Hierarchical Deep Learning Framework for Automated Marine Vegetation and Fauna Analysis Using ROV Video Data

Bjørn Christian Weinbach[a,*], Rajendra Akerkar[a], Marianne Nilsen[c], Reza Arghandeh[b]

[a] *Western Norway Research Institute, Røyrgata 4, Sogndal, 6856, Vestland, Norway*
[b] *Western Norway University of Applied Sciences, Inndalsveien 28, Bergen, 5063, Norway*
[c] *Western Norway University of Applied Sciences, Røyrgata 6, Sogndal, 6856, Norway*

Abstract

The convergence of deep learning with remotely operated underwater vehicles (ROVs) offers potential for detailed marine biodiversity studies. This paper contributes to the understanding of underwater object detection by focusing on marine vegetation and fauna and static objects. We introduce the Esefjorden Marine Vegetation Segmentation Dataset (EMVSD) and a comprehensive framework named FjordVision for automated marine vegetation and fauna analysis. FjordVision consists of iterative dataset creation and improvement using object detectors for instance segmentation, augmentations, and semi-automated annotations. Initially, YOLOv8 was used for constructing the dataset. However, later experiments showed that Mask R-CNN performs better at all hierarchical levels. The framework also incorporates a hierarchical Convolutional Neural Network (CNN) model that refines classification based on outputs from the object detector. The object detector model trained on the EMVSD dataset, known as the EMVSD Convolutional Object Detector, serves as the backbone for the hierarchical CNN model, which classifies marine entities across four hierarchical levels: binary, class, genus, and species. Our approach with Mask R-CNN outperforms the baseline YOLOv8 model, demonstrating improvements in hierarchical classification accuracy. Notably, the RT-DETR model also showed improvements with our reclassification algorithm. This study presents the new dataset and explores AI applications in the marine environment, offering insights into the challenges and opportunities in marine biodiversity conservation. The code and substantial parts of the data used in this research are available here.

Keywords: Esefjorden Marine Vegetation Segmentation Dataset (EMVSD),

*Corresponding author. ORCID: 0000-0002-9812-1008
Email addresses: bcw@vestforsk.no (Bjørn Christian Weinbach), rak@vestforsk.no (Rajendra Akerkar), Marianne.Nilsen@hvl.no (Marianne Nilsen), Reza.Arghandeh@hvl.no (Reza Arghandeh)

Hierarchical Deep Learning Framework, Esefjorden, Norway, Marine biodiversity, Underwater imagery, YOLO v8, Object detection, Instance segmentation, Hierarchical CNN, Biodiversity conservation, Deep learning applications, Marine species classification

1. Introduction

Over the last decade, computational capabilities have significantly expanded, enabling solutions to a broader range of tasks. At the same time, Remotely Operated Vehicles (ROVs) have become market-ready, and many actors are investing in ROVs [37]. Marine biologists are now recognizing the potential that ROVs offer and actively use them in their research to collect data in underwater environments [13]. Researchers working with this kind of data are eager to automate parts of the workflow that are currently done manually, such as identifying the time points at which objects of interest appear in the video.

According to Fayaz et al. [9], a computer vision approach employed for identifying and locating underwater objects in a subaquatic picture or video is known as underwater object detection. González-Sabbagh and Robles-Kelly [12] state in their comprehensive review of computer vision in the underwater environment that despite its critical importance, underwater computer vision presents significant challenges due to the inherent difficulties associated with color cast, image degradation, loss of contrast, and low-light conditions often encountered in underwater scenes. These challenges arise from the scattering, distortion, attenuation, and other optical effects inherent to light propagation in water columns [12, p.1]. Using Esefjorden as our case in an investigation into the field of underwater computer vision, with a focus on marine vegetation and fauna.

Object detection and segmentation are fundamental tasks in computer vision, aiming to identify and delineate objects within images. These tasks encompass several challenges, including variability in object sizes, occlusion, diverse and complex backgrounds, and varying lighting conditions. Object detection focuses on identifying objects and their bounding boxes, while segmentation goes further to classify each pixel of the object, with instance segmentation distinguishing individual instances of the same class. Recent advancements in datasets, such as FishNet [18] and WildFishNet [41], have contributed significantly to training models that can handle the complexity of underwater environments.

Before the advent of You Only Look Once (YOLO), many object detection frameworks relied on region proposal algorithms as a critical step in their pipeline. These methods, such as R-CNN and its successors, Fast R-CNN, and Faster R-CNN, employed a two-stage approach where the first stage generated potential object boundaries (region proposals), and the second stage classified these regions into object categories and refined their bounding boxes [11, 10, 31]. While effective, these approaches were computationally intensive and slow, hindering their applicability for real-time detection tasks.

The introduction of YOLO by Joseph Redmon et al. marked a significant departure from region proposal-based methods. YOLO's innovative design allowed it to predict bounding boxes and class probabilities in a single forward pass of the network, dramatically improving detection speeds and enabling real-time applications [28]. This innovation significantly accelerated object detection, making it feasible for real-time applications. Subsequent iterations, including YOLOv2 and YOLOv3, introduced anchor boxes and multi-scale detection, respectively, enhancing accuracy and speed [29, 30]. The development continued with YOLOv4, which optimized the balance between speed and accuracy further [2]. Ultralytics' contributions from YOLOv5 to YOLOv8 incorporated advancements in training techniques and model scalability, with YOLOv8 achieving significant performance improvements [16]. Most recently, YOLOv9 introduces the concept of programmable gradient information (PGI) and the Generalized Efficient Layer Aggregation Network (GELAN) architecture, which addresses the information bottleneck in deep learning by improving parameter utilization and enabling better performance in object detection tasks, demonstrating the ongoing evolution and versatility of the YOLO architectures in addressing complex object detection tasks in real-time [36]. Additionally, specific adaptations of YOLO, such as those proposed by Yang et al. [40], have shown improvements in challenging conditions like foggy maritime environments, further emphasizing YOLO's robustness.

One notable example of early work in underwater object detection is presented by Elawady [7], where they used a preliminary version of the CoralNet dataset [4], which at the time consisted of corals in Moora, French Polynesia. Cui et al. [6] used data of fish collected in the Gulf of Mexico and employed a convolutional neural network (CNN) to detect fish in blurry underwater environments. Xu and Matzner [39] used three different datasets of fish to train a You Only Look Once (YOLO) classifier. The datasets originated from Scotland, Washington, and Alaska, and in a hydroelectric setting. Some of the impactful works in this field have been done by researchers at the University of Agder, Norway [24, 20, 17]. In the first study, Olsvik et al. [24] used CNN-SENet to classify images from Fish4Knowledge as well as a self-made dataset of some of the most abundant fish species in Northern Europe. In the second study by Knausgård et al. [20], a two-stage approach is proposed, with YOLO v3 being utilized for object detection and CNN-SENet being used for classification. This is in contrast to most available literature and is done for a much more fine-grained classification of species. The last study by Kalhagen et al. [17] goes away from the two-stage approach and instead employs the YOLO algorithm with hierarchical classification to solve the challenge with fine-grained classifications. Since the underwater environment comes with visibility and illumination variations, uncertainty in the class labels arises. In instances where the model is uncertain, a higher-order class can be used instead. They name their YOLO algorithm "YOLO Fish."

Despite significant advancements, current research exhibits some limitations that restrict its application to broader marine biology studies. Previous methods have mainly focused on fish detection. While this is very useful and has use cases

in industry and especially aquaculture, it is a limitation. Biologists use ROVs to collect more diverse biological data than that limited to fish. To our knowledge, established ontologies are mostly limited to fish detection. If one is to employ a classification scheme similar to that of the YOLO Fish [17], an ontology must be defined for the use case of interest. There are limited datasets outside that of fish detection in a Norwegian setting. While the iterative approach described by [5] gives many opportunities to define new datasets for underwater object detection in Norway, this remains to be done at the time of writing this review. Notably, datasets like FishNet and models such as FishDETECT [14] and the YOLOv5-CNN model discussed by Patro et al. [25] have provided foundational frameworks, yet challenges remain in the effective application of these models in more complex marine environments.

To address these limitations, we present FjordVision, a hierarchical deep learning framework for automated marine vegetation and fauna analysis in Esefjorden, Norway. FjordVision consists of two main components: data preprocessing and dataset construction, and hierarchical post-processing convolutional network. In data preprocessing and dataset construction, we introduce the Esefjorden Marine Vegetation Segmentation Dataset (EMVSD), an open-access dataset of labeled segmentations for marine life observed in Esefjorden, Norway. This dataset contains 17,362 images and over 30,000 objects. We use YOLO v8 for instance segmentation, augmentations, and semi-automated annotations to construct and enhance the EMVSD dataset. The YOLO v8 model, known as Yolo v8 Convolutional Object Detector, is trained on the EMVSD dataset.

In the hierarchical post-processing convolutional network, we train on Yolo v8 Convolutional Object Detector outputs, including class and confidence. This network refines marine entity classification into four taxonomic levels: binary, class, genus, and species.

The contributions of this paper include:

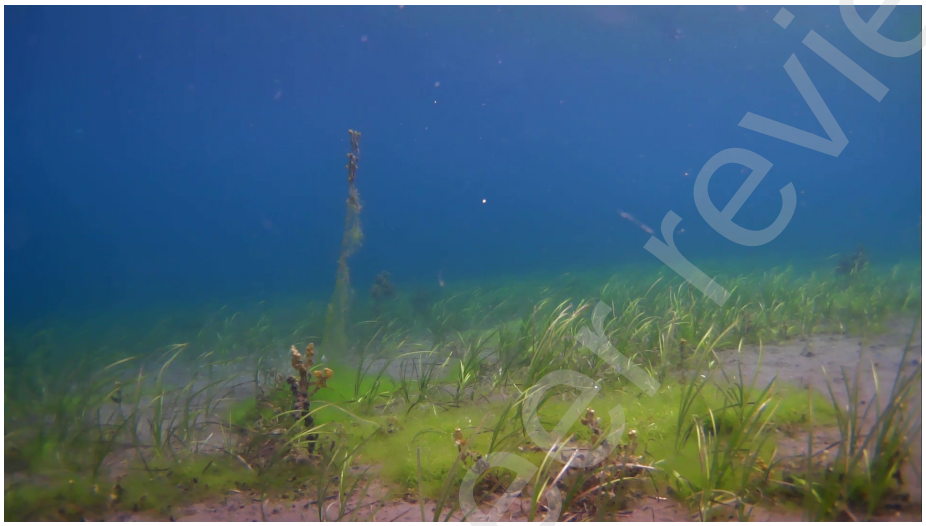
1. Creating a new open-access dataset of labeled segmentations for marine life observed in Esefjorden, the Esefjorden Marine Vegetation Segmentation Dataset (EMVSD).
2. Developing a framework for training underwater models that combines subsea drones, Meta's Segment Anything Model, and YOLO v8.
3. Developing a hierarchical CNN post-processing block that classifies outputs of YOLO into hierarchical classes.
4. Offering insights into the challenges of hierarchical classification in the marine environment, noting that reclassification generally did not improve over the baseline model except in specific cases like RT-DETR.

2. Use Case and Data Description

This study utilizes two primary datasets. The Esefjorden Marine Vegetation Segmentation Dataset (EMVSD), which includes 17,000 annotated images which is used to train all the models in this paper. This dataset is publicly available

here (DOI: 10.6084/m9.figshare.24072606). The experimental data and trained models from this paper is available here (DOI: 10.6084/m9.figshare.25688718).

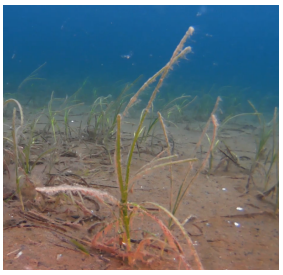
2.1. Location and Data Collection



(a) Image of Eelgrass meadow.

(b) Bladder wrack (*Fucus vesiculosus*)

(c) *Urospora*



(d) Eelgrass (*Zostera marina*)

Figure 1: Different observed organisms in Esefjorden.

According to a Huseklepp [15], Esefjorden, is a smaller fjord within the larger Sognefjorden, spanning approximately 4 kilometres. The fjord harbours several crucial natural habitats, a selection is shown in figure 1, including a patch area for Eelgrass (*Zostera marina*) and spawning grounds for various aquatic animals, such as Atlantic Cod (*Gadus morhua*), Haddock (*Melanogrammus aeglefinus*), European Pollock (*Pollachius pollachius*), Whiting (*Merlangius merlangus*), Turbot (*Scophthalmus maximus*), and potentially European Plaice (*Pleuronectes platessa*) and European Flounder (*Platichthys flesus*) [20]. The relatively shallow depth of the fjord makes it suitable for subsea drone dives,

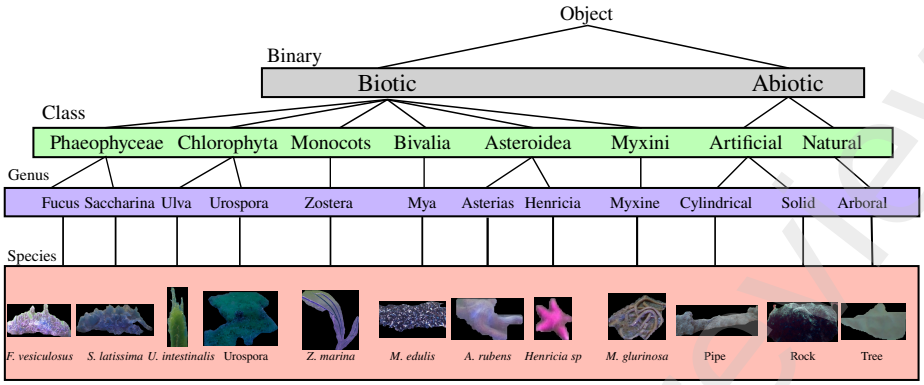


Figure 2: Ontology of Esefjorden. The ontology is used for training the hierarchical machine learning model described in section 3. In the dataset the four hierarchical labels *binary*, *class*, *genus*, and *species* are used. To make all classes fit the hierarchy during training, a creative approach was taken in the abiotic class with the constructed taxonomic ranks *Artifical*, *Natural*, *Cylindrical*, *Solid* and *Arboral* to incorporate the classes *Rock*, *Pipe*, *Tree* into our ontology.

as it ensures better video illumination and reduces the risk of tether breakage. In 2015, The Institute of Marine Research (IMR) identified a patch of eelgrass in the inner parts of the Esefjorden [15]. Eelgrass meadows play a crucial structural and functional role in many coastal ecosystems [3]. They are also important indicator species, sensitive to eutrophication, and can reflect and integrate water quality over extended periods [3]. Additionally, eelgrass meadows play a significant role in carbon sequestration, with studies highlighting their capacity to store substantial amounts of blue carbon, which contributes to mitigating climate change [32, 26]. Human-mediated factors, such as overfishing, eutrophication, and habitat destruction, pose both local and global threats to eelgrass ecosystems [3]. Given these characteristics, Esefjorden was deemed suitable for our research. We conducted a dive to assess the spatial extent of Eelgrass, other species, and objects in situ, and to collect data for segmentation, localization, and classification by our computer vision framework. Manual inspection of drone footage identified several species for classification, leading to the development of the ontology depicted in section 2.2 which is used for training the hierarchical machine learning model described in section 3.

2.2. Organisms and Ontology

This study was initiated to map the distribution of *Zostera marina*, as described by Huseklepp [15], with a keen interest in automating aspects of the post hoc analysis of observations from the study site. Our exploration encompasses a diverse array of organisms, including marine vegetation and mussels, alongside abiotic objects. A summary of the organisms and objects segmented in this study is presented in Table 1, highlighting the broad scope of our dataset EMVSD which is crucial for developing a comprehensive ontology for machine learning and hierarchical classification.

Class	Segmented Objects
Zostera marina	5994
Urospora	4187
Mytilus edulis	3808
Tree	2626
Fucus vesiculosus	2910
Pipe	1916
Myxine glurinosa	1737
Asteroidea	1740
Saccharina latissima	1612
Asterias rubens	1650
Henricia sp.	1515
Ulva intestinalis	1513
Rock	1320

Table 1: Identified species and objects segmented in the study. Text in italics indicates identified organisms. Other classes represent abiotic objects or unidentified organisms.

The numbers in the table reflect the count of segmentation masks in the dataset, which does not necessarily equate to the number of samples or specimens. This distinction is crucial for understanding the dataset's structure and for informing the development of machine learning models capable of hierarchical classification. The inclusion of both abiotic and abiotic classes underlines the complexity of the marine ecosystem and the challenges posed in automating the classification and analysis of underwater imagery. In the development of our ontology for machine learning and hierarchical classification, we employed a simplified taxonomy that includes binary classification (distinguishing marine life from abiotic objects), class, genus, and species. This approach was designed to balance the need for detailed biological classification with the practical considerations of machine learning model training and performance. The binary classification serves as the foundational layer of our ontology, enabling the initial separation of biotic entities from abiotic objects within the underwater imagery. This distinction is critical for focusing subsequent, more granular classification efforts on biologically relevant data. At the broadest level of biological classification within our dataset, we use the class label. This level allows for the grouping of organisms into categories that represent significant differences in biological characteristics, providing a manageable and informative level of diversity for ecological analysis and machine learning applications. The genus and species levels offer the most fine-grained classification, enabling precise identification and analysis of marine life. These levels are crucial for ecological studies aiming to understand species distribution, habitat preferences, and biodiversity within specific environments, such as Esefjorden. By focusing on these levels, we aim to contribute detailed, species-specific insights to the body of marine ecological knowledge.

3. Methodology

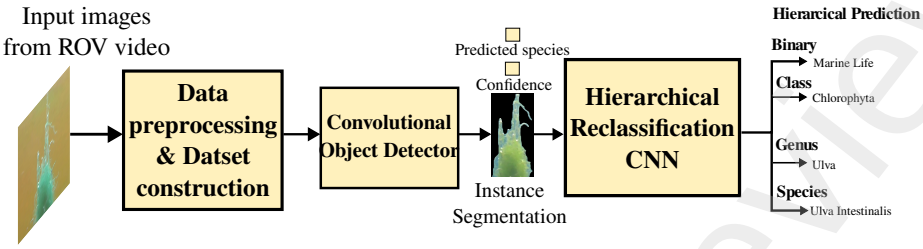


Figure 3: FjordVision Framework: This diagram illustrates the comprehensive process of data collection, preprocessing, and analysis within the FjordVision project. It highlights key components such as manual annotation, data augmentation, model training, and hierarchical CNN training. The flow from raw video data collected via ROV to final predictions made by the hierarchical CNN is depicted, showcasing the iterative improvements and the reclassification of predictions across multiple taxonomic levels.

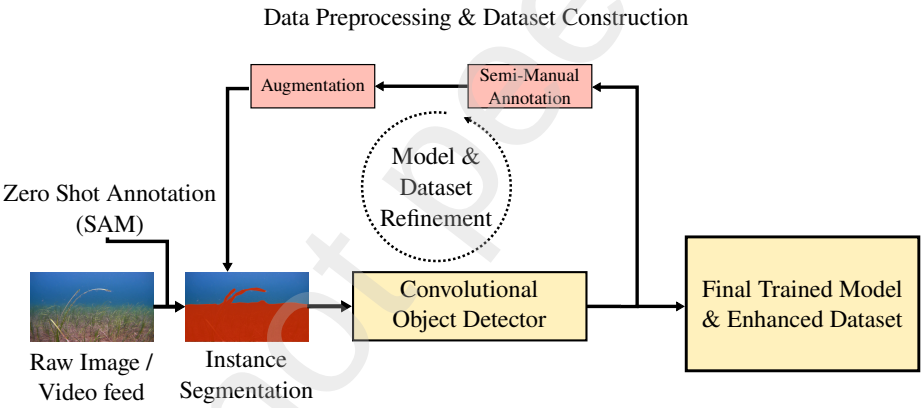


Figure 4: Pipeline for dataset construction and model training: This diagram outlines the step-by-step process of creating a robust dataset and training a convolutional object detector. It starts with raw image data from ROVs, followed by zero-shot annotation using the Segment Anything Model (SAM) by Meta. The process includes augmentation and semi-manual annotation, with iterative loops for model and dataset refinement, leading to the final trained model and enhanced dataset.

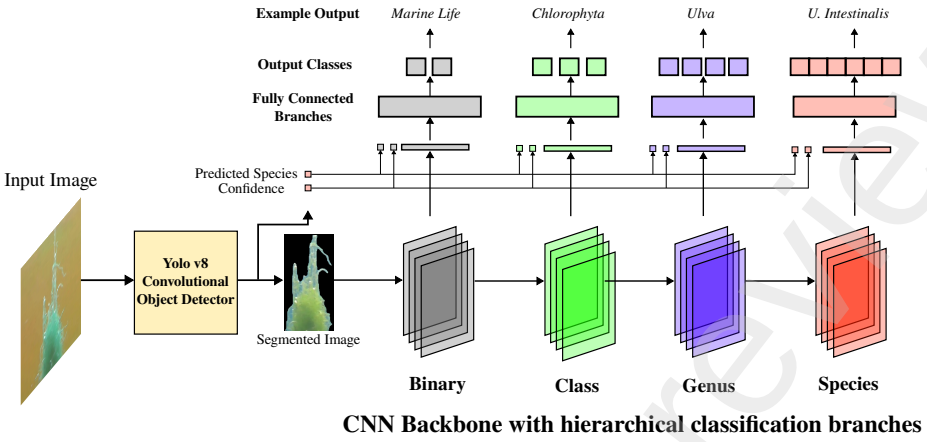


Figure 5: Architecture of Hierarchical CNN for Post-Processing: This figure details the architecture used for the post-processing of segmented images produced by the convolutional object detector. The CNN backbone is followed by fully connected branches that classify the images across four hierarchical levels—Binary, Class, Genus, and Species. Each branch in the network is designed to consider both global and contextual features, ensuring accurate hierarchical classification. The diagram also illustrates the flow of information from input images to final hierarchical predictions.

The methodology for our study involves a series of structured steps aimed at developing an isntance segmentation based hierarchical reclassification framework. This section outlines our approach from data acquisition to our hierarchical reclassification method. Our framework, depicted in Figure 3, is designed to enhance the classification and monitoring of marine biodiversity using machine learning techniques integrated with remotely operated vehicle (ROV) technology. A cornerstone of our research is the introduction of the Esefjorden Marine Vegetation Segmentation Dataset (*EMVSD*). We deployed ROVs to gather video streams from the marine environment of Esefjorden, capturing a diverse range of marine vegetation and species. The first step involves annotating images using Meta's Segment Anything Model (SAM) [19], which facilitates rapid labeling by leveraging zero-shot and promptable segmentation capabilities. The initial labeling process yielded 400 labeled images, encompassing over 3000 segmented objects. h Following the initial labeling, we applied data augmentation techniques to increase the robustness and volume of our training data. Augmentations included vertical flip, gamma transform, Gaussian noise, advanced blur, random brightness and contrast, and color jitter. Horizontal flips were excluded as they were deemed unrealistic for the underwater drone's perspective. Initially, we used the SAM-labeled dataset to apply transfer learning and fine-tune the YOLOv8n-seg model [16], resulting in our specialized EMVSD Convolutional Object Detector model. This model was employed to pre-annotate new, unseen data or re-annotate existing data, iteratively enhancing both the model and the dataset, as shown in Figure 4. We designed our framework to be flexible, allowing the integration of various instance segmentation models. While YOLOv8 was used initially, we also experimented with other models, including YOLOv9,

RT-DETR, and Mask R-CNN. Each model was evaluated for its performance in segmentation and classification tasks. We defined a hierarchical ontology to categorize the segmented objects according to different biological and artificial classes. Our Hierarchical CNN model, depicted in Figure 5, was trained on the segmented images along with the class predictions and confidence scores from the chosen segmentation model to reclassify images across all hierarchical levels defined in the ontology.

3.1. Instance segmentation and Taxonomies

Instance segmentation and classification in images are crucial applications of deep learning models with significant implications for various fields, including biology and environmental science. Previous research has demonstrated the potential of leveraging the hierarchical properties of biological taxonomy to improve the classification capabilities of object detectors [17]. Inspired by this approach, we sought to explore its application in the context of underwater image segmentation and detection for remotely operated vehicles (ROVs). The marine environment of Esefjorden, Norway, presents a diverse array of species, each belonging to different taxonomic ranks as described in section 2.2. Accurately identifying and classifying these organisms is essential for monitoring biodiversity, assessing environmental changes, and conducting ecological research. However, the underwater environment poses unique challenges for image segmentation and object detection, including variations in lighting, water turbidity, and the presence of floating particles. In this study, we aimed to assess the adequacy of current technology, specifically various instance segmentation models, for performing taxonomic classification of specimens observed in the subsea environment of Esefjorden. We were particularly interested in determining whether these models could achieve Average Precision (AP) scores ≥ 0.75 and if we could get similar results on different hierarchical levels of classification and to improve classifications by proposing a post-processing CNN model that can be used to classify segmentations from these models into different taxonomic ranks. Recent advancements, particularly the Segment Anything Model (SAM) [19] and its successor SAM 2 [27] by Meta AI, provide promising opportunities for incorporating unsupervised methods into our approach. By leveraging SAM 2's ability to perform segmentation with minimal prompts in both images and videos, future work could explore unsupervised learning pipelines that significantly reduce the need for extensive manual labeling in marine biodiversity studies.

Evaluating the performance of object detection and instance segmentation models is crucial for understanding their effectiveness and applicability to real-world scenarios. Several metrics have been established in the literature to provide comprehensive assessments of these models. Among the most widely used are Average Precision (AP), Intersection over Union (IoU), and Non-Maximum Suppression (NMS), along with other relevant metrics for instance segmentation. AP is a popular metric used to evaluate the accuracy of object detectors, summarizing the precision-recall curve into a single value. It calculates the average precision values for recall levels over [0, 1]. The AP for a single class or

category is calculated as the mean precision at a set of eleven equally spaced recall levels $[0, 0.1, \ldots, 1]$:

$$AP = \frac{1}{11} \sum_{r \in \{0, 0.1, \ldots, 1\}} p(r) \tag{1}$$

where $p(r)$ is the precision at recall r. This measure provides a comprehensive assessment of model performance across different thresholds, making it especially useful in comparing the effectiveness of various detection algorithms [8, 21]. NMS is an algorithm used to select one bounding box out of several overlapping boxes for a particular object. It is described by ranking the detection boxes based on their scores, selecting the top-ranked detection box, and eliminating all other boxes that have an IoU value greater than a predefined threshold with this box, and repeating this process until no boxes remain. This process effectively reduces the number of redundant bounding boxes, retaining only the most accurate one per object. NMS plays a critical role in object detection pipelines, ensuring that each detected object is represented by a single, precise bounding box [23]. In hierarchical classification tasks, achieving an optimal balance across different levels of the hierarchy is paramount. Inspired by the works on branch CNNs [42], leveraging class hierarchies [1], and hierarchical loss applications in astrophysics [35], we extend the concept to introduce the Hierarchical Cross-Entropy Loss. This novel loss function incorporates a dynamic weighting mechanism, governed by a parameter α, to modulate learning emphasis across hierarchical levels—from the most general to the most specific categories. Given a dataset with N samples organized into a hierarchy of L levels, let $y_i^{(l)}$ denote the prediction for the ith sample at level l, with the corresponding ground truth label $t_i^{(l)}$. The Cross-Entropy loss for a single level l and sample i is defined as:

$$\mathrm{CE}_i^{(l)} = -\sum_{c=1}^{C_l} t_{i,c}^{(l)} \log(y_{i,c}^{(l)}) \tag{2}$$

where C_l is the number of classes at level l, $t_{i,c}^{(l)}$ is a binary indicator for whether class c is the correct classification for observation i at level l, and $y_{i,c}^{(l)}$ is the predicted probability of sample i belonging to class c at level l. To adaptively emphasize the importance of each hierarchical level, we introduce weighting factors λ_l, dynamically adjusted based on the parameter α:

$$\lambda_l = \exp\left(-\alpha \cdot (L - l)\right) \tag{3}$$

This equation is refined to accurately reflect the intended emphasis decay when traversing up the tree from the leaf nodes towards the root. With a positive α, the model places more weight on the finer, more specific classifications at the lower levels, with the emphasis decreasing as we move towards the more general categories at higher levels. The total Hierarchical Cross-Entropy Loss for all samples across all levels is computed as:

$$\text{HCE} = \sum_{i=1}^{N} \sum_{l=1}^{L} \lambda_l \cdot \text{CE}_i^{(l)} \tag{4}$$

This loss function enables the model to learn relevant features across the hierarchy by adaptively balancing the importance of accuracy at each level through α. Training the model with different α values facilitates exploration of the optimal balance between hierarchical levels for the task at hand, essential for enhancing performance in hierarchical classification tasks. Exploring various α values during training allows for fine-tuning the model's focus on different hierarchical levels. A lower α results in more uniform weighting across levels, while a higher α prioritizes accuracy at the lower, more detailed levels. This adaptability is crucial for achieving the best possible performance in tasks with hierarchical class structures.

3.2. Architecture

To effectively tackle hierarchical classification tasks, we have developed a novel Hierarchical CNN architecture. Inspired by advancements in branch CNNs [42], the utilization of class hierarchies [1], and specific applications of hierarchical loss functions [35], our model is designed to learn and emphasize features relevant to each level of a given hierarchy. The Hierarchical CNN shown in figure 5 and detailed in Table 2 is composed of a series of convolutional layers designed for feature extraction, followed by a set of branches, each corresponding to a different level of the hierarchical classification task. Each branch employs a Branch CNN, a specialized sub-network designed to handle the classification at its respective level of the hierarchy. This structure allows for the extraction and utilization of features pertinent to each hierarchical level, facilitating a nuanced understanding and classification of input data. The architecture utilizes the EMVSD Convolutional Object Detector, which is trained on our dataset EMVSD created by the iterative annotation method described in Figures 3 and 4. This reclassification algorithm enhances the accuracy and reliability of hierarchical classifications by leveraging the robust initial detections provided by the chosen instance segmentation model, such as YOLOv8, YOLOv9, RT-DETR, or Mask R-CNN. The model incorporates Mish [22], which has been shown to improve the generalization capabilities of neural networks. Attention mechanisms, including Channel and Spatial Attention, are used to enhance feature representation and focus selectively on more informative parts of an input image [38]. Furthermore, our architecture introduces an innovative approach to enhancing the model's sensitivity to hierarchical information. By registering hooks on the last layer of each convolutional block, we capture intermediate activations that are then utilized by the branches. This technique allows branches to leverage both the global features extracted by the entire convolutional stack and the more localized features pertinent to their specific hierarchical level. The Hierarchical CNN also includes a mechanism for dynamically adjusting the emphasis placed on each level of the hierarchy during training, governed by the parameter α. By modulating this parameter, the model can adapt its learning process to

Table 2: Hierarchical CNN Architecture

Layer	Type	Channel	Filter	Stride
conv1_1	Conv2d	3→32	5×5, p2	1
BN1_1	BatchNorm2d	32	-	-
conv1_2	Conv2d	32→32	3×3, p1	1
BN1_2	BatchNorm2d	32	-	-
MaxPool1	MaxPool2d	32	2×2	2
ChAtt1	Channel Attention	32	-	-
SpAtt1	Spatial Attention	-	5×5, p2	1
conv2_1	Conv2d	32→64	3×3, p1	1
BN2_1	BatchNorm2d	64	-	-
conv2_2	Conv2d	64→64	3×3, p1	1
BN2_2	BatchNorm2d	64	-	-
MaxPool2	MaxPool2d	64	2×2	2
ChAtt2	Channel Attention	64	-	-
SpAtt2	Spatial Attention	-	5×5, p2	1
conv3_1	Conv2d	64→128	3×3, p1	1
BN3_1	BatchNorm2d	128	-	-
conv3_2	Conv2d	128→128	3×3, p1	1
BN3_2	BatchNorm2d	128	-	-
MaxPool3	MaxPool2d	128	2×2	2
ChAtt3	Channel Attention	128	-	-
SpAtt3	Spatial Attention	-	5×5, p2	1
conv4_1	Conv2d	128→256	3×3, p1	1
BN4_1	BatchNorm2d	256	-	-
conv4_2	Conv2d	256→256	3×3, p1	1
BN4_2	BatchNorm2d	256	-	-
MaxPool4	MaxPool2d	256	2×2	2
ChAtt4	Channel Attention	256	-	-
SpAtt4	Spatial Attention	-	5×5, p2	1
GlobalPool	AdaptiveAvgPool2d	256	-	-
BranchCNN after each Conv Block for each Level				
Linear1	Linear	-	$256 \rightarrow 4096$	-
BN_1	BatchNorm1d	4096	-	-
Linear2	Linear	-	4096→2048	-
BN_2	BatchNorm1d	2048	-	-
Linear3	Linear	-	2048→1024	-
BN_3	BatchNorm1d	1024	-	-
Linear4	Linear	-	1024→512	-
BN_4	BatchNorm1d	512	-	-
Linear5	Linear	-	512→256	-
BN_5	BatchNorm1d	256	-	-
Linear6	Linear	-	256→128	-
BN_6	BatchNorm1d	128	-	-
Linear7	Linear	-	128→N cls	-

prioritize accuracy at different hierarchical levels, from general to specific, according to the task's requirements. The model is trained on a dataset organized according to a predefined hierarchy, with each branch of the model responsible for classifying one level of the hierarchy. During training, we explore various values of α to find the optimal balance between hierarchical levels, ensuring that the model achieves high accuracy across all levels of classification.

3.3. Activation Visualization Methodology

To understand the internal workings of our Hierarchical CNN model and how it processes information at various levels of the classification hierarchy, we employed a method to visualize the activations within the network. This visualization helps illustrate the model's focus and the features it extracts at different hierarchical levels, offering insights into its decision-making process. For a detailed analysis, we also generated class-activation maps for specific classes, such as Mytilus edulis, using an optimization process described below. The generation of class-activation images involved an iterative process where an initial image tensor is optimized to maximize the activation of a target class. This process, detailed mathematically below, helps highlight the specific features within the input images that the model deems significant for classification. Mathematically, we initialize an initial image tensor I with random noise, where $I \in \mathbb{R}^{C\times H\times W}$, with C representing the number of channels and H, W representing the height and width, respectively. The objective is to modify the image tensor to maximize the activation of a target class. This is achieved by computing the gradient of the loss with respect to the image tensor and adjusting the tensor using gradient ascent. The loss $\mathcal{L}$ for a target class c is defined as $\mathcal{L}(I, c) = -\text{output}_c$, where output_c is the activation of the class c obtained from the model for image I. At each upscaling step, the image tensor is resized to increase its dimensions, enabling the visualization of more detailed features. The new size at step s is given by $\text{new\_size} = \text{size} \times (\text{upscaling\_factor})^s$. After the specified number of upscaling steps, the final image tensor represents the class-activation map, highlighting the features most relevant to the target class.

The algorithm for class-activation image generation is designed to iteratively enhance an image tensor, emphasizing the features most pertinent to a target class within a model's classification scheme. This is accomplished through a series of optimization steps where the image tensor is adjusted to accentuate the class's activation, followed by upscaling to refine the visualization. The resulting class-activation map provides a visually interpretable representation of the model's classification focus, offering insights into the model's decision-making process. The activation and class-activation visualizations serve as a bridge between the model's abstract understanding and human interpretability, allowing us to validate the model's focus and its alignment with biologically relevant features. These visualizations are not only instrumental in verifying the model's functionality but also provide a foundation for further research into model interpretability and improvement in the context of hierarchical classification in marine biodiversity. These visualizations serve as a crucial interpretative tool, bridging the gap between the model's high-dimensional understanding and

Algorithm 1 Class-Activation Image Generation

```
Initialize image tensor I with random values
for each upscaling step s from 1 to upscaling_steps do
    for each iteration i from 1 to 30 do
        Forward pass: Compute model output for I
        Compute loss L(I, c)
        Backward pass: Compute gradient of L with respect to I
        Update I using gradient ascent
    end for
    if s < upscaling_steps then
        Upscale I to new size
    end if
end for
Save the final image tensor as the class-activation map
```

human interpretable representations. They enable us to assess the model's focus and validate whether it aligns with biologically significant features, thus ensuring its reliability in practical applications. Beyond validation, these visualizations also pave the way for further explorations into enhancing model transparency and effectiveness, particularly in hierarchical classification tasks within the realm of marine biodiversity studies.

3.4. ablation studies

The removal of additional features such as confidence scores and predicted class outputs from YOLO evaluates the model's dependency on auxiliary information. By disabling these input pathways, the model relied solely on raw image data processed through its convolutional layers. Training procedures, hyperparameters, and dataset partitions remained consistent with our main experiments to ensure comparability. Testing the impact of varying the complexity within the branches of our hierarchical model involved experimenting with different configurations of branches by varying the number of layers and neurons within each branch to determine the optimal complexity for achieving the best classification performance at each hierarchical level. Increasing the complexity of processing additional features aims to assess the impact of more sophisticated feature processing on model performance. Additional layers with increased complexity were introduced to evaluate whether more intricate feature processing improves the model's ability to extract relevant information from auxiliary data. Investigating the influence of varying the depth of convolutional layers helps us understand how layer depth affects the model's ability to capture and process features at different scales and complexities. The model was tested with fewer and additional convolutional layers to observe changes in feature extraction capabilities and overall performance. Attention mechanisms are hypothesized to enhance the model's focus on relevant features within data. Removing these mechanisms evaluates their actual impact on model performance. Attention

Table 3: Comparison of Original and Ablated Model Configurations

Ablation	Component	Original Configuration	Modified Configuration
Remove Additional Features	Input Features	Image, Confidence, IoU, Predicted Class	Image only
	BranchCNN Input	Combined features with additional context	Image features only
	Feature Processing	Includes additional feature layers	No additional feature layers
Branch Architecture	BranchCNN Input	Standard complexity	Varied complexity
	Feature Processing	Standard layer configuration	Adjusted layer depth
Increased Features Complexity	Additional Feature Processing	Standard layers	Increased complexity
	Number of Layers	Standard	Additional layers
Convolutional Layer Depth	Convolutional Layers	Standard depth	Adjusted depth
Attention Mechanism	Attention Modules	Channel and Spatial Attention	Removed
	Feature Processing	Includes attention modules	No attention modules

modules (e.g., channel and spatial attention) were removed to observe how the model processes and prioritizes information without these mechanisms.

4. Results and Discussion

Table 4: Performance comparison of instance segmentation models across hierarchical levels for EMVSD. The column on the right represents the baseline F1 scores with no reclassification (manual reclassification using the ontology).

Performance Comparison						
Level	α values					No Reclassification
	0	0.2	0.5	0.8	1.0	
YOLOv8						
Binary	0.6460	0.6511	0.6491	0.6489	0.6462	0.6612
Class	0.8685	0.8687	0.8668	0.8717	0.8694	0.8792
Genus	0.9021	0.9034	0.9008	0.9055	0.9038	0.9131
Species	0.9073	0.9101	0.9076	0.9119	0.9070	0.9185
YOLOv9						
Binary	0.6563	0.6585	0.6559	0.6552	0.6580	0.6634
Class	0.8752	0.8749	0.8789	0.8753	0.8743	0.8836
Genus	0.9083	0.9075	0.9112	0.9070	0.9097	0.9169
Species	0.9106	0.9104	0.9131	0.9076	0.9124	0.9226
RT-DETR						
Binary	**0.4344**	**0.4305**	**0.4219**	**0.4259**	**0.4346**	0.4203
Class	**0.4333**	**0.4312**	**0.4242**	**0.4378**	**0.4445**	0.4253
Genus	**0.3767**	**0.3778**	**0.3700**	**0.3718**	**0.3787**	0.3813
Species	**0.3768**	**0.3809**	**0.3667**	**0.3904**	**0.3886**	0.3540
Mask R-CNN						
Binary	**0.9312**	**0.9368**	**0.9326**	**0.9181**	**0.9230**	0.9171
Class	**0.9604**	**0.9549**	**0.9529**	**0.9441**	**0.9495**	0.9390
Genus	**0.9411**	**0.9432**	**0.9454**	**0.9419**	**0.9414**	0.9250
Species	**0.9423**	**0.9393**	**0.9459**	**0.9385**	**0.9405**	0.9308

This study explores the reclassification of object detection output by integrating a hierarchical reclassification strategy. As demonstrated by our experimental outcomes presented across Tables 4 and 5, our approach provides nuanced insights into the performance of various models in hierarchical classification tasks. Particularly notable are the outcomes from our ablation studies, detailed in Table 5, which include strategic modifications such as increased feature complexity and the removal of attention mechanisms. These modifications have led to performance enhancements over our reclassification framework, illustrating the potential of advanced machine learning techniques to improve the accuracy of hierarchical classification. For example, in EMVSD, the 'Increased Features Complexity' ablation resulted in higher F1 scores across multiple hierarchical levels, although it did not surpass the YOLOv8-Original model. Reclassification improvements were observed with the RT-DETR model. The

Table 5: Impact of ablation studies, showing variations in F1 scores with different model modifications.

Ablation study						
Level	α values					No Reclassification
	0	0.2	0.5	0.8	1	
Attention Removed						
Binary	0.6496	0.6519	0.6467	0.6481	0.6476	0.6612
Class	0.8678	0.8722	0.8663	0.8675	0.8691	0.8792
Genus	0.9013	0.9050	0.8979	0.9033	0.9015	0.9131
Species	0.9061	0.9116	0.9023	0.9109	0.9066	0.9185
Decreased Branch Complexity						
Binary	0.6467	0.6484	0.6484	0.6458	0.6496	0.6612
Class	0.8707	0.8660	0.8652	0.8612	0.8641	0.8792
Genus	0.9040	0.8985	0.8966	0.8951	0.9025	0.9131
Species	0.9046	0.9047	0.9063	0.9018	0.9073	0.9185
Increased Features Complexity						
Binary	0.6537	0.6545	0.6531	0.6578	0.6594	0.6612
Class	0.8668	0.8655	0.8696	0.8669	0.8629	0.8792
Genus	0.9002	0.8988	0.9021	0.9046	0.9058	0.9131
Species	0.9076	0.9055	0.9083	0.9108	0.9075	0.9185
Removed Features						
Binary	0.6314	0.6273	0.6235	0.6292	0.6218	0.6612
Class	0.8269	0.8296	0.8363	0.8302	0.8336	0.8792
Genus	0.8686	0.8742	0.8643	0.8725	0.8699	0.9131
Species	0.8765	0.8817	0.8683	0.8794	0.8807	0.9185

RT-DETR-Reclassified model showed better F1 scores compared to RT-DETR-Original, such as increasing from 0.4203 to 0.4344 in binary classification and from 0.3540 to 0.3768 in species-level classification with $\alpha = 0$. Despite these improvements, the overall performance of the RT-DETR model remains relatively low. Mask R-CNN consistently outperformed other models across all hierarchical levels with our hierarchical reclassification framework. It achieved significant improvements in F1 scores at the binary, class, genus, and species levels compared to the baseline scores without reclassification. This consistent performance indicates that Mask R-CNN, when integrated with our hierarchical framework, can effectively capture and classify complex patterns in marine biodiversity data. Notably, Mask R-CNN also achieved higher accuracies at the higher hierarchical levels (binary and class), which is an interesting find that warrants further investigation. Conversely, YOLOv8 struggled to generalize at higher levels of the hierarchy, an observation that suggests potential limitations in its architecture for such tasks. This disparity in performance highlights the importance of selecting appropriate models based on the specific requirements of hierarchical classification tasks. The model's consistent performance across various values of α attests to its robustness—a crucial trait in real-world applications where environmental conditions are dynamic and unpredictable. This adaptability suggests that our hierarchical reclassification framework can enhance model performance across different scenarios and model architectures. In conclusion, our research demonstrates the potential of hierarchical reclassification frameworks to enhance the performance of instance segmentation models in hierarchical classification tasks. The integration of such frameworks with models like Mask R-CNN shows promise for more accurate and nuanced classifications, which are essential for advancing the field of computer vision and its applications in diverse domains.

4.1. Activation Visualization

Figure 6 illustrates the activations at various levels of the Hierarchical CNN, for all different classes at the species leve. This offers a glimpse into the model's internal workings. By examining these activations, we gain a deeper understanding of the distinctive features the network leverages at each hierarchical level, enhancing our comprehension of its classification decisions and the underlying ecological data representation. We see that for many of the classes. The edges and shape of the segmentation is of what activates the activations for many different channels in our CNN. This is especially the case for the higher levels in the hierarchy like the binary and class levels. Further down the CNN backbone we find that activations are inside the region of the segmentation. Focusing on visual attributes of the object of interest.

4.2. Detailed Activation Analysis for M. *edulis*

Figure 7 provides an in-depth look at the activation patterns and class-activation maps for the *M. Edulis* class, as detailed in Algorithm 1. The visualization in Subfigure (A) demonstrates how the hierarchical layers of the CNN

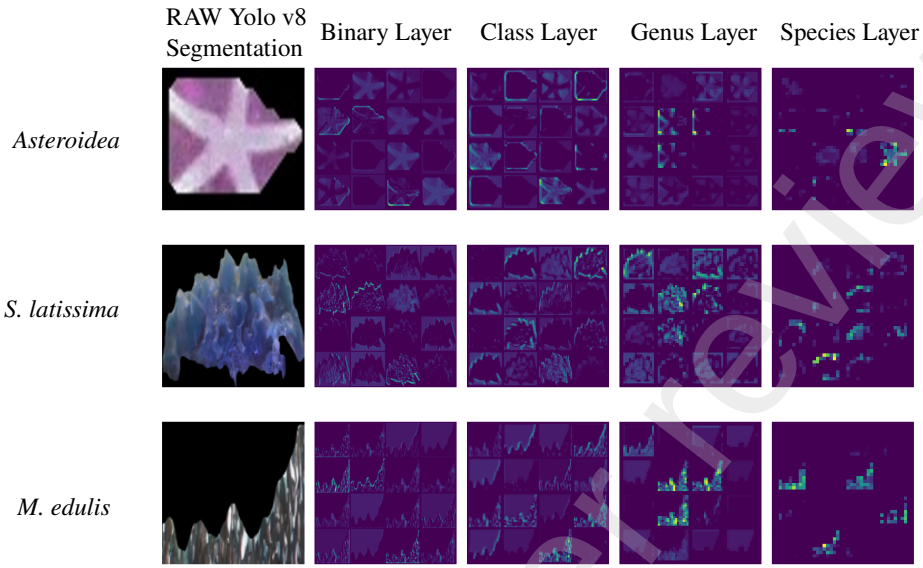


Figure 6: Visualization of activations throughout the Hierarchical CNN, showcasing how the network processes information at different levels of the taxonomy. Each panel corresponds to activations after processing inputs through the network layers designated for binary, class, genus, and species levels, providing insights into the model's focus and feature extraction at each hierarchical stage.

react to segmentation inputs, where each subsequent layer from hierarchical to species shows an increasing focus on specific anatomical features, particularly the white dots. Subfigure (B) showcases the output from our class-activation map generation algorithm, emphasizing the white dots on the shell, which are key distinguishing features for the Mytilus species. These spots are crucial for the network's decision-making process, indicating areas of high relevance for classification. Subfigure (C) provides real-world context by displaying an actual image of the Mytilus Edulis in its natural habitat, underscoring the ecological validity of the model's focus areas. This layered approach not only corroborates the model's accuracy but also enhances our understanding of its capability to adapt its recognition strategies to complex biological patterns, reinforcing the practical applications of our machine learning model in marine biodiversity conservation.

4.3. ROVs for Conservation

The integration of advanced computational techniques with Remotely Operated Vehicles (ROVs) is critical for effective marine biodiversity monitoring. This study contributes to this field by advancing the application of machine learning, specifically through the development of the EMVSD Convolutional Object Detector and a hierarchical CNN for refined classification of marine species. The EMVSD Convolutional Object Detector, coupled with our hierarchical CNN, enhances the classification accuracy of marine species, particularly

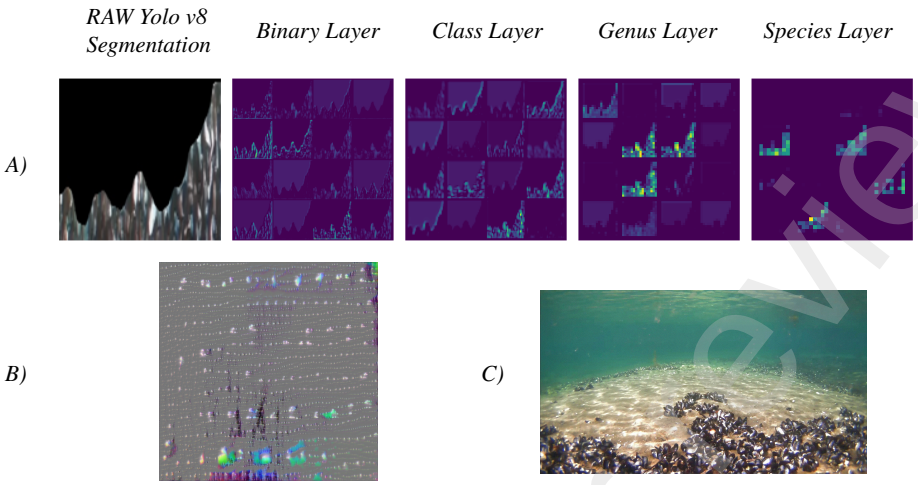


Figure 7: Detailed activation analysis for the Mytilus class, combining several types of visual data to illustrate how the model processes and identifies distinctive features crucial for classification. Subfigure (A) shows the raw YOLO v8 segmentation input from EMDVS-Net and subsequent CNN layer activations, highlighting how different layers respond to the input image. Subfigure (B) displays the class-activation maps, generated using the approach outlined in Algorithm 1, which illustrate areas most influential for classifying Mytilus. Subfigure (C) presents a raw image of Mytilus Edulis from Esefjorden captured by our ROV, showing natural habitat features. Each component provides unique insights into the model's interpretation and validation of visual features specific to *M. Edulis*

those that are visually similar within complex marine ecosystems. Such advancements are not merely technical achievements; they provide vital insights that can inform conservation strategies and ecological understanding. By accurately identifying and classifying marine species, our models facilitate detailed ecological studies and biodiversity assessments. While our primary focus has been on supervised learning models requiring labeled data, exploring unsupervised learning methods remains a promising future direction. These methods can detect patterns and anomalies without the need for pre-categorized data, offering robustness and adaptability to environmental changes, which is crucial for marine ecosystems. The potential of unsupervised methods to enhance model adaptability represents a compelling area for future research [33, 34]. Future work will focus on refining our existing architectural design to improve feature extraction and classification processes, and exploring the potential of unsupervised learning methods. Additionally, enriching the EMVSD dataset through advanced data collection techniques and leveraging multi-modal data integration will be pivotal. Interdisciplinary collaborations will be crucial in developing more comprehensive and effective models for marine biodiversity monitoring. The insights from this study emphasize the importance of continuous research and innovation to overcome the challenges in underwater detection and classification. Developing comprehensive ontologies, integrating multi-modal data, and fostering interdisciplinary collaborations will be essential for advancing the

field of ecological informatics and achieving improvements in marine biodiversity monitoring. By addressing these challenges and exploring new methodologies, our work supports the global commitment to biodiversity conservation and exemplifies the critical role of ecological informatics in sustainable ecosystem management.

5. Conclusion

This study investigates the application of hierarchical reclassification strategies to enhance the performance of object detection models, specifically in the context of marine biodiversity. Through the development and evaluation of the Hierarchical CNN model using the EMVSD dataset, we have demonstrated the potential and challenges of hierarchical classification in improving model accuracy across various taxonomic levels.

Our experimental results indicate that while the hierarchical reclassification framework provided performance enhancements for models like Mask R-CNN and RT-DETR, it did not significantly surpass the baseline YOLOv8 model in most cases. Mask R-CNN consistently outperformed other models across all hierarchical levels, achieving higher accuracies, particularly at the binary and class levels. This observation highlights the robustness and efficacy of Mask R-CNN when integrated with our hierarchical framework, suggesting that it effectively captures and classifies complex patterns in marine biodiversity data.

Conversely, YOLOv8 showed limitations in generalizing at higher hierarchical levels, an area that warrants further investigation. This disparity underscores the importance of selecting appropriate models based on the specific requirements of hierarchical classification tasks and suggests potential architectural limitations in YOLOv8 for such applications.

The ablation studies further illuminated the impact of various modifications on model performance. Strategic changes, such as increased feature complexity and the removal of attention mechanisms, demonstrated notable effects on the hierarchical reclassification framework's efficacy. These findings underscore the value of advanced machine learning techniques in refining hierarchical classification accuracy.

While our study demonstrates that AI can effectively detect and classify marine species, achieving a high level of accuracy still requires substantial human oversight, particularly from marine biologists. The integration of AI can significantly enhance their work by automating parts of the analysis and providing preliminary classifications. However, there are several key areas that require further research and development:

1. **Data Volume and Quality**: One major challenge is the sheer volume of training data required to achieve high accuracy. Our study covers a limited number of species, and scaling this up will demand much more data. Future research should focus on optimizing data collection methods and improving data annotation processes to build larger and more comprehensive datasets.

2. **Architectural Innovations and Modern Techniques**: Exploring new architectures could help reduce the amount of training data needed. Advanced models, such as transformers, Generative Adversarial Networks (GANs), and future versions of YOLO (e.g., YOLOv9, YOLOv10), have shown promise in other domains and could potentially offer improvements in underwater image classification tasks. Investigating these newer architectures and modern techniques could lead to more efficient and effective models.
3. **Unsupervised Methods**: Unsupervised learning methods hold potential for discovering patterns and insights without extensive labeled data. These methods could assist marine biologists by highlighting novel aspects of the data and uncovering complex patterns that may not be immediately apparent. Research into unsupervised learning could significantly enhance the capabilities of AI in marine biodiversity studies. The recent advancements in models like SAM 2 present opportunities for integrating unsupervised learning techniques, offering potential avenues for further exploration.

Our research contributes to the understanding and development of hierarchical classification frameworks, advancing the field of computer vision and its applications in diverse domains, including marine biodiversity. The insights gained from this study highlight the importance of continuous research and innovation in overcoming the challenges of underwater detection and classification.

Despite the limitations observed in performance improvement over the baseline YOLOv8 model with manual reclassification, our work lays a foundation for future studies. It emphasizes the need for further exploration and development in hierarchical classification and advanced machine learning techniques. The ongoing refinement of methodologies and integration of multi-modal data will be crucial for achieving significant advancements in marine biodiversity monitoring and beyond.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the author(s) used ChatGPT in order to assist in writing some sections of the paper and to fix repeated errors in the text. After using this tool, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

References

[1] Bertinetto, L., Mueller, R., Tertikas, K., Samangooei, S., Lord, N.A., 2020. Making better mistakes: Leveraging class hierarchies with deep networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12506–12515.

[2] Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M., 2020. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 .

[3] Boström, C., Baden, S., Bockelmann, A.C., Dromph, K., Fredriksen, S., Gustafsson, C., Krause-Jensen, D., Möller, T., Nielsen, S.L., Olesen, B., Olsen, J., Pihl, L., Rinde, E., 2014. Distribution, structure and function of Nordic eelgrass (Zostera marina) ecosystems: implications for coastal management and conservation. Aquatic Conservation: Marine and Freshwater Ecosystems 24, 410–434. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/aqc.2424, doi:10.1002/aqc.2424. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/aqc.2424.

[4] Chen, Q., Beijbom, O., Chan, S., Bouwmeester, J., Kriegman, D., 2021. A New Deep Learning Engine for CoralNet, in: 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), IEEE, Montreal, BC, Canada. pp. 3686–3695. URL: https://ieeexplore.ieee.org/document/9607450/, doi:10.1109/ICCVW54120.2021.00412.

[5] Crescitelli, A.M., Gansel, L.C., Zhang, H., 2021. NorFisk: fish image dataset from Norwegian fish farms for species recognition using deep neural networks. Modeling, Identification and Control: A Norwegian Research Bulletin 42, 1–16. URL: http://www.mic-journal.no/ABS/MIC-2021-1-1.asp, doi:10.4173/mic.2021.1.1.

[6] Cui, S., Zhou, Y., Wang, Y., Zhai, L., 2020. Fish Detection Using Deep Learning. Applied Computational Intelligence and Soft Computing 2020, e3738108. URL: https://www.hindawi.com/journals/acisc/2020/3738108/, doi:10.1155/2020/3738108. publisher: Hindawi.

[7] Elawady, M., 2015. Sparse Coral Classification Using Deep Convolutional Neural Networks. URL: http://arxiv.org/abs/1511.09067, doi:10.48550/arXiv.1511.09067.

[8] Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A., 2010. The pascal visual object classes (voc) challenge. International journal of computer vision 88, 303–338.

[9] Fayaz, S., Parah, S., Qureshi, G., 2022. Underwater object detection: architectures and algorithms – a comprehensive review. Multimedia Tools and Applications 81. doi:10.1007/s11042-022-12502-1.

[10] Girshick, R., 2015. Fast r-cnn, in: Proceedings of the IEEE international conference on computer vision, pp. 1440–1448.

[11] Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 580–587.

[12] González-Sabbagh, S.P., Robles-Kelly, A., 2023. A Survey on Underwater Computer Vision. ACM Computing Surveys URL: https://dl.acm.org/doi/10.1145/3578516, doi:10.1145/3578516.

[13] Gruber, D.F., Wood, R.J., 2022. Advances and future outlooks in soft robotics for minimally invasive marine biology. Science Robotics 7, eabm6807.

[14] Hamzaoui, M., Ould-Elhassen Aoueileyine, M., Romdhani, L., Boullegue, R., 2023. An Improved Deep Learning Model for Underwater Species Recognition in Aquaculture. Fishes 8, 514. URL: https://www.mdpi.com/2410-3888/8/10/514, doi:10.3390/fishes8100514. number: 10 Publisher: Multidisciplinary Digital Publishing Institute.

[15] Huseklepp, B.S., 2023. Sognefjorden i Vestland fylke. Technical Report 3933. Rådgivende Biologer AS. Bergen. URL: https://www.statsforvalteren.no/siteassets/fm-vestland/miljo-og-klima/verneomrade/marint-vern-sognefjorden/3933-sognefjorden-i-vestland-fylke.-kartlegging-av-marint-naturmangfold-i-grunne-omrader.pdf.

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

[17] Kalhagen, E.S., Olsen, Ø.L., Goodwin, M., Gupta, A., 2022. Hierarchical Object Detection applied to Fish Species: Hierarchical Object Detection of Fish Species | Nordic Machine Intelligence. Nordic Machine Intelligence URL: https://journals.uio.no/NMI/article/view/9452.

[18] Khan, F.F., Li, X., Temple, A.J., Elhoseiny, M., 2023. FishNet: A Large-scale Dataset and Benchmark for Fish Recognition, Detection, and Functional Trait Prediction, pp. 20496–20506. URL: https://openaccess.thecvf.com/content/ICCV2023/html/Khan_FishNet_A_Large-scale_Dataset_and_Benchmark_for_Fish_Recognition_Detection_ICCV_2023_paper.html.

[19] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R., 2023. Segment Anything. URL: http://arxiv.org/abs/2304.02643, doi:10.48550/arXiv.2304.02643. arXiv:2304.02643 [cs].

[20] Knausgård, K.M., Wiklund, A., Sørdalen, T.K., Halvorsen, K.T., Kleiven, A.R., Jiao, L., Goodwin, M., 2022. Temperate fish detection and classification: a deep learning based approach. Applied Intelligence 52, 6988–7001. URL: https://doi.org/10.1007/s10489-020-02154-9, doi:10.1007/s10489-020-02154-9.

[21] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft coco: Common objects in context, in: Computer Vision–ECCV 2014: 13th European Conference, Zurich,

Switzerland, September 6-12, 2014, Proceedings, Part V 13, Springer. pp. 740–755.

[22] Misra, D., 2020. Mish: A self regularized non-monotonic activation function. arXiv:1908.08681.

[23] Neubeck, A., Van Gool, L., 2006. Efficient non-maximum suppression, in: 18th international conference on pattern recognition (ICPR'06), IEEE. pp. 850–855.

[24] Olsvik, E., Trinh, C.M.D., Knausgård, K.M., Wiklund, A., Sørdalen, T.K., Kleiven, A.R., Jiao, L., Goodwin, M., 2019. Biometric Fish Classification of Temperate Species Using Convolutional Neural Network with Squeeze-and-Excitation, in: Wotawa, F., Friedrich, G., Pill, I., Koitz-Hristov, R., Ali, M. (Eds.), Advances and Trends in Artificial Intelligence. From Theory to Practice, Springer International Publishing, Cham. pp. 89–101. doi:10.1007/978-3-030-22999-3_9.

[25] Patro, K.S.K., Yadav, V.K., Bharti, V.S., Sharma, A., Sharma, A., 2023. Fish Detection in Underwater Environments Using Deep Learning. National Academy Science Letters 46, 407–412. URL: https://doi.org/10.1007/s40009-023-01265-4, doi:10.1007/s40009-023-01265-4.

[26] Postlethwaite, V.R., McGowan, A.E., Kohfeld, K.E., Robinson, C.L.K., Pellatt, M.G., 2018. Low blue carbon storage in eelgrass (Zostera marina) meadows on the Pacific Coast of Canada. PLOS ONE 13, e0198348. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0198348, doi:10.1371/journal.pone.0198348. publisher: Public Library of Science.

[27] Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., et al., 2024. Sam 2: Segment anything in images and videos. arXiv preprint arXiv:2408.00714 .

[28] 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), pp. 779–788.

[29] Redmon, J., Farhadi, A., 2017. Yolo9000: Better, faster, stronger, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7263–7271.

[30] Redmon, J., Farhadi, A., 2018. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 .

[31] Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems 28.

[32] Röhr, M.E., Boström, C., Canal-Vergés, P., Holmer, M., 2016. Blue carbon stocks in baltic sea eelgrass (*Zostera marina*) meadows. Biogeosciences 13, 6139–6153. URL: https://bg.copernicus.org/articles/13/6139/2016/, doi:10.5194/bg-13-6139-2016.

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

[34] Sonnewald, M., Dutkiewicz, S., Hill, C., Forget, G., 2020. Elucidating ecological complexity: Unsupervised learning determines global marine ecoprovinces. Science Advances 6, eaay4740. URL: https://www.science.org/doi/abs/10.1126/sciadv.aay4740, doi:10.1126/sciadv.aay4740, arXiv:https://www.science.org/doi/pdf/10.1126/sciadv.aay4740.

[35] Villar, V.A., de Soto, K., Gagliano, A., 2023. Hierarchical Cross-entropy Loss for Classification of Astrophysical Transients. URL: http://arxiv.org/abs/2312.02266, doi:10.48550/arXiv.2312.02266. arXiv:2312.02266 [astro-ph].

[36] Wang, C.Y., Yeh, I.H., Liao, H.Y.M., 2024. Yolov9: Learning what you want to learn using programmable gradient information. arXiv:2402.13616.

[37] Willners, J.S., Carlucho, I., Katagiri, S., Lemoine, C., Roe, J., Stephens, D., Łuczyński, T., Xu, S., Carreno, Y., Pairet, È., et al., 2021. From market-ready rovs to low-cost auvs, in: OCEANS 2021: San Diego–Porto, IEEE. pp. 1–7.

[38] Woo, S., Park, J., Lee, J.Y., Kweon, I.S., 2018. Cbam: Convolutional block attention module, in: Proceedings of the European Conference on Computer Vision (ECCV), pp. 3–19.

[39] Xu, W., Matzner, S., 2018. Underwater Fish Detection Using Deep Learning for Water Power Applications, in: 2018 International Conference on Computational Science and Computational Intelligence (CSCI), pp. 313–318. doi:10.1109/CSCI46756.2018.00067.

[40] Yang, D., Solihin, M.I., Zhao, Y., Cai, B., Chen, C., Riyadi, S., 2024. A YOLO Benchmarking Experiment For Maritime Object Detection In Foggy Environments, in: 2024 IEEE 14th Symposium on Computer

Applications & Industrial Electronics (ISCAIE), pp. 354–359. URL: https://ieeexplore.ieee.org/abstract/document/10576412, doi:10.1109/ISCAIE61308.2024.10576412. iSSN: 2836-4317.

[41] Zhang, X., Huang, B., Chen, G., Radenkovic, M., Hou, G., 2023. Wild-FishNet: Open Set Wild Fish Recognition Deep Neural Network With Fusion Activation Pattern. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 16, 7303–7314. URL: https://ieeexplore.ieee.org/abstract/document/10197176, doi:10.1109/JSTARS.2023.3299703. conference Name: IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing.

[42] Zhu, X., Bain, M., 2017. B-cnn: Branch convolutional neural network for hierarchical classification. arXiv:1709.09890.