supports multiple data types and accommodates diverse annotation tasks, such as object detection and classification. Additionally, Label Studio facilitates multi-user collaboration, making it well-suited for large teams working on extensive datasets. With an active community and comprehensive documentation, users have access to valuable support and troubleshooting resources.

However, there are some challenges to consider. Running Label Studio locally can be resource-intensive, particularly when dealing with large datasets or complex annotation workflows. Its performance depends on the underlying hardware and system configuration, which may vary among users. Installation and setup can also be a hurdle for those without technical expertise, as some details in the documentation may be lacking. While Label Studio offers a solid set of features, some advanced functionalities found in commercial tools, such as enhanced security, advanced user management, and priority support, are not included in the open-source version. Additionally, its collaboration features require network connectivity, which can be a limitation in environments with unreliable internet access.

5.4. VIAME

The Video and Image Analytics for the Marine Environment (VIAME) annotation tools are a suite of tools and utilities designed for the annotation, processing, and analysis of video and image data, particularly related to the marine environment. VIAME is an open-source computer vision platform built for users to develop their own AI solutions. Continuously evolving, it offers a comprehensive toolkit with workflows for creating object detectors, full-frame classifiers, image mosaics, rapid model training, image and video search, and stereo measurement techniques.

One of VIAME's biggest strengths is its specialization in underwater and marine data, making it particularly valuable for researchers and marine biologists. The platform is fully open-source under the Apache License 2.0, providing users with unrestricted access to its tools. It offers a wide range of video and image analysis capabilities, including object detection, classification, and tracking. VIAME supports both manual and automated annotation, giving users flexibility in their workflows. The platform includes both web-based and desktop annotation tools, enabling users to choose their preferred working environment. It provides robust annotation options, allowing users to define objects using polygons, lines, points, and bounding boxes. Additionally, VIAME can train models over multiple videos or image sequences using standard AI models. A built-in image enhancement feature runs under the hood, helping to improve visual data for analysis. VIAME supports both GPU and CPU installations and can be configured to run in a cloud environment, provided users set it up on services like AWS, Google Cloud, or Azure. As a self-contained platform, it can be particularly beneficial for researchers who need a dedicated system for their work.

Despite its many advantages, VIAME does have some limitations. As an open-source project, it may not have as large or active a community as some commercial alternatives, which can make troubleshooting and getting support more challenging. The software also requires high computational resources, including a powerful CPU and GPU, especially when processing large datasets or running complex algorithms. Setting up VIAME can be technically demanding, particularly for users who are not familiar with command-line interfaces or programming. Unlike some cloud-based platforms, VIAME does not provide an official cloud-hosted instance, meaning users must install and run it locally on their machines or servers. Additionally, its documentation and tutorials are partially limited compared to other annotation tools. Another limitation is its file format compatibility. VIAME only supports the import and export of its own VIAME CSV, DIVE JSON Annotation formats, and COCO Annotation format, which may restrict interoperability with other platforms.

5.5. VGG image annotator

VGG Image Annotator (VIA) (Dutta et al., 2016; Dutta and Zisserman, 2019) is a popular web-based tool used for annotating images. It is primarily designed to facilitate the creation of ground truth data for training computer vision algorithms, such as object detection, image segmentation, and classification models.

One of VIA's main advantages is that it is completely free to use under the BSD-2-Clause License. As a browser-based tool, it runs entirely within a web browser, eliminating the need for installation and making it accessible across different platforms. VIA supports multiple annotation types, including bounding boxes, polygons, points, and image classification tags, offering flexibility for various annotation tasks. Users can export annotations in multiple formats, such as JSON, CSV, and plain text, ensuring compatibility with different workflows. Additionally, VIA is open-source and allows for team collaboration, making it suitable for projects that involve multiple annotators. Since it is a client-side application, all data is processed locally in the user's web browser, meaning no server hosting is required. This also enables users to work offline, making VIA a convenient choice in environments where internet access may be limited.

Despite its strengths, VIA has some limitations. Although its interface is intuitive, it lacks some advanced features found in more specialized annotation tools. Furthermore, while it works well for small to medium-sized datasets, it may struggle with large-scale datasets due to the performance constraints of web-based applications.

5.6. Hasty

Hasty positions itself as an all-in-one solution for faster AI model creation and deployment, aiming to address three key challenges: slow manual annotations, lack of developer feedback and the need to manage multiple tools. The platform improves annotation speed by 12×, reduces data quality control costs by 90% and streamlines the entire development process.

One of Hasty's biggest advantages is its faster annotation and development process, significantly reducing the time required for data preparation and model training. A standout feature is its real-time training model, which continuously learns and improves as annotations are made. This live learning capability helps enhance both accuracy and efficiency during the annotation process. Additionally, Hasty provides an agile feedback loop, meaning the platform actively learns from user actions and offers insights to refine model performance throughout development.

However, there are some drawbacks to consider. Hasty is a proprietary platform based on closed-source software, which may limit customization and transparency for some users. Although it offers a free plan, its functionality is restricted once users exhaust their virtual credits. Another limitation is its export format options. Hasty may not support all widely used formats, such as YOLO, which can create compatibility issues with other tools. Additionally, compared to some competing platforms, Hasty lacks advanced keyboard shortcuts, which could slow down the annotation process.

5.7. CVAT

Computer Vision Annotation Tool (CVAT) is a widely used open-source annotation tool designed for labeling digital images and videos, particularly for computer vision tasks. It is available under the Apache License 2.0, and users have two main free options that are self-hosted deployment and the limited CVAT Cloud version.

One of CVAT's key strengths is its support for various annotation types, including bounding boxes, polygons, polylines, points, cuboids, and even 3D annotations. It is designed for a range of computer vision tasks, such as object detection, image segmentation, and classification. The tool offers an intuitive and user-friendly interface. A standout feature of CVAT is its automatic annotation capabilities, which leverage pre-trained models for tasks like segmentation, detection, and tracking. This includes advanced models like Segment Anything Model (SAM), YOLOv7, Text Detection v4, and TransT. Additionally, CVAT allows users to integrate their models for customized tasks. The platform also supports serverless functions, enabling users to deploy containerized applications on either GPU or CPU in a local system. CVAT is highly versatile when it comes to deployment, as it supports multiple platforms, including Docker-based installations, making it relatively easy to set up across different infrastructures.

However, despite its strengths, CVAT has some drawbacks. One major limitation of the free CVAT Cloud plan is that it restricts users to a maximum of four annotators, which may not be suitable for larger teams. Furthermore, CVAT's cloud-collaborative features require an active internet connection, which could be a limitation in environments with poor network access. Additionally, the initial setup process can be complex, especially for users unfamiliar with Docker images and containerized environments. Running CVAT can also be resource-intensive, particularly when working with large datasets or automatic annotation models, as performance depends heavily on hardware specifications and system configuration. Another potential challenge is that some Docker containers, such as SAM, are set to "ALWAYS RESTART", leading to unnecessary resource consumption. Users need to manually adjust Docker policies to optimize system performance.

5.8. Labelbox

Labelbox is a user-friendly data labeling platform designed to simplify the creation of training datasets for machine learning models. It provides an intuitive interface that supports various annotation tasks, including object recognition, image segmentation, and text classification, making it a versatile tool for different machine learning projects.

One of Labelbox's key advantages is its flexible cloud-based platform, which is available under various pricing tiers, including a free Community plan with specific terms of service. It is particularly well-suited for large-scale projects, as it allows users to efficiently manage multiple projects and large datasets. The platform offers collaborative tools with role-based access control, making it an excellent choice for team-based annotation workflows. Labelbox provides several annotation and tracking features, such as the ability to add and remove labels as needed, tag frames, and adjust bounding boxes for tracking. Users can choose to annotate videos frame by frame or play the video while labeling, offering flexibility in the annotation process. Additionally, the platform incorporates AI-assisted labeling features that help automate and accelerate the annotation process, improving efficiency. It also includes comprehensive data management tools, such as dataset versioning and quality control mechanisms, and supports multi-domain input, including images, videos, and text.

Despite its strengths, Labelbox has some notable limitations. As a proprietary, closed-source platform, it lacks the transparency and flexibility of open-source alternatives. Additionally, because it is a cloud-based tool, it requires a stable internet connection and does not support offline usage, which may be a drawback for users who need to work in disconnected environments. While Labelbox offers a free tier, many advanced features and higher usage limits require a paid subscription, which can be costly for some users. Another concern for certain organizations is data privacy and security. Since Labelbox operates in the cloud, users must upload their data, which can be a potential issue for those working with sensitive or proprietary information. Furthermore, the platform does not offer an on-premises installation option, making it unsuitable for organizations that require local data hosting.

Table 5
Overview of the annotation tools and their features.

Feature	BIIGLE	Roboflow	Label Studio	VIAME	VIA	Hasty	CVAT	Labelbox	LabelImg	Supervisely
Active learning	–	✓	✓	✓	–	✓	–	✓	–	✓
Video support	✓	–	–	✓	–	✓	✓	–	–	–
Data augmentation service	–	✓	–	✓	✓	✓	–	✓	–	✓
Workflow management	–	✓	✓	✓	✓	✓	–	✓	–	✓
Document version control	✓	✓	✓	–	–	✓	✓	✓	–	✓
Automation and AI assistance	✓	✓	✓	✓	✓	✓	✓	✓	–	✓
Collaboration and review	✓	✓	✓	✓	✓	✓	✓	✓	–	✓
Locally hosted	✓	✓	✓	✓	✓	–	✓	–	✓	✓
Open-source	✓	Paid and open-source	✓	✓	✓	–	✓	–	✓	–
Input format type	Images, videos	Images, videos, datasets	Various	Various (e.g., images, video)	Images	Images, Annotations	Images, videos	Various	Images, Annotations	Images, videos, datasets
Export format	JSON, XML, CSV, TXT etc.	JSON, CSV, XML	JSON, CSV, XML	Various (e.g., CSV, JSON)	CSV, JSON	Various (e.g., JSON, CSV)	JSON, XML, COCO, YOLO, PNG, etc.	JSON, CSV	PASCAL VOC, YOLO, CreateML, XML, TXT	JSON, CSV, XML
Extra feature	–	Model training, deployment	Customizable interfaces	Specialized for marine life	Pre-trained models for images	Automated labeling	Customizable workflows, Automated labeling	Model training, deployment	Simple UI	Model training, deployment
Supported tasks	Detection, Segmentation, Tracking	Detection, Segmentation, Classification	Detection, Segmentation, Classification, NLP	Detection, tracking	Classification, Detection	Detection, Segmentation	Classification, Detection, Segmentation, Tracking	Detection, Segmentation, Classification	Detection	Detection, Segmentation, Classification

5.9. LabelImg

labelImg (Tzutalin, 2015) is a popular open-source tool for graphical annotations, specifically designed for object localization and detection. It supports only rectangular bounding boxes, limiting its functionality to these tasks. However, we still recommend this tool because its sole focus on creating bounding boxes ensures a streamlined and user-friendly experience. labelImg provides all the essential features along with handy keyboard shortcuts for efficient annotation.

5.10. Supervisely

Supervisely is a comprehensive annotation platform designed for a wide range of data types, including images, videos, LiDAR 3D sensor fusion, and DICOM volumes. It enables users to efficiently manage datasets, collaborate on projects, and train neural networks. Unlike many other annotation tools, Supervisely serves as a unified ecosystem that integrates various open-source tools and customized solutions through Supervisely Apps, interactive web applications that run directly in the browser while being powered by Python.

One of the key advantages of Supervisely is its flexibility in exporting annotations to different models, such as YOLO, making it compatible with a broad range of machine learning tools. The platform also offers seamless integration with numerous open-source applications available on GitHub, allowing users to run models, generate synthetic images, and perform various computer vision tasks. Additionally, Supervisely provides shortcuts and advanced tools that help streamline the labeling process, enabling fast and efficient annotation.

However, Supervisely does have some drawbacks. As a proprietary, it is a closed-source platform. Additionally, while the platform is functional, its user interface is not the most intuitive or visually appealing, which may hinder usability for some users. Another limitation is the GPU dependency of its certain integrated apps, restricting access to these features for users without compatible hardware. Furthermore, Supervisely has shifted to a paid model, reducing the storage space available for free users. This change may pose challenges for projects handling large datasets.

5.11. Summary of features per annotation tool

In Table 5, we take a closer look at the comparison of the annotation tools and examine them based on several key features:

- Active learning: Some annotation tools employ active learning techniques that aim to reduce the annotation effort by strategically selecting the most informative data samples for annotation, thus optimizing the annotation process.
- Video support: Tools equipped with video annotation capabilities that allow users to efficiently annotate frames, segments, or entire videos.
- Data augmentation services: Certain annotation platforms provide data augmentation services that allow users to generate augmented versions of their annotated data.
- Workflow management: Users can define sequential or parallel annotation tasks, assign them to annotators and monitor the progress of annotation projects in real-time to ensure efficient collaboration and project management.
- Document version control: Tools that offer robust version control capabilities allow users to effectively manage different versions of their annotated datasets.
- Automation and AI assistance: Platforms that use artificial intelligence (AI) technologies to automate annotation tasks and provide intelligent assistance to annotators can significantly accelerate the annotation process, improve to minimize human error.
- Collaboration and review features: Annotation tools that facilitate collaboration and review between annotators and project stakeholders promote effective communication, feedback sharing, and quality assurance.

CVAT is the mostly used annotation tool among the iMagine project use cases due to its features, such as automatic labeling, which significantly streamlined the annotation process. This led the iMagine project to integrate CVAT into the iMagine Marketplace, making it available for users to deploy the tool directly from the platform.

6. Data repositories and open-source datasets

Open-source datasets play a crucial role in advancing scientific knowledge, supporting policy decisions, and promoting collaboration in marine research and conservation. Each use case within iMagine has created and utilized labeled datasets to train their applications. A key objective of iMagine is to make these image training datasets accessible to external users. This effort aims to enhance transparency by showcasing the images used to train AI models, thereby increasing user confidence and understanding of these models. Additionally, it seeks to enable the reuse of these images for various purposes, such as retraining existing models or training new AI models beyond the consortium (Kozlov et al., 2024).

To achieve this goal, the labeled images used for training the iMagine use cases are cataloged and made available for download via Zenodo. This is the preferred platform for data storage and sharing due to several compelling features. First, it offers long-term storage, supported by the EU, ensuring the preservation and dissemination of research results over time. Second, Zenodo assigns each publication

Table 6
Overview of open-source datasets for aquatic sciences.

Dataset name	Description	Application
FathomNet	It serves as a valuable resource for training, testing, and validating cutting-edge artificial intelligence algorithms aimed at exploring our ocean and its diverse inhabitants. Drawing inspiration from well-known annotated image databases like ImageNet and COCO, FathomNet strives to create a comprehensive reference dataset specifically focused on images depicting marine life (Katija et al., 2022).	Detection
MINKA	It is a citizen science platform designed to collect and share biodiversity and environmental data, with a strong focus on advancing the Sustainable Development Goals (SDGs). It empowers citizen scientists through user-friendly tools, expert-validated observations, and accessible data for researchers and policymakers, fostering community engagement and knowledge sharing (MINKA, 2024). Martínez (2024a) created Python scripts to download pictures from MINKA.	Various tasks
Marine litter assessment ZooScanNet	Aquatic plastic litter dataset developed for APLASTIC-Q publication (Wolf et al., 2021). ZooScanNet: plankton images captured with the ZooScan (Elineau et al., 2024).	Classification Classification
Segmentation masks for ZooScan	Segmentation masks of ZooScan images focusing on images with several objects separated by a human operator (Jalabert et al., 2024).	Detection, Segmentation
EMSO OBSEA EMSO Azores	Labeled Images at OBSEA for Object Detection Algorithms (Baños Castelló et al., 2024). Deep-sea observatories images labeled by citizens for object detection algorithms (Lebeaud et al., 2024).	Detection Detection
EMSO SmartBay	Nephrops (Nephrops norvegicus) Burrow object detection simple training dataset from Irish Underwater TV surveys (Melvin, 2024), SmartBay Marine Species Object Detection Training dataset (Cullen, 2024b), SmartBay Marine Types Object Detection Training dataset (Cullen, 2024a).	Detection
Oil spill detection	Segmented oil spills (Ferrer et al., 2024).	Optimization
FlowCam plankton identification	LifeWatch observatory data: phytoplankton annotated trainingset by FlowCam imaging in the Belgian Part of the North Sea (Decrop et al., 2024), LifeWatch observatory data: phytoplankton annotated image library by FlowCam imaging for the Belgian part of the North Sea (Lagaisse et al., 2024).	Classification
Shoreline extraction	SCLabels: Labelled rectified RGB images from the Spanish CoastSnap network (Soriano-González et al., 2023).	Segmentation
Beach seagrass wrack identification	BWILD: Beach seagrass Wrack Identification Labelled Dataset (Soriano-González et al., 2024b).	Detection, Segmentation
Rip currents detection	RipAID: Rip current Annotated Image Dataset (Soriano-González et al., 2025).	Detection
Freshwater diatoms identification	Large image dataset of freshwater diatoms for training deep learning models (Venkataramanan et al., 2024), Usefulness of synthetic datasets for diatom automatic detection using a deep-learning approach (Laviale and Venkataramanan, 2023).	Detection, Classification
EyeOnWater	EyeOnWater training dataset for assessing the inclusion of water images (Krijger, 2024).	Classification

a Digital Object Identifier (DOI), making the datasets easy to locate and cite. Additionally, the platform provides robust version control, allowing researchers to track changes and to access specific dataset versions as needed. Zenodo also accommodates generous storage needs, offering 50 GB of free storage per publication with the option to request additional space. Finally, it provides detailed usage insights, including download and access statistics, enabling researchers to gauge the reach and impact of their data effectively (Kozlov et al., 2024).

One drawback is that Zenodo serves as a general repository for a wide range of EU results across various disciplines. Its publication template is primarily designed for reports and papers, making it less suitable for describing datasets supported by domain-specific vocabularies. However, iMagine has collaborated with Zenodo as part of the EU-funded Zenodo-ZEN project to develop a more domain-specific approach and created a dedicated template for its training image datasets, structured as a DCAT profile and supported by aquatic vocabularies (Kozlov et al., 2024). This profile was used in dialogue with the Zenodo team, to create extensions to the generic Zenodo metadata template. As a result, more options are now available in the Zenodo upload form, allowing for capturing in more detail important information for AI training datasets.

In Table 6, we provide an overview of the available open-source datasets for aquatic scientists used or produced by use cases of the iMagine project.

7. Preprocessing techniques

Raw data often include noise, corruption, missing values, and inconsistencies, making preprocessing essential. Poor data quality can affect accuracy and lead to false predictions. For example, Gaussian noise, which adds random variations in pixel values, can blur the objects' edges and fine details. Salt-and-pepper noise introduces random black and white pixels, which can obscure important parts of the image. An AI model trained on noisy images might misclassify images because the noise alters the texture and outline, which are crucial for the model's recognition process.

Therefore, preprocessing through cleaning, integration, transformation, and reduction is vital for facilitating and more accurate knowledge extraction. Techniques such as classification, clustering, and association, among others, enhance data quality and improve the performance of AI models (Maharana et al., 2022). These techniques are frequently used in combination, depending on the specific requirements of the AI model, the characteristics of the imaging data and their domain. Here are some common preprocessing techniques for imaging data:

7.1. Data cleaning

Raw data is prepared for analysis through data cleaning techniques that address issues which could negatively impact the model's performance. Data cleaning is an essential part of data preprocessing, focused on fixing or removing errors, inconsistencies, and irrelevant information. It ensures the dataset is accurate, consistent, and complete, helping the model perform better and generalize effectively to new data.

Images might come in different sizes. Resizing them to a uniform size reduces computational complexity and ensures consistency. Furthermore, already trained neural networks require a certain input image size (e.g., ResNet50 requires images of 224×224), that should be fulfilled to use them for prediction of the new images.

7.2. Scaling

Images are represented as matrices of pixel values that are usually unsigned integers ranging from 0 to 255. While these raw values can be fed directly into neural networks, doing so may hinder model performance, often resulting in slower training. Preprocessing of pixel values, such as scaling them to a specific range, centering, or standardizing, helps in numerical stability during training and can improve model efficiency and training outcomes.

7.2.1. Normalization and standardization

Scaling techniques such as normalization and standardization improve model performance, reduce the impact of outliers, and ensure that the data is on the same scale contributing equally to the model. Generally, there is no method other than trial and error to know which technique is the best for a specific dataset.

Normalization scales the pixel values to a range, usually [0, 1] or [−1, 1]. This ensures that all the input data is consistent and within a range that the neural network can process effectively. It reduces computational complexity and helps in faster convergence during training.

Standardization adjusts the pixel values so that they have a mean of zero and a standard deviation of one. This is particularly useful when the dataset has varying lighting conditions and contrasts. It helps in faster and more stable training by ensuring the data distribution is consistent. Standardization can lead to faster and more stable convergence during training.

7.2.2. Traditional and just-in-time scaling

Traditionally, the images would have to be scaled before the development of the model and stored in memory or on disk in the scaled format. The latter approach requires preprocessing the entire dataset before training, which can be time-consuming and resource-intensive, especially with large datasets, but has the advantage that scaling does need to be repeated for every experiment

An alternative approach is to scale the images using a preferred scaling technique just-in-time during the training or model evaluation process. Keras supports this type of data preparation for image data via the ImageDataGenerator class and API. PyTorch supports this type of data preparation for image data via the ‘‘torchvision.transforms’’ module and its various transformation functions. This method allows for on-the-fly data augmentation and scaling, reducing the need for extensive preprocessing and storage.

7.3. Handling imbalanced data

Class imbalance occurs when the number of instances of one class is significantly higher or lower than the instances of other classes. This can cause the model to be biased toward the majority class and perform poorly on the minority class.

To handle imbalanced classes, several techniques are used, such as resampling methods:

- Resampling (Oversampling and Undersampling): Oversampling increases the number of instances in the minority class by duplicating existing instances, augmentation or creating new instances synthetically. Undersampling reduces the number of instances in the majority class by randomly removing instances.
- Class weight adjustment: Modifying the cost function to give importance to the minority class.
- Synthetic data generation: Creating synthetic samples for the minority class using techniques like SMOTE (Synthetic Minority Oversampling Technique).

7.4. Data augmentation

Augmenting data by applying transformations like rotation, flipping, scaling, and cropping helps in increasing the diversity of the dataset, thereby improving the generalization capability of the model and reducing overfitting (when the model memorizes the specific features of the training data set, rather than the underlying patterns of the broader data set). It is important to note that the augmentation technique should transform the training dataset by reflecting realistic and relevant variations of the original dataset. This can then help the model to better generalize without deviating far away from the real scenarios that might be encountered.

7.5. Noise reduction

Noise in images can adversely affect model performance. Techniques such as Gaussian blurring, median filtering, or denoising autoencoders can be used to reduce noise as described in the following.

Gaussian blur (Eswar, 2015) is an image processing algorithm used to smooth the image. The blurring effect reduces sharp edges and creates smooth color transitions at the edges. Gaussian blur, achieved by applying the Gaussian function to an image, creates a normal distribution of pixel values, smoothing out some randomness, reduces noise and minimizes details. It creates an effect similar to viewing the image through a translucent lens. It is often used in preprocessing to enhance image structure and involves convolving the image with a Gaussian kernel matrix.

The median filter (Zhu and Huang, 2012) is a non-linear digital filtering technique commonly used to remove noise from images or signals. It is widely used in digital image processing because it preserves edges while effectively reducing noise, making it a valuable preprocessing step for tasks like edge detection. Unlike a linear filter that averages values, the median filter sorts the pixel values within a specified neighborhood around each pixel and replaces the central pixel with the median value of that group. This technique is especially effective at removing ‘‘salt-and-pepper’’ noise, which manifests as random black and white specks, while preserving the sharpness of edges in the image.

Denoising Autoencoders (GeeksforGeeks, 2024) are a variation of the traditional autoencoder, where the input to the encoder is a noisy or corrupted version of the original data. The encoder, a neural network with hidden layers, processes this noisy input to generate a low-dimensional encoding. The decoder, another neural network, then reconstructs the original data from this encoding. The loss is calculated by comparing the reconstructed output with the original, uncorrupted input. Training with noisy data as input and clean data as the target helps the model to focus on learning meaningful features in the latent space, effectively ignoring the noise, which allows it to reconstruct a clean version of the input.

7.6. Contrast enhancement

Enhancing contrast can improve the visibility of features in the image. Techniques like histogram equalization or adaptive histogram equalization can be employed for this purpose (Raveendran et al., 2021).

7.7. Feature extraction

Depending on the task, extracting relevant features from images can be beneficial. Techniques like edge detection (e.g., Sobel, Canny), texture analysis (e.g., Gabor filters), or deep feature extraction using pre-trained CNNs can be applied.

7.8. Dimensionality reduction

For high-dimensional imaging data, techniques like Principal Component Analysis (PCA) or t-distributed Stochastic Neighbor Embedding (t-SNE) can be used to reduce dimensionality while preserving important information.

7.9. Image segmentation

Segmenting images into meaningful regions can facilitate object detection, recognition, or tracking tasks. Techniques like thresholding, region-growing, or deep learning-based segmentation models can be employed.

7.10. Artifact removal

In aquatic imagery, artifacts like motion artifacts, scanner or underwater camera artifacts can degrade image quality. Data collected during the study of aquatic environments often contains unwanted elements like noise, distortions, or errors, referred to as artifacts. These artifacts can originate from various sources, including measurement equipment, environmental conditions, or data processing techniques, potentially obscuring or distorting the true characteristics of the aquatic system under investigation. Removing these artifacts improves the quality and accuracy of the data, enabling more precise analysis, interpretation, and decision-making. Techniques like interpolation, artifact detection, and correction algorithms can be applied.

8. Performance metrics and evaluation methods

Performance metrics and evaluation methods are essential components of any machine learning project, as they provide insights into the effectiveness and accuracy of the trained models. These metrics quantify the performance of the model and help to assess its suitability for the intended task. In the following, common performance metrics are described.

8.1. Confusion matrix

A confusion matrix is a table that illustrates the performance of an algorithm, typically used in supervised learning classification tasks. It compares the actual target values with those predicted by the machine learning model, presenting the number of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). The rows of the matrix represent the actual classes and the columns represent the predicted classes, or vice versa.

The confusion matrix provides a detailed insight into the performance of a classification model, showing not just overall accuracy but also how well the model distinguishes between different classes, such as correctly identifying positives versus negatives.

- TP: These are the cases in which the model correctly predicts the positive class. If the actual class is positive (e.g., the presence of a disease), and the model also predicts it as positive, it is a true positive.
- FP: These are the cases in which the model incorrectly predicts the positive class. If the actual class is negative, but the model predicts it as positive, it is a false positive.
- TN: These are the cases in which the model correctly predicts the negative class.
- FN: These are the cases in which the model incorrectly predicts the negative class.

8.2. Accuracy

The ratio of correctly predicted instances to the total instances.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{1}$$

8.3. Precision

The ratio of correctly predicted positive observations to the total predicted positives.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

8.4. Recall

The ratio of correctly predicted positive observations to all observations in the actual class.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

8.5. F1 score

The harmonic mean of precision and recall, useful for imbalanced datasets.

$$F1score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

8.6. ROC-AUC

ROC Curve (Receiver Operating Characteristic Curve): A graphical plot that illustrates the diagnostic ability of a binary classifier system. AUC (Area Under the ROC Curve): Measures the entire two-dimensional area underneath the entire ROC curve.

8.7. Mean Square Error (MSE)

The average of the squared differences between predicted and actual values. A lower value indicates better performance.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{5}$$

8.8. Mean average precision (mAP)

It measures the accuracy of a model in detecting objects and localizing them within an image. A higher mAP score signifies better performance.

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i \tag{6}$$

, where AP_i is the average precision for the ith class. AP is the area under the precision–recall curve for a single class. It summarizes the precision–recall curve into a single value representing the average precision at different recall levels.

8.9. Intersection over Union (IoU)

Intersection over Union (IoU) measures the overlap between two boxes, with greater overlap indicating a higher IoU. It is primarily used in object detection to train models to accurately predict bounding boxes around objects. For instance, a green box (correct) and a blue box (predicted) should ideally overlap perfectly, achieving an IoU of 1. IoU is also employed in non-max suppression to eliminate redundant boxes around the same object. In this method, the bounding box with higher confidence is selected and all remaining bounding boxes that have an IoU less than a predefined IoU threshold with the selected box are removed.

8.10. Minimal Intersection over Union (Min IoU)

Minimal Intersection over Union (Min IoU) is the area of overlap of two bounding boxes divided by the minimum area of the two bounding boxes. This measure is helpful while trying to understand how bounding boxes are overlapping each other, and execute a cleaning algorithm similar to Non-Maximum Suppression.

8.11. Cross-Entropy (CE)

It quantifies the difference between the predicted probability distribution generated by the model and the true probability distribution,

which is usually represented by a one-hot encoded vector indicating the correct class. CE is often used as a loss function during model training. For a single sample, the CE loss can be expressed as:

$$H(p,q) = \sum_{i=1}^{c} p_i log(q_i) \tag{7}$$

, where p_i is the true distribution, q_i is the predicted probability for class i, and c is the total number of classes. For a dataset with multiple samples, the CE loss is averaged across all samples.

8.12. The Fraction Skill Score (FSS)

It was originally introduced in the context of probabilistic weather forecasting analysis (Roberts and Lean, 2008). It evaluates forecast accuracy by accounting for the uncertainty and variability of weather events. Specifically, the FSS measures the skill of probabilistic predictions by comparing them to a ground truth (e.g., satellite observations). Unlike a traditional overlay method, this metric considers not only the common area between simulation and observation but also the spatial distribution within that area. The FSS is calculated as:

$$FSS = 1 - \frac{\sum_{i=1}^{n}(f_i - o_i)^2}{\sum_{i=1}^{n}(f_i^2 + o_i^2)} \tag{8}$$

, where f_i is the forecast fraction, o_i is the observed fraction, and the summation runs over all grid points i. The FSS ranges from 0 to 1, where 1 indicates a perfect match between the forecast and observation, and 0 indicates no skill.

9. Monitoring model performance

While assessing various preprocessing techniques, AI models, choosing right metrics etc., it is essential to keep track of these experiments and compare them. Experiment tracking tools are software platforms or frameworks designed to help researchers and data scientists efficiently manage and monitor their machine learning experiments (Berberi et al., 2025). These tools provide features for organizing, recording, visualizing, and analyzing experimental data, making it easier to track the progress of experiments, compare results, and reproduce findings. Here are some popular tools for tracking experiments.

9.1. TensorBoard

TensorBoard is a powerful tool that provides the visualization and tooling needed for machine learning experimentation. It was originally developed for use with TensorFlow, but it has since been adapted to work with other deep learning frameworks, including PyTorch. TensorBoard allows you to visualize various metrics, graphs, and other details about your model.

- Visualization: Provides detailed visualizations of metrics such as loss and accuracy to help understand model performance.
- Graph Analysis: Allows you to view the calculation graph and helps with troubleshooting and optimization.
- Embeddings: Projects embeddings into low-dimensional spaces for easier interpretation.
- Flexibility: Supports visualizations of histograms, images, text and more.
- It can be self-hosted solutions.

9.2. MLflow

MLflow simplifies machine learning development by integrating experiment tracking, packaging code into reproducible runs, and facilitating model sharing and deployment. It provides lightweight APIs that are compatible with any ML application or library (such as TensorFlow, PyTorch, XGBoost) and can be used in various environments such as Jupyter notebooks, standalone applications, and the cloud. Key features of the tool are:

- Model Registry: MLflow includes a centralized model registry for managing and versioning models.
- Experiment Tracking: With MLflow, you can log parameters, metrics, artifacts (e.g., model files) and other metadata from your machine-learning experiments.
- MLflow Projects: Packaging ML code in a reusable and reproducible form to share with other data scientists or transfer to production.
- MLflow Models: Managing and deploying models from various ML libraries to multiple model serving and inference platforms.
- It can be self-hosted.

In the context of iMagine, we have deployed an MLflow tracking server primarily for experiment tracking. We added another layer on top of the MLflow server that enables users to self-register in MLflow via the project common AAI (EGI Check-In) and to manage permissions to their experiments and models with other users. We implemented automatic backup of the in-use databases and enabled manual restore operations. The code related to the dockerized solution is available at Laures et al. (2024).

9.3. Weights & Biases

Weights & Biases (W&B) provides a comprehensive platform that facilitates experiment tracking, visualization and collaboration on machine learning and artificial intelligence projects.

- Experiment Tracking: W&B enables users to comprehensively track machine learning experiments. It logs metrics, hyperparameters, and other relevant information during model training, providing a clear record of experiment configurations and results.
- Visualization: The platform provides interactive visualizations that help users identify trends in model performance, compare experiments and identify areas for improvement.
- Collaboration: Teams can collaborate effectively via W&B by sharing experimental results, visualizations, and insights. This promotes transparency and knowledge sharing within research and development teams.
- Integration: W&B integrates seamlessly with popular machine learning frameworks such as TensorFlow, PyTorch and scikit-learn. This integration simplifies the logging and visualization of experiments directly from these frameworks.
- Dashboard and project management: Users can manage their machine learning projects and experiments via a user-friendly dashboard. It provides tools to efficiently organize, search, and analyze experiment data.
- Hyperparameter optimization: it provides capabilities for hyperparameter optimization (HPO) as part of its suite of experiment tracking and management tools.
- It cannot be self-hosted solutions.
- It has various payment plans, the Free one is limited, but there is one for academia.

TensorBoard and MLflow are flexible tools that allow for self-hosting, offering users enhanced control over their deployment and data management. In contrast, W&B does not offer a self-hosting option, as it operates as a cloud-based service. W&B offers a range of payment plans tailored to different user requirements, including a limited free tier. Additionally, W&B provides a dedicated plan for academia, specifically designed to support researchers and students working on machine learning projects.

10. Data biases and fairness in models and data

Generally, scientific models rely heavily on data to make predictions, draw conclusions and make decisions. However, these models are

not immune to biases inherent in the data on which they are based, which can lead to unfair predictions. Understanding data bias and ensuring fairness in models is important to the integrity and reliability of the predictions and recommendations they provide.

Data biases arise when modeling or analysis data fail to accurately reflect reality. In aquatic sciences, for instance, Phenotypic variability of natural systems, where a species may have different morphologies depending on its age and location, can introduce substantial biases into aquatic datasets. These biases affect population estimates, community analyses, and the evaluation of ecosystem health. Data biases can stem from:

- Sampling Bias: Certain regions or species may be over- or under-represented, leading to inaccurate depictions of less-studied or remote areas. In other words, data collection does not incorporate adequate randomization that results in class imbalance. This imbalance biases the model toward the majority class, as it encounters and learns from it more frequently. This leads to underperformance when predicting the minority class, resulting in poor generalization and skewed predictions.
- Measurement Bias: Errors in data collection methods, such as sensor inaccuracies or inconsistent sampling, can distort results.
- Temporal Bias: Data collected at specific times may overlook seasonal variations or long-term trends, impacting models that do not account for these changes.
- Geographical Bias: Some areas may be over-represented due to ease of access or historical research focus, leading to an incomplete understanding of broader or regional conditions.
- Group attribution bias: The tendency to generalize observations or characteristics from individual aquatic organisms or specific events to an entire species or ecological group, that can result in oversimplified or inaccurate conclusions.
- Morphological Bias: Ontogenetic alterations, can introduce size bias in datasets, as sampling may favor certain age groups. Phenotypic, morphology or behavior variability in response to environmental conditions, can lead to misinterpretations of size or age distributions if these environmental effects are not accounted for. Additionally, sexual dimorphism, with males and females differing in size or behavior, can cause sex-based bias if sampling methods do not differentiate between sexes or favor one over the other.

10.1. Addressing fairness and reducing bias

Fairness in aquatic sciences models involves making sure that predictions and analyses are just and do not disproportionately impact any group or area. Implementing strategies to ensure fair research practices minimizes the influence of subjective biases. These strategies include:

- Equitable Representation: Ensuring that data from all relevant regions, species, and conditions are inclusively represented to create models that generalize well.
- Impact Assessment: Analyzing how model results influence different stakeholders, such as local communities and conservation efforts.
- Bias Mitigation: Identifying and addressing biases through techniques like re-sampling, applying domain expertise, or using statistical adjustments to ensure a balanced dataset. Conducting studies across multiple populations and habitats that include a wide range of species to provide a more comprehensive understanding of aquatic ecosystems. Standardization of sampling methods across space and time, collecting environmental data (e.g., temperature, salinity) alongside biological samples, and using multiple sampling techniques to capture a full range of sizes and life stages can minimize morphological biases. For class imbalance bias reduction, see Section 7.3.
- Transparency and Open Science: Openly available research allows verification and further study by the scientific community, facilitating thorough and unbiased peer review processes to critically assess research findings and methodologies.

In 2016, a basic framework of principles known as FAIR (Findable, Accessible, Interoperable, and Reusable) was introduced to enhance the management and governance of data and facilitate the reuse of scientific information (Wilkinson et al., 2016). Originally designed for scientific data, these principles have since extended to various digital assets, including AI models and datasets.

To improve the management and sharing of research data, FAIR EVA (Aguilar Gómez and Bernal, 2023), has been developed within the European Open Science Cloud context. It is designed for specific data management systems, such as open repositories, which can be tailored to individual use cases in a scalable and automated environment. The tool aims to be flexible and adaptable, supporting various environments, repository software, and disciplines, in line with the flexibility of the FAIR Principles. FAIR EVA tool in the context of iMagine has been used to improve FAIRness of published training datasets.

11. Model delivery

The accessibility and reusability of trained AI models are important for researchers to foster collaboration and accelerate innovation. Once the AI model is trained and validated, it can be shared on the iMagine Marketplace and deployed to the aquatic image service in the production environment. In this section, we have identified six different model deployment patterns to offer AI models in production. For each one, we detail the best practices to help use cases with the adoption of the tools and services available in the iMagine AI Platform that best fits their needs. Additionally, the trained and validated models can be offered and made accessible to external users through different approaches. We refer to them in this section.

11.1. Model sharing via iMagine marketplace

The simplest option is to provide the trained models via the iMagine Marketplace component of the Platform. The trained and validated models are integrated with API (recommended is DEEPaaS API (López García, 2019), REST API for AI/ML/DL) and dockerized. This enables external users to download the AI modules as Docker images and run them on their own or third-party external compute resources. This pattern corresponds with approach 2, “Marketplace download service delivery”. In addition, the Marketplace offers “Marketplace inference service delivery” to run trained AI models for inference on the connected back-end cloud resources. The latter is also available for non-partners via short-lived (10 min) “Try” endpoints.

11.2. Processing files in an event-driven approach

To support this pattern, the iMagine project provides the OSCAR framework (Pérez et al., 2019; Risco et al., 2021). OSCAR is an open-source platform built on Kubernetes for event-driven data processing of serverless applications packaged as Docker containers. The execution of these applications can be triggered both by detecting events from object-storage systems, such as MinIO or dCache (asynchronous calls) or by directly invoking them (synchronous calls). The main benefit of this approach is the scalability of the inference jobs, as the OSCAR cluster can autoscale, adapting its size to the workload transparently for the users (and freeing up the resources when they are not needed). For this pattern, producers and providers of the aquatic analysis service can choose between two options:

- Deploy a model in the project's OSCAR serverless inference platforms: the iMagine OSCAR clusters support multitenancy and offer accounting thanks to the Prometheus and GoAccess services. They can collect metrics like resources consumed by inference executions, the number of deployed services over a period of time or the geolocation of the users interacting with the services. With this approach, models can be made accessible through the iMagine Marketplace component of the Platform. This allows users to choose and run the trained AI models for inference on the connected back-end cloud resources of the iMagine infrastructure (corresponds with approach 1, "Marketplace inference service delivery"). Moreover, the main advantage of this option is that the operational responsibility of the platform is on the project. However, users will need an EGI Check-in account to access the OSCAR platform and the computing resources provided for inference are limited. Currently, there are two OSCAR clusters available for the iMagine project, called OSCAR Inference and OSCAR Inference-walton.
- Deploy your OSCAR cluster: use case partners can decide to have their instance of the OSCAR framework, deployed by the project partner on computational resources from the iMagine consortium or even on third-party external compute resources. Support and documentation are offered in this case to deploy the OSCAR open-source platform, but the management of the platform relies on the project partner.

11.3. Processing streams of data

For applications that need to process streams of data, and want to avoid the phase where the algorithm loads the weights of the AI model for each file, the iMagine project offers mainly two options:

- OSCAR exposed services: OSCAR allows the secure deployment of auto-scaled services that expose their API, so users can directly interact with it. This is recommended for applications already integrated with the DEEPaaS API. Nevertheless, partners can also decide to design their own service presentation layer and model execution scheme and offer it through OSCAR exposed services. With the latest developments, authentication via Ingress has been enabled to better secure access to exposed services.
- The training Nomad cluster: partners can also deploy an inference endpoint at the training Nomad cluster.

With these approaches, models will only load the weights once. Thus, the primary advantage of this approach is its readiness; once the endpoint is established, the model remains loaded in memory, enabling rapid responses to inference queries. However, these methods do not release CPU and RAM resources, as they remain allocated even when the service is idle. Therefore, these approaches are particularly useful for applications that require real-time predictions rather than batch processing. Notice that this pattern is related to approach 1, "Marketplace inference service delivery" and 3, "Inference service delivery".

11.4. Processing historical data

Another pattern we have detected is the need to process historical data, consisting typically of a significant number of files that need to be processed with the AI model. For this approach, partners can benefit from OSCAR Batch. This is a tool designed to perform batch-based processing using the OSCAR framework. OSCAR Batch includes a coordinator service where the user provides a MinIO bucket containing files for processing. This service calculates the optimal number of parallel service invocations that can be accommodated within the cluster and distributes the image processing workload accordingly among the services. It ensures the efficient use of available CPU and memory resources in the OSCAR cluster. Notice that both the MinIO storage instance and the OSCAR cluster are provided for iMagine partners.

11.5. Composing inference pipelines involving several AI models

Additionally, iMagine project partners can also benefit from low-code composition tools, to graphically compose inference AI pipelines through the AI4Compose component. AI4Compose is a framework responsible for supporting composite AI by allowing the workflow composition of multiple inference requests to different AI models. This solution relies on Node-RED and Elyra, two widely adopted open-source tools for graphical pipeline composition, employing a user-friendly drag-and-drop approach. Node-RED, in combination with FlowFuse to support multitenancy, serves as a powerful graphical tool for rapid communication between different services; meanwhile, Elyra provides a visual Notebook Pipeline editor extension for JupyterLab Notebooks (support available in EGI Notebooks) to build notebook-based AI pipelines, simplifying the conversion of multiple notebooks into batch jobs or workflows.

AI4Compose is integrated with OSCAR, being able to invoke the services pre-created in the platform (users can use for that the project's OSCAR serverless inference platforms, or their OSCAR instances). The integration with OSCAR is made through flow and node implementations offered as reusable components inside both Node-RED and Elyra visual pipeline compositors. With AI4Compose, users will gain agility and resource efficiency as they can leverage the management of the computing platform to OSCAR, which provides a highly scalable infrastructure to support complex computational tasks. Furthermore, AI scientists can easily design, deploy and manage their workflows using an intuitive visual environment, reducing the time and effort required for the maintenance of inference pipelines.

11.6. Retraining AI models

Finally, users may request a retraining of an already trained AI model using their data to make the model more precise for the specific classification/prediction cases they are facing. This can also be supported by the project, which corresponds with approach 4, "Retraining service delivery".

12. Drift tools

In aquatic imaging services, such as those used to monitor marine life or underwater environments, conditions such as lighting, water clarity, dirt on the camera and species behavior can vary significantly. These variations can lead to a deterioration in model performance if not properly controlled. Drift tools are important in AI, particularly in production settings and aquatic image services, as they identify and manage the gradual decline in model performance. In AI models, data drift happens when the distribution of input data shifts from the model's original training data, causing a drop in accuracy and reliability. Drift tools enable continuous monitoring and allow early detection of shifts in data distribution or model performance. This ensures that models remain accurate and reliable, leading to more effective AI systems in such a dynamic environment.

Frouros (Sisniega and García, 2024) is a drift detection tool for machine learning systems that was developed as part of the AI4EOSC project. It aims to identify significant changes during inference, such as concept drift (changes in the learned concept) or data drift (changes in feature distributions), which can degrade model performance. By detecting these changes, users can determine whether the model predictions are unreliable due to changing feature-target relationships, data changes or problems in the data acquisition pipeline.

Frouros is implemented in Python and supports both concept and data drift detection with 32 detectors, including classical and state-of-the-art methods. It outperforms existing libraries in the variety and number of detectors offered. Frouros is designed for simplicity and offers:

- Datasets: Real or synthetic datasets for testing purposes.

- Detectors: Organized into categories for concept drift and data drift.
- Callbacks: Allows execution of custom code at key stages, similar to PyTorch Lightning and Keras.
- Metrics: Contains evaluation metrics such as prequential error metrics to assess detector performance.

Frouros provides an easy-to-use, comprehensive toolkit for ensuring the reliability of AI models in dynamic environments.

It was also found that for the case of the Video Quality Assessment, the pre-trained DOVER VQA model (Wu et al., 2022, 2023a; Wu, 2022) may be of great use.

13. Real-world use cases

In this section, we provide an overview of our mature and prototype real-world use cases, carried out by relevant user communities, and their experience in developing and implementing the best practices outlined in this manuscript.

13.1. Marine litter assessment

In this use case, the main objective is to classify drone images using a tile-wise approach. This process consists of two steps: first, detecting litter accumulations, and then identifying the specific litter categories present. A key aspect of this task was aligning the original classification categories with EU guidelines for litter monitoring strategies, while refining the previously existing processing methodology.

For data labeling, a pre-labeled dataset was used, and tests were conducted to align these initial categories, designed for AI applications and drone images, with official EU litter lists, particularly the JLIST (MSFDTechnicalGroup, 2024). The original objective was to translate the existing labels into JLIST categories. However, test results revealed that the detailed JLIST, tailored for manual on-site beach litter assessment, is neither suitable nor translatable for use with drone footage. This is due to factors such as image resolution, which is insufficient for detailed manual identification. Additionally, the images contained numerous overlapping objects and object fragments that could not be identified solely from the images.

The main performance metrics considered were the standard ones for classification tasks, including Accuracy, Precision, Recall, and F1-Score. In terms of preprocessing, the input tiles were resized to standard dimensions of 128×128 and 64×64 pixels. Augmentations were also applied, including horizontal flips, vertical flips, and random rotations. No specific cleaning of the dataset or handling of noise and anomalies was performed. The impact of the dataset's unbalanced nature on model performance was investigated by comparing the Precision, Recall, and F1-Score for each individual class. The results showed stable and consistent values for both underrepresented classes and those with a larger number of images in the dataset. Therefore, no data balancing techniques were applied.

For classification tasks, two CNNs were used to detect and identify plastic litter. Four state-of-the-art CNN models for plastic litter detection were tested, comparing larger models with smaller ones in terms of their number of parameters. Among the smaller, more compact models, MobileNetV2 demonstrated higher performance than SqueezeNet1.1. In the category of more complex CNN models with more tunable parameters, the results indicated that DenseNet121 outperformed ResNet50.

The training datasets are published on Zenodo, and the trained AI models are shared via the iMagine Marketplace. For deployment, standard strategies within the iMagine project are utilized, with the processing methodology and model inference available as a Docker container on AI4OS DockerHub and GitHub. Additionally, the models are integrated into OSCAR for user-friendly inference runs. Currently, a drift detection tool has not been implemented, as a continuous data stream requiring such monitoring is not anticipated.

Analogous to the improvement of the plastic litter detection model through the use of more modern CNN architectures, the plastic litter quantification model will similarly be improved in upcoming work by adopting more up-to-date model architectures. In the longer term, expanding the datasets may represent an important direction with the aim of providing a comprehensive and diverse dataset to facilitate broader use by external users. This includes the use of different imaging systems, such as different drones or stationary cameras, with varying resolutions and orientations to the water surface. The identification of individual litter objects could also be a useful extension for environmental conditions that do not contain large accumulations of litter. However, detecting individual objects requires alternative machine learning approaches, such as segmentation or object detection, which could be explored using additional and more diverse datasets in the future.

13.2. ZooScan - EcoTaxa pipeline

The service is a complete rewrite of the existing ZooProcess software, utilizing modern libraries. It incorporates AI components to 1. sort images containing multiple objects and 2. separate touching plankton objects in such images, as the subsequent analysis, classifying images into detailed taxonomic groups and measuring specific features, requires only one object per image.

Processing begins with a large scanned image of a preserved plankton sample on a dedicated instrument (the ZooScan). Initial preprocessing steps on the full images, such as background subtraction, simplify segmentation by enabling the use of basic thresholding techniques. However, this approach can lead to under-segmentation when objects are in contact. In such cases, semantic, instance, or panoptic segmentation models offer improvements over regular computer vision methods.

The first step is to identify under-segmented images by training a binary classifier to distinguish between images containing a single object and those containing multiple objects. A MobileNet network pretrained on ImageNet was used and fine-tuned with the use case dataset. MobileNet was selected based on prior experience in classifying such images, which indicated that small and efficient networks are sufficient to capture the necessary characteristics in relatively small, sparse, and grayscale plankton images.

To prepare the training set, we took advantage of existing datasets in which images of multiple objects were already sorted separately from the rest. The preprocessing steps, before the network, were standard data augmentation (rotation, resizing, cropping, etc.) and the resizing to a common input size was done dynamically, as part of this data augmentation. We used just in time scaling because it was easy to implement that as one operation within the data augmentation chain. We repeated the experiments a few times, but since model training was still relatively short, it was not relevant to scale all images beforehand.

The dataset was imbalanced, containing significantly more images of single objects than of multiple objects. To address this, a combination of resampling and weights was used to bias the classifier toward predicting multiple objects. The objective was to maximize recall for these instances, ensuring that as many as possible were correctly identified for the next step, that is, segmentation.

The evaluation metrics included binary cross-entropy (the loss function used for model training), overall accuracy, recall, and precision of the "multiple" class. No specific tool was used to track the experiments, as the training process was relatively straightforward.

For segmentation, instance segmentation using MaskRCNN and panoptic segmentation using Mask2Former were tested, with the latter yielding better results. However, the masks produced by deep networks do not align perfectly with the object contour pixel-per-pixel, which poses challenges for extracting contour-based shape information. Additionally, they are less reproducible than simple threshold-based segmentation. To combine the intelligence of a deep model with the

reliability of a deterministic computer vision approach, deep masks were used to define the centroids of regions. These regions were then expanded using the Watershed algorithm to cover all non-background pixels based on simple threshold segmentation. This approach ultimately produces masks with borders that match threshold-based segmentation, while ensuring that different regions, corresponding to the original detections of the deep network, remain properly separated as distinct objects.

For classification, the training dataset was assembled from existing data, in which human operators had manually traced white lines to separate touching objects. This provided a ground truth indicating multiple objects and their correct separations. All examples in the dataset were utilized, as they represented realistic scenarios, including more extreme cases. Furthermore, it was acknowledged that rarer cases (e.g., large clusters of objects sticking together) were not well handled in the predictions. However, rather than removing these cases from training, the preferred approach is to include more examples of such instances to create a more balanced dataset. No data augmentation has been applied, and no weighting schemes for segmentation are currently in use.

The dataset was published on the SEANOE platform, with a metadata record in Zenodo (Elineau et al., 2024; Jalabert et al., 2024). The evaluation metrics included Intersection Over Union (IoU) relative to the manually defined ground truth, as well as the difference in the number of retrieved objects between the ground truth and the results after applying the Watershed algorithm. No specific experiment tracking tool was used, instead, the results of various attempts were recorded in a comprehensive spreadsheet.

For both components (classification and segmentation), the trained models and associated code will be shared following standard procedures in iMagine, including a GitHub repository within the ai4os-hub organization and the Marketplace. For the production service, the dedicated inference service through OSCAR is utilized, as it provides the fast and scalable inference capabilities required for this use case. Users will access this service through a third software component, which is also under development. This component will feature a Graphical User Interface (GUI) to facilitate scan acquisition, metadata entry, AI component invocation, and result upload for further processing in the EcoTaxa tool (Picheral et al., 2017).

Drift detection tools have not yet been considered. However, at a later stage, the model trained on ZooScan images from one type of plankton net will be applied to images from another plankton net. While the imaging instrument and settings remain unchanged, ensuring consistent output, the two nets capture different taxa and organisms of varying sizes. The extent of the expected performance decline will provide insights into the generalization potential of the current model. If performance decreases by more than a few percentage points in IoU, a single model will be retrained using example images from several nets.

13.3. Marine ecosystem monitoring at EMSO OBSEA

OBSEA is an underwater observatory that has been acquiring multiparametric data since 2009. Over the years, various cameras have been deployed, resulting in a vast archive of images dating back to 2011. Historically, this data has been manually analyzed to extract ecological information about the fish community in the area, a process that is highly time-consuming and repetitive.

The primary objective of this use case is to apply AI techniques to automate the detection and classification of fish specimens captured by OBSEA's cameras and generate a time series of fish detections. This approach enables scientists to efficiently process the data, allowing them to focus on research rather than manually counting fish in images. Despite the availability of extensive historical data, many earlier cameras suffered from biofouling issues and low resolution. However, in July 2023, several new HD cameras were deployed. Leveraging these cameras, a new dataset with high-quality images has been generated.

For image labeling, several tools were tested, including Roboflow, Label Studio, and BIIGLE. Although some of these tools offer useful features, most exhibited issues such as bugs, slow performance, or subscription requirements. Consequently, LabelImg was selected as the preferred solution. This simple yet effective Python package facilitates image labeling. While LabelImg does not offer advanced features at first glance, its simplicity allows for easy integration with custom scripts for dataset management and semi-automatic labeling.

After labeling a small dataset, several AI algorithms were tested, including Faster RCNN and YOLOv8. The latter provided an optimal balance between precision and speed, making it well-suited for this application. Additionally, the different variations of the algorithm (Nano, Small, Medium, Large, and XLarge) allow for training models with varying speed-to-precision ratios. In this case, the Nano version is used for real-time video inference for dissemination purposes, while the XLarge version is employed for slower but more precise inference in images for scientific applications.

To cover the seasonality of the fish community and minimize biases in the training data, a dataset of one year is being collected from July 2023 to July 2024 using newly installed HD cameras. Due to the nature of the ecosystem, the dataset is initially highly imbalanced, as species that form the base of the trophic chain appear hundreds or even thousands of times more frequently than large predators. To mitigate this imbalance, the dataset has been supplemented with images of uncommon species from the MINKA community (MINKA, 2024). Currently, the dataset consists of 5354 images classified into 21 classes, with more than 35,000 labeled instances (Baños Castelló et al., 2024).

The algorithm was trained on the iMagine platform using the default module and the command-line interface. To enhance model performance, YOLOv8's built-in data augmentation techniques were applied. The hyperparameters for data augmentation were optimized according to the official documentation. Once an initial functional version of the algorithm was developed, an inference module named "OBSEA Fish Detection" (Martínez, 2024b) was uploaded to the iMagine Marketplace.

To compare different versions of the algorithm, the mAP@50 metric was selected. This metric represents the mean of the average precision for all classes with an IoU threshold of 50%. In this application, both precision and recall are equally important, making mAP@50 a suitable metric for the evaluation of overall performance. To follow the experiments, ClearML and MLflow platforms were used to monitor the training process. The latest experiment achieved an mAP@50 score of 0.83. The trained model is publicly available at Zenodo (Baños et al., 2025).

The most challenging aspect of data preparation was managing the characteristics of underwater images, particularly water turbidity. Turbidity in seawater causes objects located a few meters away from the camera to appear blurred, making it difficult even for the human eye to clearly identify fish specimens. As a result, fish often appear as indistinct smudges in the background. To address this challenge, a trade-off was made by labeling only those fish that exhibited identifiable features suitable for species classification (e.g., tail shape, stripe patterns, etc.).

The next steps involve deploying two models into production: a full-fledged version for scientific data production and the Nano version optimized for faster inference, enabling real-time video analysis at the cost of reduced performance. In addition, inference is planned for the entire data archive, starting from 2011, to extract biological information. For this task, the OSCAR framework will be utilized for large-scale batch inference.

13.4. Marine ecosystem monitoring at EMSO azores

The dataset is from the "Deep Sea Spy" project (DeepSeaSpy, 2024), in which images of deep-sea habitats are annotated by citizen scientists.

The objective is to automatically identify animals based on these annotations by training a deep learning neural network (YOLOv8). Since the annotations were provided by citizens, it is necessary to address challenges specific to citizen science. Therefore, a data cleaning step is essential to ensure that the YOLOv8 model can effectively learn from the dataset.

The untouched dataset consists of 3979 images and 253,323 annotations, with 3403 images originating from the Juan de Fuca Ridge and 576 from the Azores. With 15 different species represented, a certain degree of data imbalance is unavoidable, as citizen annotations tend to favor species that are easier to identify. Therefore, the next steps will focus on two specific species, Buccinidae and Bythograeidae, for several reasons. First, these species belong to different habitats, with Buccinidae found in the Juan de Fuca Ridge and Bythograeidae in the Azores, making them the most annotated species in their respective regions. Second, the dataset imbalance makes it challenging to train a multiclass model that performs well across all species. Additionally, the quality of the images may lead to identification confusion between species within the same habitat. Third, species that are more difficult to identify tend to have a higher rate of incorrect annotations from citizen scientists, which negatively impacts the accuracy of the trained model. Furthermore, because the images differ significantly between the two habitats in terms of layout, composition, and species distribution, it is essential to have one representative species from each habitat. Lastly, each species may require specific data cleaning, as bounding box areas and annotation densities vary. Focusing on a limited number of species will allow for more effective cleaning strategies without compromising data quality.

Corrections were made to the original annotations to address identified issues and enhance the model's performance. Specifically, the labeling of bounding boxes was adjusted from lines to rectangles while ensuring that line angles did not result in unusable rectangular bounding boxes. Additionally, a padding of 25 pixels was applied to one of the two target species, Bythograeidae, as the trained YOLOv8 models demonstrated improved performance with this adjustment.

The primary challenge encountered was annotation redundancy. To address this, a two-step cleaning approach was implemented, leveraging redundancy as a means of cross-validating the presence of an animal. First, the pipeline unifies bounding boxes based on an IoU threshold. Bounding boxes with an IoU of 0.6 or higher relative to any other bounding box are retained, while the others are discarded. Next, the minimal IoU between overlapping bounding boxes is measured. If the IoU is 0.4 or higher, the bounding boxes are kept in the final dataset. Through this process, the dataset was reduced to 20,979 annotations. These specific IoU thresholds were determined through extensive testing, visual validation, and model performance evaluation. The best-performing models were selected by experimenting with different cleaning parameters. Model experiments are tracked by storing files locally and maintaining documentation of the work performed.

It is planned to conduct automatic identification on other species. However, it may be more effective to wait for additional images or annotations before proceeding. The untouched Buccinidae dataset originally contained 98,282 annotations, which were reduced to 14,662 after the two-step cleaning process. The untouched Bythograeidae dataset initially had 2426 annotations, which were reduced to 305 following the same process. The difference in the number of annotations between Bythograeidae and Buccinidae is primarily due to the disparity in the quantity of available images.

To evaluate performance, the results of YOLOv8 model training were analyzed. The metrics used for evaluation included Precision, Recall, and mAP@50-95.

For the training on Buccinidae, expert annotations were used as validation for the model trained on citizen annotations. The best-performing model achieved a Precision/Recall of 0.6 but exhibited low confidence in its detections, with an mAP@50-95 of 0.15 compared to an mAP@50 of 0.5, indicating higher precision at lower confidence levels.

For the training on Bythograeidae, no expert annotations were available for validation; therefore, a portion of the citizen-annotated dataset was used instead. The best model achieved a precision/recall of 0.4 and, similar to the Buccinidae model, demonstrated low confidence in its detections, with an mAP@50-95 of 0.1 compared to an mAP@50 of 0.3, again suggesting higher precision with lower confidence.

The dataset and trained models are available on SeaNoe platform, with a metadata record in Zenodo (Lebeaud et al., 2024) to ensure accessibility. The AI models will be deployable through the iMagine module named Deep Sea Detection. The use of a data drift detection tool is not being considered, as species detection is highly specific to the sample location. No other deep-sea sample location has the same species repartition as the one in this use case.

13.5. Marine ecosystem monitoring at EMSO SmartBay

In EMSO SmartBay, three use cases are being examined: marine species detection at the SmartBay Underwater Observatory, video quality assessment of the SmartBay Observatory video feeds and archive, and Nephrop burrow detection for prawn fishery surveys. In the marine species detection and prawn burrow use cases, a self-hosted instance of CVAT has been used for image annotation. Images are collected and uploaded to a self-hosted instance of MinIO object store (MinIO, 2024). In MinIO, the images are stored in S3-compatible storage buckets, which function as cloud storage accessible via HTTP URLs and are added to CVAT as S3 storage locations. Folders of images are then used to create annotation projects and annotation tasks. The images in a task are manually annotated within the CVAT web interface using bounding boxes labeled according to the object classes defined in the CVAT project. Once annotation tasks are completed, the annotation data (images and labels) can be exported from CVAT in COCO format. The task datasets are then uploaded to Roboflow projects for collation, training dataset analysis, and preparation for export in YOLOv8 format.

The focus has been on training YOLOv8 object detection models for marine species and Nephrop burrow data. An on-premise GPU has been used for training, with plans to utilize the iMagine platform for additional model training runs. Additionally, the MLflow instance on the iMagine platform has been introduced to record model training runs. This tool facilitates the collection of metrics on training and model performance, as well as to explore which dataset and training optimizations can improve model performance.

In EMSO SmartBay, obtaining published versions of annotated North East Atlantic marine species datasets or prawn burrow datasets in usable annotated formats has proven challenging. To supplement local imagery, CC-BY licensed images from the MINKA portal, a citizen science-based public repository of marine species pictures, have been utilized. A copy of the MINKA downloader tool (Martínez, 2024a), developed by EMSO-OBSEA, has been used to download images of target species. Although this data lacks bounding box annotations, it remains valuable as it provides community-identified reference images of species, which can be annotated and incorporated into training datasets.

Student summer bursars have conducted the majority of the annotation work. However, the Marine Institute aims to develop an accessible on-premise platform or solution for collating and annotating imagery datasets. The "Histogram Equalization" tool in CVAT has been used during data annotation. This tool, implemented in CVAT using a JavaScript port of OpenCV, enhances image contrast, allowing features to be more easily distinguished in overexposed or underexposed (very bright or very dark) images. The enhancement is applied solely for annotation purposes and does not alter the image for training. However, bursar students utilized "noise" and "exposure" augmentation techniques in Roboflow to enhance certain training datasets for Nephrops and marine species.

For the Nephrops Burrow Detection use case, one of the student bursars annotated 3331 image frames from various prawn fishing ground stations. This approach was taken to capture a range of burrow images from different seabed types within the fishing area seabed types. Five annotation classes were used: prawn burrow, small prawn burrow, crab burrow, closed burrow, and gate-kept burrow (a prawn burrow with a prawn visible inside). Additionally, 1171 control images of the seabed without any labeled features were included, resulting in a total of 4502 images in the initial dataset. However, the dataset was highly imbalanced, containing over 5000 annotations of prawn burrows but only 79 gate-kept burrows. To address this issue, the annotated dataset is imported into a project on the Roboflow platform, is analyzed and refined to balance the class distribution by reducing the disparity in class numbers. This process reduced the total image count to 468. Roboflow's image augmentation features such as "noise" and "exposure" were then applied to copies of the images to further enhance and expand the dataset, increasing the final dataset for model training to 1200 images.

Initially, two "burrow complex" annotation classes were included, representing prawn burrows connected "back to back" or in a "T" shape pattern. However, these classes were removed as they appeared to cause confusion in the model. An approach in Python using YOLOv8 Oriented Bounding Boxes is explored, along with object tracking and counting functionality in YOLOv8. This approach aimed to analyze the angles of burrows in relation to each other to improve the identification and counting of "burrow complexes" in video sequences.

For the Video Quality Assessment use case, a "Proof of Concept" was implemented using the pre-trained DOVER VQA model (Wu et al., 2022, 2023a; Wu, 2022). This model and the example scripts provided by the DOVER VQA GitHub repository have proven useful "out of the box" for scoring videos based on both technical and aesthetic quality. In limited experimentation with the model so far, the scoring system has been effective in distinguishing between poor and high-quality video footage.

In future work, EMSO SmartBay and the Marine Institute aim to improve the models by supplementing the initial training datasets (Melvin, 2024; Cullen, 2024a,b) with further images and annotation classes. It is also aimed at supplementing the Nephrops burrow imagery and annotations with more imagery from Irish Nephrops fishing areas. Furthermore, the approaches developed in the iMagine project are intended to be applied to object detection tasks in various other fishery surveys and species detection, as well as to support in-house infrastructure for capturing species counts and video quality metrics.

13.6. Oil spill detection

In this use case, a consortium consisting of Orbital EOS, the University of Trento (UniTN), and CMCC is developing an end-to-end, AI-enhanced oil spill detection system. To achieve this goal, a system capable of detecting oil spills from satellites using AI (developed by Orbital EOS) was implemented, successfully recording several incidents worldwide. This dataset was organized and made available on Zenodo (Ferrer et al., 2024) and the THREDDS Data Server (Unitn, 2024), ensuring public access to these observations. Using this dataset, an AI hybrid modeling approach was implemented, integrating the MEDSLIK-II oil spill model (De Dominicis et al., 2013) with Bayesian Optimization (Frazier, 2018), an AI-driven parameter search, to optimize simulation results based on two oil spill observations.

Initially, satellite imagery is collected and labeled internally, and proprietary models developed by Orbital EOS are applied to these datasets. These models identify oil spills in the available images, generating estimates of the spill type (i.e., mineral or biogenic), location, area, and volume. Both active (i.e., Synthetic Aperture Radar) and passive (i.e., optical) sensors are used to sample the ocean surface and model outputs are delivered in standardized format. These estimations are then used in subsequent steps of the processing chain, which includes generating real or hypothetical oil spill scenarios. This process accounts for potential false positives, where an oil spill is detected but did not actually occur in the real world.

From this approach, a dataset of events ranging from 2018 to 2022 was collected and systematized into a database by the University of Trento. This dataset contains approximately 300 events related to 172 oil spill images from around the world. A THREDDS catalog has been deployed at UniTN premises to support environmental scientists from various disciplines in browsing and accessing the provided datasets as open data (Unitn, 2024), for any purpose they may have. The THREDDS catalog includes both oil spill images and NetCDF data (MEDSLIK-II simulation outputs) available also in JSON representation. The entire dataset serves as a benchmark baseline for future research activities and is expected to be further expanded over time.

Based on the oil spill image detections dataset, CMCC integrated these components in an end-to-end pipeline that includes a data-driven approach to improve the accuracy of oil spill events simulated by the MEDSLIK-II model (Accarino et al., 2025). Numerical simulations rely on a set of physical parameters whose values affect the shape of the simulated spill in space and time.

In this use case, a Bayesian Optimization approach is employed to estimate the optimal physical parameter configuration that maximizes a specific score function.

The Fraction Skills Score (FSS) is used to quantify how closely numerical simulations match the segmented oil spill images. MEDSLIK-II simulations are initialized with the physical parameters estimated by the Bayesian Optimization method. An optimization framework is established in which the simulation is repeated multiple times until a converge criteria is reached.

The individual simulations are compared, and the optimal set of parameters is selected. Due to the nature of this approach, experiment tracking was not required, as variations in image sets and environmental conditions can cause results to differ between events. At the end of the optimization process, the parameters that maximize the metric are identified, generating an optimal simulation that produces a visually representative image of the oil slick.

Due to the factors described above, the solution does not incorporate a deep learning model itself. Instead, the results are derived from a physical model with AI-enhanced parameters. Additionally, data drift is not considered in this approach, as both the images and environmental conditions contribute to uncertainty in the optimization process. Moreover, the continuous development of the oil spill model may further influence the functionality of the Bayesian Optimization algorithm.

Future work will address evaluation of improved metrics for the optimization process, as well as further validation on additional oil spill events. Possibly, explorations of fully ML-based oil spill models in place of the traditional numerical ones (e.g., Lagrangian) will also be considered.

This framework is available on the iMagine Marketplace. The hybrid model is designed for advanced users who are proficient in utilizing satellite imagery and shape files to perform the optimization independently.

13.7. Flowcam plankton identification

FlowCam is a high-throughput imaging device that produces between 300,000 and 400,000 particle images annually in the current monitoring context. To store and analyze this high volume of data, semi-automatic data pipelines were established to preprocess raw output data and store it in an internal database. Convolutional Neural Networks, based on an Xception architecture (Chollet, 2017), have been trained on manually validated FlowCam images of phytoplankton cells to accelerate the identification of particle images. After inference, images are manually checked by scientists using an in-house developed labeling tool to ensure data quality. The current model was trained

using a dataset consisting of 337,613 images across 95 classes, with the training set available on Zenodo (Decrop et al., 2024).

A module of the FlowCam phytoplankton identification service has been developed on the iMagine platform. Users can customize the train/validation/test split according to their preferences and apply data augmentation using the Python Albumentations package (Buslaev et al., 2020). The categorical Cross-Entropy loss function was used for the phytoplankton classes. The full monitoring image library, consisting of 1,865,953 manually validated FlowCam images targeting eukaryotic microphytoplankton in the 55–300 μm range, is openly available in the Marine Data Archive (MDA) and linked to an IMIS discovery dataset record (Integrated Marine Information System) available via Lagaisse et al. (2024).

A well-known challenge in working with this type of imaging dataset is the significant class imbalance, with a few overly abundant classes and many rare classes represented by only a few images. To address this issue, thresholds were established for each trained class during training set sampling from the image library, ensuring a minimum of 100 images per class and a maximum of 10,000 images. Augmentation techniques are implemented to artificially upsample rare classes and are available through the FlowCam module on the iMagine platform. Additional functionality has been included in the module to enhance the applicability of models and training datasets across different FlowCam device versions. This is achieved by providing code for image transformation to account for variations in image resolution and differences between RGB and grayscale images.

To evaluate model performance after training, Jupyter notebooks are provided for users to assess key metrics such as Precision, Recall, and F1 score at the class level. Currently, drift monitoring is not included, as models are trained on ground truth data and regularly retrained using annually validated monitoring data. However, drift monitoring could be considered for future implementation.

13.8. Underwater noise identification

The model's objective is to predict the distance to the closest vessel. However, challenges arise due to vessels that are either dark or exhibit irregularities in their Automatic Identification System (AIS) transmissions. To address this issue, a strategy was implemented to connect local minima (i.e., the closest vessels) within a specified window frame. This approach enables the continuous annotation of recordings based on the distance to the nearest AIS data point. The selection of the window frame size is crucial. A smaller window frame increases the dataset's temporal resolution by providing more data points but also raises the risk of missing the closest vessel if there are gaps in AIS data transmission. Conversely, a larger window frame reduces the likelihood of missing the closest vessel due to intermittent data, but results in decreased temporal resolution. Therefore, determining an optimal window frame is essential to balance the trade-off between data resolution and accuracy in vessel identification.

Through experimentation and filtering, a 6 min window frame was predominantly used. However, during the latter half of the year, a transition to a 5 min window frame was implemented for the Grafton data. This adjustment resulted in 240 and 288 data points for the 5 min and 6 min windows, respectively, within a 24-h period. Despite these efforts, approximately 8% of the data had to be excluded due to irregularities in AIS data. Ultimately, 27,524 WAV 10-s audio segments were generated over 116 days, including 40 days with overlapping stations and 76 unique days. All acoustic recordings were converted to single-channel, with a sampling rate of 48 kHz, and segmented into 10-s non-overlapping windows. The dataset is published at Decrop et al. (2025).

To prepare the data for distance classification, we segmented the audio into 10-s, non-overlapping windows, categorizing each segment by its proximity to the nearest vessel. Categories were divided into 1 km bins: 0–1 km, 1–2 km, 2–3 km, 3–4 km, 4–5 km, 5–6 km, 6–7 km, 7–8 km, 8–9 km, 9–10 km, and 10+km.

The Biolingual model (Robinson et al., 2024), a pre-trained version of CLAP-LAION partially trained on underwater bioacoustics data, was selected for transfer learning. Two distinct strategies were employed to leverage this model. In the first approach, feature extraction, the pre-trained layers were used to obtain high-level representations from Log-Mel spectrograms, which were then passed through three custom layers designed for the task. In the second approach, fine-tuning, the model was initialized with the pre-trained weights, but all layers were retrained from scratch, followed by a single linear output layer.

During training, the model was both trained using a costume loss function. To enhance the loss function, a weighted penalty was introduced based on the severity of the prediction error. For instance, a prediction that is off by 1 km is penalized less than one that is off by 5 km. Instead of standard one-hot encoding, the loss function was modified to reflect the degree of the error.

The dataset was initially divided into training, validation, and testing sets. To ensure data independence, full-day deployments from different locations and dates were assigned to each set (i.e., data remains independent within the same day but not across different days). For any given day and location, the data was entirely allocated to either the training, validation, or testing set, with no overlap across these sets for the same day. The distribution split was 79.4%, 10.6%, and 9.9%, reflecting the uneven availability of data points across different days. The data was collected from two stations and covered most of 2022 to mitigate data bias.

Although fine-tuning typically leads to better performance thanks to its comprehensive retraining, it also comes with significantly higher computational costs. Feature extraction, on the other hand, is more efficient, as it only updates the final layers of the model. Choosing between these approaches thus involves balancing performance improvements against computational demands—an important consideration for real-world applications.

The model has yet to be published together with the data, but is currently running on the imagine platform. Currently, drift monitoring is not included; however, it could be considered in the future. In the future, the model could be tested and/or retrained in new environments to compare the performance.

13.9. Beach monitoring

This use case aims to automate beach seagrass wracks identification, shoreline extraction, and the detection of rip currents from beach imaging systems, including operational platforms and crowdsourced acquisition initiatives, to address open research questions and contribute to current developments (Soriano-González et al., 2024a; Fernandez-Mora et al., 2025; Khan et al., 2025).

For each coastal feature and imaging system, a training dataset was created, either programmatically using R software or manually using CVAT. Datasets were published in the Zenodo iMagine community open repository (Table 6). The shoreline dataset was the simplest to generate, as it built upon previous research (González-Villanueva et al., 2023). Beach wracks and rip currents datasets required more expertise and time. The amorphous nature of both coastal features made it difficult to clearly identify their borders, and suitable images that clearly displayed them were difficult to find.

For model development, the general workflow involved training in the iMagine platform, monitoring performance and managing experiments using MLFlow, all within the iMagine ecosystem. For model evaluation, common performance metrics such as accuracy, loss, precision, recall, F1-score, and IoU were considered. However, specific models and data preprocessing steps were identified and tested for each individual case.

For beach seagrass wrack identification, the U-Net (segmentation), YOLO (object detection and mask segmentation) in their different

versions and scaled variants, and a combined approach leveraging the object detection capabilities of YOLO and the zero-shot segmentation capabilities of SAM were tested. For YOLO training, the random data augmentation setting was applied, introducing image transformations such as flips, translation, HSV (Hue Saturation Value) adjustments, scaling, and mosaicking. By enhancing the model's generalizability, this approach helped mitigate dataset imbalances and improved performance across a broader range of images. For SAM segmentation, images were cropped to the bounding boxes generated by the best-performing YOLO model. The primary limitation of this combined approach was the uncertainty introduced by error propagation from YOLO bounding boxes to SAM predictions. A comparison of the three approaches suggested that YOLO should be prioritized. However, detecting and segmenting lower density beach wracks was consistently challenging independently of the method used, indicating that it is a more complex class.

For shoreline extraction, the U-Net and Bidirectional Long Short-Term Memory (Bi-LSTM) networks were tested. For U-Net training, patchify was used to split images of varying sizes into smaller patches with 50% overlap, ensuring compatibility with the U-Net architecture (256 × 256). Rather than resizing, patchify was chosen to prevent the creation of excessively large steps in the shoreline due to pixel interpolation during the downsampling process. For Bi-LSTM training, images were split into rows, with each row processed independently by the Bi-LSTM network (589,292 rows). To maintain consistent row lengths, rows were equally padded with black pixels on both the right and left sides when necessary. This row-wise approach enabled training models individually for each beach due to its lower requirement for input images compared to U-Net. This separate training strategy significantly enhanced the performance of Bi-LSTM and demonstrated its suitability when training data is limited. From the shoreline extraction case, two major conclusions were drawn: (i) The tidal range significantly impacts model performance; microtidal beaches are less complex than mesotidal beaches, especially those with flooded terraces. (ii) While segmentation metrics typically consider the entire image, the validation for shoreline extraction methods must prioritize the areas around the shoreline, as this is where errors are most important, and likely to occur.

For rip currents' detection, both one- and two-stage detectors models are currently under analysis. From the training dataset to model development, rip currents are being treated as oriented objects, related to the waves' direction and the shoreline position, with variable width and length, and with ambiguous borders and shapes. While the rip current case is still in its early stages, preliminary results using a subset of the training dataset indicate that one-stage detectors, such as RT-DETR (Real-Time DEtection Transformer) and YOLO, may be more suitable for our rip current detection application. The processing speed advantage of these detectors is essential for the near real-time detection needed in an early warning system, and further extends their applicability to both static imagery and high frame rate video analysis.

The central challenge across all addressed cases lies in data imbalances. These imbalances are evident both in the number of images from different sites – each representing a specific beach or coastal stretch with similar geometry – and in the temporal distribution of images. These factors are influenced by uncontrollable elements such as citizen participation variations across areas and times, the concentration of operational systems in specific regions, and the natural occurrence of coastal features and processes in particular beach areas or seasons. Future efforts will focus on expanding the training datasets, and investigate data augmentation and drift monitoring techniques. Additionally, key future objectives include advancing model development, enhancing model robustness and generalizability to new scenarios, and transitioning the models into practical applications via deployable modules.

13.10. Freshwater diatoms identification

The use case aims to develop a diatom-based bioindication service using automatic pattern recognition algorithms for microscope images from freshwater benthic diatom samples. From a computer vision perspective, the main challenge for developing a pipeline for diatom identification lies in the fact that our target organisms (diatoms) include a considerable number of often very subtly different categories, in comparison with more standard problems. In comparison to existing alternatives, the aim was to provide a tool that is fast and robust while being reliable with a high level of repeatability, but also with training sets allowing transferability to the greatest number of end-users.

In addition, the primary bottleneck was the limited size of the initial training dataset, since the COVID-19 crisis strongly constrained the acquisition of new imaging data. Yet, it was also an opportunity to test the use of synthetic datasets. The proof of concept for the prototype was thus established by pre-training the models using a synthetic dataset (simulated microscope images) and then fine-tuning them with a limited real dataset (real microscope images). The synthetic dataset was created by: 1. collecting individual images of diatoms from atlases (approximately 15,000 individual diatom thumbnails representing around 200 diatom species, with at least 30 images per species), 2. applying data augmentation by varying the size and orientation of the thumbnails, and 3. generating virtual microscope images using seamless techniques to paste the thumbnails onto a gray background, thereby mimicking a realistic set of microscope images containing various diatoms.

Using the synthetic dataset, the performance of the detection network (YOLOv5) improved by up to 25% in precision and 23% in recall at an IoU threshold of 0.5 (Venkataramanan et al., 2023b). The current online version now uses YOLOv8 for both detection and classification.

The diatom thumbnails dataset was also used to train a deep learning classifier with EfficientNet as the backbone. The classifier was evaluated based on the accuracy score, which was 94%. Of the approximately 200 diatom species included in the dataset, 113 were classified with 100% accuracy. Additionally, other classification methods were explored to better address the high inter-class similarity and intra-class variance among the different diatom species. These factors can make it challenging for traditional classification methods to accurately distinguish between species. To tackle this issue, a method was proposed for learning feature representations that group visually similar images of each class together while ensuring that inter-class features are widely separated. This approach was also tested using a standardized plankton image dataset (WHOI-Plankton Dataset) (Venkataramanan et al., 2021).

Furthermore, a method for estimating uncertainty in classification performance was introduced. This method uses the proximity of a data point to different class features to estimate uncertainty in the network's prediction. It also demonstrates how this approach can provide a reliable estimate of prediction confidence and detect out-of-distribution samples (Venkataramanan et al., 2023a). This method has also been tested on another dataset of diatom images recently published (Venkataramanan et al., 2024).

Standard evaluation metrics were used, including classification accuracy, Expected Calibration Error (ECE), Negative Log-Likelihood (NLL), Area Under the Receiver Operating Characteristics curve (AUROC), and Area Under the Precision-Recall curve (AUPR). The effectiveness and generalizability of the proposed feature representation and uncertainty estimation methods were demonstrated through testing on various standard datasets (COCO, MNIST, CIFAR, SVHN).

To further improve the current models and develop new ones, a larger dataset of real diatom microscope images is being consolidated. Highly trained diatom experts are focusing on the taxonomic classification task using BIIGLE (rotating bounding boxes), while less experienced experts, such as students, are working on object segmentation using Labelbox (instance segmentation masks), and more recently SAMv2. These datasets will then be used to retrain the detection and classification pipeline.

The new dataset will also be employed to test a pipeline based on instance segmentation using YOLO and morphological parameter extraction from the obtained segmentation masks. The results will

further support biologists in their morphometric analyses (Chéron et al., 2025). This approach will be compared with unsupervised methods (e.g., GANs) capable of exploring the morphological variability of diatoms in an unsupervised manner.

To date, the diatom thumbnails dataset has been published in the institutional repository (DOREL), which is connected to the national repository (Laviale and Venkataramanan, 2023). This dataset is currently updated with more images and more species. The different models have been published on GitHub and are available on the Marketplace of the iMagine Project. The real image dataset will be released in the future. Currently, there is no need for drift monitoring of the developed model, but it may be considered in the future.

14. Results and discussion

Through collaboration across a wide range of real-world applications (Table 1), from marine litter detection to freshwater diatom classification, the iMagine project established best practices and guidelines for producers and providers of image sets and image analysis applications, and adapted its AI platform to support their diverse tasks.

During the project, we identified several key challenges, such as limited data availability, time-consuming labeling, proper selection of AI algorithms, resource constraints for training and deploying models, ensuring the reproducibility of trained models, managing data and model drift in production environments. Data preparation, particularly handling the characteristics of underwater images, such as water turbidity, proved to be a significant hurdle. Image annotation is a time-consuming task in producing training datasets. Annotation redundancy and inconsistencies in citizen science data also posed challenges, along with the need for expert intervention in labeling highly specialized datasets (e.g., plankton taxonomists and ictiologists). Data imbalance was another common issue, requiring strategies such as resampling, class weight adjustment, applying data augmentation techniques, and exploring the use of generative AI for synthetic data generation. These factors required more time and expertise, creating bottlenecks in the development of various use cases.

For every challenge and step in the AI development process (Fig. 1) we proposed a solution. Data annotation tools were suggested and provided. To tackle data scarcity and promote transparency, labeled datasets have been published on public repositories such as Zenodo, following the FAIR principles. Resource constraints were mitigated by providing access to federated cloud infrastructures, allowing users to utilize the computing power of multiple providers. All trained models are publicly available, and the corresponding code has been containerized using Docker to ensure reproducibility. For model deployment, the OSCAR system offers a scalable and reproducible inference environment that supports AI models in containers with minimal setup. The Frouros framework is recommended to support continuous monitoring and detection of data and model drift during operation. Although the resources on the iMagine platform are primarily designated for the use cases within the project, everybody can try the modules for 10 min. Furthermore, all the code for modules and tools developed is publicly available and open-source. This accessibility makes it easier for marine scientists and other researchers to adopt and integrate advanced AI workflows into their own work using their resources.

While the iMagine platform provides a comprehensive framework for AI-based image analysis in aquatic sciences, one limitation of the current approach is the relatively small number of use cases. Although the selected use cases span various areas of marine and freshwater sciences, they may not fully represent the wide range of challenges and data types encountered across the broader aquatic science community. We addressed this limitation, in part, through open calls during the project, which allowed for the inclusion of additional use cases.

15. Conclusion and outlook

This paper summarizes the experiences we have gained from various use cases in aquatic sciences, all related to the development of image analysis applications that take advantage of AI techniques. The sharing of knowledge, challenges, acquired know-how, and solutions were established through the Competence Center, a distributed support team within the iMagine project. The work covered all stages of AI/ML life-cycle. This paper presented these in dedicated sections: neural networks relevant for image and video analysis, annotation of images for constructing training datasets with the following open access publication, data preprocessing methods necessary to apply before AI model training can start. It also included an overview of the evaluation metrics and experiment tracking tools, highlighting FAIRness aspects and addressing data biases, as well as how trained AI models can be shared with the community on the iMagine platform and made available for inference. Finally, all iMagine use cases provided insights into their application of the aforementioned techniques.

Based on the study and insights from the iMagine use cases, CVAT was the tool most commonly used by the use cases due to its user-friendly interface, robust annotation features, and flexibility.

In terms of the deep learning model, YOLO was the most commonly used model for object detection, and it demonstrated strong performance across many use cases. For segmentation, the ZooScan - EcoTaxa Pipeline service found that in aquatic imagery, panoptic segmentation using Mask2Former achieved better performance than instance segmentation with Mask R-CNN.

Regarding data availability, Zenodo offers a robust solution by providing an open access platform for researchers to share, store, and manage datasets. Moreover, the FAIR EVA tool in the context of iMagine, has been used to improve the FAIRness of published training datasets concerning the metadata used in Zenodo, along with the training dataset.

MLflow was found to be the most practical solution for AI/ML experiment tracking. As a result, an MLflow server is provided to the project use cases for the efficient management and tracking of the machine learning experiments.

To make the AI models developed during the iMagine project available to other scientists, the models are coupled with DEEPaaS API and published as Docker images on the iMagine platform, while the source code is shared on GitHub. For the inference services, the OSCAR serverless system provides a reliable solution, offering efficient and scalable deployment capabilities.

Future work should focus on addressing the identified bottlenecks and limitations of the approach. For underwater image analysis, exploring advanced image enhancement techniques to mitigate the effects of turbidity would be beneficial. The use of automatic and semi-automatic annotation with pretrained models in annotation tools would help streamline the labeling process. Expanding datasets with more diverse and balanced samples will further improve model performance. The use of synthetic datasets generated with the help of generative AI may also be considered. Implementing data drift monitoring systems could help ensure the long-term reliability of both the models and the deployed data acquisition processes. Exploring additional deep learning architectures and data preprocessing steps could lead to further performance gains. The Federated Learning technique (McMahan et al., 2017) could be applied to distributed aquatic datasets. iMagine has started collaborating with the DEAL project to explore this decentralized training technique in real-world scenarios. Leveraging foundation models may be considered the next step beyond CNNs for AI-based aquatic services, wherever applicable. For citizen science data, developing more robust data-cleaning pipelines and potentially incorporating expert validation could enhance data quality. Furthermore, continued development and refinement of the iMagine platform and its Marketplace will facilitate broader access to and utilization of the AI models and tools developed. Cooperation with other EU projects in the field (such as Blue-Cloud), and the expansion of the variety of use cases will further enrich the study and promote best practices.

CRediT authorship contribution statement

Elnaz Azmi: Writing – review & editing, Writing – original draft, Software, Methodology, Investigation. **Khadijeh Alibabaei:** Writing – review & editing, Writing – original draft, Software, Methodology, Investigation. **Valentin Kozlov:** Writing – review & editing, Writing – original draft, Software, Methodology, Investigation. **Tjerk Krijger:** Writing – review & editing, Software, Methodology, Data curation, Conceptualization. **Gabriele Accarino:** Writing – review & editing, Software, Methodology, Data curation, Conceptualization. **Sakina-Dorothée Ayata:** Writing – review & editing, Software, Methodology, Data curation, Conceptualization. **Amanda Calatrava:** Writing – review & editing, Software, Resources, Methodology. **Marco Mariano De Carlo:** Writing – review & editing, Software, Methodology, Data curation, Conceptualization. **Wout Decrop:** Writing – review & editing, Software, Methodology, Data curation, Conceptualization. **Donatello Elia:** Writing – review & editing, Software, Methodology, Data curation, Conceptualization. **Sandro Luigi Fiore:** Writing – review & editing, Software, Methodology, Data curation, Conceptualization. **Marco Francescangeli:** Writing – review & editing, Software, Methodology, Data curation, Conceptualization. **Jean-Olivier Irisson:** Writing – review & editing, Software, Methodology, Data curation, Conceptualization. **Rune Lagaisse:** Writing – review & editing, Software, Methodology, Data curation, Conceptualization. **Martin Laviale:** Writing – review & editing, Software, Methodology, Data curation, Conceptualization. **Antoine Lebeaud:** Writing – review & editing, Software, Methodology, Data curation, Conceptualization. **Carolin Leluschko:** Writing – review & editing, Software, Methodology, Data curation, Conceptualization. **Enoc Martínez:** Writing – review & editing, Software, Methodology, Data curation, Conceptualization. **Germán Moltó:** Writing – review & editing, Software, Resources, Methodology. **Igor Ruiz Atake:** Writing – review & editing, Software, Methodology, Data curation, Conceptualization. **Antonio Augusto Sepp Neves:** Writing – review & editing, Software, Methodology, Data curation, Conceptualization. **Damian Smyth:** Writing – review & editing, Software, Methodology, Data curation, Conceptualization. **Jesús Soriano-González:** Writing – review & editing, Software, Methodology, Data curation, Conceptualization. **Muhammad Arabi Tayyab:** Writing – review & editing, Software, Methodology, Data curation, Conceptualization. **Vanessa Tosello:** Writing – review & editing, Software, Methodology, Data curation, Conceptualization. **Álvaro López García:** Writing – review & editing, Software, Resources, Methodology. **Dick Schaap:** Writing – review & editing, Project administration, Conceptualization. **Gergely Sipos:** Writing – review & editing, Project administration, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The project iMagine receives funding from the European Union's Horizon Europe research and innovation program under grant agreement number 101058625.

The AI4OS software stack is developed in the AI4EOSC project, which receives funding from the European Union's Horizon Europe research and innovation program under grant agreement number 101058593.

The use case "Freshwater Diatoms Identification" described in Section 13.10 receives funding from ANR, France (ANR-20-THIA-0010).

Data availability

The implementations and datasets of use cases are all publicly available at the following addresses:

- The tools available on the iMagine platform as a part of the AI4OS software stack:
  https://github.com/ai4os
- The code sources of applications and models implemented through the iMagine use cases:
  https://github.com/ai4os-hub
- The ready to use applications of the use cases on the iMagine Marketplace:
  https://dashboard.cloud.imagine-ai.eu/marketplace
- The training datasets of the iMagine use cases:
  https://zenodo.org/communities/imagine-project/records?q=&f=resource_type%3Adataset.

References

Accarino, G., De Carlo, M.M., Atake, I., Elia, D., Dissanayake, A.L., Sepp Neves, A.A., Peña Ibañez, J., Epicoco, I., Nassisi, P., Fiore, S., Coppini, G., 2025. Improving oil slick trajectory simulations with Bayesian optimization. http://dx.doi.org/10.48550/arXiv.2503.02749, arXiv:2503.02749.

Aguilar Gómez, F., Bernal, I., 2023. FAIR EVA: Bringing institutional multidisciplinary repositories into the FAIR picture. Sci. Data 10 (1), 764. http://dx.doi.org/10.1038/s41597-023-02652-8.

Baños, P., Prat, O., Martínez, E., del Río Joaquín, 2025. OBSEA fish detector AI model (YOLO). http://dx.doi.org/10.5281/zenodo.14910365.

Baños Castelló, P., Prat Bayarri, O., Martínez Padró, E., Francescangeli, M., Aguzzi, J., del Rio, J., 2024. Labeled images at OBSEA for object detection algorithms. http://dx.doi.org/10.5281/zenodo.10809433.

Berberi, L., Kozlov, V., Nguyen, G., Sáinz-Pardo Díaz, J., Calatrava, A., Moltó, G., Tran, V., López García, Á., 2025. Machine learning operations landscape: platforms and tools. Artif. Intell. Rev. 58 (6), 167. http://dx.doi.org/10.1007/s10462-025-11164-3.

Bonino, G., Galimberti, G., Masina, S., McAdam, R., Clementi, E., 2024. Machine learning methods to predict sea surface temperature and marine heatwave occurrence: a case study of the Mediterranean Sea. Ocean. Sci. 20 (2), 417–432. http://dx.doi.org/10.5194/os-20-417-2024.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al., 2020. Language models are few-shot learners. Adv. Neural Inf. Process. Syst. 33, 1877–1901. http://dx.doi.org/10.48550/arXiv.2005.14165.

du Buf, H., Bayer, M.M., 2002. Automatic Diatom Identification. WORLD SCIENTIFIC, http://dx.doi.org/10.1142/4907.

Buslaev, A., Iglovikov, V.I., Khvedchenya, E., Parinov, A., Druzhinin, M., Kalinin, A.A., 2020. Albumentations: fast and flexible image augmentations. Information 11 (2), 125. http://dx.doi.org/10.3390/info11020125.

Chen, G., Huang, B., Chen, X., Ge, L., Radenkovic, M., Ma, Y., 2022. Deep blue AI: A new bridge from data to knowledge for the ocean science. Deep. Sea Res. Part I: Ocean. Res. Pap. 190, 103886. http://dx.doi.org/10.1016/j.dsr.2022.103886.

Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R., 2022. Masked-attention mask transformer for universal image segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1290–1299. http://dx.doi.org/10.1109/CVPR52688.2022.00135.

Chéron, S., Felten, V., Venkataramanan, A., Wetzel, C.E., Heudre, D., Pradalier, C., Usseglio-Polatera, P., Devin, S., Laviale, M., 2025. Long-term effects of the herbicide glyphosate and its main metabolite (aminomethylphosphonic acid) on the growth, chlorophyll a and morphology of freshwater benthic diatoms. Heliyon 11 (13), e43680. http://dx.doi.org/10.1016/j.heliyon.2025.e43680.

Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 1800–1807. http://dx.doi.org/10.1109/CVPR.2017.195.

Cullen, E., 2024a. Smartbay marine species object detection training dataset. http://dx.doi.org/10.5281/zenodo.13989649.

Cullen, E., 2024b. Smartbay marine types object detection training dataset. http://dx.doi.org/10.5281/zenodo.13989526.

De Dominicis, M., Pinardi, N., Zodiatis, G., Lardner, R., 2013. MEDSLIK-II, a Lagrangian marine surface oil spill model for short term forecasting – part 1: theory. Geosci. Model. Dev. 6, 1851–1869. http://dx.doi.org/10.5194/gmd-6-1851-2013.

Decrop, W., Deneudt, K., Parcerisas, C., Schall, E., Debusschere, E., 2025. AIS-annotated hydrophone recordings for vessel classification. http://dx.doi.org/10.14284/723.

Decrop, W., Lagaisse, R., Mortelmans, J., Muyle, J., Amadei Martínez, L., Deneudt, K., 2024. LifeWatch observatory data: phytoplankton annotated trainingset by FlowCam imaging in the Belgian Part of the North Sea. http://dx.doi.org/10.5281/zenodo.10554844.

DeepSeaSpy, 2024. Deep Sea Spy. URL: https://www.deepseaspy.com/en.

Devlin, J., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. http://dx.doi.org/10.48550/arXiv.1810.04805, arXiv preprint arXiv:1810.04805.

Dutta, A., Gupta, A., Zissermann, A., 2016. VGG image annotator (VIA). URL: http://www.robots.ox.ac.uk/~vgg/software/via.

Dutta, A., Zisserman, A., 2019. The VIA annotation software for images, audio and video. In: Proceedings of the 27th ACM International Conference on Multimedia. MM '19, ACM, New York, NY, USA, http://dx.doi.org/10.1145/3343031.3350535.

Elineau, A., Desnos, C., Jalabert, L., Olivier, M., Romagnan, J.-B., Brandao, M.C., Lombard, F., Llopis, N., Courboulès, J., Caray-Counil, L., Serranito, B., Irisson, J.-O., Picheral, M., Gorsky, G., Stemmann, L., 2024. ZooScanNet: plankton images captured with the ZooScan. http://dx.doi.org/10.17882/55741.

Eswar, S., 2015. Noise Reduction and Image Smoothing Using Gaussian Blur (Ph.D. thesis). California State University, Northridge.

Fernandez-Mora, A., Gomez-Pujol, L., Coco, G., Orfila, A., 2025. Hydrodynamic conditions of Posidonia oceanica seagrass berm formation and dismantling events. Sci. Total Environ. 958 (178005), http://dx.doi.org/10.1016/j.scitotenv.2024.178005.

Ferrari, M., D'Agostino, D., Aguzzi, J., Marini, S., 2025. Underwater Mediterranean image analysis based on the compute continuum paradigm. Future Gener. Comput. Syst. 162, 107481. http://dx.doi.org/10.1016/j.future.2024.107481.

Ferrer, S., Mengual, A., Sepp Neves, A.A., 2024. iMAGINE UC4 - Segmented oil spills. http://dx.doi.org/10.5281/zenodo.11354662.

Frazier, P.I., 2018. A tutorial on Bayesian optimization. http://dx.doi.org/10.48550/arxiv.1807.02811, arXiv (Cornell University).

Gaur, A., Pant, G., Jalal, A.S., 2023. Comparative assessment of artificial intelligence (AI)-based algorithms for detection of harmful bloom-forming algae: an eco-environmental approach toward sustainability. Appl. Water Sci. 13 (5), 115. http://dx.doi.org/10.1007/s13201-023-01919-0.

GeeksforGeeks, 2024. Denoising AutoEncoders in machine learning. URL: https://www.geeksforgeeks.org/denoising-autoencoders-in-machine-learning.

Girshick, R., 2015. Fast R-CNN. In: 2015 IEEE International Conference on Computer Vision. ICCV, pp. 1440–1448. http://dx.doi.org/10.1109/ICCV.2015.169.

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. pp. 580–587. http://dx.doi.org/10.1109/CVPR.2014.81.

González-Villanueva, R., Soriano-González, J., Alejo, I., Criado-Sudau, F., Plomaritis, T., Fernàndez-Mora, À., Benavente, J., Del Río, L., Nombela, M.Á., Sánchez-García, E., 2023. SCShores: a comprehensive shoreline dataset of Spanish sandy beaches from a citizen-science monitoring programme. Earth Syst. Sci. Data 15 (10), 4613–4629. http://dx.doi.org/10.5194/essd-15-4613-2023.

Gunda, N.S.K., Gautam, S.H., Mitra, S.K., 2019. Artificial intelligence based mobile application for water quality monitoring. J. Electrochem. Soc. 166 (9), B3031. http://dx.doi.org/10.1149/2.0081909jes.

Guo, Y., Liu, Y., Georgiou, T., Lew, M.S., 2018. A review of semantic segmentation using deep neural networks. Int. J. Multimed. Inf. Retr. 7, 87–93. http://dx.doi.org/10.1007/s13735-017-0141-z.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 770–778. http://dx.doi.org/10.1109/CVPR.2016.90.

Heredia, I., Kozlov, V., 2025. iMagine D4.4 best practices and guideline updated for developers and providers of AI-based image analytics services. http://dx.doi.org/10.5281/zenodo.14961558.

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Andreetto, M., Adam, H., 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. http://dx.doi.org/10.48550/arXiv.1704.04861, arXiv preprint arXiv:1704.04861.

Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q., 2017. Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 2261–2269. http://dx.doi.org/10.1109/CVPR.2017.243.

Jalabert, L., Amblard, E., Berrenger, H., Bourhis, A., Desnos, C., Llopis, N., Martins, E., Merland, C., Serranito, B., Elineau, A., Irisson, J.-O., 2024. Segmentation masks of ZooScan images focusing on images with several objects separated by a human operator. http://dx.doi.org/10.17882/99663.

Jocher, G., Qiu, J., Chaurasia, A., 2023. Ultralytics YOLO. URL: https://github.com/ultralytics/ultralytics.

Katija, K., Orenstein, E., Schlining, B., Lundsten, L., Barnard, K., Sainz, G., Boulais, O., Cromwell, M., Butler, E., Woodward, B., et al., 2022. FathomNet: A global image database for enabling artificial intelligence in the ocean. Sci. Rep. 12 (1), 15914. http://dx.doi.org/10.1038/s41598-022-19939-2.

Khan, F., Stewart, D., De Silva, A., Palinkas, A., Dusek, G., Davis, J., 2025. RipScout: Realtime ML-assisted rip current detection and automated data collection using UAVs. IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens. 18, 7742–7755. http://dx.doi.org/10.1109/JSTARS.2025.3543695.

Kirillov, A., He, K., Girshick, R., Rother, C., Dollár, P., 2019. Panoptic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9404–9413. http://dx.doi.org/10.1109/CVPR.2019.00963.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.-Y., et al., 2023. Segment anything. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4015–4026. http://dx.doi.org/10.48550/arXiv.2304.02643.

Kozlov, V., Sipos, G., Schaap, D., 2024. Imagine D3.3 AI application upgrade, deployment, and operation plan. http://dx.doi.org/10.5281/zenodo.11520845.

Krijger, T., 2024. EyeOnWater training dataset for assessing the inclusion of water images. http://dx.doi.org/10.5281/zenodo.10777440.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2017. ImageNet classification with deep convolutional neural networks. Commun. ACM 60 (6), 84–90. http://dx.doi.org/10.1145/3065386.

Lagaisse, R., Amadei Martínez, L., Decrop, W., Mortelmans, J., Muyle, J., Deneudt, K., 2024. LifeWatch observatory data: phytoplankton annotated image library by FlowCam imaging for the Belgian part of the North Sea. http://dx.doi.org/10.14284/680.

Langenkämper, D., Zurowietz, M., Schoening, T., Nattkemper, T.W., 2017. BIIGLE 2.0 - Browsing and annotating large marine image collections. Front. Mar. Sci. 4, 83. http://dx.doi.org/10.3389/fmars.2017.00083.

Laures, C., Esteban Sanchis, B., Berberi, L., Kozlov, V., 2024. MLflow auth GUI. URL: https://codebase.helmholtz.cloud/m-team/ai/mlflow-auth-gui.

Laviale, M., Venkataramanan, A., 2023. Dataset for publication: Usefulness of synthetic datasets for diatom automatic detection using a deep-learning approach. http://dx.doi.org/10.12763/UADENQ.

Lebeaud, A., Tosello, V., Borremans, C., Matabos, M., 2024. Deep-sea observatories images labeled by citizen for object detection algorithms. http://dx.doi.org/10.17882/101899.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft COCO: Common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (Eds.), Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. Springer International Publishing, pp. 740–755. http://dx.doi.org/10.1007/978-3-319-10602-1_48.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C., 2016. SSD: Single shot MultiBox detector. In: Computer Vision – ECCV 2016. Springer International Publishing, Cham, pp. 21–37. http://dx.doi.org/10.1007/978-3-319-46448-0_2.

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3431–3440. http://dx.doi.org/10.1109/CVPR.2015.7298965.

López García, Á., 2019. DEEPaaS API: a REST API for machine learning and deep learning models. J. Open Source Softw. 4 (42), 1517. http://dx.doi.org/10.21105/joss.01517.

Maharana, K., Mondal, S., Nemade, B., 2022. A review: Data pre-processing and data augmentation techniques. Glob. Transitions Proc. 3 (1), 91–99. http://dx.doi.org/10.1016/j.gltp.2022.04.020.

Marrable, D., Barker, K., Tippaya, S., Wyatt, M., Bainbridge, S., Stowar, M., Larke, J., 2022. Accelerating species recognition and labelling of fish from underwater video with machine-assisted deep learning. Front. Mar. Sci. 9, 944582. http://dx.doi.org/10.3389/fmars.2022.944582.

Martínez, E., 2024a. MINKA downloader. URL: https://github.com/obsea-upc/minka-downloader?tab=readme-ov-file.

Martínez, E., 2024b. OBSEA fish detection. URL: https://dashboard.cloud.imagine-ai.eu/marketplace/modules/obsea-fish-detection.

McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.y., 2017. Communication-efficient learning of deep networks from decentralized data. In: Singh, A., Zhu, J. (Eds.), Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. In: Proceedings of Machine Learning Research, vol. 54, PMLR, pp. 1273–1282, URL: https://proceedings.mlr.press/v54/mcmahan17a.html.

Melvin, É., 2024. Nephrops (Nephrops norvegicus) Burrow object detection simple training dataset from Irish Underwater TV surveys. http://dx.doi.org/10.5281/zenodo.13987957.

MinIO, 2024. Open source cloud storage software MinIO. URL: https://min.io.

MINKA, 2024. A community to get the SDG. URL: https://minka-sdg.org.

Moniruzzaman, M., Islam, S.M.S., Bennamoun, M., Lavery, P., 2017. Deep learning on underwater marine object detection: A survey. In: Advanced Concepts for Intelligent Vision Systems: 18th International Conference, ACIVS 2017, Antwerp, Belgium, September 18-21, 2017, Proceedings 18. Springer, pp. 150–160. http://dx.doi.org/10.1007/978-3-319-70353-4_13.

MSFDTechnicalGroup, 2024. Joint list of litter categories for marine macrolitter monitoring. URL: https://mcc.jrc.ec.europa.eu/main/dev.py?N=41&O=459.

Nagpal, M., Siddique, M.A., Sharma, K., Sharma, N., Mittal, A., 2024. Optimizing wastewater treatment through artificial intelligence: recent advances and future prospects. Water Sci. Technol. 90 (3), 731–757. http://dx.doi.org/10.2166/wst.2024.259.

Pérez, A., Risco, S., Naranjo, D.M., Caballer, M., Moltó, G., 2019. On-premises serverless computing for event-driven data processing applications. In: 2019 IEEE 12th International Conference on Cloud Computing. CLOUD, IEEE, pp. 414–421. http://dx.doi.org/10.1109/CLOUD.2019.00073.

Picheral, M., Colin, S., Irisson, J.-O., 2017. EcoTaxa, a tool for the taxonomic classification of images. URL: http://ecotaxa.obs-vlfr.fr.

Qin, H., Li, X., Yang, Z., Shang, M., 2015. When underwater imagery analysis meets deep learning: A solution at the age of big visual data. In: OCEANS 2015 - MTS/IEEE Washington. pp. 1–5. http://dx.doi.org/10.23919/OCEANS.2015.7404463.

Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al., 2021. Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning. PMLR, pp. 8748–8763. http://dx.doi.org/10.48550/arXiv.2103.00020.

Rath, S.R., 2024. fasterrcnn_pytorch_training_pipeline. URL: https://github.com/sovit-123/fasterrcnn-pytorch-training-pipeline.

Raveendran, S., Patil, M.D., Birajdar, G.K., 2021. Underwater image enhancement: a comprehensive review, recent trends, challenges and applications. Artif. Intell. Rev. 54, 5413–5467. http://dx.doi.org/10.1007/s10462-021-10025-z.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 779–788. http://dx.doi.org/10.1109/CVPR.2016.91.

Ren, S., He, K., Girshick, R., Sun, J., 2016. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. 39 (6), 1137–1149. http://dx.doi.org/10.1109/TPAMI.2016.2577031.

Risco, S., Moltó, G., M., N.D., Blanquer, I., 2021. Serverless workflows for containerised applications in the cloud continuum. J. Grid Comput. 19 (30), http://dx.doi.org/10.1007/s10723-021-09570-2.

Roberts, N.M., Lean, H.W., 2008. Scale-selective verification of rainfall accumulations from high-resolution forecasts of convective events. Mon. Weather Rev. 136 (1), 78–97. http://dx.doi.org/10.1175/2007MWR2123.1.

Robinson, D., Robinson, A., Akrapongpisak, L., 2024. Transferable models for bioacoustics with human language supervision. In: ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 1316–1320. http://dx.doi.org/10.1109/ICASSP48485.2024.10447250.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. Springer, pp. 234–241. http://dx.doi.org/10.1007/978-3-319-24574-4_28.

Rubbens, P., Brodie, S., Cordier, T., Destro Barcellos, D., Devos, P., Fernandes-Salvador, J.A., Fincham, J.I., Gomes, A., Handegard, N.O., Howell, K., Jamet, C., Kartveit, K.H., Moustahfid, H., Parcerisas, C., Politikos, D., Sauzède, R., Sokolova, M., Uusitalo, L., Van den Bulcke, L., van Helmond, A.T.M., Watson, J.T., Welch, H., Beltran-Perez, O., Chaffron, S., Greenberg, D.S., Kühn, B., Kiko, R., Lo, M., Lopes, R.M., Möller, K.O., Michaels, W., Pala, A., Romagnan, J.-B., Schuchert, P., Seydi, V., Villasante, S., Malde, K., Irisson, J.-O., 2023. Machine learning in marine ecology: an overview of techniques and applications. ICES J. Mar. Sci. 80 (7), 1829–1853. http://dx.doi.org/10.1093/icesjms/fsad100.

Saleh, A., Sheaves, M., Rahimi Azghadi, M., 2022. Computer vision and deep learning for fish classification in underwater habitats: A survey. Fish Fish. 23 (4), 977–999. http://dx.doi.org/10.1111/faf.12666.

Sharma, R., Saqib, M., Lin, C.-T., Blumenstein, M., 2022. A survey on object instance segmentation. SN Comput. Sci. 3 (6), 499. http://dx.doi.org/10.1007/s42979-022-01407-3.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. http://dx.doi.org/10.48550/arXiv.1409.1556, arXiv preprint arXiv:1409.1556.

Sisniega, J.C., García, Á.L., 2024. Frouros: An open-source Python library for drift detection in machine learning systems. SoftwareX 26, 101733. http://dx.doi.org/10.1016/j.softx.2024.101733.

Soriano-González, J., Català-Gonell, A., González-Pérez, L., Criado-Sudau, F., Sánchez-García, E., Fernández-Mora, A., 2025. RipAID: Rip current annotated image dataset. http://dx.doi.org/10.5281/zenodo.15082426.

Soriano-González, J., González-Villanueva, R., Sánchez-García, E., 2023. SCLabels: Labelled rectified RGB images from the Spanish CoastSnap network. http://dx.doi.org/10.5281/zenodo.10159977.

Soriano-González, J., Sánchez-García, E., González-Villanueva, R., 2024a. From a citizen science programme to a coastline monitoring system: Achievements and lessons learnt from the Spanish CoastSnap network. Ocean & Coastal Management 256 (107280), http://dx.doi.org/10.1016/j.ocecoaman.2024.107280.

Soriano-González, J., Sánchez-García, E., Oliver-Sansó, J., Pérez-Cañellas, J.D., Criado-Sudau, F., Gómez-Pujol, L., Fernández-Mora, A., 2024b. BWILD: Beach seagrass wrack identification labelled dataset. http://dx.doi.org/10.5281/zenodo.12698763.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2016. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 2818–2826. http://dx.doi.org/10.1109/CVPR.2016.308.

Tan, M., Le, Q., 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning. pp. 6105–6114. http://dx.doi.org/10.48550/arXiv.1905.11946.

Tan, M., Pang, R., Le, Q.V., 2020. Efficientdet: Scalable and efficient object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10781–10790. http://dx.doi.org/10.48550/arXiv.1911.09070.

Tzutalin, 2015. LabelImg. URL: https://github.com/tzutalin/labelImg.

Unitn, 2024. Oil spill catalog: THREDDS data server top-level TDS catalog. URL: http://thredds.imagine.disi.unitn.it.

Venkataramanan, A., Benbihi, A., Laviale, M., Pradalier, C., 2023a. Gaussian latent representations for uncertainty estimation using mahalanobis distance in deep classifiers. In: 2023 IEEE/CVF International Conference on Computer Vision Workshops. ICCVW, pp. 4490–4499. http://dx.doi.org/10.1109/ICCVW60793.2023.00483.

Venkataramanan, A., Faure-Giovagnoli, P., Regan, C., Heudre, D., Figus, C., Usseglio-Polatera, P., Pradalier, C., Laviale, M., 2023b. Usefulness of synthetic datasets for diatom automatic detection using a deep-learning approach. Eng. Appl. Artif. Intell. 117, 105594. http://dx.doi.org/10.1016/j.engappai.2022.105594.

Venkataramanan, A., Kloster, M., Burfeid-Castellanos, A., Dani, M., Mayombo, N.A.S., Vidakovic, D., Langenkämper, D., Tan, M., Pradalier, C., Nattkemper, T., Laviale, M., Beszteri, B., 2024. "UDE DIATOMS in the Wild 2024": a new image dataset of freshwater diatoms for training deep learning models. GigaScience 13, giae087. http://dx.doi.org/10.1093/gigascience/giae087.

Venkataramanan, A., Laviale, M., Figus, C., Usseglio-Polatera, P., Pradalier, C., 2021. Tackling inter-class similarity and intra-class variance for microscopic image-based classification. In: International Conference on Computer Vision Systems. Springer, pp. 93–103. http://dx.doi.org/10.1007/978-3-030-87156-7_8.

Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L.B., Bourne, P.E., et al., 2016. The FAIR Guiding Principles for scientific data management and stewardship. Sci. Data 3 (1), 1–9. http://dx.doi.org/10.1038/sdata.2016.18.

Wolf, M., van den Berg, K., Gnann, N., Sattler, K., Stahl, F., Zielinski, O., 2021. Aquatic plastic litter dataset developed for APLASTIC-Q publication. http://dx.doi.org/10.1088/1748-9326/abbd01.

Wu, H., 2022. Open source deep end-to-end video quality assessment toolbox. URL: http://github.com/timothyhtimothy/fast-vqa.

Wu, H., Chen, C., Hou, J., Liao, L., Wang, A., Sun, W., Yan, Q., Lin, W., 2022. FAST-VQA: Efficient end-to-end video quality assessment with fragment sampling. In: Proceedings of European Conference of Computer Vision. ECCV, http://dx.doi.org/10.1007/978-3-031-20068-7_31.

Wu, Y., Chen, K., Zhang, T., Hui, Y., Berg-Kirkpatrick, T., Dubnov, S., 2023b. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 1–5. http://dx.doi.org/10.1109/ICASSP49357.2023.10095969.

Wu, H., Zhang, E., Liao, L., Chen, C., Hou, J.H., Wang, A., Sun, W.S., Yan, Q., Lin, W., 2023a. Exploring video quality assessment on user generated contents from aesthetic and technical perspectives. In: International Conference on Computer Vision. ICCV, http://dx.doi.org/10.1109/ICCV51070.2023.01843.

Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K., 2017. Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 5987–5995. http://dx.doi.org/10.1109/CVPR.2017.634.

Xu, S., Zhang, M., Song, W., Mei, H., He, Q., Liotta, A., 2023. A systematic review and analysis of deep learning-based underwater object detection. Neurocomputing 527, 204–232. http://dx.doi.org/10.1016/j.neucom.2023.01.056.

Zhou, P., Bu, Y.-X., Fu, G.-Y., Wang, C.-S., Xu, X.-W., Pan, X., 2024. Towards standardizing automated image analysis with artificial intelligence for biodiversity. Front. Mar. Sci. 11, 1349705. http://dx.doi.org/10.3389/fmars.2024.1349705.

Zhu, Y., Huang, C., 2012. An improved median filtering algorithm for image noise reduction. Phys. Procedia 25, 609–616. http://dx.doi.org/10.1016/j.phpro.2012.03.133.