Interreg Co-funded by the European Union

NEXT Black Sea Basin

EfxINNOs

Establishing and Operating an Innovative Marine Technology Transfer Network for Enhancing the Transition to a Sustainable Blue Economy in the Black Sea Basin

Processing of

Process Side Scan Sonar data, images and videos using Machine Learning and AI tools and delineate key habitats

8/2026

https://envrio.eu/efxinnos/

Table of Contents

1. Introduction 5
2. Side Scan Sonar Imagery 6
2.1. Methodology 7
2.2. Dataset Preparation 9
2.3. Machine Learning Models 10
2.3.1. YOLO based object detection 11
2.3.2. U-Net and U-Net++ based seagrass semantic segmentation 12
2.4. Training Configuration 14
2.4.1. YOLO 15
2.4.2. U-Net / U-Net++ 16
2.5. Results 17
2.5.1. YOLO 17
2.5.2. U-Net / U-Net++ 22
2.6. Limitations 24
2.6.1. Dataset scale 24
2.6.2. Ground truth reliability 24
2.6.3. No independent test set 25
2.6.4. Architecture choice is data-scale dependent 25
2.7. Future work 25
3. GoPro Footage 26
3.1. Data acquisition 26
3.1.1. Video files 27
3.1.2. SRT telemetry files 28
3.2. Video processing pipeline 29
3.2.1. Processing steps 30
3.2.2. Configuration file 31
3.3. Underwater video enhancement (Step 0) 33
3.3.1. Overview of the enhancement methods 35
3.3.2. Physically based methods 35
3.3.3. Pretrained networks 38
3.3.4. Classical methods 39
3.3.5. Processing options 40
3.4. Evaluation of the enhancement methods 41
3.4.1. Evaluation metrics 41
3.4.2. Results 42

3.4.3. Development of the Natural method 42
3.4.4. Performance in turbid water 43
3.4.5. Calibration from motion 43
3.4.6. Comparison of all methods and processing cost 44
3.5. Training data and segmentation models (Steps 1 to 3) 47
3.5.1. Frame extraction and annotation 48
3.5.2. Model training 48
3.6. Detection, depth and georeferenced observation points (Steps 4, 4a and 5) 50
3.6.1. Frame-by-frame detection 50
3.6.2. Depth viewer 52
3.6.3. Georeferenced observation points 52
3.7. Seagrass coverage mapping (Step 6) 53
3.7.1. Projection method 53
3.7.2. Outputs 54
3.7.3. Results 54
3.8. Limitations, validation and future work 57
4. Publicly available code 59
4.1. Side Scan Sonar 59
4.2. GoPro footage 62
5. Bibliography 67
6. Introduction 70
6.1. Available Datasets, Data Preparation and Harmonisation 70
6.1.1. Overview of Available Underwater Datasets 70
6.1.2. NEMOSENS Micro-AUV and SSS Observations 70
6.1.3. Complementary Coastal Video Investigations 70
6.1.4. Underwater Video Datasets Provided by Istanbul University 70
6.1.5. Metadata and Quality-Assurance Procedures 71
6.1.6. Partner Data Exchange and Harmonisation 71
6.2. Underwater Image Enhancement and AI-Based Preprocessing 71
6.2.1. Rationale and Objectives 71
6.2.2. Initial Image-Quality Assessment 71
6.2.3. Conventional Enhancement Techniques 71
6.2.4. Deep Learning-Based Underwater Restoration 71
6.2.5. Comparative Interpretation and Limitations 72
6.2.6. Proposed Evaluation and Quality-Control Framework 72
6.2.7. Relevance for Romanian Pilot Areas 72

7. Machine Learning and AI-Based Object Detection, Segmentation and Habitat Delineation 73
7.1. Objectives and Approach 73
7.1.1. Detection, Segmentation and Mapping 73
7.1.2. Exploratory Models 73
7.1.3. Preliminary AI Findings 73
7.1.4. Seagrass and Benthic Habitat Classes 73
7.1.5. Marine Litter Detection 73
7.1.6. Annotation and Validation Framework 73
7.1.7. Georeferencing and GIS Conversion 74
7.1.8. Relationship to SSS 74
7.1.9. Interoperability 74
7.1.10. Side Scan Sonar Data Processing and Acoustic–Optical Integration 74
7.1.11. Preliminary Operational SSS Observations 74
7.1.12. Proposed SSS Processing Workflow 74
7.1.13. Potential Habitat Transitions 75
7.1.14. Acoustic–Optical Integration 75
7.1.15. ML-Assisted Acoustic Classification 75
7.1.16. Potential GIS integrated data 75
7.1.17. Limitations and Requirements 75
7.1.18. Preliminary Results, Methodological Limitations and NIMRD Contribution 75
7.1.19. Underwater Image Processing Results 75
7.1.20. Detection and Segmentation Results 76
7.1.21. Habitat Delineation Framework 76
7.1.22. Principal Limitations and contribution to project methodology 76
7.1.23. Conclusions, Recommendations and Next Steps 76
7.1.24. Main Methodological Contributions 76
7.1.25. Proposed Next Steps 77

1.Introduction

This report presents the work carried out within the project on processing side-scan sonar data, images, and videos using machine learning (ML) and artificial intelligence (AI) tools to delineate key benthic habitats. The work focuses on seagrass meadows, which are among the most ecologically valuable coastal habitats and are protected under the EU Habitats Directive (92/43/EEC).

The data were collected using the NemoSens autonomous underwater vehicle (AUV), which carries a side-scan sonar and a GoPro camera mounted under its nose, facing forward and downward toward the seabed. The two sensors provide complementary information. The side-scan sonar records a wide acoustic swath on both sides of the vehicle and produces georeferenced mosaics of large areas, but seagrass has to be recognised from the texture of the acoustic image, which remains uncertain without visual confirmation. The camera covers only a narrow strip along the vehicle's track and is limited by turbidity and light, but it shows the seabed directly and allows *seagrass* to be identified with confidence. For this reason, ML methods were developed for both data sources, and the video footage was also used as ground truth for interpreting the sonar mosaics.

2.Side Scan Sonar Imagery

Side-scan sonar is widely used to survey previously unmapped areas of the seabed, producing images from reflections of acoustic pulses. The resulting data can be processed into high-resolution, georeferenced mosaics in any map projection, examined in detail through waterfall displays and contact zooms, and used to classify seabed sediments and habitats. In EfxINNOs, the main targets were the seagrass meadows of the Black Sea and the North Aegean Sea. The side-scan sonar data were processed using SonarWiz, a widely used software package for sonar data acquisition and post-processing.

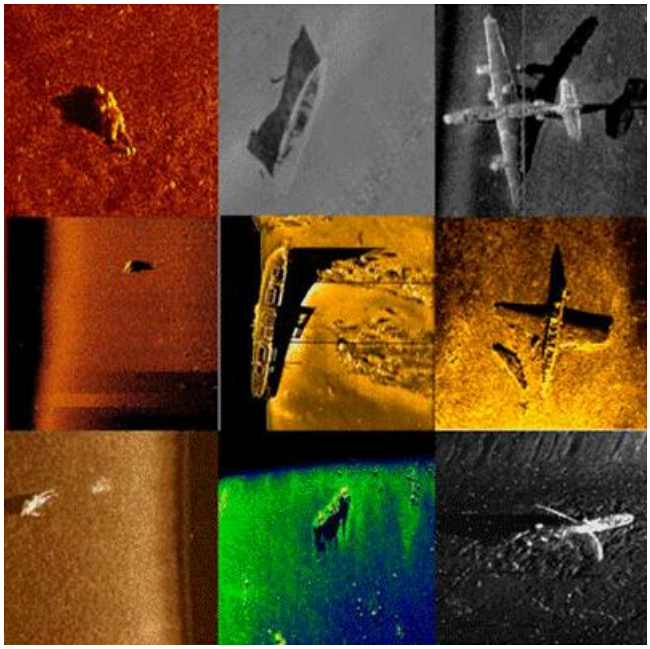


***Figure 1**. Example side-scan sonar images [1].*

In parallel, Python algorithms were used to enhance analysis and delineate seagrass meadows. Machine Learning (ML) tools were also used for image segmentation and synthesis, leading to efficient underwater object detection and recognition. Advancements in ML and Artificial Intelligence (AI) have revolutionised the way we process and analyse underwater data collected with side-scan sonar. By leveraging these powerful tools, we can derive insights from the vast amounts of sonar data, images, and videos captured by AUVs and ROVs in the areas of interest (AOIs).

The process of using ML and AI techniques to analyze side-scan sonar data, images, and videos highlights their role in delineating key habitats in the underwater environment. Processing side-scan sonar data involves extracting valuable information from collected data and using trained ML algorithms to automatically detect and classify objects, such as seabed geological features and marine life habitats.

Thus, ML techniques can efficiently process large volumes of sonar data, aiming to identify significant features and create detailed maps of the underwater terrain. AI tools can also facilitate the identification of potential threats to marine ecosystems, such as invasive species and pollution, by analyzing video footage, thereby enabling a deeper understanding of benthic habitats and their complex dynamics. ML and AI techniques play a role in delineating key habitats, identifying and classifying them, assessing habitat health, monitoring changes over

the project period, and thus identifying potential critical areas for biodiversity conservation efforts, including marine spatial planning and ecosystem-based management. While ML and AI tools offer significant capabilities for data processing and analysis, the project addressed several challenges, including the need for high-quality training datasets and the validation of ML models with ground-truth data.

2.1. Methodology

This project employs an approach that begins with acquiring Side Scan Sonar data using the NemoSens AUV to detect and map seagrass. The NemoSens is a compact autonomous underwater vehicle that follows pre-programmed survey lines, so that the seabed can be covered systematically and the surveys can be repeated over the same area. Seagrass meadows are well suited to acoustic mapping, since their canopy generally produces stronger, more textured backscatter than the surrounding bare sediment, allowing individual patches to be distinguished in sonar imagery. An example of an SSS image depicting seagrass is shown in Figure 2 below.

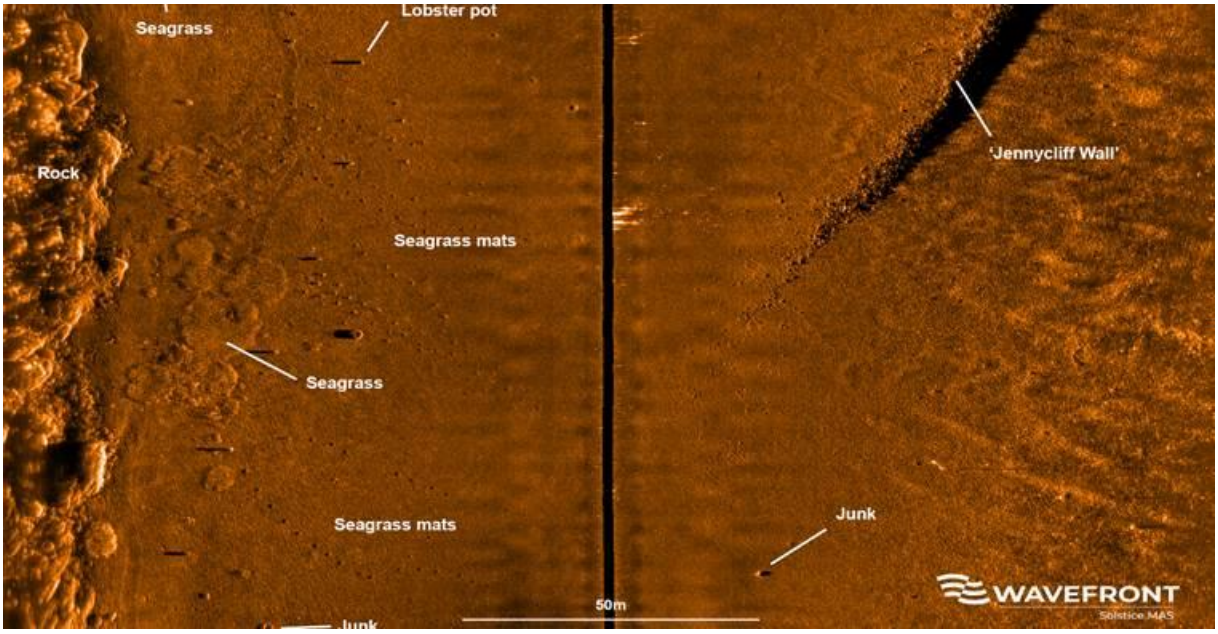


Figure 2. Example of Side Scan Sonar image, depicting seagrass patches.

The overall workflow is illustrated in Figure 3. After acquiring the data, a series of preprocessing steps were performed. This included data normalization, conversion of the raw images to GeoTIFF format, and image enhancement to reduce noise (e.g., CLAHE contrast filter). The raw sonar records were first processed in SonarWiz into georeferenced mosaics, so that each pixel corresponds to a known position on the seabed and the results can be overlaid directly on other spatial data in GIS software. Normalization brings the pixel intensities of the mosaics to a common range, which reduces the effect of variations in gain and signal strength across the survey and helps the models train in a stable way. Contrast Limited Adaptive Histogram Equalization (CLAHE) enhances local contrast while limiting the amplification of noise, making the texture of the seagrass canopy easier to distinguish from the surrounding seabed. Finally, the mosaics were divided into fixed-size, overlapping tiles suitable as input to the machine learning models, as described in Section 2.2.

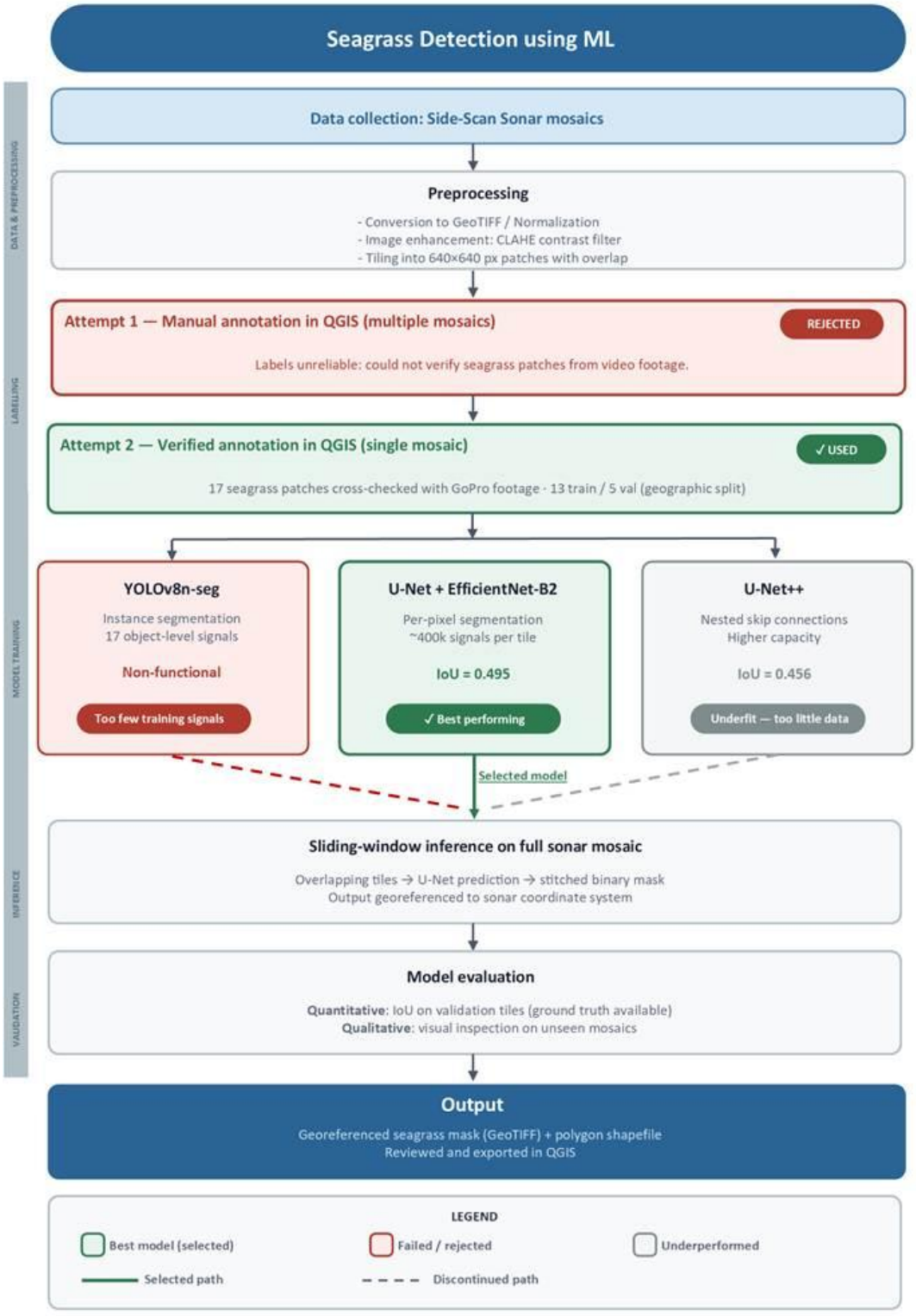


***Figure 3**. Workflow for object detection using Side Scan Sonar data and ML algorithms.*

The next step involved manually labelling the dataset, that is, annotating the seagrass present in the images. This was done in QGIS by visually examining the georeferenced side scan sonar

mosaic. Polygons were drawn in a vector layer wherever seagrass patches were present, producing a shapefile in the same coordinate system as the mosaic. Displaying the GoPro track alongside the mosaic in QGIS allowed each polygon to be checked against the video footage before it was retained. The labelled tiles were then split into training and validation subsets, as described in Section 2.2. Despite this verification, the labels carry a degree of uncertainty, since they rely on visual interpretation of the sonar texture and on the correlation with the GoPro footage. This uncertainty remains because the GoPro and the Side Scan Sonar do not have the same field of view and therefore do not cover the exact same area as can be seen in Figure 5.

2.2. Dataset Preparation

The raw Side Scan Sonar mosaics generated from the preliminary AUV survey were tiled into smaller, fixed-size image patches to make them suitable for training and to increase the effective number of training samples. A tile size of 640 × 640 pixels was chosen to balance spatial context and sample count. Each tile is large enough to contain entire seagrass patches along with the surrounding seabed, giving the models the context needed to separate the two classes, while remaining small enough to produce multiple tiles from a single mosaic and fit in GPU memory during training. The size also matches the standard input resolution of the YOLO models and allows the U-Net models to be trained at the tiles' native resolution without resizing, which would otherwise blur the fine acoustic texture that distinguishes seagrass from sand. Neighbouring tiles were allowed to overlap, so that patches cut by the edge of one tile appear completely in an adjacent tile.

The first survey mosaic labelled for this project (~104 hand-drawn polygons) was digitised by visual interpretation of sonar texture alone, without a corresponding video ground truth check. Models trained on it failed to learn because, upon closer review, those polygons were found not to reliably correspond to real seagrass presence, and the mosaic, its labels, and the models trained on it were excluded from further use. Therefore, a smaller mosaic was chosen that included corresponding GoPro camera footage to validate the presence of seagrass (Figure 4). Ground truth from GoPro correlation remains an estimate due to differences in field of view. By this we mean that the GoPro track and the Side Scan Sonar track are not identical, resulting in the GoPro partially covering the SSS track. An example of this is shown in Figure 5.

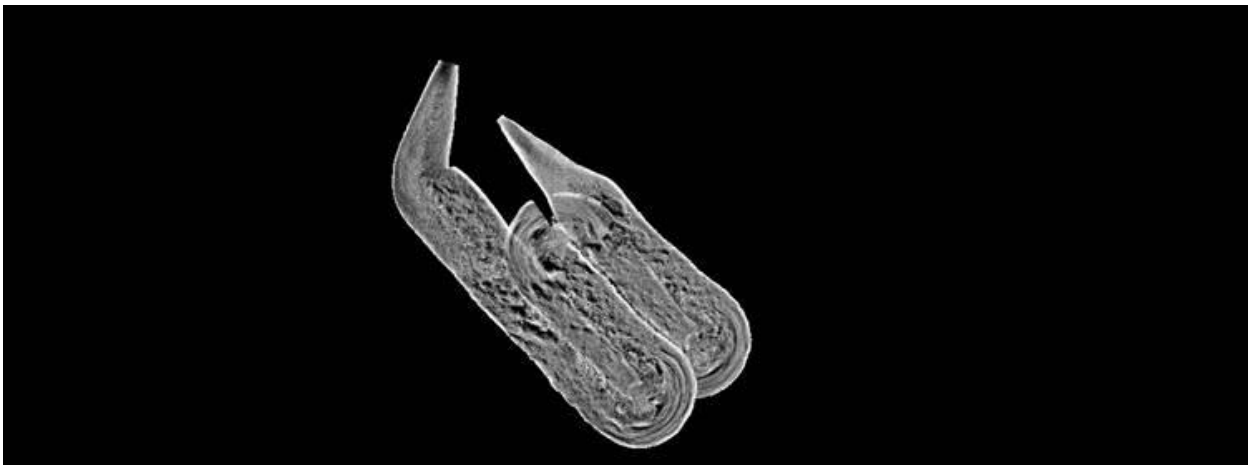


***Figure 4**. Side-scan sonar mosaic of the smaller survey area selected for training, with corresponding GoPro footage available to validate the presence of seagrass.*

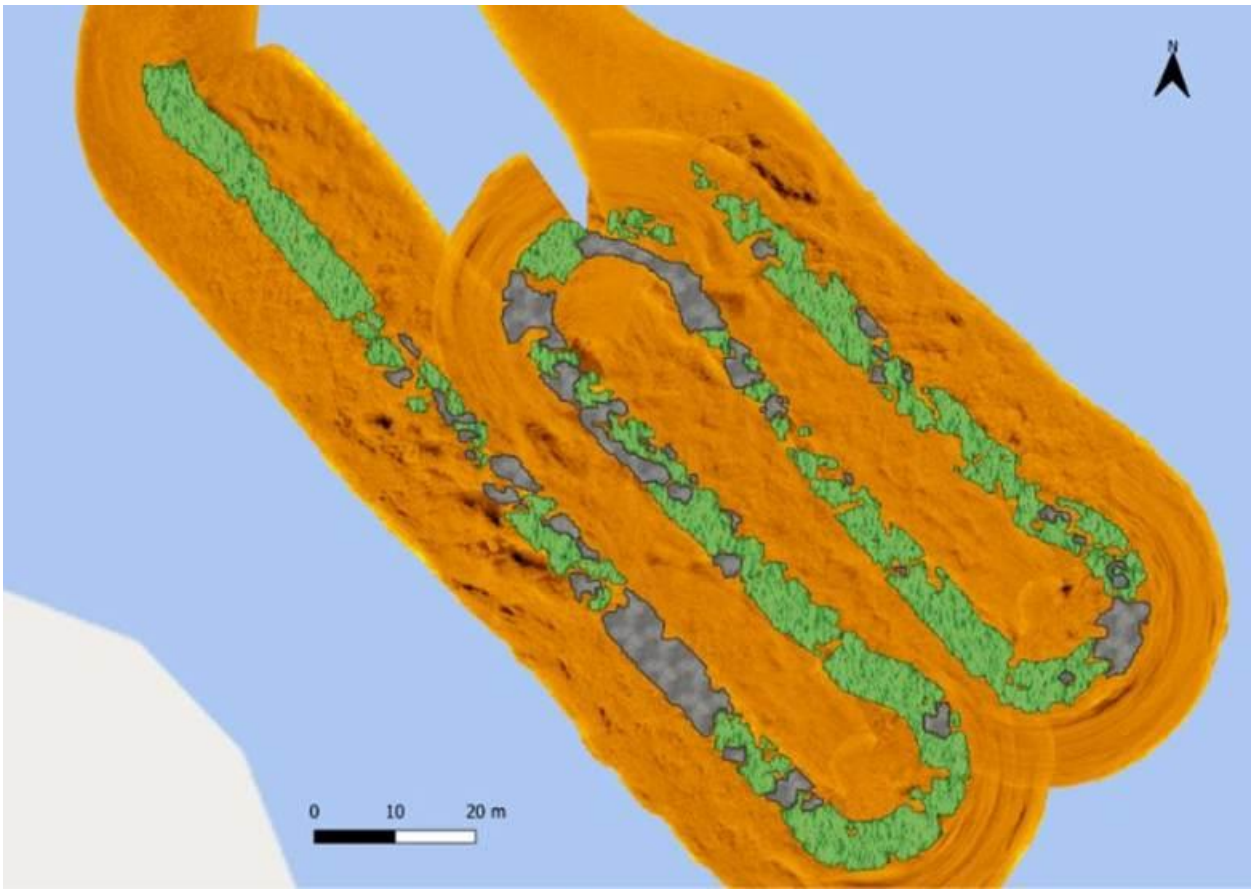


***Figure 5**. Visualisation of GoPro video coverage vs the full Side Scan Sonar Image.*

The mosaic was split into 640x640 pixel tiles, creating 27 valid tiles (24 containing seagrass, 3 containing backgrounds only). The shapefile contained 17 hand-drawn seagrass polygons, cross-referenced against GoPro footage. Because tiles overlap by 62.5%, a single polygon is typically clipped into several tile-level instances. The 17 polygons produce 143 tile-level training instances. An overlap of 62.5% corresponds to a stride of 240 px between neighbouring tiles, so each part of the seabed appears in several tiles at different positions within the frame. This serves as a form of data augmentation, exposing the models to the same seagrass patches with different surrounding contexts, which is valuable when the number of independent annotations is so small. Given the small number of tiles, the spatial train/validation split had to use a reduced buffer in order to leave some tiles strictly for validation. A spatial split was required because overlapping tiles share large portions of the same seabed. A random split would place near-identical image content in both the training and validation sets, allowing the models to be evaluated on content they had already effectively seen, inflating the reported accuracy. The buffer tiles, located between the training and validation areas, overlap with both and were therefore excluded from either set. This resulted in 13 training and 5 validation tiles (9 tiles dropped as the spatial buffer). The validation set is consequently very small, and the resulting metrics should be read as indicative rather than definitive, as discussed further in Section 2.6.

2.3. Machine Learning Models

Artificial Intelligence (AI) broadly refers to computer programs that can sense, reason, act, and perform tasks that would normally require human intelligence. Machine Learning (ML) is a

subset of AI in which algorithms learn patterns from data rather than being explicitly programmed, so that their performance improves as they are exposed to more data over time. Deep Learning (DL) is in turn a subset of ML that uses multilayered artificial neural networks to learn hierarchical representations directly from raw inputs, such as image pixels, and typically relies on large amounts of training data. The relationship between the three is shown in Figure 6 below.

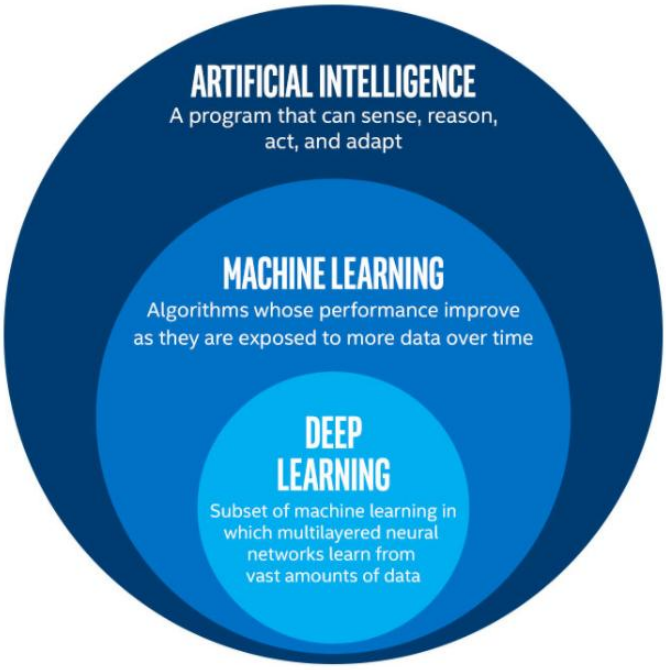


***Figure 6**. Relationship between Artificial Intelligence, Machine Learning and Deep Learning.*

All three models used in this work are deep learning models built on Convolutional Neural Networks (CNNs), which are particularly well-suited to image analysis because they learn local spatial features such as edges and textures directly from the data. YOLO (You Only Look Once) is an object detection and instance segmentation model that locates each seagrass patch in a single pass through the network, returning a bounding box and a mask for every detected object. U-Net is an encoder-decoder network originally developed for biomedical image segmentation, which classifies each pixel in the image as either seagrass or background, while U-Net++ extends it with nested skip connections to refine the predicted boundaries. The two approaches therefore address the mapping task differently: YOLO treats seagrass patches as discrete objects, whereas U-Net and U-Net++ produce a continuous, pixel-level map of seagrass cover. Each model is described in more detail in the following subsections.

2.3.1. YOLO based object detection

YOLO (You Only Look Once) was the first model used to detect and map seagrass in the sonar tiles. First introduced by Redmon et al. [2], YOLO is a single-stage object detector designed for real-time operation, meaning that it can process images quickly enough to keep pace with live data streams, such as video or large batches of images collected during a survey. In its original form, YOLO divides the input image into a grid of equal-sized cells, each responsible for detecting objects whose centres fall within it and predicts for every cell a set of bounding boxes

together with class probabilities and confidence scores, which express both the likelihood that an object is present and how well the predicted box fits it. Since then, the architecture has evolved through a series of versions that improved both accuracy and speed through stronger backbones, multi-scale feature fusion and improved training strategies. The version used in this work, YOLOv8 [3], developed by Ultralytics, uses an anchor-free detection head that predicts object centres directly instead of adjusting predefined anchor boxes, and includes a segmentation branch that outputs a pixel mask for each detected object alongside its bounding box. YOLOv8 is available in several sizes, from nano (n) to extra-large (x), offering a trade-off between speed and accuracy, and YOLO-based models have already been successfully applied to target detection in Side Scan Sonar imagery [1], [4].

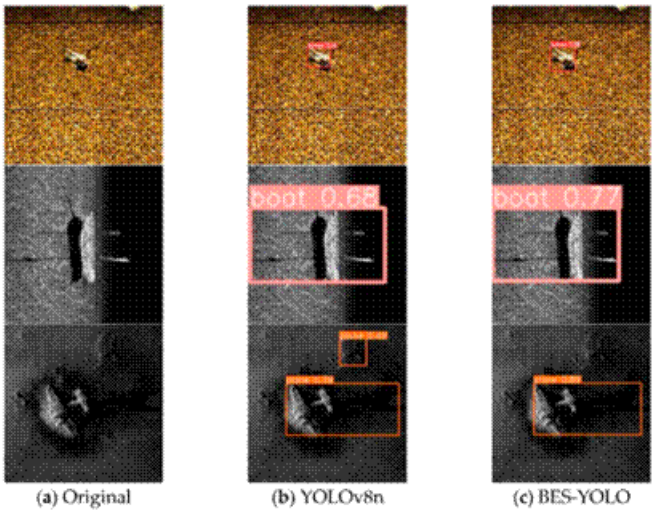


***Figure 7**. Multi-Scale Marine Object Detection in Side-Scan Sonar Images Based on BES-YOLO [1].*

All predictions occur in a single forward pass through a Convolutional Neural Network (CNN), making YOLO both fast and efficient. Unlike traditional methods that first generate region proposals and then classify them, YOLO treats detection as a single regression problem, predicting object positions and classes directly from the input pixels. This enables it to detect multiple objects in an image at high speed and with reasonable accuracy, making it ideal for tasks such as marine monitoring, where thousands of images need to be processed quickly and efficiently.

2.3.2. U-Net and U-Net++ based seagrass semantic segmentation

Given YOLO's limitations on a smaller dataset, we tested two additional models based on dense pixel-wise semantic segmentation. U-Net with an EfficientNet-B2 encoder and U-Net++. The same tiles and labels were used to train those models. Unlike instance detection, which must learn to localize and separate individual objects largely from scratch, semantic segmentation benefits from a strong ImageNet pretrained encoder and a dense per-pixel training signal. Every pixel in every one of the 13 training tiles contributes to the loss, rather than only the 17 labelled object instances we used for YOLO. This makes it substantially more data efficient for very small datasets.

2.3.2.1. U-Net and U-Net++ Architecture Overview

Unlike instance detection models such as YOLO, semantic segmentation models work at the

pixel level, assigning every pixel in the image to a class, in this case either seagrass or background. This is particularly advantageous when training data are scarce: a single 640 × 640 pixel tile provides about 400,000 labelled pixels, each contributing to the training loss, whereas YOLO learns only from the 17 annotated polygons, which yield 143 object instances after tiling.

U-Net [5] is a fully convolutional encoder-decoder network originally developed for biomedical image segmentation, a field where annotated data are typically scarce, and it has since become one of the most widely used architectures for semantic segmentation in remote sensing and environmental applications. Its name comes from the symmetric, U-shaped structure of the network. The encoder, or contracting path, progressively downsamples the input image through successive convolutional and pooling stages, building an increasingly abstract and context-rich representation at the expense of spatial resolution. The decoder, or expanding path, then progressively upsamples this representation back to the input resolution, producing a class prediction (seagrass or background) for every pixel rather than a bounding box around an object, as illustrated in Figure 8.

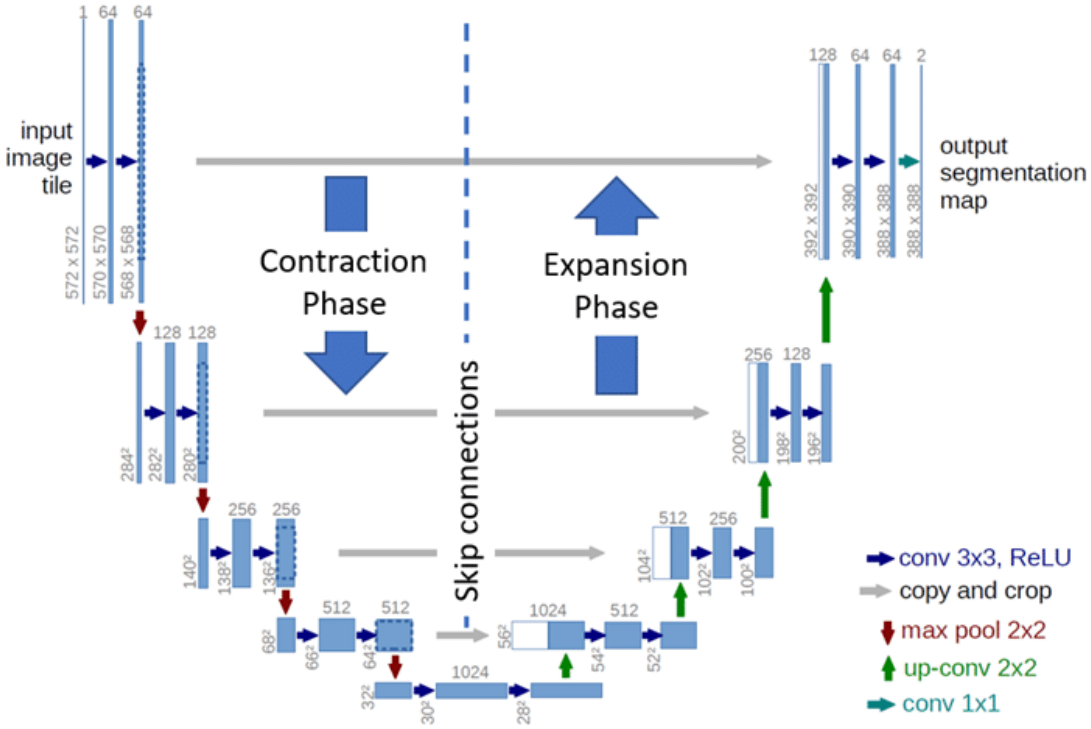


Figure 8. U-Net architecture, showing the contraction (encoder) and expansion (decoder) phases linked by skip connections, adapted from Ronneberger et al. [5].

The key feature of U-Net is the set of skip connections that link each decoder stage directly to the encoder stage of matching resolution, reinjecting fine spatial detail, such as edges and texture, that would otherwise be lost during downsampling. Because the network contains no fully connected layers, it can be applied to images of arbitrary size, and its authors showed that, combined with strong data augmentation, it can be trained effectively from very few annotated images. In this work, the original encoder was replaced with an EfficientNet-B2 network pretrained on ImageNet, so the model starts with general-purpose visual features rather than learning them from scratch. This combination of a pretrained encoder and a dense, per-pixel training signal is what allowed U-Net to learn effectively from the small sonar dataset, where YOLO's instance-based approach did not.

U-Net++ [6] is a refinement of U-Net that redesigns the skip connections, as shown in Figure 9. In U-Net, each encoder level is directly connected to the corresponding decoder level at the same resolution. U-Net++ replaces these direct links with nested, densely connected skip pathways: a grid of intermediate convolutional blocks sits between encoder and decoder, and each block combines the outputs of all preceding blocks at its level with an up-sampled feature map from the level below. Feature maps are therefore refined step by step through several short hops rather than one long jump. This is intended to narrow the “semantic gap” between the encoder's detailed but shallow features and the decoder's coarser but more context-rich features, so that the two are more similar when merged, which, in principle, improves the precision of segmentation boundaries.

U-Net++ also supports deep supervision, in which a loss is computed at the output of each nested sub-network (L1 to L4 in Figure 9). This allows a trained model to be pruned to a shallower sub-network at inference time, trading a small loss of accuracy for faster prediction. The cost of these additions is a larger decoder with more parameters, which needs more memory and training data than the original U-Net.

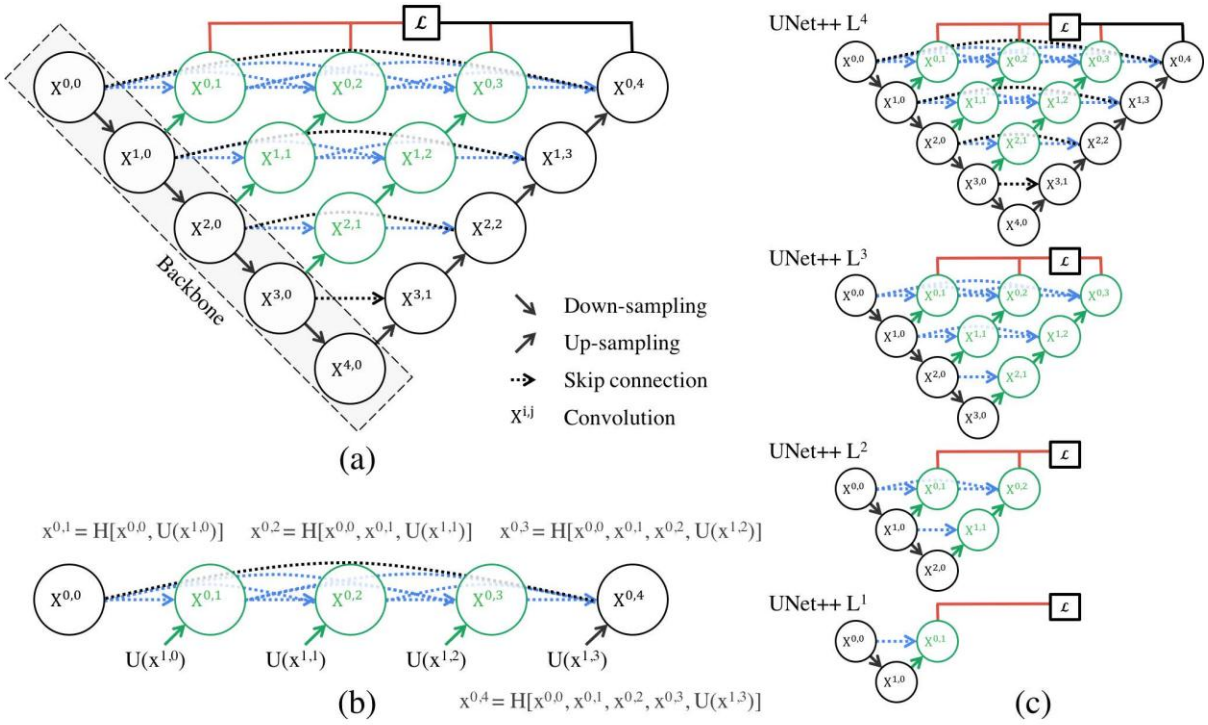


Figure 9. U-Net++ architecture: (a) nested encoder-decoder with dense skip pathways and deep supervision; (b) detail of the first skip pathway; (c) the pruned sub-networks L1 to L4 that can be selected at inference, adapted from Zhou et al. [6].

2.4. Training Configuration

Building on the methodology described above, the first version of the YOLO detection and segmentation model was trained on the annotated Side Scan Sonar dataset collected during the survey in the Nea Peramos Gulf. This section presents the dataset used for this initial training round, the training configuration, and the resulting performance. It should be noted that these results are based on a limited, early-stage dataset and are intended to demonstrate the feasibility of the workflow described in the Methodology section. The model will be retrained and refined as additional annotated data becomes available throughout the project.

2.4.1. YOLO

A YOLOv8 nano segmentation model (YOLOv8n-seg) was selected for this preliminary round, as its lightweight architecture is well-suited to a small initial dataset and enables fast iteration while the annotated dataset is still growing. The model was trained for 200 epochs on the seagrass class only, optimizing simultaneously for bounding-box detection and instance mask segmentation.

2.4.1.1. Algorithm Interface

The YOLOv8n-seg model introduced in Section 2.3.1 was configured and trained using the script shown in Figure 10. Using the Ultralytics YOLO library, the model is initialised from the pretrained "yolov8n-seg.pt" weights and trained with a single call to model.train(), which reads the dataset definition from "seagrass.yaml". Training ran for 200 epochs at an input resolution of 960 pixels. The batch size was set to 4 to fit within the available GPU memory, and early stopping was applied with a patience of 50 epochs, so training would end if validation performance did not improve for 50 consecutive epochs.

```python
from ultralytics import YOLO

def main():
    model = YOLO("yolov8n-seg.pt")  # nano

    model.train(
        data="seagrass.yaml",
        epochs=200,
        imgsz=960,  # ↑ bigger input
        batch=4,  # ↓ reduced for VRAM
        device=0,
        patience=50,
        save=True,
        project="runs/seagrass",
        name="v_paper_inspired",
        # Augmentation (close to paper's recipe)
        hsv_h=0.0,
        hsv_s=0.0,
        hsv_v=0.4,  # ↑ more brightness variation
        degrees=180,
        translate=0.1,
        scale=0.3,
        shear=0.0,
        flipud=0.5,
        fliplr=0.5,
        mosaic=1.0,
        mixup=0.1,
        copy_paste=0.2,
        erasing=0.4,  # ↓ cutout-like (paper uses cutout)
        # Loss weights — leave default
        box=7.5,
        cls=0.5,
        dfl=1.5,
        # Optimizer — switch to SGD per paper
        optimizer="SGD",  # ↓ was AdamW
        lr0=0.01,  # ↑ was 0.001 (SGD needs higher)
```

[Comment Translate] - ♡ | Google
Augmentation (close to paper's recipe)

***Figure 10**. Training script for the YOLOv8n-seg model, showing the dataset configuration, training hyperparameters, and augmentation settings used for the preliminary training round.*

The augmentation parameters were configured to closely follow the recipe reported in the publication of Yang et al. [4]. Hue augmentation was disabled (hsv_h = 0.0) to preserve the radiometric characteristics of the sonar imagery, while saturation was left unchanged and brightness variation was increased (hsv_v = 0.4) to improve robustness to differences in sonar gain and backscatter intensity. Geometric augmentations included full-range rotation (180°), translation, scaling, and both vertical and horizontal flipping, reflecting the fact that seagrass patches in Side Scan Sonar imagery lack a fixed orientation. Mosaic augmentation was applied at full probability, together with mix-up, copy-paste, and random erasing, the latter used as a

stand-in for the cutout augmentation described in the reference paper.

Loss weighting was left at its default configuration (box = 7.5, cls = 0.5, dfl = 1.5), and the SGD (Stochastic Gradient Descent) optimizer was used, as in the reference paper. The optimizer updates the model's weights during training based on the gradient of the loss computed for each batch of images, so that the model gradually improves its predictions. The initial learning rate was set to 0.01 to suit SGD, with a final learning rate ratio of 0.1 and a cosine learning rate schedule, momentum set to 0.937, and weight decay to 0.0005, again matching the values reported in the reference paper, with the first 3 epochs used as a warm-up period.

2.4.2. U-Net / U-Net++

Both models used an EfficientNet-B2 encoder pretrained on ImageNet, reading the same 13 train and 5 validation tiles used for YOLO. The only difference between the two runs is the model class (smp.Unet) vs. (smp.UnetPlusPlus). All other settings, data loading, and augmentation are identical, so that any difference in outcome reflects architecture alone.

2.4.2.1. Algorithm Interface

The Figure 11 below shows the training script, which is built with the segmentation_models_pytorch library and used unchanged for both runs apart from the model class. The reasoning behind the chosen settings is explained in the following paragraph.

```
model = smp.Unet(                    # or smp.UnetPlusPlus for the second run
    encoder_name="efficientnet-b2",
    encoder_weights="imagenet",
    in_channels=3,
    classes=2,                       # background, seagrass
)

# 150 epochs, batch size 4, input size 640 px (native tile resolution)
optimizer = torch.optim.AdamW(model.parameters(), lr=1e-4, weight_decay=1e-4)
scheduler = torch.optim.lr_scheduler.CosineAnnealingLR(optimizer, T_max=150)

class_weights = torch.tensor([0.3, 1.0])   # background, seagrass
loss = 0.5 * DiceLoss(mode="multiclass") + 0.5 * CrossEntropyLoss(weight=class_weights)
```

***Figure 11**. Training configuration of the U-Net and U-Net++ models, showing the EfficientNet-B2 encoder pretrained on ImageNet, the two classes (background and seagrass), the AdamW optimiser with a cosine learning rate schedule and the combined Dice and weighted cross-entropy loss.*

Both models were trained for a fixed 150 epochs without early stopping, as the validation IoU curve levelled off smoothly rather than showing sharp overfitting. Training used the tiles' native resolution of 640 px, because dense pixel-wise prediction did not require upscaling. The batch size of 4 and the AdamW optimiser, in place of the SGD optimiser used for YOLO, follow common practice for encoder-decoder segmentation networks with a pretrained backbone, where a lower, adaptive learning rate helps preserve the pretrained features during fine-tuning. Dice loss directly measures mask overlap and is robust to class imbalance; it was combined with a weighted cross-entropy term in which the seagrass class was weighted 1.0 and the background 0.3, since background pixels far outnumber seagrass pixels in every tile. Data augmentation followed the same procedure as for YOLO.

2.5. Results

The models trained on the verified side scan sonar dataset are evaluated below. Results for the YOLO instance segmentation model come first and cover its training curves, detection and mask metrics, confusion matrix and sample predictions. The U-Net and U-Net++ semantic segmentation models follow, assessed using the Intersection over Union (IoU) metric.

2.5.1. YOLO

The 143 tile-level seagrass instances used to train the YOLO model, described in Section 2.2, are summarised in Figure 12 below, which shows the number of instances, their positions within the tile, and the relative sizes of their bounding boxes.

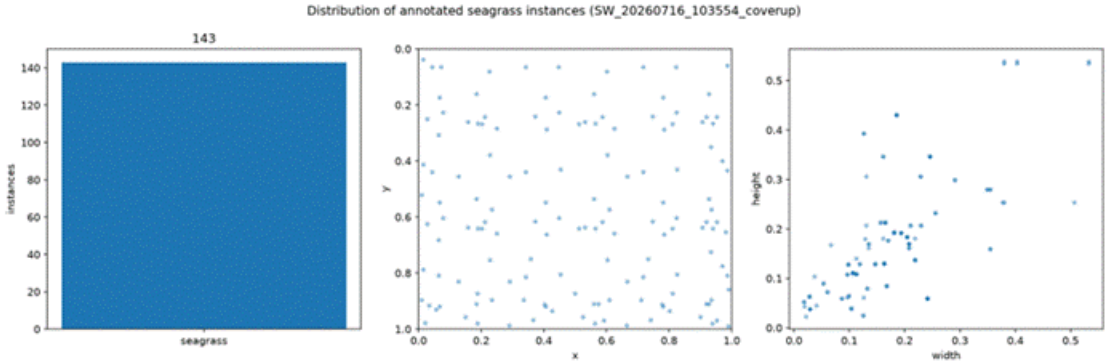


***Figure 12**. Distribution of the 143 tile-level seagrass instances used for training, showing instance counts, spatial locations within the image frame (x, y), and relative bounding box sizes (width, height).*

During training, the YOLO model optimises four loss terms: the box loss, which measures how well the predicted bounding boxes match the annotated ones; the segmentation loss, which measures the pixel-level agreement between predicted and annotated masks; the classification loss, which reflects how confidently each instance is assigned to the seagrass class; and the distribution focal loss, which refines the placement of the bounding-box edges. Decreasing losses on the training set show that the model fits the data, while the validation losses indicate whether this learning generalises to unseen tiles. The evolution of the training and validation losses (box, segmentation, classification, and distribution focal loss), together with precision, recall, and mAP metrics for both the bounding-box (B) and mask (M) heads, is summarized in Figure 13 below.

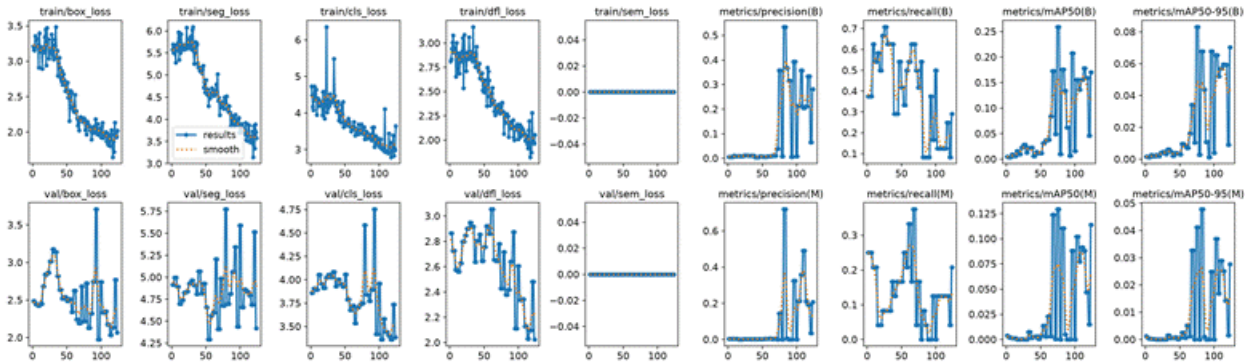


***Figure 13**. Training and validation loss curves and evaluation metrics (precision, recall, mAP50, mAP50-95) over 124 training epochs, for both the bounding-box detection head (B) and the instance-*

segmentation head (M).

In a converging run, precision, recall, and mAP would keep rising as the losses fell, before levelling off at high values. Here, these metrics rise only modestly over the first ~75 epochs, then plateau at a low level for the remainder of the 124 epochs rather than continuing to climb. This is an early indicator that the model is not learning a generalisable representation of seagrass at this dataset size: with only 143 tile-level instances, the validation set contains few examples, so the metrics are both low and sensitive to individual predictions. The same pattern is observed for the bounding-box (B) and mask (M) heads, suggesting that the limitation lies in the amount of training data rather than in either output head specifically, a point discussed further in Section 2.6.1.

2.5.1.1. Detection (Bounding Box) Performance

The bounding-box detection head reached a best mAP50 of 0.259 (epoch 74) on the validation set. By the final epoch (124), it had declined to mAP50=0.170, with precision 0.281 and recall 0.292. For comparison, the earlier attempt on the unverified mosaic never exceeded mAP50 ≈ 0.08 for the bounding-box head, but both are far below what would be considered a usable detector. The corresponding curves are shown below in Figures 14 to 17.

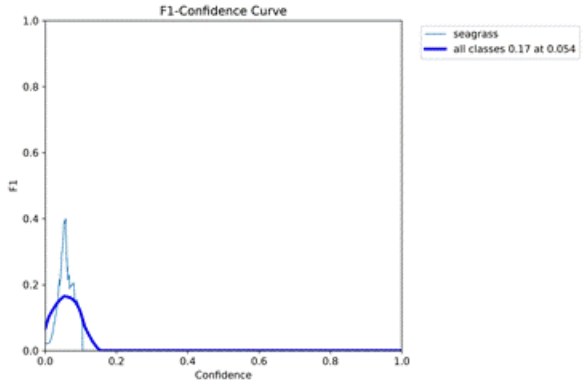


***Figure 14**. F1-Confidence curve for the bounding-box detection head.*

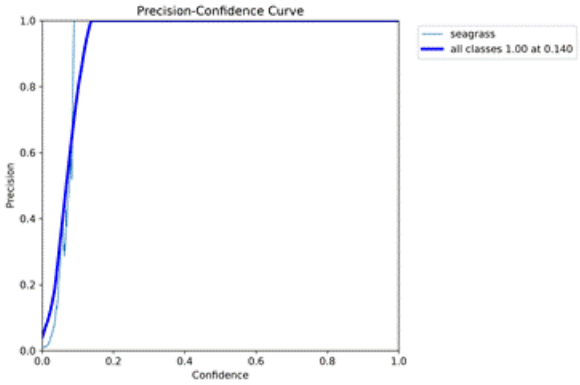


***Figure 15**. Precision-Confidence curve for the bounding-box detection head.*

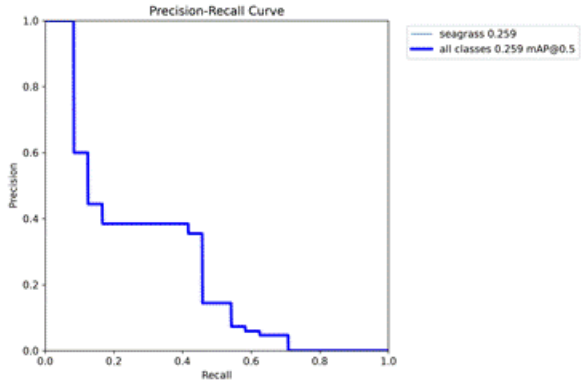


***Figure 16**. Precision-Recall curve for the*

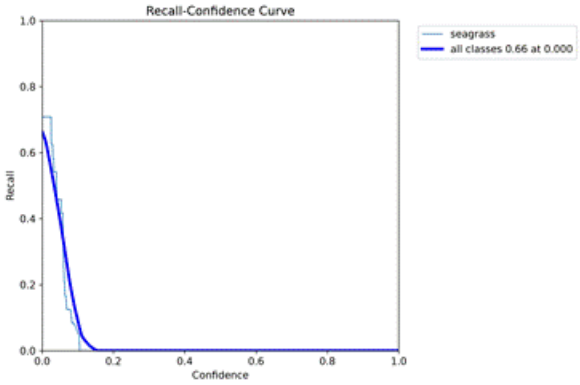


***Figure 17**. Recall-Confidence curve for the*

bounding-box detection head (mAP50 = 0.259).

bounding-box detection head.

2.5.1.2. Segmentation (Mask) Performance

The instance-segmentation head, which delineates the pixel-level extent of each seagrass patch rather than only its bounding box, reached a best mAP50 of 0.129 (epoch 74). By the final epoch, it had declined to mAP50 = 0.114, mAP50-95 = 0.028, precision = 0.207, and recall = 0.208. The mask head performs worse than the box head throughout training, which means that the model struggles even more to delineate precise boundaries than to localise a rough box. The corresponding curves are shown below in Figures 18 to 21.

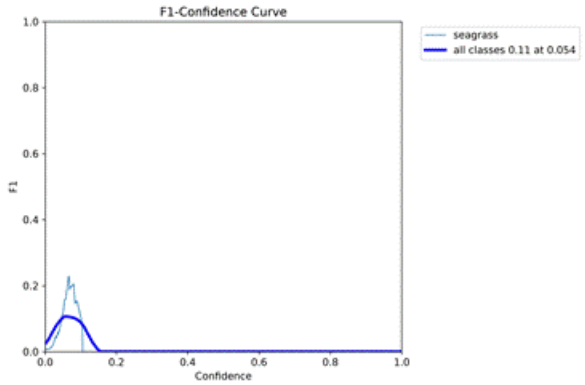


***Figure 18**. F1-Confidence curve for the instance-segmentation head.*

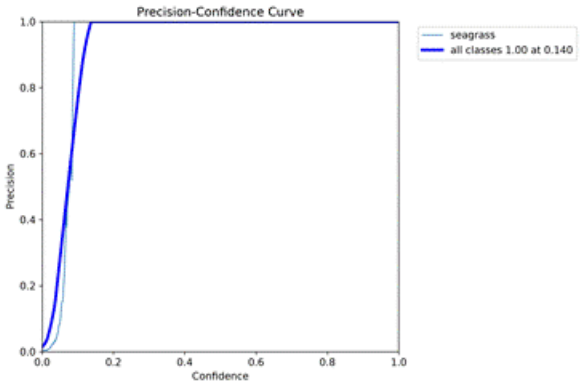


***Figure 19**. Precision-Confidence curve for the instance-segmentation head.*

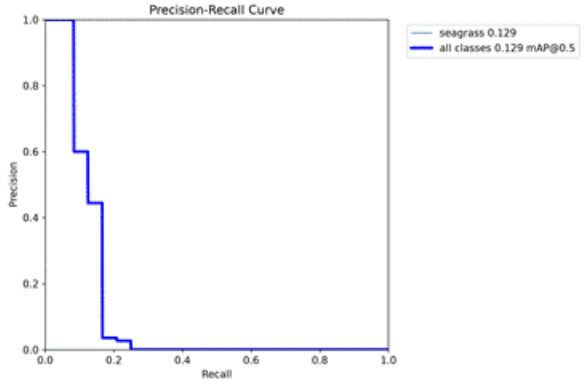


***Figure 20**. Precision-Recall curve for the instance-segmentation head (mAP50 = 0.129).*

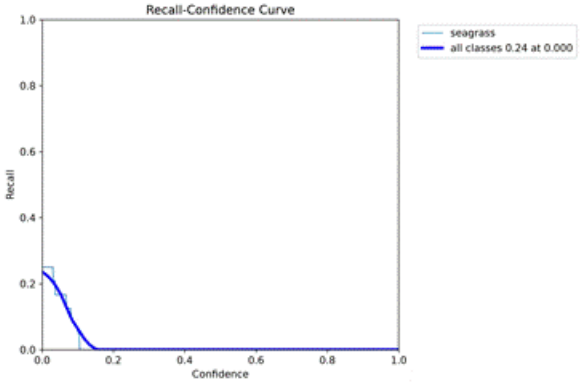


***Figure 21**. Recall-Confidence curve for the instance-segmentation head.*

2.5.1.3. Confusion Matrix

The confusion matrix below summarizes the validation results at the default confidence threshold (0.25). All 24 true seagrass instances across the 5 validation tiles were incorrectly predicted as background, with 0 True Positives (TP). This shows that the model completely fails at the threshold a user would actually run it at, as it detects nothing at all.

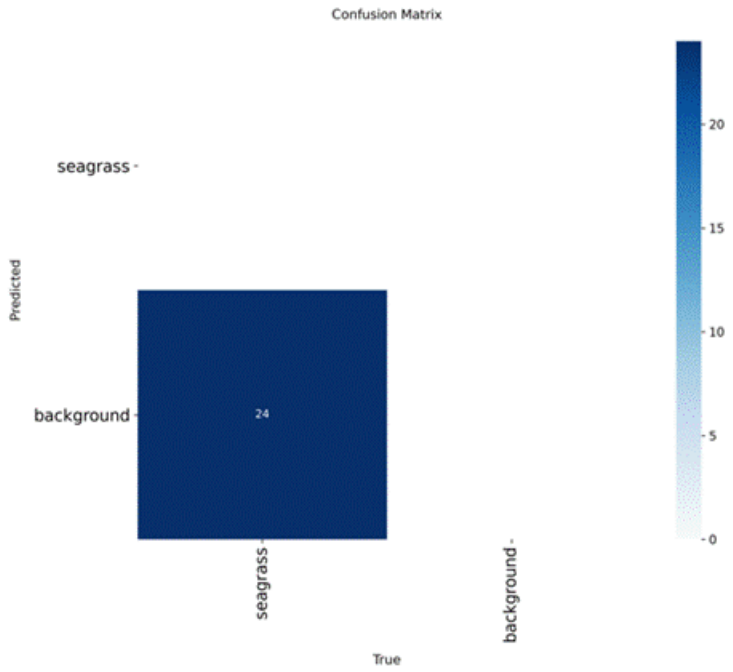


***Figure 22**. Confusion matrix for the seagrass class on the validation set, at the default confidence threshold (predicted vs. true labels).*

2.5.1.4. Sample Predictions on Validation Data

The model's predictions on the validation tiles at the default confidence threshold (0.25) are shown in Figure 23 below. No boxes or masks are drawn, because no prediction passed the threshold on any of the 5 tiles.

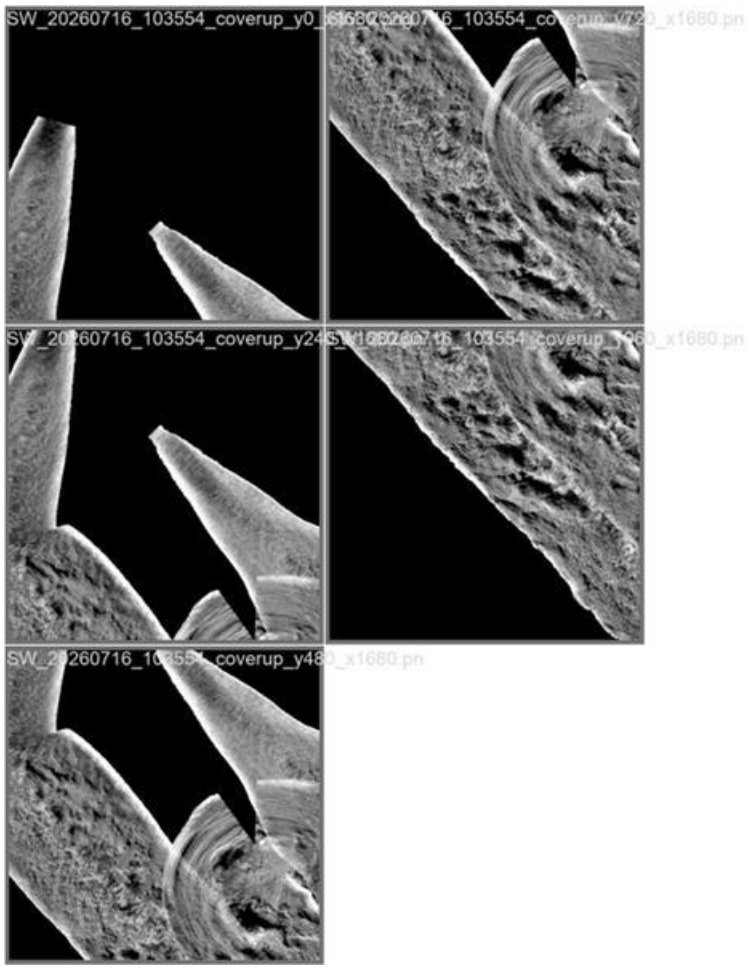


Figure 23. Validation batch with predicted seagrass instances overlaid, at the default threshold. No detections are drawn because none exceeded the threshold.

To see what the model had actually learned, rather than only what passed the default threshold, the predictions were repeated at a very low confidence threshold (0.01), which forces the model to output everything it considers seagrass, regardless of confidence. At this threshold, the model produced several hundred overlapping boxes and masks per tile, all labelled as seagrass with confidence scores no higher than ~0.13, which together covered almost the entire frame, including regions with no sonar data at all (Figure 24). This behaviour is typical of a detector that has not learned a real decision boundary, rather than one that is simply imprecise.

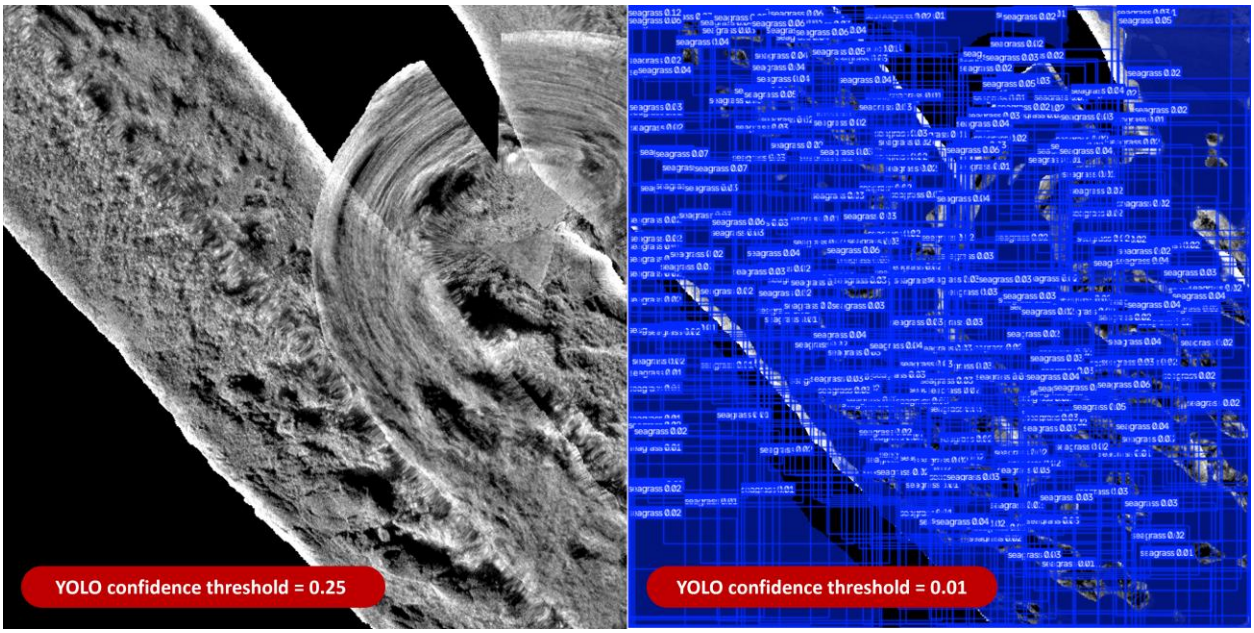


Figure 24. Same validation tile at default confidence threshold (0.25) on the left, and at confidence threshold 0.01 on the right.

2.5.2. U-Net / U-Net++

Both models were trained on 13 image tiles from a single verified sonar mosaic and evaluated on 5 validation tiles. A balance aware geographic train-validation split was applied to prevent leakage caused by overlapping tiles. Contrast Limited Adaptive Histogram Equalization (CLAHE) was applied during preprocessing to enhance texture contrast in the sonar imagery.

Performance was evaluated using the Intersection Over Union (IoU) metric, which measures the overlap between the predicted mask and the ground truth. U-Net achieved best IoU of 0.495 (epoch 95) and slightly outperformed U-Net++ which had a best IoU of 0.456 (epoch 93). Both converge smoothly, in clear contrast to YOLO with U-Net++ scoring slightly lower than the plain U-Net on every validation file. Figure 25 shows the training loss and validation seagrass IoU curves for both the U-Net and U-Net++ models.

The lower performance of U-Net++ is attributed to its greater number of parameters which refers to the additional intermediate nodes that require more training data to converge effectively. With only 13 training images, these nodes introduced noise rather than improved precision. Both models produced spatially coherent predictions that followed the general shape of the seagrass patches. These results indicate that for datasets of this scale, a simpler dense segmentation architecture is preferable. Unet++ and YOLO remain viable candidates in the case that a larger, more extensive, labeled dataset is available.

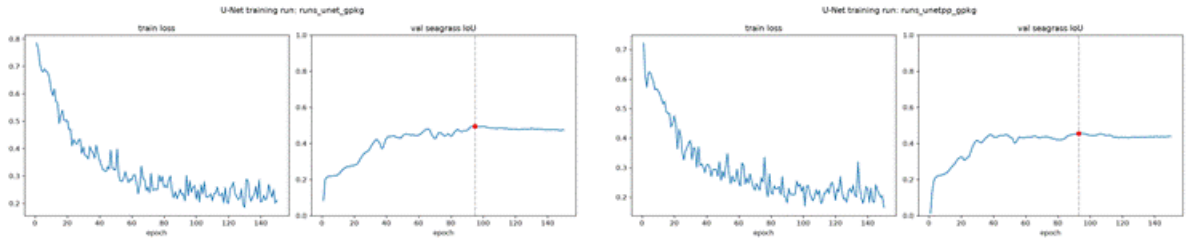


***Figure 25**. Training loss and validation IoU curves for plain U-Net (left, best IoU 0.495) and U-Net++ (right, best IoU 0.456).*

2.5.2.1. Sample predictions on validation data

To illustrate the segmentation quality, the ground truth (green) and the U-Net prediction (blue) are overlaid on a validation tile in Figure 26. The predicted mask closely follows the true seagrass boundaries and correctly excludes a swirl-shaped sonar artifact caused by the AUV turning (top right of Figure 27), even though the model was never explicitly trained to ignore such features.

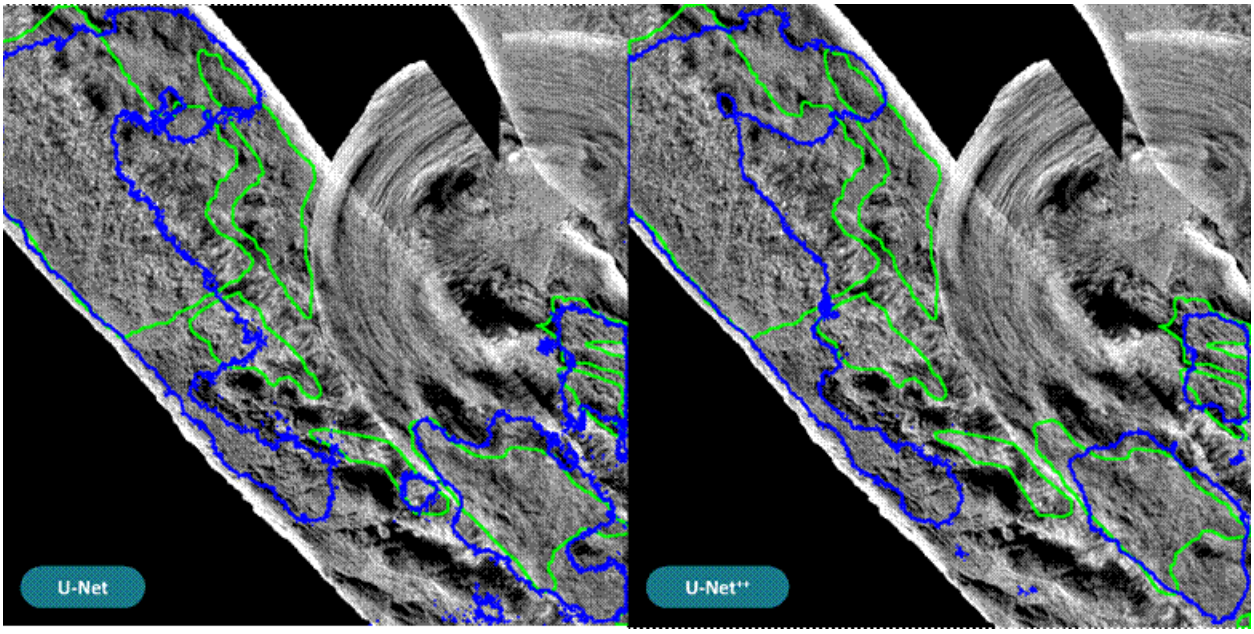


***Figure 26**. Validation tile with ground truth (green) and model prediction (blue) overlaid. On the left can be seen the prediction of the plain U-Net and on the right the prediction of the U-Net++.*

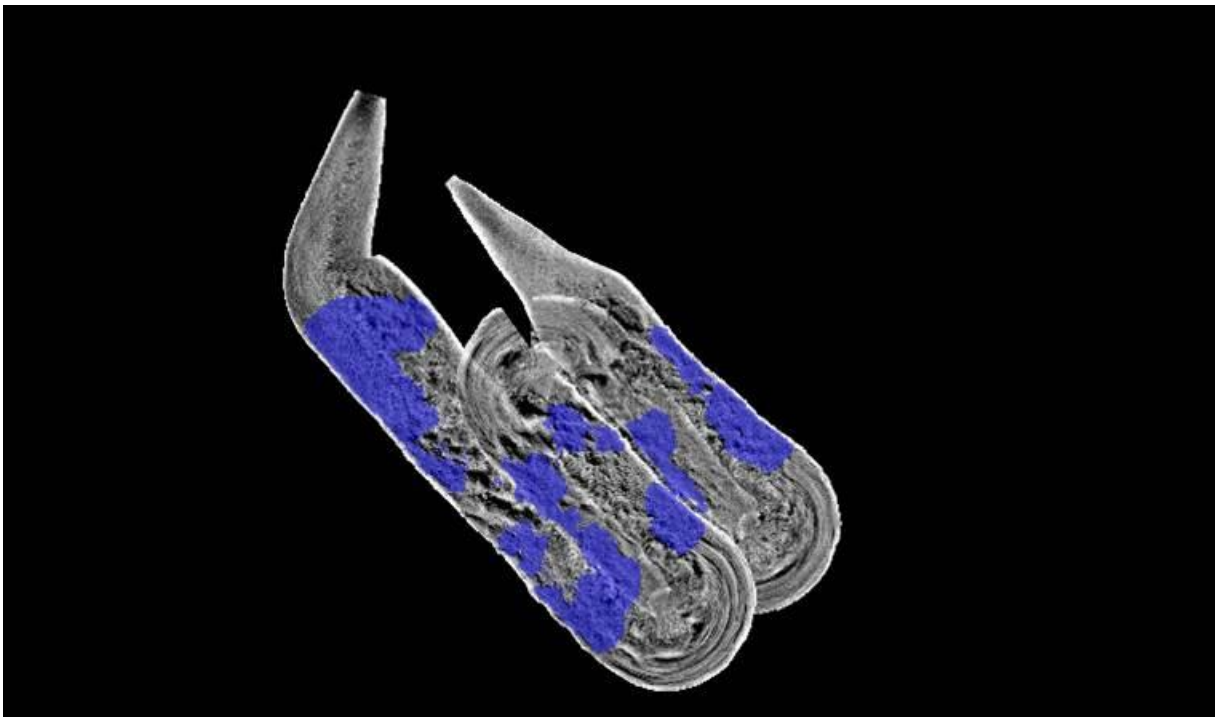


Figure 27. U-Net's seagrass prediction overlaid on the side scan sonar mosaic.

2.6. Limitations

The side scan sonar results presented above are subject to several limitations, which mainly stem from the small amount of verified data available. These concern the scale of the training dataset, the reliability of the ground truth derived from the GoPro footage, the absence of an independent test set and the extent to which the choice of model architecture depends on the size of the dataset. Each of these is discussed below.

2.6.1. Dataset scale

The verified dataset used for model comparison consisted of a single mosaic, tiled into 27 overlapping 640x640 px tiles, split into 13 training, 5 validation and 9 images dropped as buffer. This is a small sample by any standard, and the reported validation IoU (0.495 for the plain U-Net) should be read as a promising first result rather than a statistically reliable performance estimate. A handful of validation tiles can shift the metric substantially, and the 62.5% tile overlap used to maximize training coverage means the 13/5 split, while spatially buffered (to avoid leakage), still draws from a single continuous survey area rather than independent sites.

2.6.2. Ground truth reliability

Ground truth polygons were digitized in QGIS and cross-referenced against GoPro footage to confirm seagrass presence. This was a meaningful improvement over visual labeling but still an approximate field correlation which carries an irreducible uncertainty because of the different field of view between the side scan sonar and the GoPro. This sets a ceiling on how closely any model's output can realistically be expected to match the labeled boundaries.

2.6.3. No independent test set

All results reported here come from a train/validation split of the same survey mosaic. No separate, independently verified site has been held out as a test set due to the lack of verified data. Until that exists, reported metrics measure how well the model fits this one site's label distribution, not how well it generalizes to new survey locations.

2.6.4. Architecture choice is data-scale dependent

YOLOv8n-seg trained on the same 13 images but did not produce a usable model at any confidence threshold. This is consistent with instance detection's lower supervision density. Each image contributes only a handful of labeled object instances, versus every pixel contributing to a dense segmentation loss. This means that YOLO needs substantially more labeled data in order to produce a usable model whereas U-Net does not. Therefore, the architecture choice made is specific to the current data scale and should be revisited as the dataset grows.

2.7. Future work

- More GoPro verified survey sites to serve as an independent test set and to grow the training pool beyond a single mosaic.
- Re-evaluating YOLO as a viable choice once a larger labeled dataset is available, since its current failure is attributed to data scale.
- Revisiting U-Net++ (or larger segmentation architectures) once more labeled data exists since its extra capacity was not an advantage at the current scale but may become one.
- Use of SAM/SAM2 as a boundary refinement layer on top of U-Net's output, to investigate whether promptable segmentation can sharpen boundary precision. This would not address the underlying data scarcity constraint but could improve output quality independently of it.

3.GoPro Footage

Underwater video cameras, such as action cameras mounted on diver-held rigs or ROV systems, have been widely used as a complementary tool in benthic habitat surveys, providing direct visual documentation of the seafloor. Unlike acoustic methods, optical imaging allows for species level identification and visual confirmation of habitat types, making it particularly valuable for providing ground truth data. However, optical surveys are constrained by water turbidity, light availability, and limited optical coverage, making them unsuitable as standalone mapping tools over large study areas.

In EfxINNOs, GoPro underwater footage was collected with the NemoSens AUV at targeted locations within the North Aegean Sea. The footage served two purposes. It was used as a visual ground-truth reference to verify seagrass presence at locations identified in the Side Scan Sonar mosaics, and also as an independent image dataset for seagrass detection using YOLO-based object detection. By applying ML directly to the video frames, the project explored the potential of optical data as a parallel detection pathway alongside the SSS pipeline.

3.1. Data acquisition

The optical dataset used for the video-based seagrass mapping consists of AUV dives recorded with a GoPro HERO13 Black action camera mounted under the nose of the NemoSens AUV, looking forward and downward at the seabed through a flat port (Figure 28).

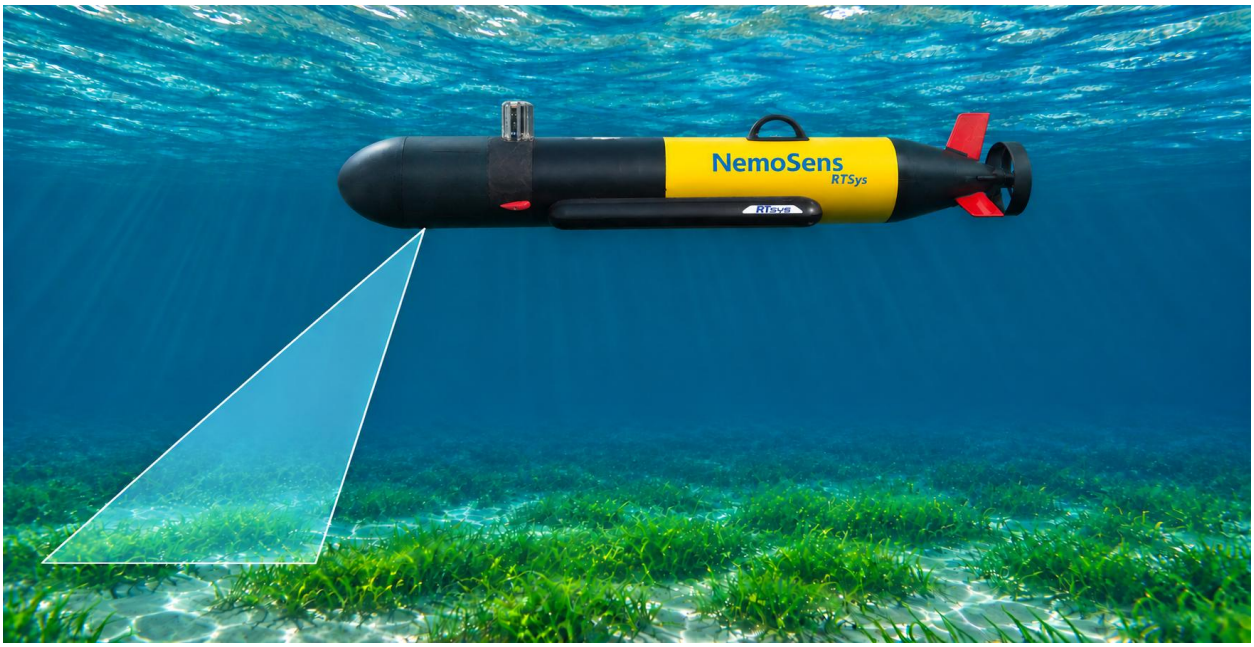


***Figure 28**. Configuration of the NemoSens AUV with the GoPro camera looking forward and downward at the seagrass meadow. The shaded area indicates the field of view of the camera.*

Each dive is stored as a pair of files, namely the video itself and a SubRip (.srt) telemetry file with the same name, in which the vehicle logs its position and sensor state synchronised with the video time. The main test dive, GX010287, was recorded on 16 July 2026, starting at 10:48, near 40.828° N, 24.340° E. A second dive, GX010175, recorded in markedly more turbid water, was used to test the robustness of the processing chain under poor visibility. The main characteristics of the test dive are summarised in Table 1.

Table 1. Main characteristics of the test dive GX010287.

Item	Value
Video	GX010287.MP4, 4K portrait (2160 × 3840 px), 29.97 fps, 14,537 frames, 8.1 min
Camera view	Forward and downward at the seabed, with the top of the frame showing the farthest part of the scene
Telemetry	GX010287.srt, one entry every 0.5 s (966 entries)
Second dive	GX010175.MP4, turbid water, 4 min 50 s
Annotation classes	Seagrass, fish, rock, debris

3.1.1. Video files

Typical raw frames of the two dives are shown in Figure 29. In both, the green haze of the water column dominates the image, and in the turbid dive GX010175 the seabed is barely visible beyond the lower part of the frame.

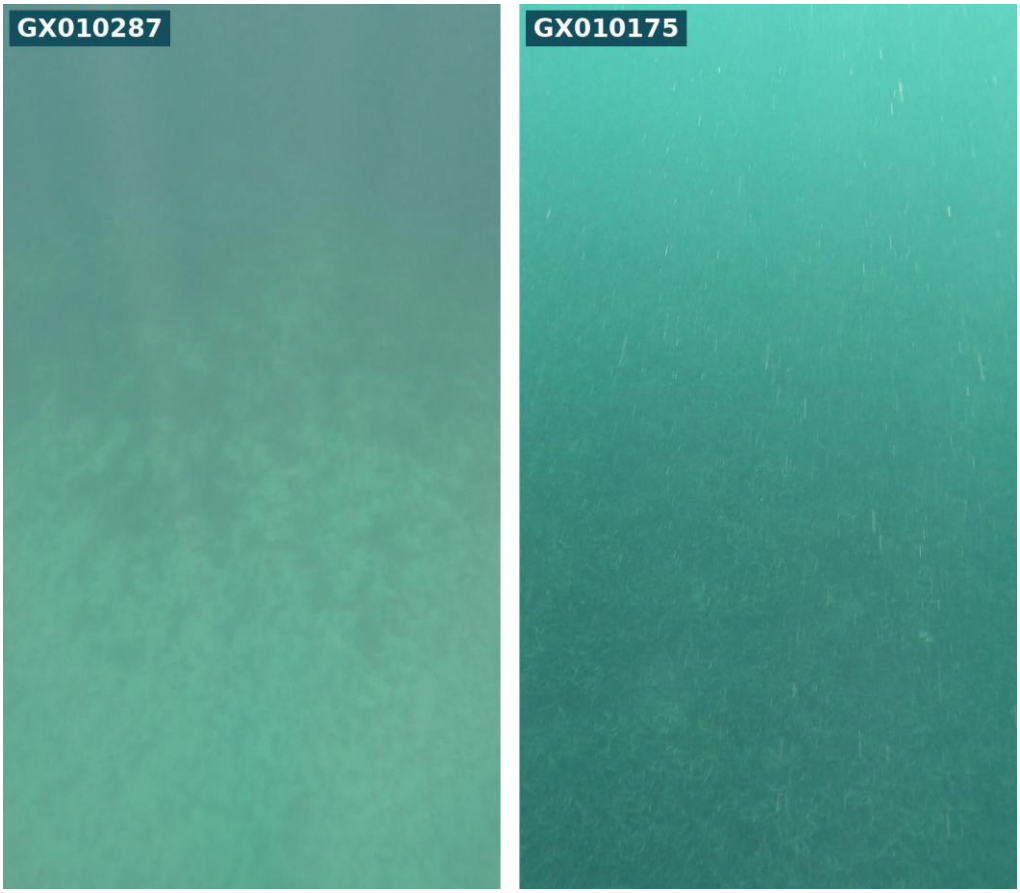


Figure 29. Raw GoPro frames of the clear-water dive GX010287 (left) and the turbid-water dive GX010175 (right), before enhancement.

Both the physically based colour correction and the projection of detections onto the seabed depend on the geometry of the camera. Since the tilt of the camera below the horizontal had not been recorded during the survey, it was estimated from the video itself, using the fact that the apparent motion of the seabed in the image (optical flow) is governed by the tilt, the field of view, the altitude and the speed of the vehicle. Initial estimates based on the optical flow

alone were 60.5° for GX010287 and 62° for GX010175, and a rounded value of 60° was adopted in the early stages of processing. This value was later refined to 64° by matching the seabed motion in GX010175 to the vehicle speed derived from the GPS track, using the field of view described below. The metadata of both video files (Table 2) show that they were recorded in 4K 16:9 with the Wide digital lens and HyperSmooth stabilisation, so the field of view was taken from the GoPro HERO13 Black action camera specifications for these settings (horizontal field of view (HFOV) of 109° and vertical field of view (VFOV) of 63° in air, for a camera held in landscape orientation).

Table 2. Camera settings recorded in the GoPro metadata.

Tag	Value	Meaning
EISE	Y	Electronic image stabilization - **on**
EISA	HS EIS	Stabilisation type - **HyperSmooth**
VFOV	W	Digital lens - **Wide**
ZFOV	123.636°	Diagonal field of view of the Wide lens, as reported by the camera
HCTL	OFF	Horizon levelling - **off**
HSGT	OFF	HindSight - **off**

The GoPro specifications refer to the field of view in air. Underwater, the camera views the scene through a flat port [7], and light is refracted as it passes from the water into the air inside the housing, so the field of view becomes narrower than specified. Using the in-air values would therefore make each frame appear to cover a larger area of seabed than it actually does, which would distort both the colour correction and the coverage maps. For this reason, the specified angles were corrected with Snell's law [8]:

$$\sin \theta_{air} = n \sin \theta_{water}$$

where θ_{water} is the angle of incidence, θ_{air} is the angle of refraction and n is the refractive index of the seawater, taken as 1.333.

Since the video was recorded in portrait orientation, the long side of the frame is vertical, which gives a vertical field of view of 75.3° and a horizontal field of view of 46.2° underwater (Table 3). These values replace the earlier provisional estimates of 60° and 72°, which were derived from the optical flow and the GoPro metadata.

Table 3. Camera geometry adopted for the video processing.

Camera parameter	Value	Source
Tilt below horizontal	64°	Best fit of the seabed motion in GX010175 against the GPS speed, using the field of view below. Earlier optical-flow estimates were 60.5° (GX010287) and 62° (GX010175)
Vertical field of view (underwater)	75.3°	GoPro specifications for 4K 16:9 Wide with HyperSmooth (109° × 63° in air), narrowed by the flat port (Snell's law, n = 1.333), for portrait video with the long side vertical
Horizontal field of view (underwater)	46.2°	As above, used by the detection overlay, the depth viewer and the coverage mapping

3.1.2. SRT telemetry files

Each telemetry entry recorded in the .srt file holds the time stamp, the position of the vehicle and three sensor values (Table 4). The altitude above the seabed, measured by the Doppler Velocity Log (DVL) of the vehicle, and the underwater latitude and longitude, provided by its navigation system, are both recorded into the .srt file. A navigation log with heading, pitch and

roll was not available for these dives, so the heading of the camera was derived from the direction of travel of the vehicle.

Table 4. Fields of the .srt telemetry log and their range on the test dive.

Field	Meaning	Range on GX010287
lat, lon	Position of the vehicle (WGS84)	An area of approximately 150 × 190 m
imm.	Camera depth below the sea surface (m)	1.2 to 9.9 m
alt.	Camera height above the seabed (m), from the DVL	2.0 to 13.3 m
light state	Camera lights on (1) or off (0)	1 throughout the dive

An excerpt of the telemetry log of GX010287 is shown in Figure 30. Each entry, recorded every 0.5 s, holds the video time interval, the date and time, the position and the sensor values.

```
1
00:00:00,000 --> 00:00:00,500
2026/07/16-10:48:02 -- lat:40.828139 lon:24.340438
imm.:4.585355 alt.:10.893332 -- light state:1

2
00:00:00,500 --> 00:00:01,000
2026/07/16-10:48:03 -- lat:40.828140 lon:24.340447
imm.:4.464695 alt.:11.919866 -- light state:1

3
00:00:01,000 --> 00:00:01,500
2026/07/16-10:48:03 -- lat:40.828141 lon:24.340456
imm.:4.283705 alt.:11.919866 -- light state:1

4
00:00:01,500 --> 00:00:02,000
2026/07/16-10:48:04 -- lat:40.828142 lon:24.340465
imm.:4.213320 alt.:11.919866 -- light state:1
```

Figure 30. Excerpt of the .srt telemetry log of GX010287, showing the first four entries.

During processing, it was found that the telemetry log for GX010175 labels its entries 0.5 s apart, even though it contains about 6 entries per second, so its subtitle times run 3 times too slowly (14 min 53 s of subtitles for a 4 min 50 s video). To prevent such errors from propagating into the maps, all steps that use the telemetry read the logs through a common reader which compares the subtitle times with the clock logged inside each entry. When the two disagree by more than 10%, every entry is timed by its logged clock. The test dive GX010287 is not affected by this issue.

3.2. Video processing pipeline

This section provides an overview of the processing chain developed to convert GoPro footage and telemetry data into georeferenced maps of seagrass meadows. It first outlines the individual processing steps, the software they rely on, and the outputs of each step, and then describes the configuration file that controls the whole chain. The individual steps are presented in detail in the following sections.

3.2.1. Processing steps

A dedicated processing chain was developed in Python to turn a GoPro dive video and its telemetry log into a georeferenced map of seagrass cover. The chain consists of seven individual scripts, numbered 0 to 6, which are run one after another, together with an optional depth viewer (step 4a). All settings are kept in a single, commented configuration file, config.ini, in which paths are given relative to the project folder and every parameter is accompanied by a short explanation of its effect. Because every parameter is stored in one place, a run can be reproduced or repeated on a new dive by editing config.ini alone, without changing the code. Each script reads the outputs of the previous ones, so that a step only needs to be rerun when its inputs or its section of config.ini change. The processing was developed and tested on a standard consumer laptop with an NVIDIA RTX 4060 GPU, using PyTorch 2.6 with CUDA, rather than on a dedicated workstation or computing cluster. The processing times reported in the following sections therefore reflect what can be achieved with commonly available hardware and demonstrate the pipeline's efficiency.

The processing chain relies on a set of open-source libraries, using Ultralytics for the detection and segmentation models, OpenCV for reading and writing video and computing optical flow, imageio-ffmpeg for hardware-accelerated video input and output in the enhancement step, labelme for annotation, GeoPandas, Rasterio and pyproj for the geospatial outputs, SciPy and Matplotlib for analysis and plotting, and Hugging Face Transformers for the monocular depth model used in steps 0 and 4a. The code is organised so that shared helper modules, model weights and third-party networks each have their own folder, while the outputs are grouped by processing stage (enhancement, prediction, points and coverage). The results are accompanied by a QGIS project containing the side-scan sonar mosaic of the same area, allowing the optical maps to be compared directly with the acoustic data presented in Section 1. Table 5 summarises each step and its outputs, and Figure 31 illustrates the complete workflow, from the input data to the final coverage map.

***Table 5**. Steps of the video processing pipeline.*

Step	Script	Function	Main output
0 (optional)	0.enhance_video.py	Removes the water haze and colour cast from the video	Enhanced video with a matching .srt copy
1	1.extract_frames.py	Saves one still frame every few seconds	Still frames (JPEG)
2	2.label.py	Opens labelme to outline seagrass, fish, rock and debris	One JSON label file per frame
3	3.train.py	Builds the dataset and trains a YOLO segmentation model	Versioned trained model
4	4.live_detect.py	Runs the model on the video, frame by frame	Annotated video and per-frame CSV
4a (optional)	4a.live_depth.py	Plays the video as a Depth Anything V2 depth map in meters	Depth video
5	5.make_points.py	Joins the detections with the GPS log	GPS points (CSV and GeoPackage)
6	6.make_coverage_map.py	Projects the detections onto the seabed	Coverage raster and seagrass polygons

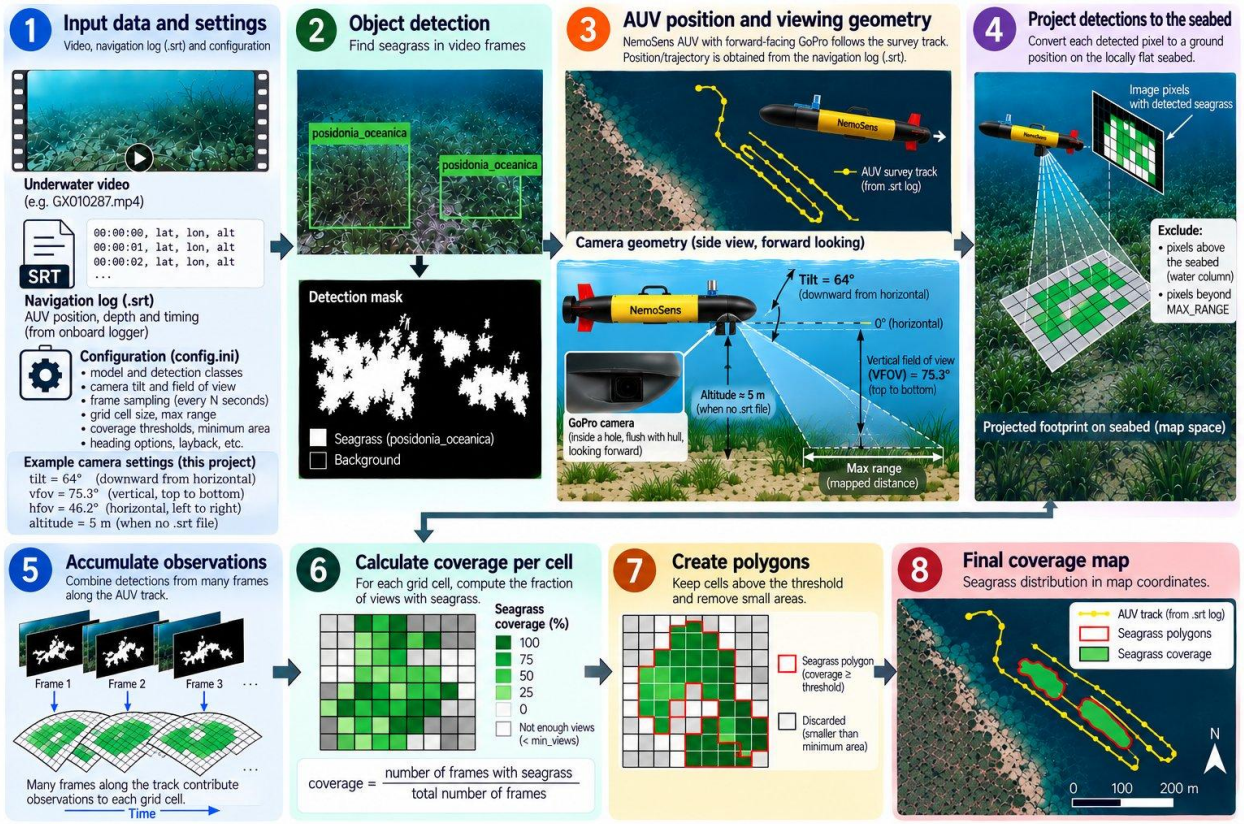


***Figure 31**. Schematic overview of the workflow from survey video to seagrass coverage map, showing (1) input video, telemetry log and configuration, (2) seagrass detection in the video frames, (3) AUV position and viewing geometry, (4) projection of the detections onto the seabed, (5) accumulation of observations from many frames, (6) coverage per grid cell, (7) polygon creation and (8) final coverage map.*

Steps 1 to 3 are only needed to train a new model. Once a trained model is available, mapping a new dive requires only steps 4 to 6, or step 6 alone if only the coverage map is needed. In its latest run on the test dive GX010287 (model v10 applied to the Natural-enhanced video), the pipeline mapped 2,431 m² of seabed, of which 1,299 m² (53%) was classified as seagrass meadows, distributed in 36 patches, and 456 m² (19%) as rock. The following sections describe each step in detail.

3.2.2. Configuration file

The configuration file (*config.ini*) is divided into nine sections, one for each script plus [general], which lists the classes used for labelling and training (Table 6). A setting left empty takes its value from an earlier section, so that shared values are defined only once. For example, the camera geometry (tilt 64°, vertical and horizontal fields of view 75.3° and 46.2°) is set in [enhance] and is used by the enhancement, the distance lines of the detection video, the depth viewer and the coverage map.

***Table 6**. Sections of the configuration file config.ini and the main settings used in this study.*

Section	Used by	Main settings (value used in this study)
[general]	Steps 2, 3	Classes offered for labelling and training (posidonia_oceanica, fish, rock and debris)

Section	Used by	Main settings (value used in this study)
[enhance]	Step 0 (geometry also Steps 4, 4a and 6)	Video and time range. Enhancement method (Natural). Range source (telemetry or Depth Anything V2). Natural tuning (sand brightness 80, haze 1.0, shadow balance, contrast and altitude fading). Particle removal, multi-frame denoising, per-scene white balance. Camera tilt 64°, vertical and horizontal fields of view 75.3° and 46.2°
[extract]	Step 1	Video to take frames from, interval between frames (8 s, or 1 s for the debris frames), optional output folder and time ranges
[label]	Step 2	Frame folder to be labelled
[train]	Step 3	Classes to train on (posidonia_oceanica, rock and debris (added from v12). Base model YOLO11s-seg. Up to 150 epochs, early stop after 40 without improvement. Image size 640 px, batch 8, validation share 20%, random seed 42
[detect]	Steps 4, 5 (and 6 by default)	Video and model version, minimum confidence 0.40, bottom zone 25% of the frame height, temporal smoothing 1 s, optional SAM2 refinement or tracking (off)
[depth]	Step 4a	Video and depth model, map type (distance or height), colour scale and display
[coverage]	Step 6	Classes to map (posidonia_oceanica, rock, debris), camera heading and offset, layback behind the GPS antenna (0 m), maximum mapped range 8 m, cell size 0.25 m, minimum views 3, class threshold 50%, minimum polygon area 1 m²
[points]	Step 5	Minimum cover of the bottom zone for a seagrass point (5%)

Every setting is preceded by a comment that explains its effect and its accepted values, as shown for the [coverage] section in Figure 32. Changing a value, for example the maximum mapped range or the class threshold and rerunning the corresponding script is enough to produce a new result without editing the code.

```
[coverage]
; 6.make_coverage_map.py: puts every analysed frame's seagrass on the map (camera position
; from the .srt, flat seabed) and makes a coverage raster + polygons (shapefile / GeoPackage).
; Empty = same as in [detect] (video, model, conf) or [enhance] (tilt, vfov).
video =
model =
conf =
; Classes to map, comma separated (each gets its own raster, shapefile and GeoPackage layer).
; Classes the model doesn't know are skipped.
classes = posidonia_oceanica, rock
; Seconds between analysed frames (0.5 = every half second)
every = 0.5
; Camera angle below horizontal and field of view (degrees), only to override [enhance] for the
; map and the distance lines of 4.live_detect.py; empty = as in [enhance]
tilt =
vfov =
hfov =
; Direction the camera looks: track = direction of travel from the GPS, or a compass value (0-360)
heading = track
; Camera direction relative to that heading in degrees (e.g. 180 = looking backwards)
heading_offset = 0
; Metres the camera is behind the GPS antenna along the track (0 = GPS on the camera)
layback = 0
; Only map pixels up to this distance from the camera (far pixels are small and less accurate)
max_range = 8
; Grid cell size in metres
cell = 0.25
; Cells seen in fewer frames than this stay empty
min_views = 3
; A cell belongs to a class when at least this % of the frames that saw it found that class there
threshold = 50
; Polygons smaller than this (m2) are dropped
min_area = 1
```

***Figure 32**. The [coverage] section of config.ini (lines 243 to 275), which controls the coverage map of Step 6. Comments (green, starting with ;) explain each setting, and empty values are taken from [detect] or [enhance].*

The file *config.ini* holds only the current settings and is overwritten when it is edited. A copy of the file should therefore be archived with the outputs of each run, so that every map can be traced back to the settings that produced it.

3.3. Underwater video enhancement (Step 0)

Underwater footage is degraded by absorption and scattering, two physical processes that act on light as it travels through water. Both depend on the wavelength of the light and the distance it travels through the water, so their effect on a video varies with the camera's depth below the surface and the distance between the camera and the seabed.

Sunlight at the surface contains all visible colours, but water absorbs long wavelengths much faster than short ones (Figure 33). Red light (about 650 nm) is absorbed rapidly and is mostly lost within the first 5 to 10 m; green light (about 550 nm) is attenuated at an intermediate rate; and blue light (about 450 nm) penetrates the deepest. As a result, the ambient light that illuminates the seabed becomes increasingly blue-green with depth. Near the surface, a scene appears in its natural colours; at around 5 m the reds begin to fade, and by 10 m the scene already looks predominantly blue-green. In clear oceanic water the remaining light is mostly blue, whereas in coastal waters such as the survey area, phytoplankton and dissolved organic matter also absorb blue light, so the cast shifts towards green. The same absorption acts a second time on the light reflected by the seabed on its way to the camera, so the colour loss in each pixel depends both on the depth of the scene below the surface and on its distance from the camera.

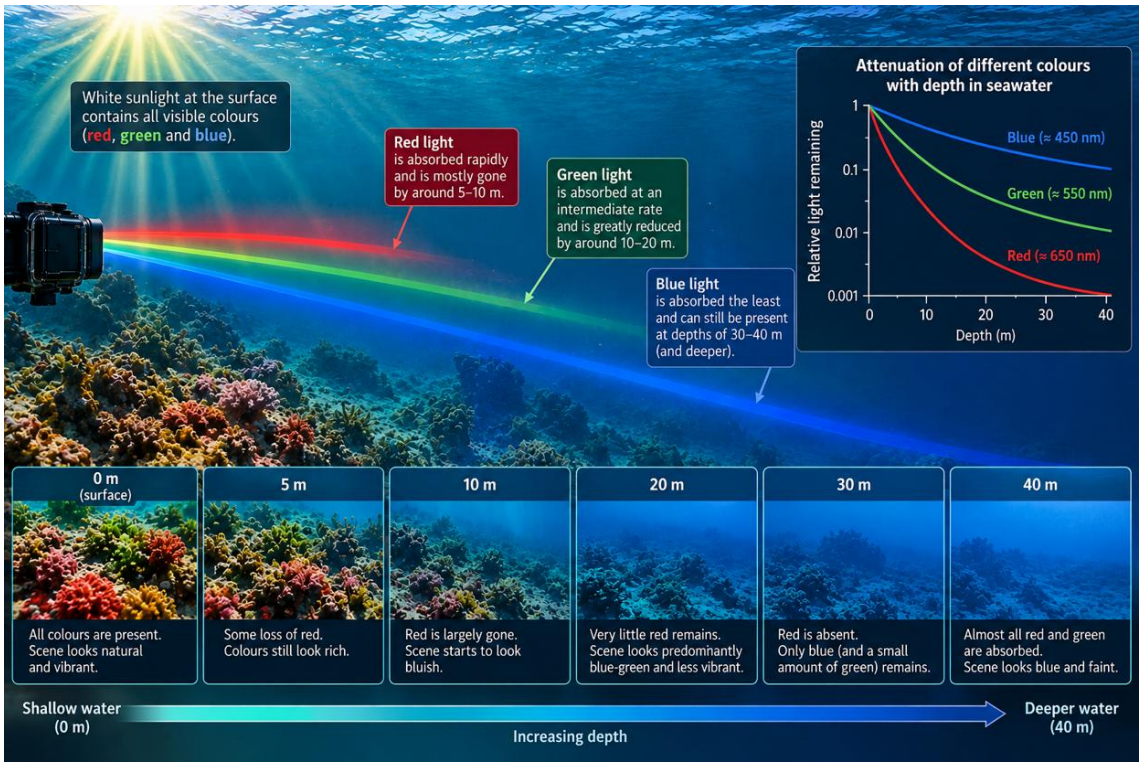


Figure 33. Attenuation of sunlight with depth in seawater. Red light is absorbed first, followed by green, while blue penetrates deepest. The panels illustrate the typical appearance of a scene from the surface down to 40 m.

In addition to absorption, seawater contains very small, suspended particles, such as plankton, fine sediment and air bubbles, which scatter light in many directions (Figure 34). Part of the ambient light scattered by the particles between the camera and the seabed is redirected into the lens and is added to the light coming from the seabed itself. This extra light, known as backscatter, carries no information about the scene and acts as a veil that reduces contrast and gives distant parts of the image a hazy, blue-green appearance. Because more particles lie along a longer line of sight, backscatter increases with the distance to the seabed, and it becomes considerably stronger in turbid water.

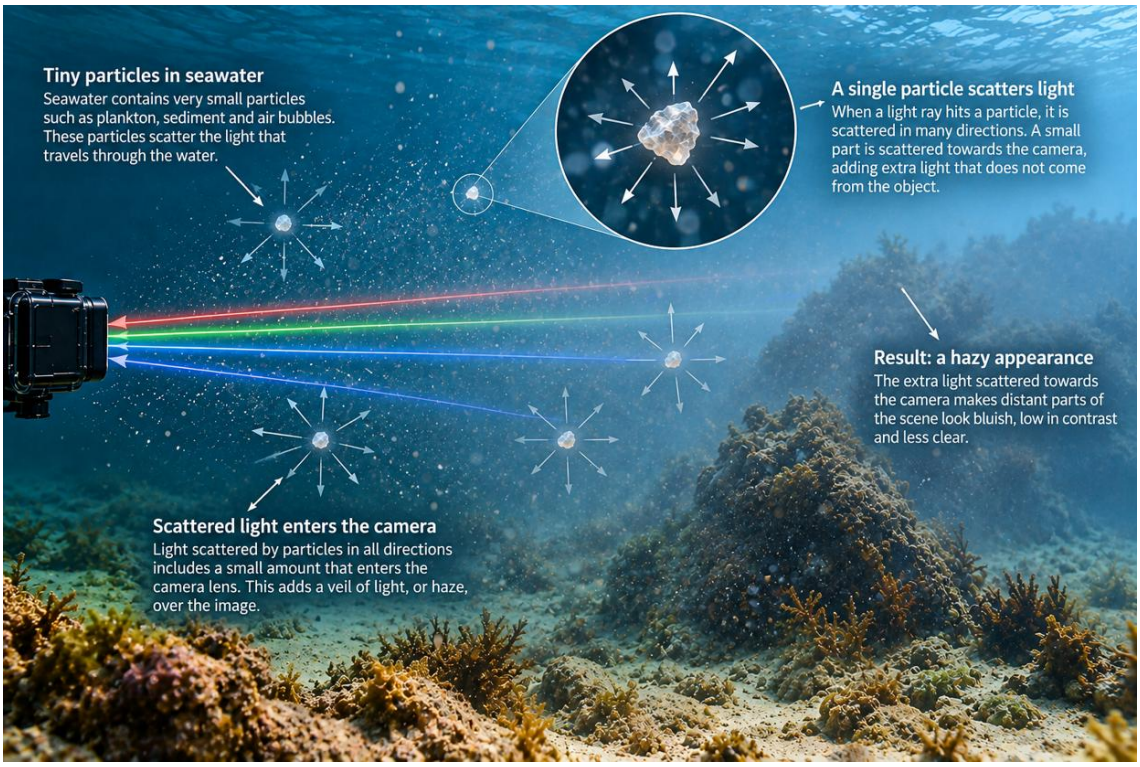


Figure 34. Scattering of light by suspended particles in seawater. The part of the scattered light that enters the camera adds a haze (backscatter) that reduces contrast and makes distant parts of the scene appear blue-green.

In this survey, the main cause of the colour cast is the depth at which the AUV flies. The AUV operates in altitude-keeping mode, holding a constant height of 3 m or 5 m above the seabed, so its immersion follows the bathymetry. During the dive, the vehicle's immersion varies from about 1.2 m to 9.9 m, a range over which most of the red light is lost (Figure 33). The colour balance of the footage therefore changes considerably within a single video. Close to the surface, the scenes retain much of their natural colour, whereas in the deepest parts they show a strong green-blue cast. Because the camera is tilted forward, the upper part of each frame views the seabed at a distance of several meters, so both colour loss and backscatter are particularly strong there, whereas the lower part of the frame, which looks at the seabed near the vehicle, is much less affected.

A single, fixed colour correction cannot compensate for effects that vary with depth along the dive and with distance within each frame. Step 0 removes the haze, restores the lost colour and saves a new enhanced video together with a matching copy of the telemetry log. The step is optional. However, a detection model must be trained and applied to the same type of video (original or enhanced), since the appearance of seagrass differs markedly between the two.

3.3.1. Overview of the enhancement methods

Because no single published method was known to perform well on shallow, seagrass-covered seabed viewed obliquely from an AUV, sixteen methods were implemented and compared on the project footage. They fall into three groups: six physically based methods that fit an image formation model to each video (Section 2.3.2), eight pretrained deep learning networks (Section 2.3.3), and two classical single-frame dehazing methods (Section 2.3.4). Two hybrid combinations, in which a physical correction is followed by WaterNet, were also tested. Combined with the optional Depth Anything range, denoising and local contrast enhancement, this gave 27 distinct configurations, all evaluated on the same clips (Figure 37).

3.3.2. Physically based methods

Six of the methods are physically based: Sea-thru, DeepSeeColor and OceanLens, adapted from the literature, and Physics, Natural and Motion, developed in this project. Rather than learning a correction from example images, they model how the water attenuates the light reflected by the seabed and adds haze, estimate the parameters of this model from each video, and invert it to recover the true colours of the scene. They therefore need no training data and adapt to the water conditions of each clip, but they require the distance to every pixel. The following subsections describe the common image formation model, how the range is estimated and the safeguards that prevent the seabed from distorting the fit. All six methods rest on this model but use it in different ways. Physics simplifies it to a single attenuation coefficient per channel, Sea-thru uses it in full with range-dependent attenuation, DeepSeeColor and OceanLens fit its parameters with an optimiser or small neural networks, Natural keeps its haze term but replaces the attenuation with a smooth curve fitted to the data, and Motion fits its parameters from seabed patches observed at two distances. Each method is then described individually.

3.3.2.1. Image formation model

Light reflected by the seabed is attenuated on its way to the camera, red first, then green and then blue, while the water itself scatters ambient light towards the camera and adds a haze known as backscatter. The Sea-thru model of Akkaynak and Treibitz [9] describes the signal recorded in each colour channel c as:

$$I_c = J_c\, e^{-\beta_{D,c} d} + B_c\left(d\right)$$

$$B_c\left(d\right) = B_{\infty,c}\left(1 - e^{-\beta_{B,c} d}\right)$$

where I is the colour recorded by the camera, J is the true colour of the scene, d is the range from the camera to the scene, βD is the attenuation coefficient of the direct signal and B is the backscatter, which grows with range towards its asymptotic value B_∞. The physically based methods estimate B and β_D from the video itself and then invert the equation to recover J. Physics, Sea-thru, Natural and Motion carry out all fits on linear light, not on gamma-encoded pixel values. DeepSeeColor and OceanLens, like their original implementations, work on gamma-encoded values.

3.3.2.2. Range estimation

Physically based methods require the range for every pixel. Two options were implemented (Table 7). The first derives the range from the altitude in the telemetry log, the camera tilt and the vertical field of view, assuming a flat seabed. The second uses the monocular depth estimation network Depth Anything V2 (Small) [10], which predicts, for each pixel, the distance to the camera. Despite the term 'depth', this is not water depth below the surface, and the values are only relative (which points are nearer and which are farther), without units. They are converted to meters using the altimeter. The bottom 10% of image rows show the seabed closest to the camera, and their values are scaled to match the range computed for them from the AUV's altitude above the seabed. To avoid flicker, the distance map is smoothed over time, with each new frame contributing 30% of the result.

***Table 7**. Range estimation options for the physically based enhancement.*

Range source	How the range is obtained	Strength	Weakness
Telemetry (.srt)	Altitude from the .srt log plus camera tilt and vertical field of view	Fast, no additional model	Assumes a flat seabed. The seagrass canopy, which rises above the seabed, is treated as farther away than it is and is over-corrected to grey-olive
Depth Anything V2	Relative distance from the camera per pixel, with the bottom 10% of rows scaled to the range derived from the AUV's altitude	Follows slopes, rocks, fish and open water	Slower, and not trained on underwater imagery

3.3.2.3. Robust fitting

Since the seabed itself is part of the signal, the fit of the water parameters must be protected from scene content. In the Physics and Sea-thru methods, attenuation samples closer to the camera than the brightest sample are discarded because the increase in signal there is caused by the scene (dark seagrass directly below the camera) rather than by the water. Samples deviating by more than three robust standard deviations from the fitted curve (four for the smooth curve of the Natural method) are removed, and the fit is repeated; outside the fitted distance range, the attenuation is held constant. For each run, a diagnostic chart of the fit is saved, in which the rejected samples are drawn as hollow symbols.

3.3.2.4. Individual methods

Physics. The Physics method is the simplest form of the image formation model and was developed within the project as a baseline for the other physically based methods. The water is fitted once for the whole video, so the correction does not flicker from frame to frame. It is fitted on sample frames spread over the video, in linear light, with the range obtained as described in Section 2.3.2.2. The haze is fitted first, through the darkest 1% of the pixels in each of 15 distance classes, which are assumed to contain almost nothing but haze. Through these pixels, a single saturating curve per channel is fitted with a robust loss and, unlike in Sea-thru, without a residual term:

$$B(d) = B_{\infty}\left(1 - e^{-\beta_B d}\right)$$

After the haze is subtracted, the average signal is taken in 20 distance classes. Assuming that the seabed has the same average colour near and far, a straight line fitted to the logarithm of this average against distance gives one attenuation coefficient per channel, constant at all distances, with the outlier rules of Section 2.3.2.3. Each pixel is corrected by removing the

haze and dividing by the transmission:

$$J = \frac{I - B(d)}{e^{-\beta_D d}}$$

The gain is limited to 5× so that dark areas are not turned into noise. An optional correction compensates for the bluer, dimmer daylight when the camera depth changes during the dive. Finally, each channel is stretched separately between the 0.5th and 99.5th percentiles of the corrected sample frames. With the telemetry range, the correction reduces to a single gain and a single offset per image row, so it is fast.

Sea-thru. Sea-thru [9] is based on the authors' revised underwater image formation model, in which the colour loss of light reflected by the scene and the build-up of haze are governed by different coefficients, and the attenuation varies with distance rather than remaining constant. It was designed for still images with a known distance map, which, in the original work, was obtained via structure-from-motion from several overlapping photographs. The haze is first estimated from the darkest pixels at each distance, which are assumed to contain almost nothing but haze. The attenuation is then estimated from the local average colour of the image, which indicates how much light of each colour reaches each point, and is described as a function of distance. Removing the haze and compensating for the attenuation recovers the colours as they would appear without water. In this project, Sea-thru uses the same sample frames, distance classes, and outlier rules as the Physics method, with three differences that align with the original method. First, the residual term is added to the haze fit. Second, the attenuation is estimated from the local average colour, computed with a blur that extends mainly along the image rows, which lie at similar distances, so that near and far parts of the scene are not mixed. Third, the attenuation varies with range, in the form $a + b\,e^{-cd}$, fitted to the median of each distance class. As in Physics, the attenuation is held constant outside the fitted range, the gain is limited to five times, and each channel is then stretched separately.

DeepSeeColor. DeepSeeColor [11] was developed for real-time colour correction on board autonomous underwater vehicles. It expresses the Sea-thru image formation model as small neural network modules, one estimating the haze and one the attenuation, and fits them to the incoming images by gradient descent, so that the model adapts continuously as the water and lighting change. No reference images are needed, because the losses are computed on the corrected image itself. The original work uses a distance map provided by the vehicle's sensors. In this project, DeepSeeColor fits 25 parameters with the Adam optimiser on a copy of each frame reduced to 480 px, using 500 iterations on the first frame and 5 on each subsequent frame, starting from the previous values, and drives the average of each channel towards mid-grey, with the gain limited to three times. It was reimplemented in the project code because the original implementation is distributed under the AGPL-3.0 licence.

OceanLens. OceanLens [12] extends DeepSeeColor. It keeps the same physically based structure, with network modules for the haze and the attenuation, but makes these modules deeper and changes the losses. A more robust loss is used for the haze, and additional edge losses are intended to preserve and sharpen object boundaries in the corrected image. Like DeepSeeColor, it is trained on images without reference data. In this project, OceanLens uses the DeepSeeColor physics networks with an additional layer each, and Huber loss for the haze. Since its published checkpoints belong to other scenes, it was fitted on 24 frames of each clip. Its edge losses do not train any parameter in the original code and were left out, and its networks saturate when the range is expressed in meters, so that in practice the range has no effect on its output.

Natural. Developed in this project based on the evaluation of the methods above, the Natural

method addresses the specific causes of their failure in seagrass footage. It subtracts only the true haze term, follows the haze of each frame over time, derives a single brightness gain common to all channels from a smooth fit of the signal with range, gives every range the colour balance seen nearest the camera, balances the brightest surfaces (sand and pale rock) to light beige, and reduces the correction in open water and at altitudes above 6 m. Its development is described in Section 2.4.3.

Motion. This experimental method estimates the water properties from the same seabed patch observed at two distances. As the AUV moves forward, patches tracked with optical flow pass from the top of the frame (far) to the bottom (near), and the attenuation and haze are fitted so that each patch has the same true colour at both distances. The remaining processing follows the Natural method. The method is described and evaluated in Section 2.4.5.

3.3.3. Pretrained networks

Eight of the methods used in the project are deep neural networks pretrained by their authors on large collections of underwater photographs. Unlike physically based methods, they do not model water or require distances to each pixel. They apply the correction learned from their training images directly to each frame. They were used as published by their authors, without retraining on the survey footage, so their results depend on how closely that footage resembles their training data. All of them run on a reduced copy of each frame (720 px for WaterNet, 512 px for both versions of FUnIE-GAN and for UIR-PolyKernel, 288 × 512 for SCNet, USUIR and UDnet, and 256 × 256 for U-shape Transformer). The resulting colour change is then applied to the full-resolution frame and smoothed over time.

WaterNet [13] was introduced together with the Underwater Image Enhancement Benchmark (UIEB), a set of 950 real underwater images, 890 of which have a reference image chosen by volunteers as the best among the outputs of 12 existing enhancement methods. It is a gated fusion network that receives the frame along with white-balanced, contrast-equalised, and gamma-corrected copies, and learns, per pixel, how to blend them. On the GPU, it runs in bfloat16 precision, since the network overflows in float16.

FUnIE-GAN [14] is a lightweight conditional generative adversarial network (GAN) designed for real-time use on underwater robots. Its generator is a small encoder–decoder network, and it was trained on the EUVP dataset, which contains paired and unpaired collections of poor- and good-quality underwater images. The survey footage reproduces the warm colours of its training photographs, producing sepia seagrass on GX010287 and grey-teal tones on GX010175 with visible grid artefacts, and was not usable for colour restoration.

FUnIE-GAN-UP, the unpaired version of FUnIE-GAN [14], learns from two separate, unmatched collections of poor- and good-quality images instead of matched pairs. It is trained in a CycleGAN fashion, in which one network converts poor images into good ones, a second converts them back, and the round trip must return the original image. It was used in place of UGAN [15], which uses a CycleGAN to create artificially degraded copies of clear underwater images and then trains a GAN on the resulting pairs, because the UGAN authors did not publish weights. Since the weights of FUnIE-GAN-UP exist only for Keras, the Keras graph was re-implemented in PyTorch. It produces the most even colour of all methods and is very stable over time, but with a sepia or dusty-pink appearance and low contrast.

UIR-PolyKernel [16] is a compact convolutional network built around very large convolution kernels, used in several forms, that allow each layer to take a wide region of the image into account while keeping the network small. On the survey footage, it gives a clean, artefact-free output, but turns the seagrass mauve-grey on GX010287. Its network definition was adapted to

run without external dependencies.

SCNet, **USUIR**, **UDnet**, and U-shape Transformer were selected from a publicly available list of underwater image enhancement methods. As SCNet and USUIR are published without a licence, their code was not copied into the project. Instead, the repositories and weights are downloaded at their tested versions on first use. A depth-aware transformer (DAUT) was also considered, but its published weight folder is empty.

SCNet [17] was designed to work across different types of water. The colour of underwater images varies widely with the water, and a network trained on one type often fails on another. SCNet therefore normalises its internal features across both image locations and colour channels, so that they depend less on the water colour and more on the scene itself.

USUIR [18] is unsupervised, so it needs no reference images for training. It combines a network with the underwater image formation model. Each image is split into estimates of the clean scene, the transmission, and the backscatter; the clean estimate is degraded again, and the result must be consistent with the original. This shared origin of the two images is the 'homology' referred to in the paper's title.

UDnet [19] is also unsupervised. It is based on a conditional variational autoencoder, which can produce several plausible enhanced versions of each image rather than a single one, reflecting the uncertainty in the correction. Training is guided by conventional contrast and colour stretching of the input image, so no reference images are needed.

U-shape Transformer [20] combines a U-shaped encoder–decoder network with transformer modules that can relate distant parts of the image. One module fuses features at several scales, and the other directs the network towards the colour channels and image regions that are most degraded. Its authors also released LSUI, a dataset of more than 4,000 paired underwater images, on which it was trained.

3.3.4. Classical methods

The two classical methods, the underwater dark channel prior (UDCP) and blue-green channel dehazing with red channel correction (GB dehazing), are image-processing techniques that require neither training data nor distance information for each pixel. Both rely on the simple haze model used for dehazing outdoor photographs, in which each pixel is a mix of the scene and a uniform background light, weighted by a single transmission value. The transmission is estimated from the image itself, using a statistical assumption about what clear images look like, and the model is then inverted to recover the scene. Unlike the Sea-thru model used by the physically based methods, this model does not separate the colour loss of the light reflected by the seabed from the build-up of haze. Both methods operate on single frames. For video, only the background light and, for GB dehazing, the red gain are smoothed over time.

UDCP [21] adapts the dark channel prior, originally proposed for dehazing outdoor photographs. The prior states that in most small patches of a clear image, at least one colour channel is very dark due to shadows, dark objects, or strongly coloured surfaces. Haze brightens this dark channel, so its brightness indicates how much haze is present. Underwater, red light is absorbed so strongly that the red channel is dark almost everywhere, making every patch appear haze-free. UDCP therefore computes the dark channel from the green and blue channels only. The background light is taken from the brightest pixels of this dark channel, the transmission is derived from it patch by patch, and the haze model is inverted to recover the scene.

GB dehazing [22] treats the colour channels separately. The blue and green channels, which are degraded mainly by haze, are dehazed with a modified form of the dark channel prior. The

red channel, which is degraded mainly by absorption, is corrected instead with the grey-world assumption, which states that the average colour of a scene is neutral grey. The red channel is therefore scaled to balance the other two. In the original method, an adaptive exposure map finally corrects regions that have become too dark or too bright. This step was not implemented in the project, where the red gain is instead limited to four times.

On the survey footage, bright sand, which has no dark pixels in any channel, violates the dark channel assumption and is mistaken for dense haze, so both methods saturate the sand (to yellow or neon green) and change appearance abruptly between consecutive frames.

The implemented methods are summarised in Table 8.

Table 8. Underwater enhancement methods implemented and compared.

Method	Type	Uses range	Fitting
Natural (recommended)	Sea-thru haze model tuned for natural colours, developed in this project	Yes	Once per video, with the haze followed per frame
Motion (experimental)	As Natural, with the water calibrated from seabed patches tracked as the AUV moves	Yes	Once per video, from tracked patch pairs
Physics	Simple physical model, one attenuation coefficient per channel	Yes	Once per video
Sea-thru	Range-dependent attenuation and additional haze term [9]	Yes	Once per video
DeepSeeColor	Sea-thru model fitted by gradient descent [11]	Yes	Continuously, frame by frame
OceanLens	DeepSeeColor-style physics networks with additional layers [12]	Yes (no effect in practice)	Once, on 24 frames of the clip
Sea-thru + WaterNet, Physics + WaterNet	Physical correction followed by WaterNet	Yes	Both
WaterNet	Pretrained fusion network [13]	No	Pretrained
FUnIE-GAN	Pretrained generative adversarial network [14]	No	Pretrained
FUnIE-GAN-UP	Pretrained unpaired (CycleGAN-style) FUnIE-GAN, used in place of UGAN [15]	No	Pretrained
UIR-PolyKernel	Pretrained polymorphic large-kernel CNN [16]	No	Pretrained
SCNet	Pretrained network, desensitised to water type [17]	No	Pretrained
USUIR, UDnet	Pretrained unsupervised networks [18], [19]	No	Pretrained
U-shape Transformer	Pretrained transformer network [20]	No	Pretrained
UDCP, GB dehazing	Classical dark channel prior [21] and blue-green dehazing with red correction [22]	No	Per frame

3.3.5. Processing options

The enhancement can be restricted to part of a video by a start and end time, either keeping only that part (with the telemetry copy shifted accordingly) or copying the rest unchanged, and the water parameters can be fitted on the selected part or on the whole video. A preview mode writes only a four-frame before and after comparison, the fit chart and the fitted values, which allows settings to be checked within a few minutes. Three further options were added to improve clarity and colour. Multi-frame denoising aligns the neighbouring frames (for example five) to each frame using optical flow and takes the per-pixel median before the colour correction. Noise and suspended particles largely disappear, as do fish and drifting debris, at a

processing speed of about 1.5 frames per second. Local contrast enhancement applies Contrast Limited Adaptive Histogram Equalization (CLAHE) [23] to brightness only, so that seagrass leaves and sand edges stand out without altering colours. Finally, for the Natural method, sand and pale rock are mapped to a configurable CIE Lab lightness (default 80) instead of full white. Previously, the sand appeared almost white (L between 89 and 91), whereas it now lies between L 78 and 81. On the RTX 4060 the Natural method with Depth Anything processes about 2 to 2.5 frames per second, and WaterNet about 5 frames per second, including the reading of 4K input and the writing of 1920 px output. The name of each output video records the main settings (method, range source, denoising and local contrast).

Further options were added from mid-September 2026, mainly to address problems observed on the full dives. A fast particle filter removes the thin, near-vertical streaks produced by suspended particles lit by the vehicle lamps, which the colour correction had turned into red dots on GX010175. It costs about 15 ms per frame, roughly eight times less than multi-frame denoising, and is now enabled by default. Multi-frame denoising gained a particle mode, in which only pixels clearly brighter than in the neighbouring frames are replaced, so that the texture of the seagrass keeps its detail. For the Natural and Motion methods, a haze multiplier scales the haze removed after the clipping limit has been applied (values above 1 give a clearer image but turn dark seagrass grey-purple), and a shadow balance removes less haze in the channels that the shadows lack, until the darkest 20% of the pixels come out dark olive (CIE Lab $a \approx -4$, $b \approx 8$), so that dense meadow does not turn mauve. A deflicker option brings the mean colour of each frame to its running average over about 0.7 s, against the red pulses caused by bands of sunlight in shallow water (GX010287 at 3:30). Finally, a per-scene colour balance can either set the palest surfaces of each scene to light beige (Natural and Motion methods) or apply the Deep White Balance network of Afifi and Brown [24] to every 15th frame, with any method. The deflicker and per-scene balance options are disabled by default. The effect of the particle filter on the turbid dive is illustrated in Figure 35.

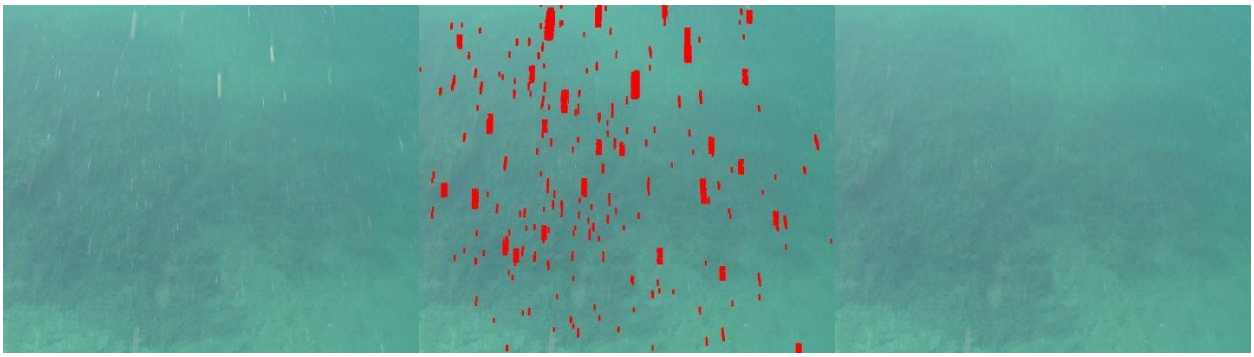


***Figure 35**. Particle removal on the turbid-water dive GX010175, showing the original frame with suspended particles (left), detected particles marked in red (center) and cleaned frame (right).*

3.4. Evaluation of the enhancement methods

The comparison showed that the Natural method gives the most natural colours of all methods tested, with olive-green seagrass, light beige sand, and colour that remains as stable from frame to frame as in the original footage. Each of the other methods failed at least one of the checks described below.

3.4.1. Evaluation metrics

Each method was applied to 8 frames from the clip between 1:21 and 1:43 of GX010287 and to

8 frames spread over the whole dive (from 0:30 to 7:50), using the Depth Anything range. Colours were measured in the CIE Lab colour space, where negative *a* values indicate green and positive *b* values indicate yellow. Four indicators were computed. The colour change from top to bottom is the spread of the mean colour over six horizontal bands of the frame. High values indicate coloured upper parts of the frame (red, purple or yellow casts at long range). The frame-to-frame jump is the spread of the mean colour across the eight frames. High values indicate unstable colours. The seagrass colour is the mean *a*/*b* of the darker pixels (between the 5th and 40th brightness percentiles), and the sand colour is the mean *a*/*b* of the brightest 8% of pixels. The standard deviation of lightness was used as an indicator of contrast.

3.4.2. Results

Table 9 compares the original footage with four representative methods, three physically based (Natural, Sea-thru and DeepSeeColor) and one pretrained network (WaterNet), on the clip from 1:21 to 1:43 of GX010287. The comparison of all methods follows in Section 2.4.6. Natural is the only method that turns the seagrass olive green and the sand light beige. Its colour change from top to bottom (4.4) and its frame-to-frame jump (4.2) remain close to those of the original footage (3.7 and 3.3), while the contrast more than doubles (from 6.5 to 14.9). Sea-thru removes the colour from the seagrass, which turns grey, and is the least stable from frame to frame (15.3). DeepSeeColor gives the highest contrast (24.6), but also the largest colour change from top to bottom (13.5) and a green-yellow sand. WaterNet is the most stable method (2.5) but leaves the seagrass bluish.

***Table 9**. Colour indicators for the clip 1:21 to 1:43 of GX010287 (CIE Lab).*

Method	Colour change top to bottom	Frame-to-frame jump	Seagrass a / b	Sand a / b	Contrast (L std)
Original	3.7	3.3	−19 / 12 (green haze)	−23 / 19	6.5
Natural	**4.4**	**4.2**	**−14 / 9 (olive green)**	**0 / 6 (light beige)**	**14.9**
Sea-thru	9.2	15.3	0 / −1 (grey)	−1 / 4	19.3
DeepSeeColor	13.5	3.6	−8 / 1	−19 / 27 (green-yellow)	24.6
WaterNet	10.9	2.5	−4 / −4 (bluish)	−2 / 7	17.6

Over the whole dive the Natural method again showed the lowest colour change from top to bottom (4.8, against 7.7 for Sea-thru and more than 15 for the other methods). Its frame-to-frame jump over the whole dive (10.0) reflects the real changes of scene between open water, sand and meadow rather than instability of the correction. DeepSeeColor broke down on frames showing only haze, producing neon green and red colours (frame-to-frame jump of 33.5).

3.4.3. Development of the Natural method

The Natural method was built by identifying why the established methods failed on the project footage and correcting each cause in turn.

Grey or purple seagrass. The Sea-thru haze fit includes a residual term that represents scene colour remaining in the darkest pixels. In a meadow this term corresponds to the green of shaded seagrass (about 0.08 of the green channel at 5 m, nearly half of what was removed as haze), so removing it strips the seagrass of its colour. The Natural method subtracts only the true haze term, lowered until at most 0.5% of the pixels would fall below zero.

Colour fluctuating between purple, yellow and blue. A single fixed haze estimate for a whole

clip leaves small errors in each frame, which the subsequent contrast stretch amplifies about four times. The Natural method follows the haze strength and colour of each frame relative to its typical value, smoothed over about 0.3 s.

Coloured upper parts of the frame. The attenuation of each channel, and especially of blue, is poorly constrained by the data. Instead, the Natural method fits a smooth curve (a smoothing spline over the logarithm of the range, within about 1.5%) through the average signal of each channel at each range. A single brightness gain common to all channels is derived from this curve (capped at a factor of three), and every range is given the colour balance observed nearest to the camera.

Colour casts from stretching each channel separately. The brightest surfaces (the 2nd to 8th brightest percent of pixels, typically sand and pale rock) are balanced to light beige, and a single contrast stretch is then applied to all channels.

Artificial colours in open water. Frames whose contrast is less than 45% of the usual contrast of the dive are corrected less strongly, and those below 20% are not corrected at all, so that open water remains blue-green. Earlier thresholds of 80% and 45% also weakened the correction of hazy meadow seen from higher above the seabed (GX010287, 2:48 to 2:55). In addition, the correction now fades with altitude. It is applied in full up to 6 m above the seabed and not at all from 10 m, because frames recorded from higher up show mostly water and had turned magenta (GX010287 at 0:40, at 8 to 9 m).

Two further modifications were tested and rejected. Capping the far-range colour correction at 12 or 20 m made the upper parts of the frame worse (the colour difference between the top band and the rest of the frame was 27 and 17 respectively, against 8.8 without a cap), and leaving 10 to 20% of the haze in place did not make the seagrass appear greener.

3.4.4. Performance in turbid water

The dive GX010175 contains about 3.5 times less red signal than GX010287 (the 1st to 99th percentile range of the linear red channel is 0.066, against 0.23), so its true colours cannot be fully recovered by any method. Two additional corrections made the Natural method usable under these conditions. First, the haze is measured in eight vertical strips per frame, and the left-to-right colour gradient of the illumination, estimated from all sample frames, is evened out column by column, which reduced the left-to-right colour difference from 27.0 to 3.5. Second, when the sand calls for a very strong white balance correction (a factor of 5.8 between channels on GX010175), noise and particles turn pink, so the balance is capped at a factor of 3.5, leaving a slight green tint. The clearest result on GX010175 was obtained by combining the Natural method with five-frame denoising and local contrast enhancement.

3.4.5. Calibration from motion

The experimental Motion method estimates the water properties from the same surface observed at two different distances. The idea follows Vlachos et al. [25], who used a colour chart photographed from many distances in combination with depth maps. Here, the seabed itself acts as the chart, since as the AUV moves forward a patch seen at the top of the frame (far) reaches the bottom of the frame (near) a few seconds later. The procedure has four stages. First, the range of each image row is computed from the camera tilt (64°), the vertical field of view (75.3°, Table 3) and the DVL altitude, which together reproduce the motion of the seabed observed in the optical flow. Depth Anything only adds relief within a row (±30%), since its far range proved about three times too long (30 m instead of about 10 m). Second, seabed patches are tracked with pyramidal Lucas-Kanade optical flow [26], verified forwards and backwards,

for up to 8 s in 12 sections of the clip, giving 5,281 patch pairs on GX010287 and 1,033 on GX010175. Third, the attenuation ß and the haze parameters B_∞ and $ß_B$ are fitted per channel so that the true colour of each patch comes out equal at both distances, which removes the scene from the fit. The haze may not exceed the darkest pixels, and pairs deviating by more than three robust standard deviations (1 to 2%) are discarded. Finally, the remaining processing (sand white balance, per-frame haze, frame-to-frame correction and a final evening-out of any residual colour change with range) is carried out as in the Natural method.

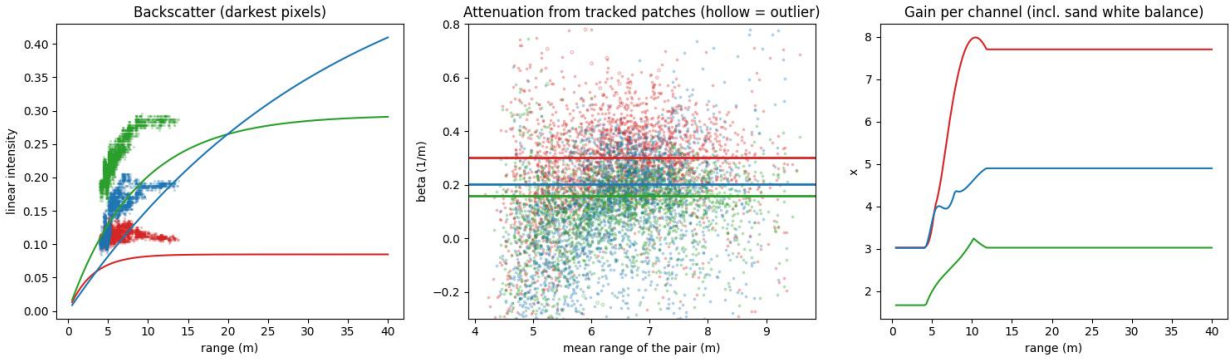


***Figure 36**. Calibration from motion on GX010287, showing the haze curves, attenuation implied by each tracked patch pair with the fitted values, and gain per colour channel.*

***Table 10**. Comparison of the Motion and Natural methods on 10-second clips.*

10 s clip	Colour change top to bottom	Frame-to-frame jump	Seagrass a / b	Fitted ß R / G / B (1/m)
GX010287, Motion	**2.7**	6.3	0 / 3 (olive-grey)	0.30 / 0.16 / 0.20
GX010287, Natural + depth	4.1	4.5	−12 / 11 (green)	n/a
GX010175, Motion	5.5	4.6	+10 / 7 (pink)	0.04 / 0.05 / 0.00
GX010175, Natural + depth	3.6	3.1	−20 / 1	n/a

On the clear dive, the Motion method gives the most even colour from near to far of all methods (Table 10), with physically plausible attenuation values in which red is attenuated fastest. On the turbid dive, however, the tracked patches only come a few meters closer to the camera (from about 6 m to 4 m), so the attenuation cannot be determined reliably and red is over-amplified, producing a pink cast. The calibration adds about five minutes per run (452 s in total for the 10 s clip of GX010287).

3.4.6. Comparison of all methods and processing cost

Finally, every method was run on a 10-second clip of each dive (GX010287 from 1:30 to 1:40 and GX010175 from 1:21 to 1:31) with the provisional geometry of that time (tilt 64°, vertical field of view 75.3°), producing 27 outputs per dive. Figure 37 shows the same frame of GX010287 as rendered by the original footage and all 27 outputs. The Natural method with the Depth Anything range is the only method that keeps the seagrass green and the sand beige on both dives. The Motion method gives the most even colour from near to far on the clear dive, and SCNet, combined with local contrast enhancement, is the best of the pretrained networks.

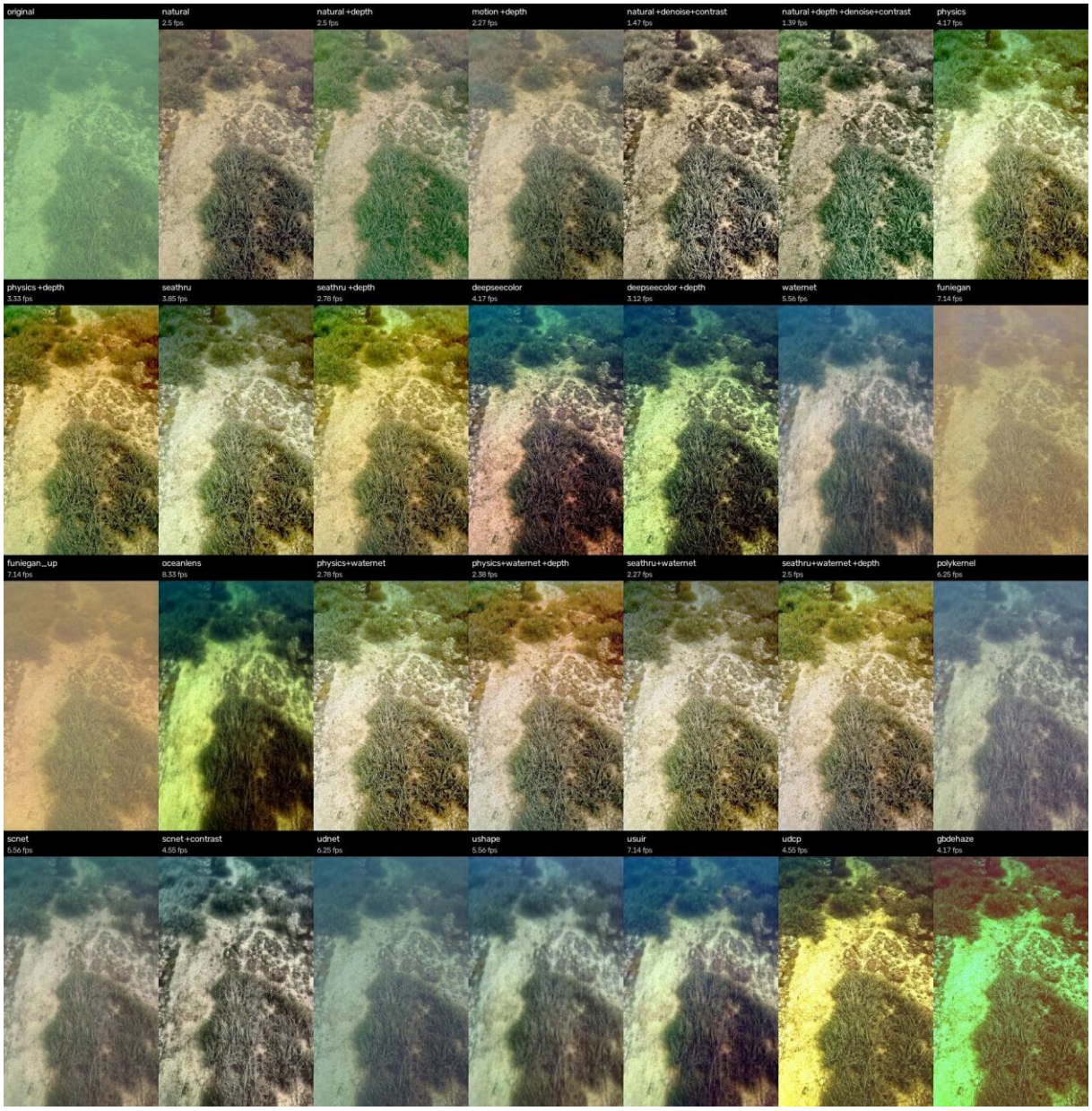


***Figure 37**. The same frame of the clear-water dive GX010287 as rendered by the original footage (top left) and all 27 enhancement outputs, with the processing speed of each method on the RTX 4060 in frames per second.*

The processing time of each configuration is given in Table 11 and shown in Figure 38. The total time includes fitting the water parameters once per run, whereas the video-only time grows with the length of the clip.

***Table 11**. Processing time for 10 s of video (300 frames) on the RTX 4060, sorted by speed.*

Output	GX010287 total / video only	GX010175 total / video only	fps (GX010287 / GX010175)
OceanLens	118 s / 36 s	110 s / 42 s	8.3 / 7.1
FUnIE-GAN	49 s / 42 s	48 s / 42 s	7.1 / 7.1

Output	GX010287 total / video only	GX010175 total / video only	fps (GX010287 / GX010175)
FUnIE-GAN-UP	53 s / 42 s	52 s / 42 s	7.1 / 7.1
USUIR	53 s / 42 s	49 s / 42 s	7.1 / 7.1
UIR-PolyKernel	63 s / 48 s	53 s / 42 s	6.3 / 7.1
UDnet	62 s / 48 s	53 s / 42 s	6.3 / 7.1
WaterNet	64 s / 54 s	61 s / 48 s	5.6 / 6.3
SCNet	68 s / 54 s	53 s / 42 s	5.6 / 7.1
U-shape Transformer	69 s / 54 s	63 s / 54 s	5.6 / 5.6
UDCP	75 s / 66 s	67 s / 60 s	4.6 / 5.0
SCNet + contrast	86 s / 66 s	72 s / 60 s	4.5 / 5.0
GB dehazing	76 s / 72 s	71 s / 66 s	4.2 / 4.6
DeepSeeColor	97 s / 72 s	81 s / 60 s	4.2 / 5.0
Physics	168 s / 72 s	152 s / 66 s	4.2 / 4.6
Sea-thru	173 s / 78 s	167 s / 66 s	3.9 / 4.6
Physics + depth	215 s / 90 s	200 s / 90 s	3.3 / 3.3
DeepSeeColor + depth	138 s / 96 s	110 s / 72 s	3.1 / 4.2
Sea-thru + depth	293 s / 108 s	212 s / 102 s	2.8 / 2.9
Physics + WaterNet	199 s / 108 s	198 s / 114 s	2.8 / 2.6
Natural	221 s / 120 s	200 s / 102 s	2.5 / 2.9
Natural + depth	**265 s / 120 s**	**318 s / 150 s**	**2.5 / 2.0**
Sea-thru + WaterNet + depth	295 s / 120 s	243 s / 126 s	2.5 / 2.4
Motion + depth	452 s / 132 s	413 s / 126 s	2.3 / 2.4
Physics + WaterNet + depth	251 s / 126 s	312 s / 180 s	2.4 / 1.7
Sea-thru + WaterNet	263 s / 132 s	206 s / 114 s	2.3 / 2.6
Natural + denoise + contrast	340 s / 204 s	288 s / 192 s	1.5 / 1.6
Natural + depth + denoise + contrast	**358 s / 216 s**	**330 s / 198 s**	**1.4 / 1.5**

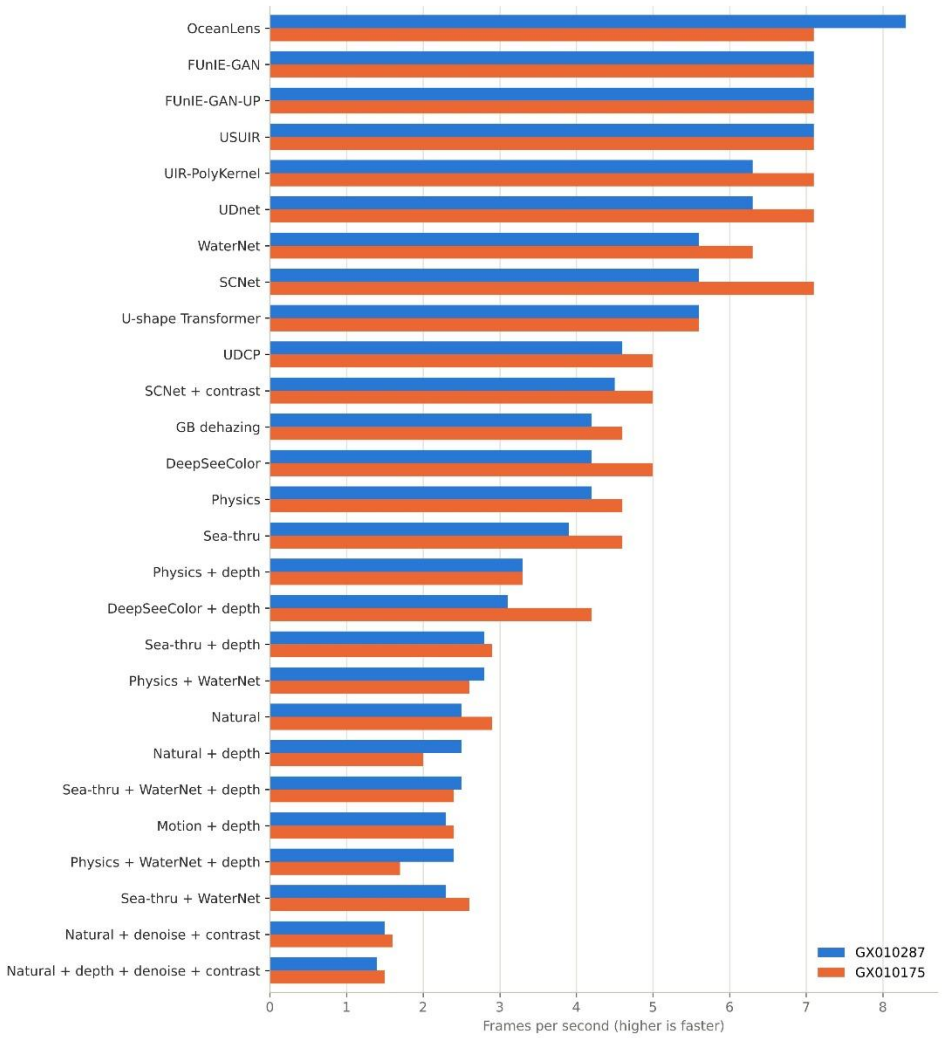


***Figure 38**. Processing speed of each configuration on the RTX 4060, in frames per second, for GX010287 (blue) and GX010175 (orange).*

On this basis, the Natural method with the Depth Anything range is recommended when natural colours are the priority, at about 2.5 frames per second. When an .srt telemetry file is available, a faster compromise is to run the Natural method with the telemetry range instead of Depth Anything, which shortens the processing of the 10 s clip from 265 to 221 s on GX010287 and from 318 to 200 s on GX010175. The cost is the flat-seabed assumption, under which the seagrass canopy is slightly over-corrected (Table 7). Adding multi-frame denoising and local contrast enhancement gives the clearest result at about 1.5 frames per second, corresponding to roughly 20 s of processing per second of video, or 2.5 to 3 hours for an 8-minute dive.

3.5. Training data and segmentation models (Steps 1 to 3)

In total, thirteen versions of the model were trained. Versions v1 to v10 were all trained on the same 61 frames from GX010287. More specifically, v1 to v7 were trained on the original 4K frames, and v8 to v10 on frames extracted from the Natural-enhanced video, which were re-annotated from scratch. Versions v11 to v13 also included annotated frames from the turbid dive GX010175, while v12 and v13 introduced a third class, debris. The highest mask mAP50

remains 0.730, achieved by v4 (YOLO11s-seg trained on the seagrass and rock classes using the original frames). On enhanced frames, the best result is 0.703 (v8), and with both dives combined it is 0.682 (v11).

3.5.1. Frame extraction and annotation

Step 1 extracts one still frame at a fixed interval from the video. With an interval of 8 s, the 8.1-minute test dive yielded 61 frames. A shorter interval gives more frames to annotate and, in general, a better model. In Step 2 the frames were annotated in labelme [27] with the classes seagrass, fish, rock and debris, each annotated frame receiving a JSON label file. Polygons were drawn along the edge of the seagrass, which suits a segmentation model. Bounding boxes are also accepted and are converted to four-point polygons. Each separate patch was outlined individually, while a continuous meadow was outlined as one area, and hazy or distant seagrass was annotated as well. Frames containing no seagrass were left without labels, so that they teach the model what is not seagrass. For the enhanced dataset, the 61 frames were extracted from the Natural-enhanced video at 1080 × 1920 px, and 50 of them were annotated, while the original 4K frames and their labels were archived.

Frames of the turbid dive GX010175 were added in a second round, taken from its enhanced video (Natural method, telemetry range and final camera geometry) at 1080 × 1920 px. A first set of 59 frames spread over the dive was extracted into its own folder, and 27 of them were annotated, with 45 seagrass and 5 debris shapes. Because debris is rare in the footage, Step 1 was extended so that frames can be taken only from selected time ranges, at a shorter interval and into a separately named folder. Extracting one frame per second from eight short stretches in which debris is visible yielded 28 further frames, 21 of which were annotated with 25 debris shapes.

3.5.2. Model training

Step 3 rebuilds the training dataset from all annotated frame folders, converts the labelme polygons to the YOLO segmentation format and splits the frames into training and validation subsets in an 80/20 ratio with a fixed random seed. Unlabelled frames inside an annotated folder are included as negative examples, while folders without any labels are skipped. Training is then run on the GPU for up to 150 epochs, with early stopping after 40 epochs without improvement, at an input size of 640 px and a batch size of 8, which takes about 10 minutes for 61 frames. Every run is stored in a new versioned folder containing the trained weights, the training charts and a summary file with the date, a note, the data, the classes, the settings and the scores, so that earlier versions are never overwritten. Models of the YOLO11 and YOLO26 families [28] were tested, together with the transformer-based detector RT-DETR [29], which predicts boxes only (Table 12).

***Table 12**. Trained model versions and validation scores.*

Version	Frames	Base model	Classes	Shapes	Mask mAP50	Box mAP50
v1	Original	YOLO11n-seg	Seagrass	119	0.495	0.595
v2	Original	YOLO11n-seg	Seagrass, fish, rock	310	0.705	0.725
v3	Original	YOLO11n-seg	Seagrass, rock	199	0.699	0.715
v4	**Original**	**YOLO11s-seg**	**Seagrass, rock**	**199**	**0.730**	**0.763**
v5	Original	RT-DETR-L (boxes only)	Seagrass, rock	199	n/a	0.757
v6	Original	YOLO26s-seg	Seagrass, rock	199	0.717	0.704

Version	Frames	Base model	Classes	Shapes	Mask mAP50	Box mAP50
v7	Original	YOLO11s-seg	Seagrass	125	0.443	0.520
v8	Enhanced	YOLO11s-seg	Seagrass, rock	235	0.703	0.707
v9	Enhanced	YOLO11s-seg	Seagrass, rock	235	0.703	0.707
v10	Enhanced, labels revised	YOLO11s-seg	Seagrass, rock	239	0.602	0.659
v11	Enhanced, both dives	YOLO11s-seg	Seagrass, rock	284	0.682	0.653
v12	Enhanced, both dives	YOLO11s-seg	Seagrass, rock, debris	289	0.610	0.617
v13	Enhanced, both dives + debris frames	YOLO11s-seg	Seagrass, rock, debris	314	0.674	0.700

Versions v1 to v10 used 49 training and 12 validation frames, v11 and v12 used 96 and 24, and v13 used 119 and 29 (seed 42 throughout). On the same frames and with the same two classes, 235 shapes were annotated on the enhanced frames (v8), compared with 199 on the original frames (v4), an increase of about 18%. This suggests that the enhancement made the scene easier to interpret. With the colour cast and haze removed, seagrass patches and rocks that blended into the background in the original frames, especially in the farther, upper part of the image, could be distinguished and delineated. The greater number of shapes thus indicates a more complete annotation and, therefore, more training information, although part of the increase may also stem from the annotators' growing familiarity with the footage.

Version v9 repeats v8 with identical scores, while v10 follows a revision of the labels of about 20 frames and scores lower on its own revised labels. Scores obtained across several classes are averages over the classes and are therefore not directly comparable to the seagrass-only versions (v1 and v7). Scores measured on different label sets are not directly comparable either, so the comparison of v8 to v10 with v4 says little about whether enhancement helps detection. Moreover, with only 12 validation frames, the scores vary considerably between versions, and differences of a few hundredths should not be over-interpreted. Compared with the side-scan sonar models described in Section 1, the video models achieved substantially higher scores on a similarly small training set, reflecting the much clearer visual signature of seagrass meadows in optical imagery.

Versions v11 to v13 extend the training data to the turbid dive. Adding GX010175 (v11) doubled the number of frames to 120, 46 of which contain none of the classes, and gave a mask mAP50 of 0.682, close to the single-dive models, on a validation set twice as large. Adding debris as a third class with only five examples (v12) lowered the average scores (mask mAP50 0.610), since the class average includes a class the model had barely seen. With 25 further debris shapes from frames selected for it (v13), the scores recovered to a mask mAP50 of 0.674 and a box mAP50 of 0.700. Debris is represented by only 30 shapes, so its detection should be regarded as indicative. Because each version has its own validation frames, these scores are not directly comparable with those of v1 to v10.

Of the thirteen model versions listed in Table 12, v1 to v7 were trained on frames from the original footage, whereas from v8 onwards the training frames were taken from the enhanced videos, which were also used for detection. The test below shows why this change was needed, since a model trained on original frames does not carry over to enhanced video. To quantify this, model v2, trained on original frames, was applied to the same 20-time instants of every

enhanced output, and the predicted seagrass cover and its overlap (Intersection over Union, IoU) with the prediction on the original video were computed (Table 13).

***Table 13**. Seagrass detected by model v2 (trained on original frames) on the enhanced outputs.*

Dive	Output	Mean seagrass cover	IoU with the prediction on the original
GX010287	Original	33%	1.00
GX010287	UDnet	32%	0.69
GX010287	WaterNet	29%	0.67
GX010287	Natural	17%	0.42
GX010287	Natural + depth	15%	0.39
GX010287	Sea-thru	11%	0.27
GX010175	Original and all outputs	0 to 12% (no seagrass found on most)	n/a

The stronger the colour correction, the less the predictions of the original-trained model agree with those on the original video. Since this measures agreement rather than accuracy, it shows that model training and detection must use the same kind of video. Determining whether enhancement also improves detection will require v8 to v10 and a model trained on original frames to be scored on the same set of annotated frames.

3.6. Detection, depth and georeferenced observation points (Steps 4, 4a and 5)

Step 4 runs the trained model on every frame of the video and records its detections, and Step 5 converts these records into a georeferenced point every 0.5 s, with a seagrass presence/absence attribute.

3.6.1. Frame-by-frame detection

The detection script plays the video in a window, with detections drawn as coloured areas, along with a progress window, and can be paused or stopped by the operator. The model can be selected as the latest version, a specific version or a weights file, and RT-DETR models are recognised automatically. A minimum confidence of 0.40 is applied; lower values detect more faint seagrass at the cost of more false alarms. Frames are downscaled to 1920 px before detection, which does not affect accuracy since the model operates at 640 px, but speeds up drawing. The live display runs at about 7 frames per second on 4K footage, while the saved annotated video retains every frame.

The step produces the annotated video and a CSV file with one row per frame containing the sea depth (immersion plus altitude) below the camera and at the top edge of the frame and, for each class, the number of detected areas, the highest confidence, the percentage of the frame covered and the percentage of the bottom zone covered. The bottom zone is the lowest quarter of the frame, nearest to the camera, and is marked in the annotated video with a yellow line. Its cover is the most reliable measure of the seabed directly below the vehicle, since it is least affected by haze and perspective. An optional overlay draws lines at round seabed distances ahead of the vehicle, labelled with the width of the view at that distance and the distance to the top edge of the frame, computed from the altitude, a flat seabed and the same tilt and fields of view as the coverage mapping. The opacity of the coloured detections is adjustable (0.3 by default). Temporal smoothing over 1 s is applied in the current configuration. An example of the annotated video produced by model v10 on the Natural-enhanced GX010287 is shown in Figure 39. Three options were implemented to refine or stabilise the outlines (Table

14).

Table 14. Options for refining and stabilising the detections.

Option	Function	Effect	Cost per frame (RTX 4060)
SAM 2 refinement	SAM 2 [30] redraws a precise outline inside each detected box	Finer edges	+115 ms (SAM 2.1 tiny), +185 ms (SAM 2.1 base)
Temporal smoothing	Averages each frame with the surrounding frames (centred, no lag)	At 0.5 s, the mean frame-to-frame jump of bottom cover fell from 5.2 to 1.0 percentage points	Negligible
Tracking	YOLO runs every few frames (15 by default), and the SAM 2 video memory follows the regions in between	Combined with smoothing, the most stable result, with a largest jump of 3.5 points against 17 with smoothing alone	+120 ms (SAM 2.1 tiny)

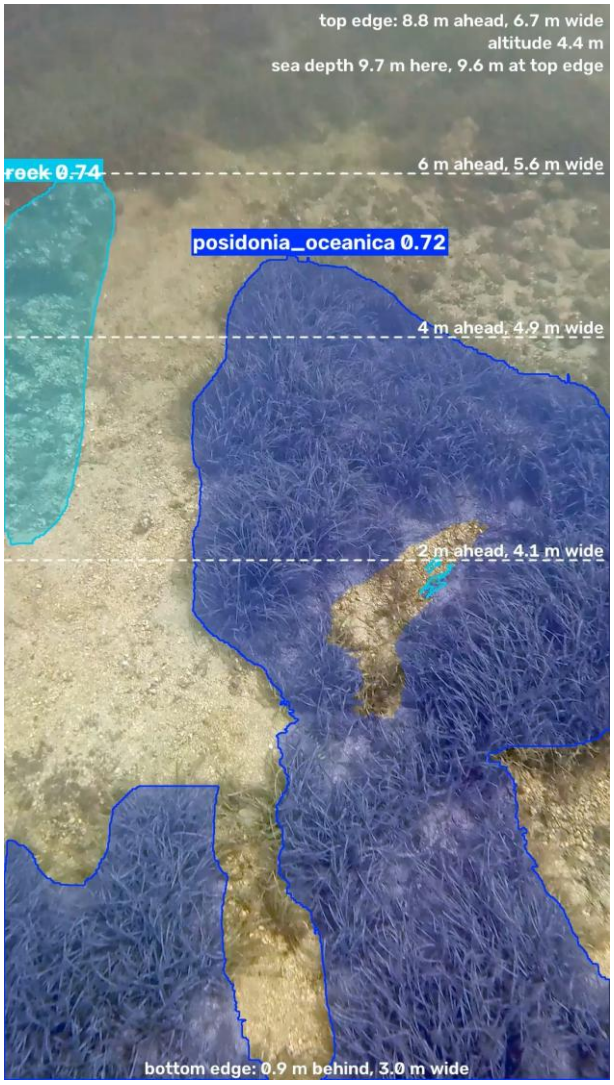


***Figure 39**. Frame of the annotated video produced by model v10 on the Natural-enhanced GX010287 video, showing* detected seagrass (dark blue) *and rock (light blue) with their confidence, distance lines with the width of the view, and the altitude and sea depth from the telemetry log.*

3.6.2. Depth viewer

The optional step 4a plays the video as a depth map computed with Depth Anything V2 [10] and scaled to meters with the DVL altitude. The scale and offset of each frame are fitted so that the map matches, on average, a flat seabed located at the measured altitude below the camera. It runs at about 9 to 12 frames per second with the Small model (about 20 to 25 minutes for GX010287) and saves a depth video, displayed either alone or side by side with the original footage. Two display modes are available. The distance mode shows the straight-line distance from the camera. On GX010287, at 4.1 m altitude, this is about 4.2 m at the bottom edge of the frame and 9 m at the top. The height mode shows the height above or below the flat seabed, up to ±0.75 m, with features standing above the seabed (rock, seagrass mat) and below it (sand channels) shown in contrasting colours. Differences below about 0.3 m should be regarded as noise. The depth viewer is a diagnostic and visualisation tool and is not used in the coverage mapping. An example in the side-by-side view is shown in Figure 40.



***Figure 40**. Depth viewer (step 4a) on the Natural-enhanced GX010287 video, showing the video frame (left) and Depth Anything V2 distance map scaled to meters with the DVL altitude (right).*

3.6.3. Georeferenced observation points

Step 5 joins the most recent detection file of a video with its telemetry log. Each telemetry entry, recorded every 0.5 s, becomes a point containing the video time, the date and time, the position, the camera depth, the altitude, and the light state. Each point also receives the

number of frames falling within its half-second (about 15), the average bottom-zone cover and the highest confidence among those frames. A point is classified as seagrass when its bottom zone is covered by at least a minimum percentage (5% by default). Since this threshold is applied to stored values, it can be changed, and the points regenerated within seconds, without running the detection again. The points are provided as both a CSV table and a GeoPackage in WGS84 (EPSG:4326) for direct use in Geographic Information System (GIS) software such as QGIS, with bottom-zone cover, the highest confidence, and a presence-or-absence attribute for every class (including rock). Only the processed part of the video receives points. On the v10 run on GX010287, 696 of the 966 points (72%) were classified as seagrass. On the v13 run on GX010175, 313 of 1,739 points (18%) were classified as seagrass. Because the log of GX010175 holds about six entries per second (Section 2.1.2), its points lie about 0.17 s apart rather than 0.5 s.

3.7. Seagrass coverage mapping (Step 6)

Step 6 maps the cover of each class, currently seagrass, rock and debris, both as a raster and as polygons. In its latest run on GX010287 (model v10, enhanced video), it identified 1,299 m^2 of seagrass meadows and 456 m^2 of rock within 2,431 m^2 of surveyed seabed (53% and 19% respectively). Full photogrammetric reconstruction by structure-from-motion (for example, with OpenDroneMap) was considered but set aside because moving leaves, haze, the repetitive texture of the meadow, and the oblique view of a single camera frequently cause such reconstructions to fail, and they require a heavy software installation. Instead, Step 6 projects every analysed frame onto the seabed using the camera position recorded in the telemetry log.

3.7.1. Projection method

Detection. A frame is read every 0.5 s (965 frames on the test dive) and the model segments the class to be mapped, using the same video, model and confidence threshold as Step 4 unless specified otherwise.

Camera placement. The position is taken from the telemetry latitude and longitude, converted to meters in the local UTM zone (zone 35N, EPSG:32635) and smoothed over about 1.5 s. The heading is derived from the direction of travel, smoothed over a period of about 3 s, and held constant when the vehicle moves at a speed below 0.15 m/s. The camera height is taken from the altitude, as a running median over 2.5 s.

Projection of the pixels. The segmentation mask is sampled on a grid of 270 × 480 points. The ray from each sample, computed for a pinhole camera tilted by θ below the horizontal, intersects a flat seabed located h meters below the camera, where h is the altitude. For normalised image coordinates *x* (to the right) and *y* (downwards), the components of the ray and the resulting ground offsets are:

$$\text{down} = y\cos\theta + \sin\theta$$

$$\text{forward} = \frac{h\left(\cos\theta - y\sin\theta\right)}{\text{down}}, \qquad \text{right} = \frac{h\,x}{\text{down}}$$

Accumulation on a grid. Pixels farther than 8 m from the camera are discarded (12 m in the first runs), since their resolution and contrast are too low. A grid of 0.25 × 0.25 m cells counts, for each cell, the number of frames that observed it and the share of those frames' samples classified as seagrass, with each frame counting once per cell.

Coverage and polygons. The coverage percentage of a cell is the seagrass share averaged over

all its views. Cells observed in fewer than three frames are left empty. Cells with a coverage of 50% or more are converted to polygons, which are smoothed and retained for areas of 1 m² or more. Averaging over many frames per cell also reduces flicker and isolated single-frame errors in the detector.

3.7.2. Outputs

For each processed video, the coverage mapping step writes four georeferenced outputs (Table 15). Two raster layers on a 0.25 m grid hold the mapped information: the coverage of each class and a mosaic of the frames placed on the map, which is used to verify that the projection is correctly aligned. From the coverage raster, polygons of seagrass meadows and rock are derived and saved as a Shapefile with the area of each polygon, and the same layers are collected in a single GeoPackage together with the surveyed area and the camera track, so that the results can be opened directly in GIS software.

***Table 15**. Outputs of the coverage mapping step.*

Output	Content
Coverage raster per class (GeoTIFF)	Coverage from 0 to 100% per 0.25 m cell, with no data where the cell was not observed often enough
Mosaic raster (GeoTIFF)	The frames placed on the map (coarse RGB), used to check the alignment
Polygons per class (Shapefile)	Seagrass meadow and rock polygons with their area in m²
Coverage GeoPackage	One polygon layer per class, together with a surveyed area layer and a camera track layer

The coverage raster and polygons are produced in OGC-compliant formats (GeoTIFF, Shapefile, and GeoPackage), consistent with the project's data harmonisation approach and suitable for integration into EMODnet Seabed Habitats.

3.7.3. Results

The first coverage map was produced for GX010287 with model v2 on the original video (Table 16). The camera track shows the vehicle's back-and-forth survey lines. Dense meadow lies along the middle and southern lines, while the northern start of the survey is mostly bare.

***Table 16**. First coverage mapping result for GX010287 with model v2 (original video).*

Measure	Value
Frames analysed	965 (one every 0.5 s), 3.9 min on the RTX 4060
Grid	596 × 767 cells of 0.25 m (149 × 192 m)
Surveyed area	5,270 m², along survey lines about 10 to 12 m wide
Seagrass (≥ 50% of views)	2,349 m², i.e. 45% of the surveyed area
Patches	25 (median area 8.3 m², largest 1,259 m²)

This first run assumed that the GPS position refers to the camera (no layback), that the camera looks along the track and that the camera had a provisional geometry (tilt 45°, vertical field of view 90°). The camera geometry has since been determined (tilt 64°, vertical field of view 75.3°, horizontal field of view 46.2°, Section 2.1), which means that this map extends too far ahead of the camera and must be regenerated.

Two later runs were carried out on the Natural-enhanced video with the models trained on enhanced frames, mapping both seagrass and rock (Table 17).

Table 17. Coverage mapping results for GX010287 on the enhanced video.

Measure	v8	v10
Surveyed area	2,368 m²	2,431 m²
Seagrass	About 1,312 m² (55%)	1,299 m² (53%), 36 patches, largest 358 m²
Rock	About 353 m² (15%)	456 m² (19%), 38 patches

Figure 41 shows the mosaic of the enhanced frames projected onto the map along the back-and-forth survey lines to check the projection alignment, and Figure 42 shows the resulting seagrass meadow and rock polygons along the camera track.



Figure 41. Mosaic of the Natural-enhanced frames of GX010287 projected onto the seabed along the survey track (coarse RGB mosaic used to check the alignment).



***Figure 42**. Coverage map of GX010287 from the Natural-enhanced video,* showing seagrass meadow polygons *(green) and rock polygons (grey) along the camera track (dashed line).*

The surveyed area of these runs is less than half that of the v2 run, mainly because the maximum range was reduced from 12 to 8 m. The coverage script does not record the camera geometry it used, and the configuration was changed after the v10 run, so it is not certain whether these maps already use the final geometry of 64°, 75.3° and 46.2°. All areas and percentages reported in this section should therefore be regarded as preliminary until the coverage maps are regenerated with the final geometry.

The turbid dive GX010175 was mapped with model v13 on its enhanced video, after the final camera geometry had been set (Table 18).

***Table 18**. Coverage mapping result for the turbid dive GX010175 with the latest model.*

Measure	Value
Surveyed area	1,880 m²
Seagrass	114 m² (6%), 15 patches, largest 39 m²
Rock	None mapped (no rock annotated on GX010175)
Debris	One polygon of 1 m²
Observation points (Step 5)	313 of 1,739 classified as seagrass (18%)

Only 6% of the surveyed seabed in GX010175 was mapped as seagrass meadows, compared with more than half in GX010287. The low visibility of this dive limits how much seabed is seen clearly in each frame, and the model was trained on only 27 annotated frames from it, so the

low cover may partly reflect missed detections rather than an absence of seagrass. The single debris polygon shows that the class can be mapped, but a meaningful debris map will require more annotated examples.

3.8. Limitations, validation and future work

The main open questions concern the camera geometry and the training data, since both determine how far the resulting maps can be trusted.

Camera geometry. The tilt (64°, from the fit of the seabed motion against the GPS speed) and the fields of view (75.3° and 46.2°, from the GoPro specifications and Snell's law) replace the earlier values of 60° and 72°, but changes in the pitch of the AUV are not corrected, the wide-angle GoPro lens is treated as a pinhole camera, and the seabed is assumed to be flat. Errors in these assumptions shift or stretch the far part of each frame, both in the enhancement (Step 0) and in the coverage mapping (Step 6).

Enhancement. Frames showing only haze contain little colour to recover; the camera lights are not modelled, and some frames retain a slight brownish-pink tint at the far top of the image.

The maps can be validated in three complementary ways. First, the mosaic raster and the polygons can be overlaid on satellite imagery in QGIS, since in clear water at 5 to 10 m depth seagrass meadows usually appear as dark patches. Second, where survey lines overlap, the same seabed is mapped twice from different passes, and agreement between the two passes provides a check of the projection. Third, the Step 5 points can be compared with the coverage raster along the track. In addition, the coverage maps can be cross-checked against the side-scan sonar mosaics described in Section 1, which provide an independent acoustic line of evidence for the extent of the meadows. Figure 43 overlays the video-derived coverage onto the side-scan sonar mosaic from the same survey. The video maps only a narrow strip along the AUV's track, whereas the side-scan sonar covers the full swath on both sides of the track, so that most of the surveyed seabed is observed by the sonar alone. The two techniques are therefore complementary. The video provides verified, species-level observations along the track, which can serve as ground truth for interpreting and machine-learning analysis of the sonar mosaics.

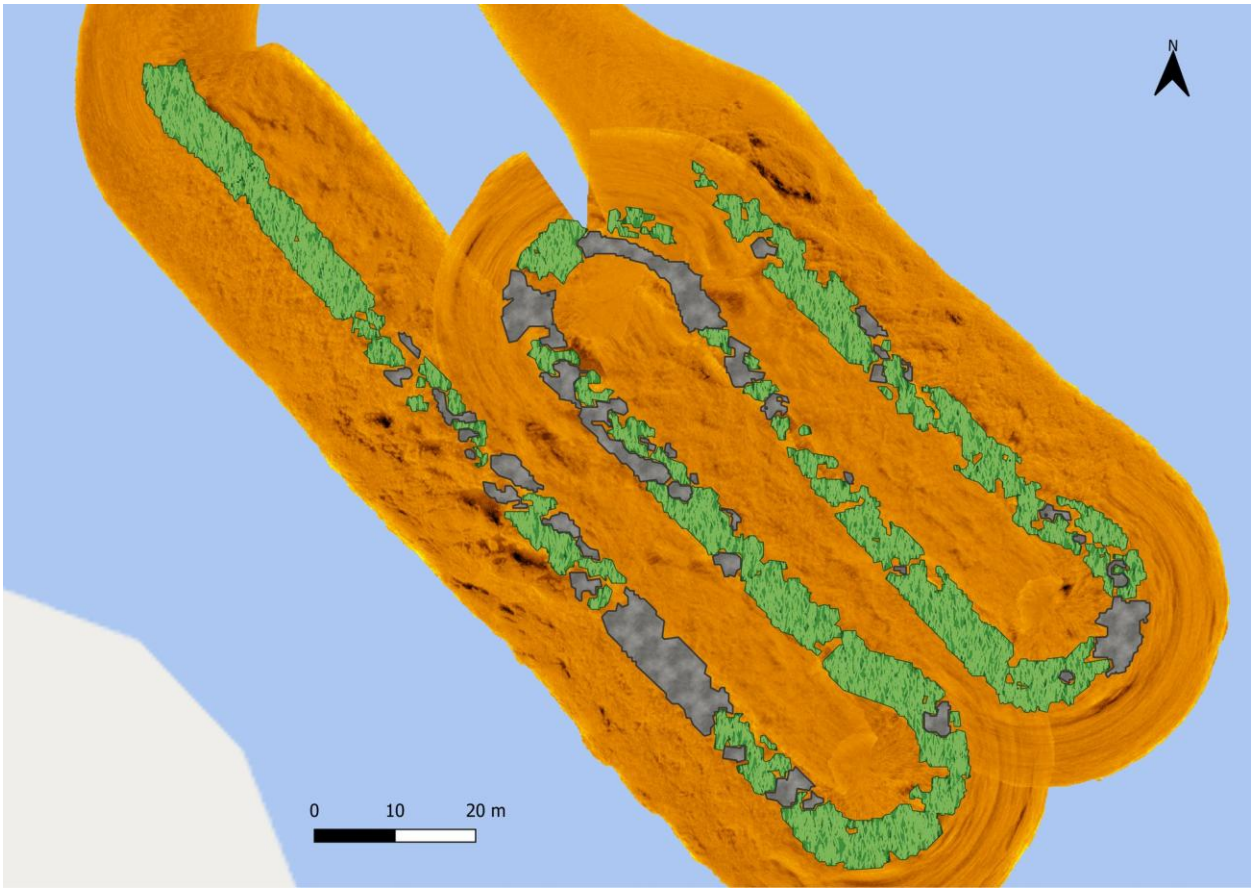


***Figure 43**. Video-derived seagrass and rock coverage of GX010287 overlaid on the side scan sonar mosaic (orange) of the same survey.*

4.Publicly available code

To support the reuse of the methods developed in this activity and the reproducibility of the results, the code is made publicly available. This section presents the available code, first for processing the side-scan sonar data and then for processing the video footage. The repository for the side scan sonar code is in preparation and will be added in a later version of this report.

4.1. Side Scan Sonar

The code for the detection of seagrass in side scan sonar mosaics with the U-Net model described in Section 2.5.2 is available in the public GitHub repository Efxinnos_ML_SSS (https://github.com/nikolaoskokkos/Efxinnos_ML_SSS).

The repository includes the Jupyter notebook seagrass_unet.ipynb, which can be opened and run directly in Google Colab (https://colab.research.google.com/github/nikolaoskokkos/Efxinnos_ML_SSS/blob/main/seagrass_unet.ipynb) or on Binder (https://mybinder.org/v2/gh/nikolaoskokkos/Efxinnos_ML_SSS/main?labpath=seagrass_unet.ipynb). The trained model, a sample mosaic and its reference seagrass polygons are included, so the notebook can be tested immediately in a web browser, without installing Python or the required libraries on a local computer.

The notebook applies the trained U-Net (model v1, a U-Net with an EfficientNet-B2 encoder, implemented with segmentation-models-pytorch, with classes: background and seagrass) to a single-band side-scan sonar mosaic in GeoTIFF format. The mosaic is stretched to 8-bit between its 2nd and 98th percentiles, cut into overlapping tiles of 640 × 640 pixels, and the seagrass probabilities across the overlapping tiles are averaged, thereby avoiding visible seams between tiles. The probability map is then thresholded and converted to seagrass polygons. Training is not part of the notebook, but further models can be added. A new folder "*models/v2/*" with the weights and an "*info.txt*" file is detected automatically and preselected as the latest model. The contents of the repository are listed in Table 19.

***Table 19**. Contents of the Efxinnos_ML_SSS repository.*

File or folder	Content
seagrass_unet.ipynb	The notebook with a settings panel and the results of each step
.code/pipeline.py	The code behind the notebook panels for tiling, prediction, polygon creation, scoring, maps and download
models/v1/	The trained U-Net (model.pt) and its info.txt, which records the EfficientNet-B2 encoder, training mosaic and labels, 640 × 640 px tiles, validation on a held-out band of the mosaic, 150 epochs (best epoch 95), AdamW optimiser with a cosine schedule, loss of 0.5 Dice and 0.5 weighted cross-entropy, validation seagrass IoU 0.495
input/	Sample mosaic SW_20260716_103554_coverup.tif (0.1 m pixels) and its reference seagrass polygons seagrass_20260716_103554.gpkg
styles/	QGIS styles (.qml) for the seagrass and surveyed area layers
requirements.txt, runtime.txt	Python packages (CPU-only PyTorch for Binder) and Python version (3.11)

File or folder	Content
README.md	User guide with a walkthrough of every step, settings with their default values, output files and use of own data

When the notebook is opened in Colab, its first cell copies the repository into the session, installs the packages that are missing from the standard Colab environment (segmentation-models-pytorch, Rasterio and pyogrio) and activates the interactive panels. On Binder or a local computer, this cell does nothing. The second cell reports the Python, PyTorch, segmentation-models-pytorch and Rasterio versions, the CPU and GPU, and the models, mosaics and QGIS styles found (Figure 44). A GPU runtime is optional, since even on Binder without a GPU the sample mosaic is processed in under a minute.

```
In [1]: # Google Colab only: download the repository once, work in its folder, install what Colab lacks,
        # turn on the panels. On Binder (or your own computer) this cell does nothing.
        import os, sys
        if "google.colab" in sys.modules:
            REPO = "/content/Efxinnos_ML_SSS"
            os.chdir("/content")
            if not os.path.isdir(REPO):
                !git clone -q https://github.com/nikolaoskokkos/Efxinnos_ML_SSS.git {REPO}
                !pip install -q segmentation-models-pytorch==0.5.0 rasterio pyogrio
            %cd {REPO}
            from google.colab import output
            output.enable_custom_widget_manager()

In [2]: import os, sys
        sys.path.insert(0, os.path.abspath(".code"))
        from pipeline import *

        check_environment()

        Python 3.13.15, PyTorch 2.6.0+cu124, segmentation-models-pytorch 0.5.0, rasterio 1.4.3
        CPU cores: 16, GPU: NVIDIA GeForce RTX 4060 Laptop GPU
        Models in models/: v1 (latest: v1)
        Mosaics in input/: SW_20260716_103554_coverup.tif
        QGIS styles in styles/: seagrass, surveyed_area
```

***Figure 44**. First two cells of the side scan sonar notebook, which copy the repository and install the packages in Google Colab and check the environment (versions, CPU, GPU, models, mosaics and QGIS styles).*

4.1.1. Workflow of the notebook

The notebook is organised in five steps, numbered 0 to 4, each with a settings panel whose defaults are the recommended values. Each prediction is stored in a folder of its own, results/<date>_<time>_<mosaic>/, together with a settings.json file that records the settings used.

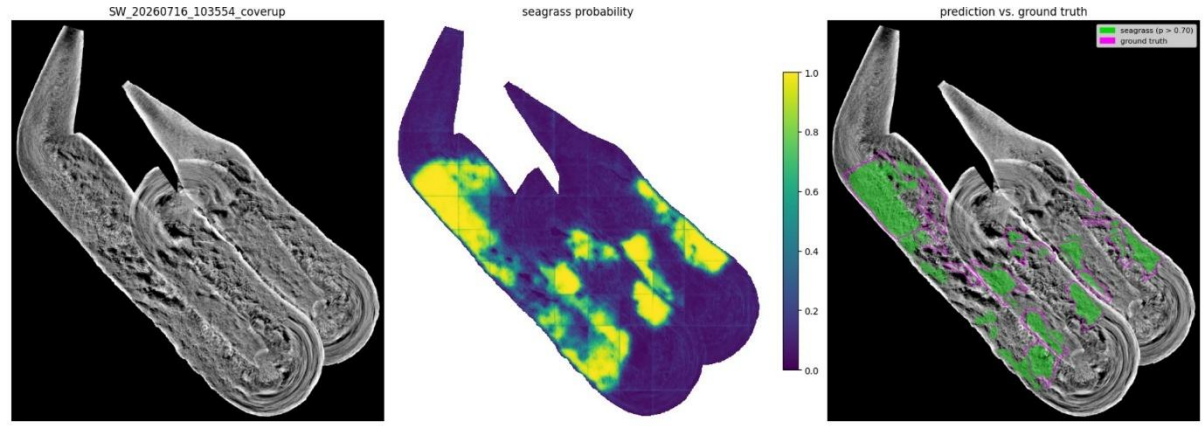
Choice of the mosaic (step 0). The sample mosaic is preselected. Users can add their own mosaic with an upload button for small files or by dragging large files into the input folder. The mosaic must be a single-band GeoTIFF with a coordinate reference system and a nodata value for the empty areas. Since the model was trained on 0.1 m pixels, mosaics with a similar pixel size are expected to give the best results. Seagrass polygons in the input folder (GeoPackage, Shapefile or GeoJSON) that overlap the mosaic are found automatically and used as ground truth for scoring.

Prediction (step 1). The panel sets the model, the seagrass probability threshold (0.7), the overlap of the tiles (384 px), the minimum share of sonar data for a tile to be processed (0.20, so that tiles that are mostly nodata are skipped), the minimum polygon area (1 m^2), the ground

truth polygons and the number of tiles predicted at once.

When the prediction is complete, the notebook shows an overview with the sonar mosaic, the seagrass probability and the predicted seagrass compared with the ground truth, followed by a summary of the seagrass area, its share of the area covered by the sonar, the covered area, the number of polygons and, when ground truth is available, the IoU, precision, recall and F1 score (Figure 45).

On the sample mosaic, 25 tiles were processed in 3 s on the RTX 4060 laptop GPU, and 1,941 m² of seagrass (19.9% of the 9,758 m² covered by the sonar, in 12 polygons) were predicted against 1,856 m² of reference polygons, with an IoU of 0.572, a precision of 0.712, a recall of 0.744 and an F1 score of 0.728. Since the sample mosaic is the one on which the model was trained, these scores are optimistic and higher than the validation IoU of 0.495. The performance of the model on new data can only be assessed on a mosaic it has not seen.



SW_20260716_103554_coverup
threshold 0.70 · model v1 · 25 tiles in 3 s

SEAGRASS	SHARE OF COVERED AREA	COVERED AREA	SEAGRASS POLYGONS
1,941 m² · 0.194 ha	19.9 %	9,758 m²	12 ≥ 1 m²

GROUND TRUTH
1,856 m²

scores against the ground truth	IoU	precision	recall	F1	predicted m²	truth m²
whole mosaic	0.572	0.712	0.744	0.728	1,941	1,856

19 files saved in `results/2026-11-08_222424_SW_20260716_103554_coverup/`

***Figure 45**. Output of the prediction step for the sample mosaic, showing the sonar mosaic, seagrass probability and predicted seagrass (green) with the reference polygons (magenta), followed by the summary of the areas and the scores against the reference polygons.*

Exploration (step 2). A threshold slider updates the image, the seagrass area and the scores immediately, so that the threshold can be adapted to a new mosaic. It is raised when the model marks too much as seagrass and lowered when it misses seagrass. The display can show the prediction, the probability, the errors against the ground truth (correct, missed and false areas in different colours) or the sonar alone, and the results can be saved again with the chosen threshold and minimum polygon area.

Map and download (steps 3 and 4). An interactive map shows the sonar mosaic, the covered area, the predicted seagrass and the ground truth on satellite imagery or a light background map, drawn with the QGIS styles of the repository. The last step creates a zip file with the selected runs. Each run contains a GeoPackage with the seagrass polygons (with their area) and the surveyed area, both with their QGIS styles, the same layers as Shapefiles with a .qml style file, the seagrass mask and probability as GeoTIFF rasters on the grid and coordinate reference system of the mosaic, a summary table, the scores when ground truth is available, the overview figure and the settings used.

4.2. GoPro footage

The code of the video processing pipeline described in Section 3 is available in the public GitHub repository Efxinnos_ML_videos (https://github.com/nikolaoskokkos/Efxinnos_ML_videos).

The repository packages the steps of the pipeline that are needed to map a new dive: the enhancement (Step 0), the detection (Step 4) and the coverage mapping (Step 6), as a tool that can be used without programming, together with the trained models and two sample clips so that it works immediately. The training steps (1 to 3), which require manual annotation of frames, are not part of the notebook. Instead, it offers the five models trained on enhanced frames (v8 and v10 to v13, Section 3.5). The contents of the repository are listed in Table 20.

Table 20. Contents of the Efxinnos_ML_videos repository.

File or folder	Content
posidonia_pipeline.ipynb	The notebook, with a settings panel and the results of each step
.code/	The processing scripts: • enhancement (0.enhance_video.py), • detection (detect_video.py), • coverage mapping (6.make_coverage_map.py), • the telemetry reader (lib/srt_log.py) and • the functions behind the notebook panels (pipeline.py)
models/	The trained YOLO11s-seg models v8, v10, v11, v12 and v13 (Section 2.5), each with its info.txt (training data, classes and scores)
input/	Two 30 s sample clips at 1080 × 1920 px with their .srt logs, covering GX010287 from 1:30 to 2:00 (seagrass and rock, 4 to 5.5 m deep) and GX010175 from 1:20 to 1:50 (seagrass meadow with sand patches in shallow, turbid water)
config_template.ini	Default settings with a comment for each. The notebook creates config.ini from it and updates it with the panel values
styles/	QGIS styles (.qml) for the posidonia_oceanica, rock, debris, surveyed area and camera track layers
requirements.txt, apt.txt, runtime.txt	Python packages, system packages (ffmpeg and the libraries required by OpenCV) and Python version
README.md	User guide describing every control of every step with its default value, tips for the most natural picture, output files and troubleshooting

The repository includes the Jupyter notebook posidonia_pipeline.ipynb, which can be opened and run directly in Google Colab (https://colab.research.google.com/github/nikolaoskokkos/Efxinnos_ML_videos/blob/main/posidonia_pipeline.ipynb). In this way, the pipeline can be tested in a web browser on cloud computing resources, without installing Python or the required libraries on a local computer.

In addition, as new data are collected from future surveys and new models are trained on them, these models can be added to the models folder of the GitHub repository, so that they become accessible to anyone and can be applied directly through the notebook.

When the notebook is opened through this link, its first cell copies the repository into the Colab session, installs the packages that are missing from the standard Colab environment (Ultralytics, imageio-ffmpeg and Rasterio) and activates the interactive settings panels. The notebook requires a GPU runtime, so T4 GPU should be selected as the runtime type in Colab (Runtime > Change runtime type) before the cells are run. The second cell reports the Python, PyTorch and OpenCV versions, the number of CPU cores, the GPU (an NVIDIA Tesla T4 in the session shown in Figure 46), the ffmpeg build used to read and write the videos, and the videos found in the input folder. The repository can also be launched on Binder, which provides a CPU-only environment. In this case, each 30 s sample takes about 8 to 10 minutes for the three processing steps. Colab and Binder sessions are temporary, so the results must be downloaded before the session is closed.

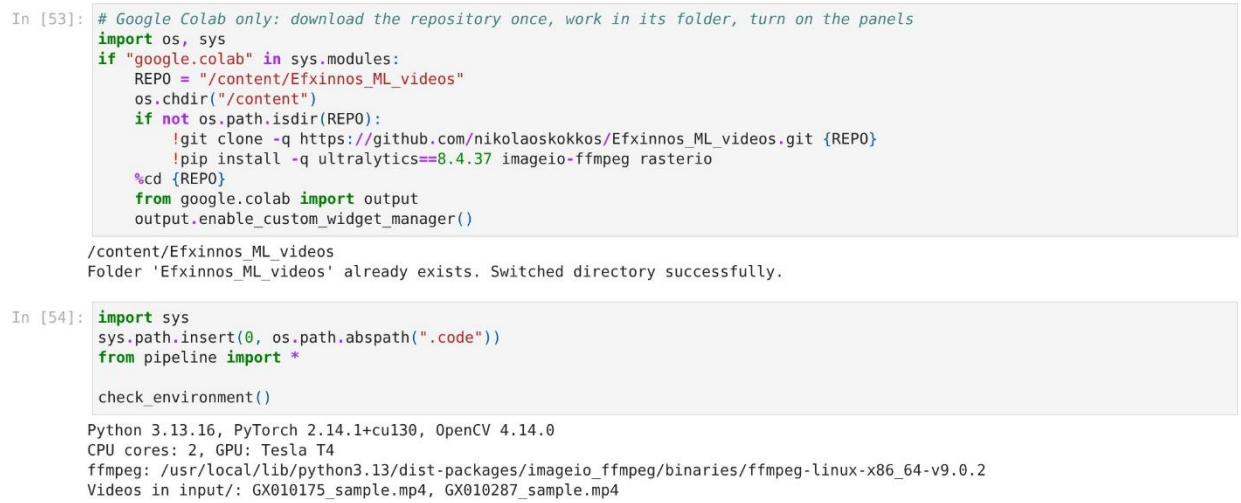


***Figure 46**. First two cells of the notebook in Google Colab, showing the repository copy, package installation, and environment check (Python, PyTorch and OpenCV versions, CPU, GPU, ffmpeg, and sample videos).*

4.2.1. Workflow of the notebook

The notebook is organised in five steps, numbered 0 to 4, which group the pipeline scripts differently from Section 3.2, and each is preceded by a short explanation. Every processing step has a settings panel whose defaults are the recommended values, so a first run only requires executing the cells from top to bottom. To change a setting, the value is edited in the panel and the cell below it is run again.

Each run is stored in a folder of its own, results/<date>_<time>_<video>/, together with a

settings.ini file that records the settings used, so that, unlike in the desktop version (Section 3.2.2), every result remains traceable to its settings. The outputs shown in Figures 47 to 49 were produced with the default settings on the GX010287 sample.

Choice of the video (notebook step 0). The video is selected from the input folder, which initially contains the two samples. Users can add their own video and its .srt log with an upload button for small files, by dragging large files into the file browser, or from a direct download link. The log must have the same name as the video. Without it, the enhancement uses a fixed altitude of 5 m and the coverage map cannot be produced.

Enhancement (notebook step 1, pipeline Step 0). The video is enhanced with the Natural method (Section 3.3.2). The panel controls the range source (flat seabed from the telemetry, or Depth Anything V2), the strength of the correction, the haze removal, a distance colour balance that gives the far part of the frame the colour, brightness and detail of the near seabed, local contrast, the brightness of the sand, the shadow balance, the correction for camera depth and the removal of floating particles (median of five aligned frames, replacement of bright pixels only, or blurring of the streaks). A preview mode fits the water and shows the result at four moments without writing the video, so settings can be tested within a few minutes. When the step is complete, the notebook reports the fitted values and shows the original and enhanced frames at four moments (Figure 47).

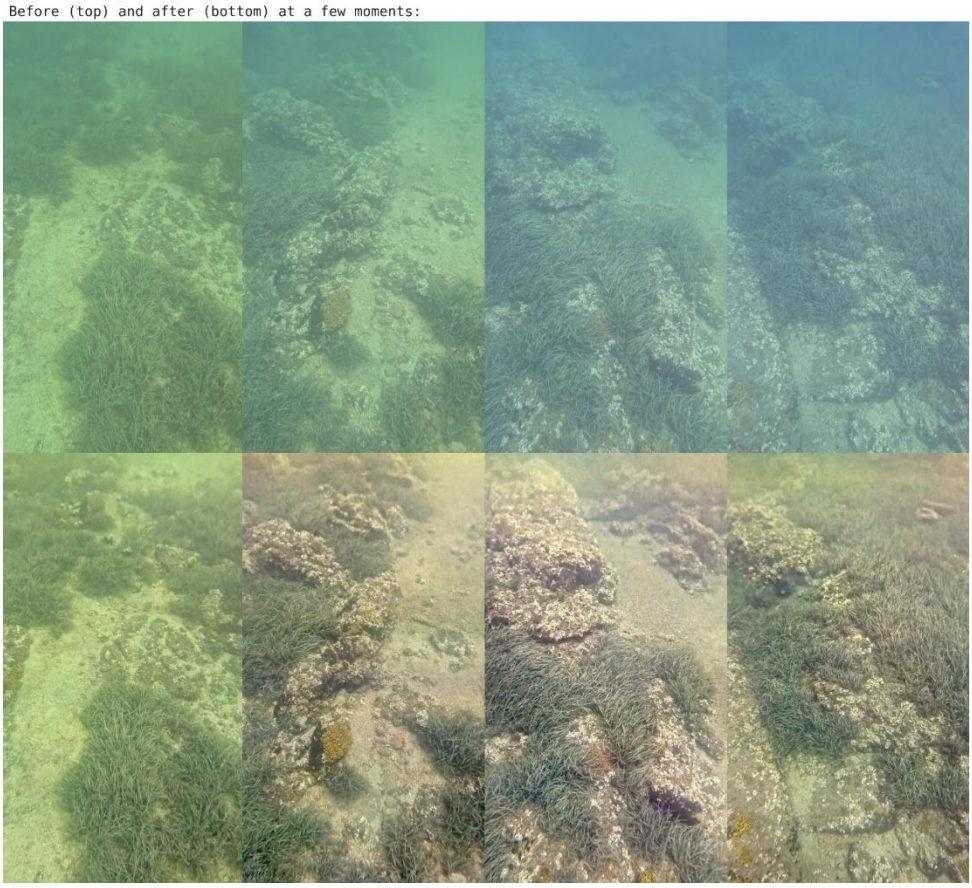


Figure 47. Output of notebook step 1 (enhancement) for the GX010287 sample, showing original frames (top) and frames enhanced with the Natural method (bottom) at four moments.

Detection (notebook step 2, pipeline Step 4). The settings panel lists the available trained models, each with its classes and mask mAP50, and preselects the latest one. It also sets the confidence threshold (0.40 by default), the opacity of the coloured detections, a colour saturation that affects only the display, the frame step (every second frame is analysed by default, which halves the processing time) and the refresh rate of the live view. Following the recommendations in the notebook, v8 is best suited to videos similar to GX010287; v11 gives the best seagrass detection in videos similar to GX010175; and v12 or v13 should be used when debris must be detected. While detection is running, the cell displays the latest annotated frame, the detected areas, their highest confidence, the coverage for each class, and a small chart of the coverage up to that point. At the end, it shows four annotated frames, a chart of the detected cover over time (Figure 48), the average cover per class and a player for the annotated video, and the next cell allows the annotated video to be browsed frame by frame. On the GX010287 sample, model v13 detected an average of 32.5% seagrass and 24.1% rock per frame, and no debris.

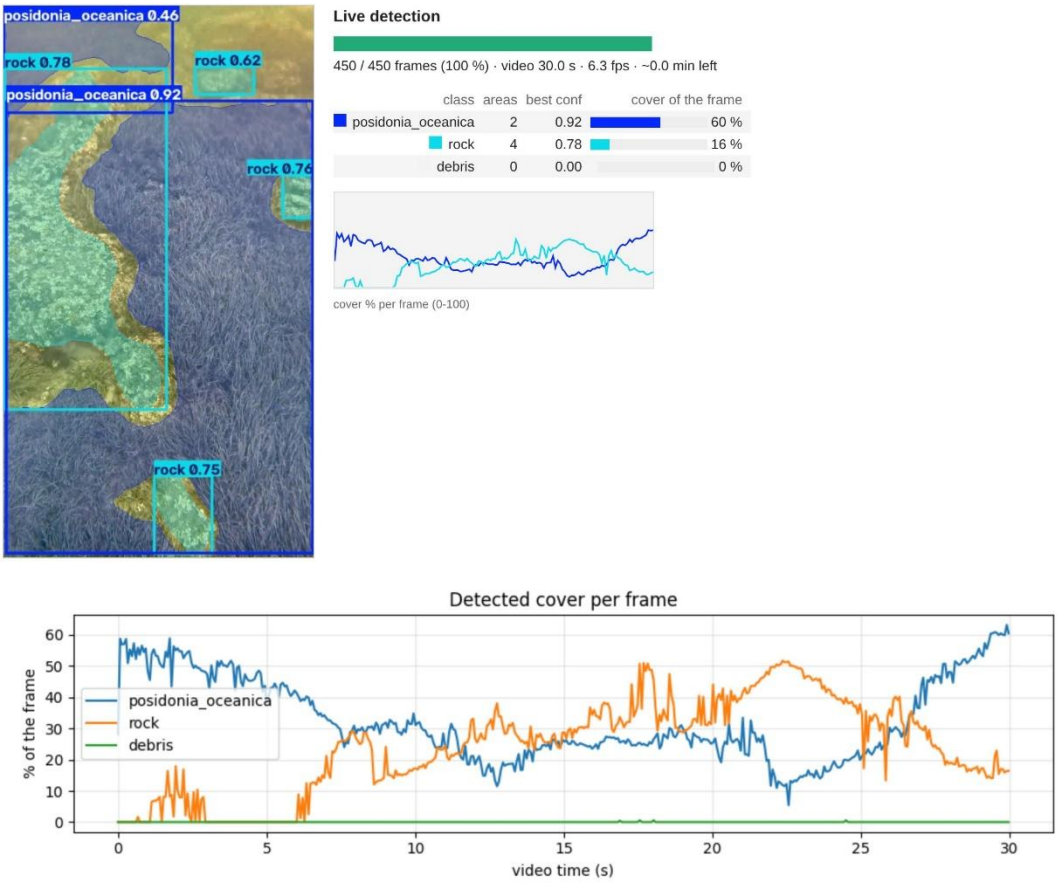
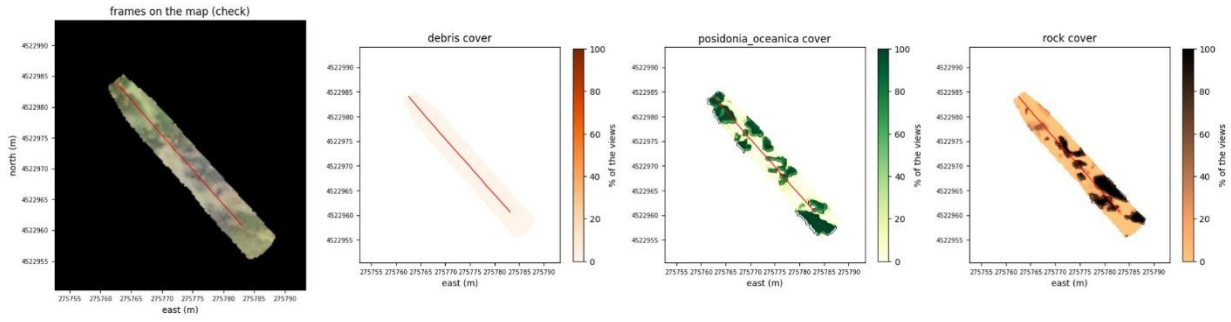


Figure 48. Output of notebook step 2 (detection) for the GX010287 sample with model v13, showing the live view of the latest annotated frame with the classes found, their best confidence and their cover (top), and detected cover of each class per frame over the 30 s clip (bottom).

Coverage map (notebook step 3, pipeline Step 6). Every analysed frame is projected onto the seabed as described in Section 3.7.1, with the same default parameters, namely one frame every 0.5 s, cells of 0.25 m, a maximum range of 8 m, at least three views per cell, a class threshold of 50% and a minimum polygon area of 1 m² (0.1 m² for debris, since a tyre covers

only about 0.3 m²). For a different camera or mounting, the camera tilt and the fields of view (64°, 75.3° and 46.2°) can be adjusted. The notebook first displays the frames placed on the map so that their alignment can be checked. It then shows a coverage raster for each class with the polygon outlines and the camera track, a table giving the surveyed area and, for each class, the number of patches, their area and their share (Figure 49), and an interactive map on OpenStreetMap or satellite imagery in which each layer can be switched on and off. In the 30 s sample, the surveyed seabed covered 198 m², of which 72.1 m² (36.3%, 10 patches) were mapped as seagrass meadows and 44.3 m² (22.3%, 6 patches) as rock.



Surveyed area: 198 m2

	class	patches	area_m2	share_of_surveyed_%
0	debris	0	0.0	0.0
1	posidonia_oceanica	10	72.1	36.3
2	rock	6	44.3	22.3

***Figure 49**. Output of notebook step 3 (coverage map) for the GX010287 sample, showing the frames placed on the map; coverage rasters of debris, seagrass meadows, and rock with polygon outlines; the camera track; and a summary of the surveyed area and the area of each class.*

Download (notebook step 4). A final panel selects the runs and the types of results to include (enhanced videos, previews, annotated videos, detection tables, coverage maps and settings) and creates a single zip file for download. The coverage maps are written as GeoTIFF rasters, a GeoPackage, and Shapefiles in the local UTM zone, as in the desktop version. The QGIS styles in the repository are stored in the GeoPackage and next to each Shapefile, so the layers open in QGIS already styled.

On the RTX 4060 laptop used for the project, the three steps took 2.7, 1.3 and 0.3 minutes for the 30 s GX010287 sample. On CPU-only environments, such as Binder or Colab without a GPU, processing is several times slower, so the notebook is best suited to short clips of up to one or two minutes, while whole dives are better processed with the full pipeline on a computer with an NVIDIA GPU.

5.Bibliography

[1] Ma, Q., Jin, S., Bian, G., & Cui, Y. (2024). Multi-Scale Marine Object Detection in Side-Scan Sonar Images Based on BES-YOLO. *Sensors*, 24(14), 4428. https://doi.org/10.3390/s24144428

[2] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779-788. https://doi.org/10.1109/CVPR.2016.91

[3] Jocher, G., Chaurasia, A., & Qiu, J. (2023). Ultralytics YOLOv8. https://github.com/ultralytics/ultralytics

[4] Yang, N., Li, G., Wang, S., Wei, Z., Ren, H., Zhang, X., & Pei, Y. (2025). SS-YOLO: A Lightweight Deep Learning Model Focused on Side-Scan Sonar Target Detection. Journal of Marine Science and Engineering, 13(1), 66. https://doi.org/10.3390/jmse13010066

[5] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, & A. F. Frangi (Eds.), Medical image computing and computer-assisted intervention – MICCAI 2015 (pp. 234–241). Springer. https://doi.org/10.1007/978-3-319-24574-4_28

[6] Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N., & Liang, J. (2018). UNet++: A Nested U-Net Architecture for Medical Image Segmentation. Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support : 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, held in conjunction with MICCAI 2018, Granada, Spain, S..., 11045, 3–11. https://doi.org/10.1007/978-3-030-00889-5_1

[7] Treibitz, T., Schechner, Y. Y., Kunz, C., & Singh, H. (2012). Flat refractive geometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(1), 51-65. https://doi.org/10.1109/TPAMI.2011.105

[8] Hecht, E. (2017). *Optics* (5th ed.). Pearson Education. ISBN 978-0-13-397722-6.

[9] Akkaynak, D., & Treibitz, T. (2019). Sea-thru: A method for removing water from underwater images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1682-1691.

[10] Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., & Zhao, H. (2024). Depth Anything V2. In *Advances in Neural Information Processing Systems (NeurIPS 2024)*. arXiv:2406.09414.

[11] Jamieson, S., How, J. P., & Girdhar, Y. (2023). DeepSeeColor: Realtime adaptive color correction for autonomous underwater vehicles via deep learning methods. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 3095-3101.

[12] Makam, R., Shankari T M, D., Patil, S., & Sundram, S. (2024). OceanLens: An adaptive backscatter and edge correction using deep learning model for enhanced underwater imaging. *arXiv preprint* arXiv:2411.13230.

[13] Li, C., Guo, C., Ren, W., Cong, R., Hou, J., Kwong, S., & Tao, D. (2020). An underwater image enhancement benchmark dataset and beyond. *IEEE Transactions on Image Processing*, 29, 4376-4389.

[14] Islam, M. J., Xia, Y., & Sattar, J. (2020). Fast underwater image enhancement for improved visual perception. *IEEE Robotics and Automation Letters*, 5(2), 3227-3234.

[15] Fabbri, C., Islam, M. J., & Sattar, J. (2018). Enhancing underwater imagery using generative adversarial networks. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 7159-7165.

[16] Guo, X., Dong, Y., Chen, X., Chen, W., Li, Z., Zheng, F., & Pun, C.-M. (2025). Underwater image restoration via polymorphic large kernel CNNs. In *ICASSP 2025, IEEE International Conference on Acoustics, Speech and Signal Processing*. arXiv:2412.18459.

[17] Fu, Z., Lin, X., Wang, W., Huang, Y., & Ding, X. (2022). Underwater image enhancement via learning water type desensitized representations. In *ICASSP 2022, IEEE International Conference on Acoustics, Speech and Signal Processing*, 2764-2768.

[18] Fu, Z., Lin, H., Yang, Y., Chai, S., Sun, L., Huang, Y., & Ding, X. (2022). Unsupervised underwater image restoration: From a homology perspective. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(1), 643-651.

[19] Saleh, A., Sheaves, M., Jerry, D., & Rahimi Azghadi, M. (2022). Adaptive uncertainty distribution in deep learning for unsupervised underwater image enhancement. *arXiv preprint* arXiv:2212.08983.

[20] Peng, L., Zhu, C., & Bian, L. (2023). U-shape Transformer for underwater image enhancement. *IEEE Transactions on Image Processing*, 32, 3066-3079.

[21] Drews, P., do Nascimento, E., Moraes, F., Botelho, S., & Campos, M. (2013). Transmission estimation in underwater single images. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 825-830.

[22] Li, C., Guo, J., Pang, Y., Chen, S., & Wang, J. (2016). Single underwater image restoration by blue-green channels dehazing and red channel correction. In *ICASSP 2016, IEEE International Conference on Acoustics, Speech and Signal Processing*, 1731-1735.

[23] Zuiderveld, K. (1994). Contrast limited adaptive histogram equalization. In P. S. Heckbert (Ed.), *Graphics Gems IV*, 474-485. Academic Press.

[24] Afifi, M., & Brown, M. S. (2020). Deep white-balance editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1397-1406.

[25] Vlachos, M., Calantropio, A., Skarlatos, D., & Chiabrando, F. (2022). Modelling colour absorption of underwater images using SfM-MVS generated depth maps. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B2-2022, 959-966. https://doi.org/10.5194/isprs-archives-XLIII-B2-2022-959-2022

[26] Lucas, B. D., & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI)*, 674-679.

[27] Wada, K. labelme: Image polygonal annotation with Python. https://github.com/wkentaro/labelme

[28] Jocher, G., & Qiu, J. (2024). Ultralytics YOLO11. https://github.com/ultralytics/ultralytics

[29] Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., Liu, Y., & Chen, J. (2024). DETRs beat YOLOs on real-time object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 16965-16974.

[30] Ravi, N., Gabeur, V., Hu, Y.-T., Hu, R., Ryali, C., Ma, T., et al. (2024). SAM 2: Segment anything in images and videos. *arXiv preprint* arXiv:2408.00714.

NIMRD Technical Contribution to Activity A1.7 Processing of Side Scan Sonar Data, Underwater Images and Videos Using Machine Learning and Artificial Intelligence for the Identification and Delineation of Key Benthic Habitats

NIMRD technical contribution to Activity A1.7. Preliminary tests and exploratory processing were documented in the proposed methodologies and in future validated outputs. No validated seagrass map, quantitative sediment-burial assessment or completed acoustic–optical fusion was approached.

6.Introduction

Within Activity A1.7 of the EfxINNOs project, NIMRD contributed to the development and preliminary testing of methodological approaches for processing underwater imagery and Side Scan Sonar (SSS) data using Machine Learning (ML) and Artificial Intelligence (AI), in continuation of the work that builds upon the acquisition, preparation and harmonisation procedures addressed under Activity A1.6, both supporting future identification, classification and spatial delineation of key benthic habitats, with particular attention to seagrass meadows, seabed substrates and marine litter.

NIMRD's contribution comprised three interconnected components: (i) preparation, quality assessment and preprocessing of underwater imagery from AUV/ROV and complementary video investigations; (ii) exploratory evaluation of AI-assisted image enhancement, object detection and segmentation; and (iii) development of requirements for future integration of optical, acoustic and environmental data into georeferenced, GIS-compatible habitat products.

The investigations were supported by external technical expertise and collaboration with project partners. High turbidity, sediment resuspension, limited visibility and incomplete supporting metadata were important constraints. Accordingly, the work was done as a preliminary methodological stage of quantitative habitat delineation and validated acoustic–optical fusion depends on additional suitable observations and ground-truth data.

6.1. Available Datasets, Data Preparation and Harmonisation

6.1.1. Overview of Available Underwater Datasets

Available datasets originated from different observation platforms (ROV and AUV) and project partners (TUV, DUTH and IU), and varied in acquisition methods, spatial coverage, quality, and associated navigation/environmental metadata. Inventory, quality assessment and harmonisation were therefore necessary before automated interpretation. The main categories were preliminary NEMOSENS micro-AUV SSS and video observations acquired by NIMRD in Constanța Port; complementary GoPro coastal recordings, particularly in the Eforie Sud–Tuzla–Costinești sector; two videos provided by Istanbul University and analyzed with external expert support/POLARIKOS, and ROV imagery considered for possible future exchange with partner P5/IU.

6.1.2. NEMOSENS Micro-AUV and SSS Observations

NIMRD conducted preliminary operational tests of the NEMOSENS micro-AUV at Constanța Port to assess platform performance, sensor configuration, acoustic acquisition, and navigation constraints. Underwater video was also recorded, but high turbidity substantially restricted its suitability for habitat interpretation.

The SSS recordings were operational test data, not a validated benthic survey of the selected EfxINNOs investigation areas. They informed future acquisition and processing requirements but did not support quantitative estimates of sediment coverage or completed habitat maps.

6.1.3. Complementary Coastal Video Investigations

Complementary GoPro observations were undertaken at selected locations, particularly along the Constanta North, and Eforie Sud–Tuzla–Costinești sector, to document seabed characteristics, potential underwater obstacles and conditions relevant to mission planning. Unfavourable marine hydro-meteorological conditions restricted systematic field campaigns and new CTD acquisition during the reporting period. Existing high-resolution recordings were examined where visibility and supporting information permitted. Variable turbidity, colour attenuation and sediment suspension limited reliable identification of benthic features and their depth distribution.

6.1.4. Underwater Video Datasets Provided by Istanbul University

Two underwater video recordings supplied by Istanbul University, totalling approximately 104.6 minutes and 8 GB, were analysed with the support of POLARIKOS S.R.L. The files were Full HD H.264 recordings. Exploratory assessment identified blue-green colour dominance, attenuation of red wavelengths, low contrast, suspended particles, uneven

illumination and other visual artefacts. The videos supported methodological testing of enhancement and detection approaches. The supplied recordings lacked the accompanying synchronised SSS, CTD, navigation, and sufficiently detailed acquisition metadata required for validated acoustic–optical fusion, quantitative georeferencing, and spatial habitat delineation.

6.1.5. Metadata and Quality-Assurance Procedures

The harmonised preparation framework considered the preservation of raw files, encompassing documentation of data provenance and partner responsibility, acquisition time and position where known, platform and sensor settings; image/video format and resolution, navigation availability, environmental conditions, and the history of preprocessing operations. Also, the quality checks addressed corrupted or incomplete files, missing timestamps, inconsistent metadata, optical degradation and positioning uncertainty. Missing information was to be documented rather than replaced by unsupported assumptions.

6.1.6. Partner Data Exchange and Harmonisation

NIMRD discussed potential ROV data exchange with P5/IU and the application of compatible organisation, metadata and quality-control practices to AUV- and ROV-derived observations. The approach was aligned with the general traceability, reproducibility, and interoperability principles of the project Data Management Plan, but complete cross-platform integration and validated georeferenced products have not been demonstrated in the materials currently available.

6.2. Underwater Image Enhancement and AI-Based Preprocessing

6.2.1. Rationale and Objectives

Optical imagery supports the identification of benthic vegetation, substrate features, and anthropogenic objects, but its reliability is strongly affected by water optical properties, illumination, turbidity, and recording geometry. The exploratory processing therefore assessed conventional and learning-based approaches for improving image interpretability, with the principal objectives of identifying degradation mechanisms, evaluating candidate enhancement methods, documenting artefacts and limitations, and establishing requirements for subsequent automated detection and segmentation.

6.2.2. Initial Image-Quality Assessment

Observed degradation included spectral colour distortion and red-wavelength attenuation, reduced local contrast, suspended particles, non-uniform illumination, optical interference and restricted visibility. These characteristics suggested that image-specific processing was preferable to a single uniform correction applied to all recordings.

6.2.3. Conventional Enhancement Techniques

Contrast Limited Adaptive Histogram Equalization (CLAHE) was considered for improving local contrast while limiting excessive amplification. It can make features more visible but may also amplify suspended particles and image noise. Colour correction and white-balance adjustment were examined to mitigate blue-green dominance and improve visual consistency, although the corrections do not restore physically lost spectral information and should not be interpreted as measurements of true benthic reflectance. Dehazing and visibility enhancement methods were considered to reduce haze-like scattering effects. They may improve apparent clarity, but severe turbidity can conceal information irretrievably; artificial edges and textures can also be introduced.

6.2.4. Deep Learning-Based Underwater Restoration

PhISH-Net, U-shape Transformer, Semi-UIR and USUIR were included among the learning-based approaches considered in the exploratory analysis, as alternative strategies for restoring images affected by underwater colour distortion, contrast loss and reduced visibility. These methods were examined as potential preprocessing tools, not

established as validated habitat-classification solutions. The available technical documentation does not provide a quantitative benchmark demonstrating that any single method consistently improves the accuracy of seagrass, substrate, or marine-litter classification under western Black Sea conditions.

6.2.5. Comparative Interpretation and Limitations

Improved visual appearance must be distinguished from validated recovery of scientific information. Enhancement may alter the colour, texture, and boundaries used by downstream classifiers, as a benchmark or ground-truth dataset to quantitatively rank all methods by performance in downstream ecological detection and segmentation, not only by perceived image quality.

6.2.6. Proposed Evaluation and Quality-Control Framework

Considering that a reproducible evaluation should preserve original recordings to extract representative frames across visibility conditions and to apply and log candidate enhancement methods, the evaluation inspected items, compared detection and segmentation on original versus enhanced frames and tried to validate against independently annotated reference imagery. Processing parameters, model versions and uncertainty were recorded for a quantitative image-quality measures were added in complementarity, but should not replace ecological validation.

6.2.7. Relevance for Romanian Pilot Areas

Faleza Nord, Cazino Constanța, and Eforie Sud were designated priority areas for investigating seabed conditions potentially suitable for seagrass occurrence or development, under the influence of beach nourishment and sediment redistribution, which affect transparency/visibility, substrate composition, and benthic communities. In these conditions, where optical imagery is inadequate, SSS may provide complementary seabed-pattern information, while biological identification still requires adequate optical or direct reference observations.

As a preliminary conclusion, the exploratory work identified promising enhancement approaches and key limitations related to visibility, colour attenuation, data provenance, and missing metadata. It established preprocessing requirements and did not quantitatively validate improvements in habitat classification.

7. Machine Learning and AI-Based Object Detection, Segmentation and Habitat Delineation

7.1. Objectives and Approach

NIMRD and external technical expertise explored computer vision approaches for underwater object recognition, segmentation, and future habitat delineation. Candidate methods included SAM 3, YOLO E, Grounding DINO, and YOLO-based models evaluated in connection with the SeaClear and J-Litter marine-litter datasets. The aim was to assess methodological suitability and identify annotation, validation and transferability requirements, not to perform a completed ecological mapping.

7.1.1. Detection, Segmentation and Mapping

Object detection locates discrete targets, often through bounding boxes and confidence scores. Segmentation assigns pixels to objects or classes and is better suited to irregular vegetation patches and substrate boundaries. Habitat mapping additionally requires georeferencing, ecological class definitions, validation and uncertainty assessment. An image mask, even if visually plausible, is not by itself a verified seagrass habitat polygon.

7.1.2. Exploratory Models

SAM 3 was considered for prompt-based segmentation and possible generation of preliminary object masks. Ecological identity cannot be inferred from a mask alone, and no validated habitat-specific accuracy was established.

YOLOE was considered for flexible detection of underwater objects and visually distinguishable features; local performance remains unverified. Grounding DINO was considered for text-guided detection and exploratory annotation, but generic textual prompts cannot replace ecological ground truth.

YOLO-based models associated with SeaClear/J-Litter were considered for marine-litter detection. Their transferability to turbid Black Sea imagery, and particularly to seagrass mapping, requires separate validation.

7.1.3. Preliminary AI Findings

The examined models displayed variable qualitative behaviour under underwater image degradation. The available technical report does not establish class-specific detection accuracy, validated seagrass delineation or completed georeferenced habitat maps, but the work identified potentially useful algorithms and the need for representative local annotations, objective evaluation and better acquisition metadata.

7.1.4. Seagrass and Benthic Habitat Classes

Faleza Nord, Cazino Constanța, and Eforie Sud were selected as priority investigation areas due to their potential relevance to seagrass occurrence or development, and seagrass presence across all polygons has not yet been established in the analyses performed. A proposed annotation scheme may distinguish confirmed seagrass, other benthic vegetation, exposed hard substrate, unconsolidated sediment, mixed substrate, marine litter and uncertain areas, considering several alga species-level identification, including Zostera, for appropriate visual or direct ecological verification.

7.1.5. Marine Litter Detection

Automated litter detection supported inventories of plastic debris, fishing-related objects, and other anthropogenic items, but partial burial, biofouling, fragmentation, turbidity, and visual similarity to natural objects limited the reliability of automated predictions, which were distinguished from manually confirmed observations linked to trustworthy position and acquisition metadata.

7.1.6. Annotation and Validation Framework

Future datasets should include representative examples across seabed classes, lighting conditions, water clarity, and

acquisition platforms, with expert-reviewed labels and a clear annotation protocol being essential. Training, validation and independent test partitions should avoid near-duplicate frames from the same video appearing across splits. Object detection may be evaluated using precision, recall and mean average precision and segmentation using Intersection over Union and Dice coefficient, preferably by class and environmental condition.

7.1.7. Georeferencing and GIS Conversion

Converting image detections into spatial habitat layers required reliable navigation, timestamps, sensor geometry, orientation, and observation altitude, along with a suitable projection and uncertainty treatment. The Istanbul University videos analysed exploratorily lacked the complete accompanying information required for defensible spatial projection. GIS-compatible habitat layers therefore remain a methodological target rather than a validated output of these tests.

7.1.8. Relationship to SSS

Optical imagery may support biological identification, while SSS contributes seabed texture and substrate-pattern information in low-visibility conditions. Acoustic classes cannot automatically be equated with ecological habitats, and the acoustic–optical integration required matching observations and ground truth.

7.1.9. Interoperability

Future derived products should retain provenance, model configuration, processing history, confidence, coordinate reference information and validation status. GIS compatibility and alignment with EMODnet-oriented practices should not be conflated with confirmed publication or acceptance by an external platform; priority actions should include locally representative imagery, expert annotation, quantitative model validation, georeferenced SSS acquisition, optical-acoustic co-registration, and uncertainty-aware spatial products.

7.1.10. Side Scan Sonar Data Processing and Acoustic–Optical Integration

SSS is a valuable complement to optical imaging in turbid nearshore waters. For the Romanian pilot areas, it is relevant to the interpretation of sediment-dominated seabed surfaces, exposed hard substrate and transitional areas potentially affected by nourishment-related sediment redistribution. Sediment deposition in shallow coastal areas, including the approximately 0–5 m depth interval of concern, covered previously exposed hard substrates and affects associated benthic communities. Demonstrating a historical hard-bottom-to-soft-bottom transition requires both temporal evidence and independent validation.

7.1.11. Preliminary Operational SSS Observations

NIMRD's NEMOSENS SSS deployments at Constanța Port were primarily for instrument and operational testing. Navigation and sonar-related technical issues were discussed with the manufacturer RTSYS. Turbid water rendered associated optical recordings unsuitable for reliable habitat interpretation. These tests did not constitute a validated habitat survey of Faleza Nord, Cazino Constanța or Eforie Sud.

7.1.12. Proposed SSS Processing Workflow

After the NOMOSENSE AUV is repaired, the raw data inventory should document the sonar configuration, range, operating frequency, navigation, platform orientation, acquisition time, and any sensor offsets. Quality checks should identify interference, discontinuities, saturation, navigation problems and incomplete coverage. Navigation processing should address timestamp synchronisation, position and attitude quality and sonar offsets. Where justified by the data, preprocessing may include slant-range correction, water-column masking, radiometric normalisation and treatment of acquisition artefacts. Processed lines may then be mosaicked into georeferenced acoustic imagery, with documented spatial resolution, overlap, backscatter consistency and positioning uncertainty.

Also, backscatter depends on sediment properties, roughness, morphology and insonification geometry; it is not a unique indicator of substrate type. Classification should combine intensity, texture, spatial context and independent

reference observations. Candidate physical classes include exposed hard substrate, predominantly sandy/unconsolidated sediment, mixed substrate, transitional or heterogeneous seabed, acoustic anomalies and unclassified areas.

7.1.13. Potential Habitat Transitions

Persistent sandy deposits over hard substrates may reduce stability/attachment surfaces for macroalgae and change the physical setting of benthic communities. Conversely, sedimentary seabeds may support different communities, but sandy substrate alone does not establish seagrass suitability in comparisons with historical maps and repeated surveys, combined with ecological reference observations, are required to assess actual changes in substrate distribution and habitat condition.

7.1.14. Acoustic–Optical Integration

The proposed workflow links georeferenced acoustic mosaics to time-synchronised underwater imagery and validated reference points. Visual observations can help interpret acoustic classes and, where water clarity permits, identify biological communities, but the currently documented exploratory datasets do not provide complete, synchronised acoustic, optical, and navigation information for validated fusion results.

7.1.15. ML-Assisted Acoustic Classification

Supervised classification, texture analysis and acoustic-image segmentation may be investigated when sufficiently representative ground-truth samples exist. Optical models should not be assumed to be transferable to sonar data without adaptation, and the three levels of approach must remain distinct: acoustic facies, interpreted physical substrate classes, and ecologically verified habitat classes.

7.1.16. Potential GIS integrated data

Future validated products may include georeferenced SSS mosaics, interpreted substrate polygons, hard/soft substrate boundaries, acoustic anomaly locations, reference observations and quality/uncertainty layers. Each GIS-integrated dataset should retain acquisition, processing, provenance and validation metadata.

7.1.17. Limitations and Requirements

Accurate navigation, adequate survey coverage, comparable acquisition settings, appropriate acoustic processing, representative ground truth, and historical information are needed to support reliable interpretation and assessment of temporal change.

So, the NEMOSENS tests helped define operational and processing requirements. The proposed SSS workflow links acoustic quality control, mosaicking, substrate interpretation and future optical validation, with particular relevance to sediment distribution and potential habitat modification.

7.1.18. Preliminary Results, Methodological Limitations and NIMRD Contribution

NIMRD's preliminary contribution comprised a dataset inventory and preparation, exploratory image enhancement and AI testing, NEMOSENS operational experience, and a proposed SSS interpretation and harmonisation with project partners. These activities established methodological requirements and will enable validated habitat maps or quantitative ecological status assessments.

7.1.19. Underwater Image Processing Results

P2/NIMRD, together with external expert/POLARIKOS, analysed two Istanbul University videos (~104.6 minutes; ~8 GB), documenting low contrast, colour distortion, suspended particles, and illumination issues. CLAHE, colour correction, dehazing and learning-based methods were examined, and a visual enhancement potential was identified, to complete quantitative benchmarking for improved habitat recognition.

7.1.20. Detection and Segmentation Results

SAM 3, YOLO, Grounding DINO, and YOLO-based marine-litter approaches were explored, yielding variable qualitative results. Neither validated seagrass delineation nor class-specific benchmark performance is established in the available technical documentation. NEMOSENS/SSS Contribution was limited to operational testing at Constanța Port, informed by sensor and navigation requirements and associated optical data, but the observations were not a complete ecological SSS survey due to a malfunction of the SS sensor.

7.1.21. Habitat Delineation Framework

The proposed framework links optical preprocessing, automated detection/segmentation, acoustic substrate interpretation, ground truth, reliable georeferencing and GIS representation. It differentiates acoustic features, physical substrate classes and ecologically validated habitats.

Partner Cooperation was related to the partner-supplied video and SSS data exchange, thus enabling the exploratory methodological testing. Discussions with P5/IU addressed potential ROV data exchange and compatible data handling. Data harmonisation emphasized provenance, metadata, QA/QC and reproducibility.

7.1.22. Principal Limitations and contribution to project methodology

Principal constraints were severe turbidity and variable illumination, limited annotated reference observations, missing or incomplete navigation and environmental metadata, lack of synchronised acoustic–optical datasets, absence of comprehensive quantitative validation, and transferability challenges across pilot environments.

Despite these constraints, the work defined a technically coherent route towards AI-assisted habitat interpretation and SSS-supported substrate mapping. It is particularly relevant for investigating sediment-related pressures and potential seagrass habitats in the Romanian pilot areas.

Thus, the current contribution is strongest as a documented exploratory and methodological effort; even the validated spatial habitat products remain dependent on additional data, expert annotations, appropriate ground truth and quantitative performance assessment.

7.1.23. Conclusions, Recommendations and Next Steps

NIMRD contributed to a methodological framework for underwater imagery and SSS processing using ML/AI approaches, building on A1.6 data acquisition and harmonisation activities. External expertise and partner collaboration supported exploratory testing under challenging conditions along the Black Sea coast.

The analysed Turkish videos demonstrated image-quality constraints and the need for careful preprocessing and independent validation. Preliminary NEMOSENS tests informed future SSS acquisition and navigation requirements but did not produce validated habitat maps.

An integrated acoustic–optical approach is particularly relevant for Faleza Nord, Cazino Constanța, and Eforie Sud, where potential seagrass habitats and nourishment-related changes in sediment motivate further investigation.

7.1.24. Main Methodological Contributions

Five contributions can be distinguished: underwater data preparation and QA; exploratory image enhancement; preliminary AI detection/segmentation assessment; SSS processing and acoustic–optical integration requirements; and partner data harmonisation.

Together these provide a foundation for future reproducible habitat mapping but should not be represented as completed quantitative ecological mapping.

Develop locally representative annotated datasets, secure independent ecological validation, benchmark detection and segmentation with class-specific metrics. The evaluation supported by AI/ML approach will improve downstream accuracy, adapt external models to Black Sea conditions and record model configurations and uncertainty, as well

as the delineation and acquisition of georeferenced SSS data within the three designated areas, using consistent line coverage and navigation quality, to process with documented corrections and mosaicking procedures, validate acoustic classes with visual and sedimentological observations, and compare with historical information where available. Any interpretation of hard-bottom burial or habitat transition should rely on evidence of change rather than present-day acoustic appearance alone, combining validated imagery and acoustic classes with bathymetry, hydrodynamic context, turbidity and other relevant environmental observations.

To preserve the distinction among image-level predictions, physical substrate interpretation, and verified ecological habitats, continued cross-partner alignment on metadata, acquisition records, annotations, quality control, and data provenance will be required under the project DMP. Also, CMEMS environmental data may provide contextual variables, while EMODnet-oriented data practices may support interoperability; neither implies that the derived habitat products have already been uploaded or accepted.

Automated image labels or acoustic classes alone cannot establish habitat health or MSFD ecological status; these require appropriate reference conditions and validated assessment criteria.

7.1.25. Proposed Next Steps

Priorities include additional collaborative consideration of georeferenced underwater and SSS observations, expert-annotated reference datasets, comparative ML/AI validation, acoustic substrate classification, co-registered optical-acoustic interpretation, documented GIS outputs, and continued partner harmonisation.

Validated distribution and pressure information can subsequently be transferred to the A1.8 assessment framework.

Thus, the work undertaken by NIMRD supports the progressive development of innovative, interoperable underwater monitoring methodologies across the Black Sea region. Its principal achievement at this stage is a documented technical and methodological basis for further data collection, algorithm evaluation and ecologically validated habitat mapping.