Detection of Corals, Seagrass, and Seaweeds Using YOLOv9 Instance Segmentation with Image Augmentation

Ken D. Gorro[1,2,*], Anthony S. Ilano[1,3], Lawrence P. Roble[1], Rue Nicole R. Santillan[1], Joseph C. Pepito[1], Elmo B. Ranolo[1], Kim D. Gorro[4], Apple Jane M. Gorro[1], Moustafa F. Ali[5], Archival J. Sebial[5], and Jezreel N. Buot[1]

[1] College of Technology, Cebu Technological University, Philippines
[2] Center for Artificial Intelligence, Cloud Computing, and Big Data, Cebu Technological University, Philippines
[3] Integrated Coastal and Marine Resources Management Center, Cebu Technological University, Philippines
[4] Department of Information Technology, Abuyog Community College, Leyte, Philippines
[5] Department of Computer, Information Sciences and Mathematics, University of San Carlos, Philippines
Email: ken.gorro@ctu.edu.ph (K.D.G.); anthony.ilano@ctu.edu.ph (A.S.I.); lawrence7roble@gmail.com (L.P.R.); ruesantillan123@gmail.com (R.N.R.S.); joseph.pepito@ctu.edu.ph (J.C.P.); elmo.ranolo@ctu.edu.ph (E.B.R.); kimgors@gmail.com (K.D.G.); applemanipis@gmail.com (A.J.M.G.); moustafing@gmail.com (M.F.A.); ajsebial@usc.edu.ph (A.J.S.); jez.buot@ctu.edu.ph (J.N.B.)
*Corresponding author

Abstract—**This study investigates the extent to which the YOLOv9e-instance segmentation model classifies and detects different types of marine objects, such as corals, marine life, seagrass, and seaweed. This study utilizes image augmentation techniques to improve the detection and classification of objects using YOLOv9. The study emphasizes the need to examine the distribution of classes within the dataset, as class imbalances can have a major impact on the model's performance. Throughout the training, the model showed a constant decrease in loss functions such as box loss, segmentation loss, and classification loss, demonstrating effective learning and generalization. The precision and recall metrics improved significantly, with a mean Average Precision (mAP) of 0.883 at an Intersection-over-Union (IoU) threshold of 0.5, validating the model's high accuracy across classes. The F1-Confidence Curve study yielded an overall F1 score of 0.84 at a confidence threshold of 0.534, highlighting the model's robustness in achieving a balance between precision and recall. The results suggest that while the model excels in detecting corals, seagrass, and seaweed, it faces challenges in accurately identifying marine life, pointing to the need for additional refinement to address class imbalances.**

Keywords—**YOLOv9, instance segmentation**

I. Introduction

The ocean, covering 70% of the Earth's surface, is a critical ecosystem that plays a vital role in regulating geophysical processes essential for life. Marine scientists face significant challenges in exploring and understanding this vast and complex environment. With technological advancements, tools such as underwater cameras have been developed to assist in this exploration. These tools enable the real-time monitoring of underwater resources, which is crucial for sustainable management and conservation efforts.

In recent years, the integration of sensors has facilitated underwater monitoring, particularly in detecting marine life. Over the past two decades, these sensors have evolved, and with the rise of computer vision, there is a growing interest in incorporating these technologies into underwater video analysis. Marine life detection using computer vision has been challenging due to several environmental factors, such as fluorescence images and similar features of corals of seaweeds and seagrass. This study explores YOLOv9 with instance segmentation and image augmentation techniques in classifying and detecting corals, seaweeds, and seagrass.

II. Related Works

A. Image Classification

Australia launched the Collaborative and Automated Tools for the Analysis of Marine Imagery and Video (CATAMI) [1] initiative in 2010 with the goal of implementing a new categorization scheme that guarantees marine species shown in underwater photos are assigned consistent names. However, the data analysis is

Manuscript received September 2, 2024; revised September 20, 2024; accepted November 11, 2024; published May 20, 2025.

not genuinely automated by this approach. All it does is simplify things by making manual data entry easier and providing a standardized procedure for assigning ground truth labels. Previous studies have demonstrated the potential of computer vision-based methods for the automated annotation of benthic data [2–4]. However, considering the variables such as shifting water turbidity, imprecise class distinctions, and deterioration of undersea color, this is an arduous process. A. Gómez-Ríos *et al.* address the challenges of classifying coral species from underwater texture images, including issues like species similarity and imbalanced datasets [5]. J. Borbon *et al.* uses Convolutional Neural Networks (CNNs) to categorize corals as healthy, dead, or bleached, highlighting the significance of precise image classification in the early identification of coral bleaching and degradation. Their results show that classification accuracy is much increased with larger datasets. Our research aims to improve image classification methods for seaweed, seagrass, and corals by building on this methodology. Our goal is to improve the accuracy of recognizing and monitoring the condition of these marine habitats, which is essential for successful conservation efforts, by utilizing sophisticated CNN architectures and extensive datasets [6].

In the following sentence, we will look at the techniques and software tools that have been created for marine creature annotation and segmentation. One well-known AI-based tool for producing manual and aided point-based annotations is the online platform CoralNet [7]. After enough photos have been labeled directly in the web browser, CoralNet trains a classifier to assist in labeling the remaining images. Friedman [8] describes Squidle+ as a cloud-based platform for georeferencing and annotating underwater visual data. It can handle images, videos, and orthomosaics (as a set of tiles) with remarkable versatility. Squidle and TagLab take distinct tacks. First, TagLab identifies areas, while Squidle+ performs point-based annotation. Point-based data is typically insufficient to pinpoint the demographic forces driving change in coral communities [9]. Second, Squidle+'s AI-assisted portion uses an active learning strategy, asking the user for further input to enhance the system's categorization performance. With its flexible working pipeline and supportive tools, TagLab enables direct modification of automated predictions. In terms of tracing, DeepSegment [10] uses a superpixel-based picture segmentation technique. To get great precision, parameters for every colony need to be carefully adjusted. DeepSegment divides the picture into tiny subregions. It takes a lot of time for the user to add semantics individually to each one. Alonso *et al.* [11] describe CoralSeg as an additional new approach that uses superpixels in a hierarchical fashion to broaden the sparse labeling and provide a coherent semantic segmentation. Repeatable surveys of benthic communities have been successfully conducted using this approach [12]. The Geodesic star convexity method [13] is modified for coral segmentation in CoralMe [14].

Strengths:

1. Standardized Marine Classification: The CATAMI initiative [1] standardized marine species naming in underwater imagery, streamlining manual data entry and ensuring consistency across studies.
2. Advances in Automated Annotation: Computer vision-based methods, especially CNNs, have shown promise in improving the accuracy of marine species classification, which is crucial for monitoring and conservation, with larger datasets enhancing results [2–5].
3. AI-Assisted Annotation Tools: Tools like CoralNet and Squidle+ enhance annotation efficiency with AI-assisted labeling and versatile capabilities for various visual data, improving performance over time [7, 8].

Weaknesses:

1. Limited Automation in CATAMI: While CATAMI standardizes classification, it lacks full automation, limiting its scalability for large datasets [1].
2. Challenges in Underwater Classification: Underwater image classification is hindered by factors like water turbidity and color deterioration, reducing accuracy, especially with imbalanced datasets [2–5].
3. Limitations of Point-Based Annotation: Point-based tools like Squidle+ may not capture enough detail for demographic analysis, and manual refinement in tools like DeepSegment is time-consuming [9, 10].
4. Complexity in Segmentation Methods: Segmentation methods like DeepSegment and CoralSeg require extensive manual input, limiting scalability, and fully automated solutions like CoralMe still face challenges [10, 11, 13, 14].

B. Image Processing

Due to the unique nature of the underwater imaging environment, underwater image processing is critical for detecting marine biological objects. It is commonly recognized that light scattering and absorption are the main causes of blur and color distortion in most underwater photos. Light in water scatters as it comes into contact with suspended particles; this process is known as forward scattering, and it distorts images. Backscattering lowers contrast in the picture and produces hazy blur [15]. The underwater environment presents two challenges: (i) wavelength-selective attenuation lowers and distorts the contrast between objects and backgrounds [16], and (ii) polarization and atomization processes might result in false positives [17]. These aforementioned challenges have led to a markedly subpar performance in the majority of underwater object-detecting cases [18]. Typically, object detection and image preprocessing are the two main technologies used in underwater object detection based on optical images. There are two categories of underwater picture preprocessing techniques: techniques for enhancing underwater images and techniques for restoring underwater images [19]. The underwater image enhancement approach does not require an algorithm to be implemented in order to take into account the particular physical imaging process; instead, it simply employs a computer graphics method to increase picture clarity. In order to produce the restored underwater picture, the underwater image restoration algorithms seek to solve the underwater imaging model inversely [20]. The underwater picture degradation model is a prerequisite for the algorithm's operation. Conventional object detection algorithms and convolution neural network-based object detection algorithms are the two main categories into which marine biometric detection techniques fall. Three steps make up the standard object identification process: preprocessing the picture, extracting features from the image, and classifying the image. The intricacy of the undersea environment makes marine biological identification more challenging in real-world applications [21]. Blanchet *et al.* [22] employed Histogram Equalization to enhance undersea photos. Coral images using color schemes including RGB, LAB, and HSV have been used by Beijbom *et al.* [23] for upgrades such as color channel stretching and intensity stretching. In order to improve contrast, Eduardo *et al.* [24] measured the coral image's pixel intensity values throughout a range using the normalizing procedure. Normalization has been utilized by Mohammad *et al.* [25] to eliminate the effect of global light in coral photos. Contrast Limited Adaptive Histogram Specification (CLAHS) is a significant enhancement approach that yields superior picture enhancement outcomes, according to Shihavuddin *et al.* [26].

Strengths:

1. Targeted Image Enhancement: Techniques like Histogram Equalization and CLAHS significantly improve underwater image clarity and contrast, aiding in the detection of marine life despite challenges like light scattering and absorption [15, 26].
2. Restoration Algorithms: Advanced restoration techniques reconstruct images by addressing the underwater imaging model, providing a more accurate representation of the underwater environment for reliable detection [20].
3. Versatile Color Adjustment: Using color schemes (RGB, LAB, HSV) and methods like color channel stretching enhances underwater images, making classification more accurate [23].
4. Normalization for Consistency: Normalization techniques effectively counteract global light variations, ensuring more consistent detection across different underwater conditions [24, 25].

Weaknesses:

1. Complex Real-World Challenges: The underwater environment's complexity, including factors like wavelength-selective attenuation and polarization, leads to false positives and hinders accurate detection [16, 17].
2. Limited Enhancement Impact: While enhancement techniques improve clarity, they don't fully address the physical causes of image degradation, limiting their effectiveness in challenging conditions [19, 26].
3. Preprocessing Dependence: Conventional object detection heavily relies on preprocessing, which may not be sufficient to overcome underwater imaging challenges [21].
4. Restoration Complexity: Effective implementation of restoration algorithms requires accurate modeling of the underwater environment, which is complex and prone to errors, affecting detection accuracy [20].

C. Object Detection

The advancement of object identification technology based on Convolutional Neural Networks (CNNs) has lately accelerated, outperforming classical approaches in many detection applications [27]. Object identification approaches based on CNN and the advancement of deep learning, have significantly increased object detection accuracy. CNN-based algorithms are broadly classified into two categories: two-stage target detection and one-stage target detection [28]. Two-stage target detection algorithms divide the detection issue into two phases: the first phase creates candidate regions, and the second phase classifies and refines those regions. These algorithms are Region-based Convolutional Neural Network (R-CNN) [29], Fast R-CNN [30], Faster R-CNN [31], and Cascade R-CNN [32]. Two-stage detection algorithms produce the most reliable detection results. Although they are accurate, the forecast speed is extremely sluggish, and the hardware requirements are enormous, making it impossible to satisfy real-time needs. One-stage object detection methods detect and classify concurrently and immediately output the classification probability and target location coordinate values. Typical algorithm models, such as the YOLO series [33–36], Single Shot MultiBox Detector (SSD) [37], RetinaNet [38], FreeAnchor [39], Feature Selective Anchor-Free (FSAF) [40], and Fully Convolutional One-Stage Object Detection (FCOS), may be somewhat less accurate than the two-stage model, but they have more real-time prediction capability. As a result, the most significant benefit of this type of network model is its high speed, even though its accuracy is somewhat lower than that of a two-stage detector. Yang *et al.* [42] employed two exemplary target identification algorithms (YOLOv3 and Fast R-CNN) to detect underwater objects. This can represent two primary types of object detection. Song *et al.* [43] suggested a technique for automatically detecting underwater objects in real time using an enhanced convolutional neural network.

Strengths:

1. High Accuracy: CNN-based object detection methods, especially two-stage detectors like R-CNN and its variants, offer high accuracy in identifying and classifying objects, making them reliable for detailed tasks [29–32].
2. Real-Time Detection: One-stage detectors like YOLO and SSD are designed for real-time applications, providing fast processing speeds that are crucial for scenarios requiring immediate analysis [33–37].
3. Broad Applicability: CNN-based object detection models can be adapted across various domains, including underwater object detection, where they have shown effectiveness despite challenging conditions [42, 43].

Weaknesses:

1. High Computational Demand: Two-stage models, while accurate, require significant computational resources, which limits their usability in real-time applications or on hardware with limited processing power [29–32].
2. Trade-off Between Speed and Accuracy: One-stage models, although faster, often sacrifice some accuracy compared to two-stage models, which may be a drawback in situations where precision is critical [33–36].
3. Performance Variability in Challenging Environments: Even with advanced CNN architectures, object detection can struggle in complex environments like underwater settings, where factors such as lighting, turbidity, and object occlusion can degrade performance [42, 43].

D. Deep Convolutional Neural Networks (DCNN)

A Deep Convolutional Neural Network (DCNN) model has been proposed in some research, and it is specifically made to identify gastrointestinal disorders from endoscopic pictures. It uses a DCNN model that is tuned for accuracy and performance to address the problems of human error, time-consuming diagnosis, and inter-laboratory inconsistencies. By utilizing sophisticated methods such as convolutional layers and different picture resolutions, the model outperforms previous approaches in terms of specificity, recall, and AUROC. In medical diagnostics, the study highlights the importance of precise and automated analysis [44]. Additionally, a paper presents a DCNN architecture for the multi-class categorization of lung illnesses, such as COVID-19, lung opacity, pneumonia, lung cancer, and tuberculosis, utilizing pictures from Chest X-Rays (CXRs). The model achieves state-of-the-art classification performance with 99.82% accuracy in detecting diseases and 98.75% accuracy in multi-class classification by implementing Grid Search Optimization (GSO). Large datasets are easily handled by the model, demonstrating the potential of DCNNs in medical diagnostics and enabling quicker and more accurate diagnosis of a variety of lung disorders [45].

Strengths:

1. High Classification Accuracy: The two studies show that DCNN designs can achieve high classification accuracy. While the study obtains nearly perfect classification accuracy for numerous lung illnesses [45], with 99.82% accuracy in disease detection and 98.75% accuracy in multi-class classification, it reports good specificity and AUROC for detecting gastrointestinal problems [44]. These findings highlight the DCNN's potential to enhance diagnosis accuracy in a range of imaging-related medical applications.
2. Optimization Techniques: The application of several convolutional layers [44] and grid search optimization [45] demonstrates how well-calibrated architectures can significantly enhance the model's performance, making DCNNs dependable tools for challenging image-based tasks.
3. Efficiency and Automation: Faster processing times are made possible by the automated nature of DCNN models, which lessens the workload for medical personnel and increases the scalability of diagnostic systems. Both papers highlight how DCNNs can eliminate time-consuming manual processes.

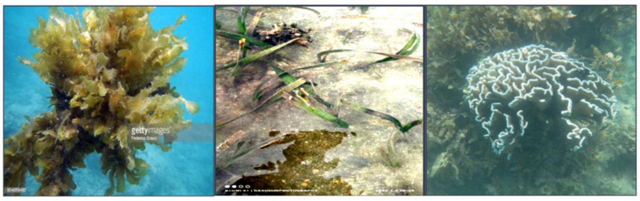
Weaknesses:

1. Class Imbalance Sensitivity: The DCNN models in both studies would have trouble addressing underrepresented classes. The model's performance in these categories may be impacted by the less common gastrointestinal disorders found [44]. Similarly, when dealing with a class distribution that is biased toward more prevalent diseases, their methodology might encounter challenges [45].
2. Computational Demands: Deep Convolutional Neural Network (DCNN) models require a lot of computation, particularly those with big architectures. This could restrict their use in contexts with limited resources, including clinics with less sophisticated computer infrastructure, or limit their real-time applicability.

III. METHODOLOGY

A. Dataset Preparation

The dataset utilized in this research comprises high-resolution images of marine ecosystems, specifically focusing on corals, seagrass, and seaweeds. These images are sourced from various underwater photography collections, marine biology research institutions, and open-source marine life databases. The dataset encompasses a diverse range of environments, including coral reefs, seagrass meadows, and coastal areas where seaweeds are prevalent. The images were captured under different lighting conditions, water clarity levels, and depths to ensure the dataset represents a wide array of real-world scenarios. This variability is crucial for training a robust YOLOv9 model capable of accurately classifying and segmenting corals, seagrass, and seaweeds in diverse marine settings. Fig. 1 depicts representative photographs from the dataset, including (a) seaweed, (b) seagrass, and (c) coral.



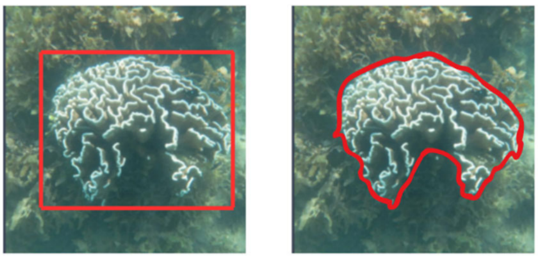
(a) (b) (c)

Fig. 1. Sample dataset: (a) Seaweeds, (b) Seagrass, (c) Coral.

1) Annotation and labeling

Annotation refers to the process of marking up or adding metadata to raw data—in this case, images—to provide meaningful information that machines can interpret. For instance, in the context of this research, annotation involves outlining or marking specific regions of an image that correspond to different marine organisms such as corals, seagrass, and seaweeds.

Fig. 2 shows that simple rectangular rectangles are drawn around the objects of interest to form bounding boxes. However, more detailed annotations are required for complex structures such as seagrass and corals. Polygon annotations are used to correctly depict the borders of objects with irregular shapes by drawing a succession of connected lines to construct a closed shape around them. This work uses polygon annotations to precisely identify the edges and contours of corals, seagrass, and seaweed.



(a) (b)

Fig. 2. The difference between (a) bounding boxing and (b) polygon annotation.

Polygon annotation is vital for this type of research because it accurately portrays the complicated and asymmetrical geometries of seaweeds, seagrass, and corals, which are difficult to depict with basic geometric shapes such as bounding boxes. These aquatic organisms usually have intricate structures that need precise boundary identification, such as coral branching or thin blades of seagrass. Polygon annotation carefully monitors the outlines of these species to offer a precise representation of their morphologies, allowing the instance segmentation model to learn and recognize them more accurately.

2) Experimental augmentation

Data augmentation is a technique used in machine learning and computer vision to artificially expand the size of a training dataset by applying various transformations to the original data. In this study, augmentations such as 90° rotation (clockwise, counter-clockwise, and upside down), rotation between −30° and +30°, and saturation adjustments between −50% and +50% are used to improve the robustness and generalization of the YOLOv9 model for classifying and segmenting corals, seagrass, and seaweeds. The graphics in Fig. 3 below show where these augmentations have been applied.

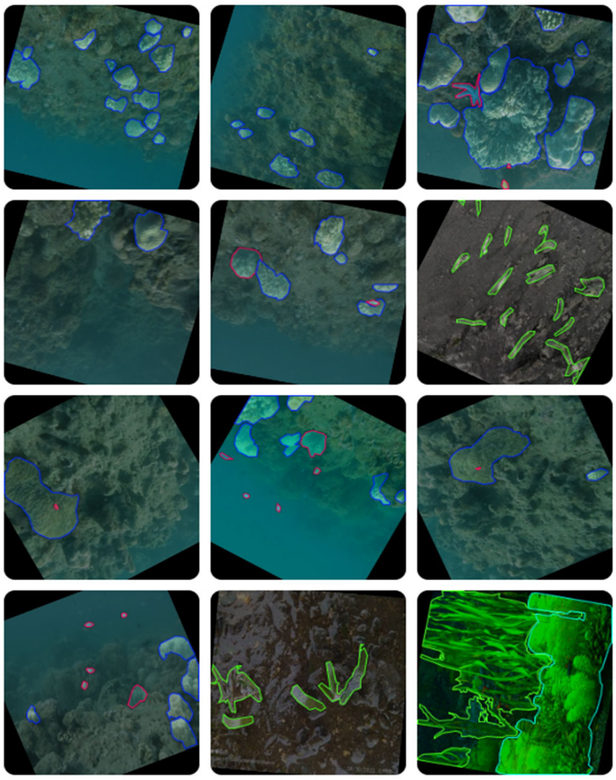


Fig. 3. Dataset with applied augmentation.

These transformations help the model learn to recognize objects under different orientations and lighting conditions, which is crucial for underwater images where the environment can vary significantly. Additionally, augmentation helps prevent overfitting by exposing the model to a broader range of variations within the dataset, ensuring that it performs well on unseen data. This step is particularly important in ecological research, where acquiring a large, diverse dataset can be challenging, making augmentation a key strategy for improving model performance.

Finally, researchers employed synthetic instance mask generation in addition to conventional data augmentation. This method combines pre-existing images with augmented masks for minority classes to produce fresh training examples using methods like CutMix and mosaic augmentation. Instead of depending just on over-sampling methods that could cause overfitting, this produced a more balanced dataset.

B. YOLO-seg Model Selection

Comparing the models' performance metrics (like mAP), computational efficiency, and resistance to class imbalances is crucial to assessing how well YOLOv9e performs in comparison to other cutting-edge instance segmentation models like YOLOv5 and YOLOv8. A thorough grasp of YOLOv9e's advantages and disadvantages in several applications, such as the categorization of seaweed, seagrass, and corals in maritime habitats, is given by this comparison.

1) YOLOv5-seg model

YOLOv5, created by Ultralytics, has strong instance segmentation capabilities because of its ProtoNet architecture, which combines the object detection head with an additional network. The five YOLOv5 variants—nano (n), small (s), medium (m), large (l), and extra-large (x)—provide versatility in a range of situations by striking a balance between computing expense and accuracy [46–48]. Table I presents an overview of YOLOv5's performance on the COCO segmentation test, highlighting its improvement from the smaller to the bigger variations. YOLOv5x-seg, for example, uses more computing resources but is more effective for large-scale detection, achieving mAP^{box} of 50.7 and mAP^{mask} of 41.4 with 88.8 million parameters.

TABLE I. YOLOV5 ON COCO SEGMENTATION TASK [49]

Model	$mAP^{box50-95}$	$mAP^{mask50-95}$	params (M)
YOLOv5n-seg	27.6	23.4	2.0
YOLOv5s-seg	37.0	31.7	7.6
YOLOv5m-seg	45.0	37.1	22.0
YOLOv5l-seg	49.0	39.9	47.9
YOLOv5x-seg	50.7	41.4	88.8

2) YOLOv8-seg model

YOLOv8, which was released in 2023, is a significant improvement over YOLOv5, improving its detection and instance segmentation capabilities. Its flexibility-focused design makes it particularly suitable for a wider range of jobs. YOLOv8 performs better than YOLOv5 in a number of areas, including higher mAP values and computational efficiency. The performance of YOLOv8 at different sizes is shown in Table II. In contrast to YOLOv5x-seg, YOLOv8x-seg achieves mAP^{box} of 53.4 and mAP^{mask} of 43.4 while keeping a comparatively smaller parameter count of 71.8 million. This shows enhanced performance without a noticeably higher computing requirement.

TABLE II. YOLOV8 ON COCO SEGMENTATION TASK

Model	$mAP^{box50-95}$	$mAP^{mask50-95}$	params (M)
YOLOv8n-seg	47.1	40.2	25.9
YOLOv8s-seg	46.6	41.4	26.7
YOLOv8m-seg	47.2	41.5	26.8
YOLOv8l-seg	49.7	42.6	27.9
YOLOv8x-seg	53.4	43.4	88.8

3) YOLOv9-seg model

YOLOv9, created by Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao, improves upon its predecessors in terms of accuracy and processing efficiency by introducing more advanced features for object detection and instance segmentation [50]. Because of its sophisticated architecture, segmentation tasks are significantly improved, particularly for complicated and intricate structures like those seen in marine habitats. With a mAP^{box} of 55.1 and a mAP^{mask} of 44.3, the YOLOv9e-seg version shows remarkable accuracy in the COCO segmentation problem. For identifying corals, seagrass, and seaweeds in this work, where segmentation precision is critical, the model with its better precision—despite having 60.5 million parameters—makes it computationally more demanding than YOLOv8.

As demonstrated by its mAP^{box} of 55.1 and mAP^{mask} of 44.3 on the COCO segmentation problem, the YOLOv9e-seg model has been selected for this research because of its higher performance in instance segmentation tasks. YOLOv9e-seg exhibits superior segmentation accuracy in comparison to other cutting-edge models like YOLOv5 and YOLOv8, which makes it particularly well-suited for intricate tasks like classifying corals, seagrass, and seaweeds. In contexts where complex item detection is required, the precision and durability of the model are essential for attaining successful classification despite its higher computing complexity and parameter count of 60.5 million. For applications where segmentation precision surpasses the trade-offs in computational demand, YOLOv9e-seg is the best option (Table III).

TABLE III. YOLOV9 ON COCO SEGMENTATION TASK [51]

Model	$mAP^{box50-95}$	$mAP^{mask50-95}$	params (M)
YOLOv9c-seg	52.4	42.2	27.9
YOLOv9e-seg	55.1	44.3	60.5

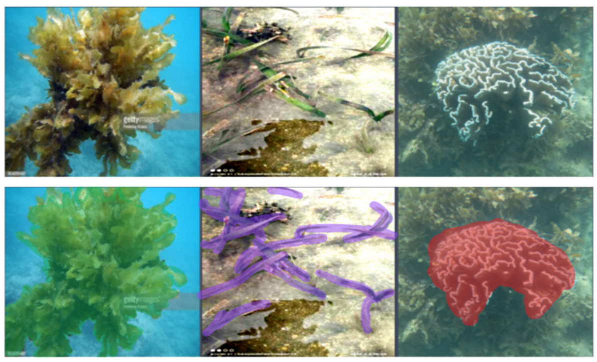


Fig. 4. Instance segmentation sample.

C. Instance Segmentation

Instance segmentation (Fig. 4) extends beyond object detection by identifying the location of objects within an image and delineating their precise shapes. This is achieved by generating masks, or contours, with associated class labels and confidence scores. In the precise classification of marine organisms, instance segmentation is crucial for accurately localizing and identifying corals, seagrass, and seaweeds, enhancing the accuracy and effectiveness of monitoring and preserving these essential marine ecosystems [52, 53]. This method, which uses bounding boxes, purely differs from the study of Elmo *et al.* [54, 55], which uses bounding boxes and focuses on object detection.

D. Evaluation Metrics

Evaluation Metrics are quantitative measures used to assess the performance of a machine learning model. They provide insights into how well the model predicts and generalizes to new, unseen data. In the context of instance segmentation, evaluation metrics such as mean Average Precision (mAP), F1-Score, and accuracy are commonly used to gauge the model's effectiveness in identifying and correctly classifying objects within an image, as well as segmenting them at the pixel level. For this research, evaluating the model using these metrics is essential, as it allows for a precise measurement of how accurately the YOLOv9 instance segmentation models detect and segment corals, seagrass, and seaweeds.

1) Precision

Precision evaluates the accuracy of the model's positive predictions, determining the proportion of correctly identified positives out of all instances that the model classified as positive. In the context of this research, precision helps to ensure that the YOLOv9-seg model accurately classifies corals, seagrass, and seaweeds without incorrectly labeling non-target objects or backgrounds as these classes. High precision is particularly critical in scenarios where false positives can have significant negative consequences, such as incorrectly identifying marine debris as coral, which could lead to inappropriate conservation measures.

$$Precision = \frac{True\ Positive}{(True\ Positive + False\ Positive)} \quad (1)$$

2) Recall

Recall measures the model's ability to identify all actual positive instances, indicating how effectively the YOLOv9-seg model detects corals, seagrass, and seaweeds that are present in the dataset. In this research, recall is crucial for assessing how well the model captures the full range of marine life within an image. A high recall ensures that the model minimizes the number of missed detections (false negatives), which is essential for comprehensive monitoring and analysis of marine ecosystems, where missing out on identifying certain species could lead to incomplete data and misinformed decisions.

$$Recall = \frac{True\ Positive}{(True\ Positive + False\ Negative)} \quad (2)$$

3) F1-Score

The F1-Score provides a balanced measure of the model's precision and recall by computing their harmonic mean. This metric is particularly useful in scenarios where there is an imbalance between the classes or when a single metric (either precision or recall) might not fully capture the model's performance. In this research, the F1-Score is essential for evaluating the YOLOv9-seg model's overall effectiveness in classifying and segmenting corals, seagrass, and seaweeds. By balancing precision and recall, the F1-Score ensures that the model is not only accurate in its positive predictions but also effective in detecting as many relevant instances as possible, leading to a more reliable and robust model for ecological studies.

$$F1 = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \quad (3)$$

By the completion of the study, the Python code for this research has been uploaded to GitHub, The following is the link to the GitHub repository: https://github.com/law4percent/detection-of-corals-seagrass-seaweeds-yolov9e-seg

4) mean Average Precision (mAP)

mean Average Precision (mAP) is a comprehensive metric that evaluates the model's overall performance by averaging the precision across different recall levels for each class and then averaging these values across all classes. In the YOLOv9-seg model used for this research, mAP serves as a key metric for determining how well the model balances precision and recall across the different classes of corals, seagrass, and seaweeds. By considering both precision and recall at various thresholds, mAP provides a holistic view of the model's accuracy and its ability to correctly segment and classify marine life across different scenarios. This makes mAP an indispensable metric for comparing the performance of different models (like YOLOv9c-seg and YOLOv9e-seg) and selecting the most suitable one for the specific needs of this research.

$$mAP = \frac{1}{k}\sum_{i=1}^{k} APi \quad (4)$$

IV. RESULTS AND DISCUSSION

A. Model Performance Metrics

The graph of instances is presented in Fig. 5, is created as soon as the model training starts. This graph depicts the distribution of different classes in the dataset, including corals, marine life, seagrass, and seaweed. In this particular graph, it is clear that there are substantially more coral examples than other types. Such information is critical because it allows researchers to understand the balance or imbalance of the dataset, which can have a direct impact on the model's performance. Additionally, it is crucial to analyze the instance graph for a number of reasons. In the first place, it makes it possible to detect class imbalances, which may result in inaccurate model predictions. For example, the model may become biased towards correctly identifying corals while doing poorly on the less represented classes if it is trained on a dataset that has a higher proportion of coral instances than examples of marine life or seaweed. Early detection of these imbalances allows researchers to address them with strategies like class weight adjustments during training or data augmentation for underrepresented classes. Additionally, the graph helps in evaluating the effectiveness of the annotation process. If certain classes have significantly fewer instances, it might indicate a need for more targeted data collection or annotation efforts to ensure that the model has enough examples to learn from.

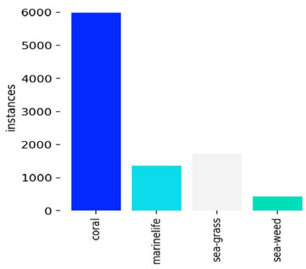


Fig. 5. Graph of instances.

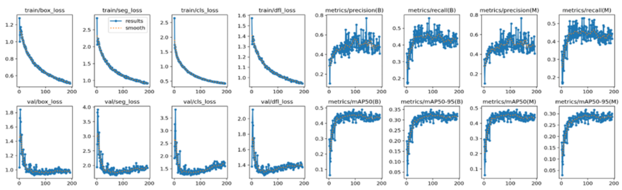


Fig. 6. Result graph for YOLOv9e-seg model.

The training result graphs provide valuable insights into the performance of the YOLOv9 model across various metrics and loss functions. The overall performance of the YOLOv9 model, as indicated by the training graphs, demonstrates a successful training process with well-tuned hyperparameters. The box loss, segmentation loss, and classification loss steadily decreased during training, reflecting the model's improved ability to accurately predict bounding boxes, segment objects, and classify instances. Additionally, the validation losses show a consistent decline, affirming that the model is generalizing well beyond the training data. The precision and recall metrics further support the model's effectiveness. The precision graphs (for both Class B and Class M) show an upward trend, indicating that the model increasingly avoids false positives as training progresses. Similarly, the recall graphs exhibit a rise, suggesting that the model becomes better at detecting true positives over time. The mAP (mean Average Precision) metrics, both mAP50, and mAP50-95, also show improvement, highlighting the model's growing accuracy in object detection across various Intersection-over-Union (IoU) thresholds.

TABLE IV. HYPERPARAMETERS

Parameter	Value
learning_rate	0.01 (default)
batch	32
weight_decay	0.001
epoch	1000

The key hyperparameters (Table IV) used for training include a learning rate of 0.01, a batch size of 32, a weight decay of 0.001, and 1000 epochs. These parameters significantly influence the model's convergence, generalization, and accuracy. The learning rate of 0.01 is a default setting that balances the trade-off between fast convergence and stability. The graphs show that the training losses, including box loss, segmentation loss, classification loss, and Distribution Focal Loss (DFL), show a steady decline as the epochs progress. This suggests that the learning rate was appropriately set, enabling the model to learn from the data without causing oscillations or diverging effectively. The batch size 32 is a common choice for deep learning models, balancing computational efficiency and stability. With this batch size, the model was able to process a reasonable number of images per iteration, which is reflected in the smooth decline of the loss curves. The choice of batch size also impacts the gradient estimates, contributing to the stability observed in the training process. Weight decay, set at 0.001, acts as a regularization technique to prevent overfitting by penalizing large weights in the model. The effectiveness of weight decay can be seen in the validation loss curves, which show that the model is generalizing well to unseen data without significant overfitting. The smooth decline in validation losses, especially for box and segmentation losses, indicates that the regularization is helping to maintain model performance on the validation set. Although the model was trained for 1000 epochs, the graphs suggest that most of the loss functions and evaluation metrics began to stabilize much earlier, around 200 epochs. This stabilization indicates that the model reached a saturation point where further training did not lead to significant improvements in performance. This could be a signal that early stopping could be considered to save computational resources and prevent potential overfitting.

In Fig. 5, the confusion matrix for the YOLOv9e-seg model provides a comprehensive overview of the model's classification performance across various classes, including corals, marine life, seagrass, seaweed, and background. The matrix reveals both the true positives and the errors the model makes, such as false positives and false negatives, which are critical for evaluating its effectiveness.

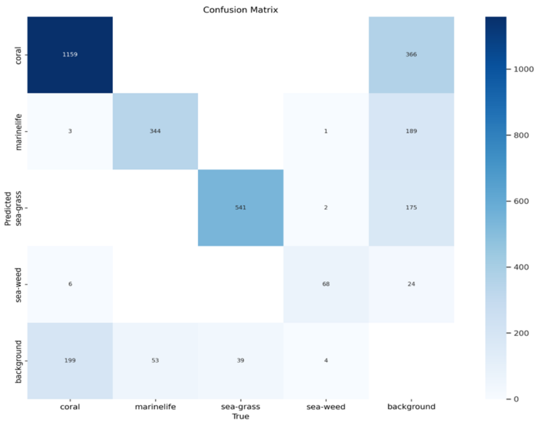


Fig. 7. Confusion matrix for YOLOv9e-seg model.

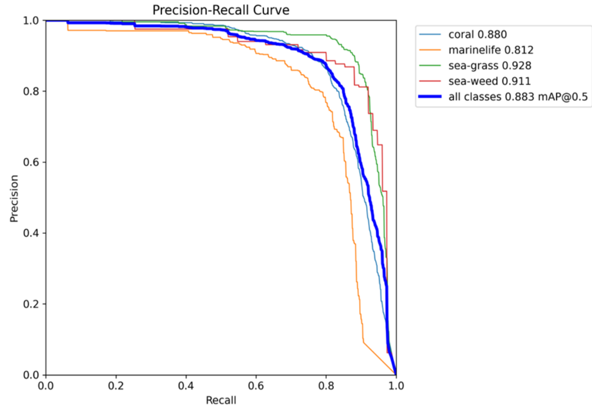


Fig. 8. Mask precision-recall curve.

Starting with the coral class, the model exhibits 1159 true positives, where the actual coral class is correctly predicted as coral. However, there are also some false negatives and false positives. Specifically, the model incorrectly predicts the coral class as marine life three times and as seaweed six times, indicating these as false negatives for the coral class. Furthermore, the coral class is incorrectly classified as background 199 times, which also contributes to the false negatives.

For the marine life class, the model demonstrates 344 true positives, accurately identifying marine life instances. However, there are false negatives, with 53 instances where marine life is incorrectly predicted as background. Additionally, there are instances where other classes are incorrectly classified as marine life, leading to false positives, though the confusion matrix details provided focus more on false negatives.

In the case of seagrass, the model achieves 541 true positives, where the actual seagrass is correctly identified. However, there are 39 false negatives, where seagrass is misclassified as background. This indicates a challenge in distinguishing seagrass from the background, especially in cases where visual features might blend with the environment.

The seaweed class has 68 true positives, correctly predicting the actual seaweed class. However, the model also produces false negatives, where seaweed is misclassified as marine life once, as seagrass twice, and as background four times. These errors highlight the model's difficulty in differentiating between visually similar classes.

Finally, the background class demonstrates several misclassifications, leading to false positives in other categories. Specifically, the background is incorrectly predicted as coral 366 times, marine life 189 times, seagrass 175 times, and as seaweed 24 times. These instances reflect false positives for the respective classes and indicate the model's struggle to accurately segment objects from the background in complex scenes.

A Precision-Recall Curve (PR Curve) is a crucial tool used to evaluate the performance of classification models, particularly in situations where the dataset has an imbalance between classes. The curve plots Precision—the ratio of true positive predictions to the total number of positive predictions—against Recall, which measures the ratio of true positive predictions to the total number of actual positives. The curve allows researchers to assess the trade-offs between precision and recall across different threshold settings, helping to understand how well the model identifies positive cases while avoiding false positives.

The results indicate varying model performance across different classes. For instance, the model demonstrates strong performance in detecting coral, with a PR curve that shows a good balance between precision and recall, resulting in fewer false positives and false negatives. However, the curve for marine life reveals lower precision and recall, suggesting that the model struggles more with accurately identifying this class compared to others. In contrast, the model excels in detecting seagrass, as indicated by the highest area under the curve, reflecting its ability to identify seagrass with high precision and recall accurately. Similarly, the model also performs well in detecting seaweed, showcasing its effectiveness in this area.

The overall performance of the YOLOv9e-instance segmentation model in this research is represented by the thick blue line, which aggregates the results across all classes. The mean Average Precision (mAP) of 0.883 at an Intersection over Union (IoU) threshold of 0.5 suggests that, on average, the model achieves a high level of precision and recall across different classes. This indicates that the model is generally reliable in its predictions, maintaining high precision even as recall increases. This reliability is particularly important in this research, where minimizing false positives is crucial for accurate environmental monitoring.

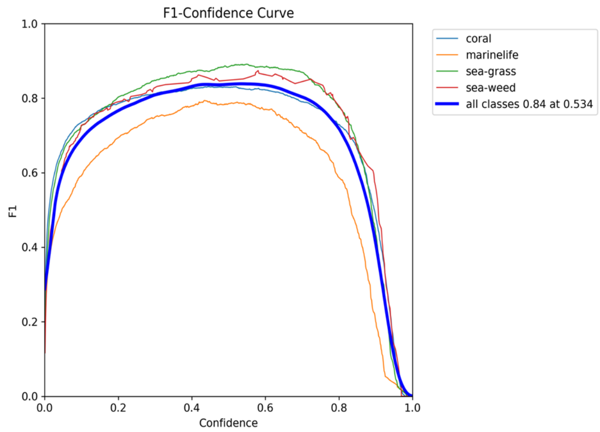


Fig. 9. Mask F1-Confidence curve.

An F1-Confidence Curve is a graphical representation that shows the relationship between the F1-Score and the confidence threshold of a classification model. The F1-Score is a crucial metric in evaluating a model's performance as it represents the harmonic mean of precision and recall, balancing these two metrics. The confidence threshold, on the other hand, is the probability threshold at which a model classifies a sample as positive. By plotting the F1-Score against different confidence thresholds, the F1-Confidence Curve helps understand how the model's overall performance changes as the confidence level varies.

The results from this research demonstrate varying model performance across different classes. For instance, the model shows a high F1-Score for seagrass (green line), indicating strong performance with an optimal balance of precision and recall at a wide range of confidence levels. Similarly, the model performs well on seaweed (red line), maintaining a high F1-Score across different confidence thresholds. Coral (blue line) also exhibits good performance, although it has a slightly lower F1-Score than seagrass and seaweed. However, the model performs less in detecting marine life (orange line), as indicated by the lower F1-Score across the confidence spectrum. This suggests that the model struggles to find an optimal balance between precision and recall for this particular class, reducing effectiveness in identifying marine life at various confidence levels.

Finally, the F1-Confidence Curve provides valuable insights into the performance of the YOLOv9e-instance segmentation model in this research. While the model performs exceptionally well in detecting seagrass and seaweed, it shows some difficulty in accurately identifying marine life, which may require further refinement. The overall F1 score of 0.84 at a confidence threshold of 0.534 suggests that the model is capable of maintaining a good balance between precision and recall, making it a reliable tool for marine object detection and segmentation.

B. Visualization of Segmentation Results

The segmented images in Fig. 10 demonstrate the YOLOv9e-seg model's excellent ability to recognize and categorize coral in a tough underwater environment. Each image features bounding boxes that outline detected coral instances, along with confidence scores that reflect the model's certainty in these detections. This visual representation underscores the model's precision, as the bounding boxes align closely with the actual coral structures, showcasing its capability to accurately localize and identify marine organisms. The high confidence scores further reinforce the model's reliability, particularly in regions where the coral is clearly defined and stands out against the backdrop. This strong performance is most evident in areas where the contrast between the coral and the surrounding environment is sharp, allowing the model to differentiate the coral with minimal ambiguity. Such results are encouraging, as they indicate that the model can effectively handle the complex task of segmenting coral in conditions where the visual features are distinct and easily recognizable.

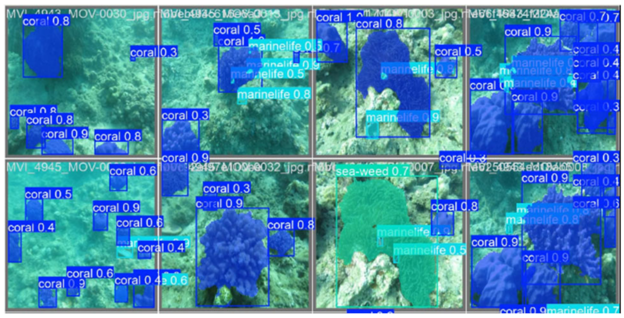
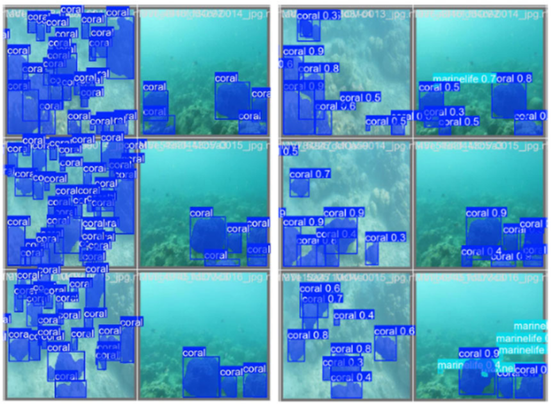


Fig. 10. Prediction of the YOLOv9e-seg.

On the other hand, the image highlights a misclassification where a region identified as "seaweed" with a confidence score of 0.7 should have been classified as coral. This misclassification could be attributed to the limited availability of training data for certain categories, as discussed in Fig. 5, which illustrates the imbalance in the dataset. The graph of instances reveals that seagrass has significantly fewer examples compared to coral, leading to the model's struggle to accurately distinguish between these similar-looking marine organisms. This lack of sufficient training examples for seaweed could cause the model to incorrectly predict it, demonstrating the need for a more balanced dataset to improve classification accuracy.

In the actual annotations (a) of Fig. 11, numerous regions are labeled as coral, indicating a dense presence of this marine organism. However, in the corresponding predictions (b) made by the YOLOv9e-seg model, there are noticeably fewer coral detections. This discrepancy could be due to the blurriness and low contrast in some underwater images, which can obscure the features of corals and make it difficult for the model to accurately identify and segment them. The unclear capture of these regions may lead to the model underestimating the presence of corals or failing to detect them entirely, resulting in a reduced number of predicted regions compared to the actual annotations.



(a) (b)

Fig. 11. The (a) actual annotation and (b) prediction results using the YOLOv9e-seg model.

The underwater environment often presents overlapping objects and complex backgrounds, which can confuse the model, leading to misclassifications or missed detections. In future studies, applying more sophisticated data augmentation techniques, such as background subtraction or using synthetic data to enhance the diversity of the training set, could help the model learn to differentiate between similar-looking objects and backgrounds. Additionally, adjusting the confidence threshold for predictions by implementing Non-Maximum Suppression (NMS) with fine-tuned thresholds can reduce redundant detections, ensuring that only the most confident prediction for each object is retained. Addressing potential class imbalances and applying regularization techniques during model training could further improve the model's generalization and accuracy, making it more reliable in complex underwater scenarios.

Despite these challenges, the YOLOv9e-seg model's predictions exhibit reasonably high confidence levels, as indicated by the confidence ratings on the anticipated bounding boxes. This suggests that while the model may miss some instances, it is generally accurate when it does detect an object. The model's ability to reliably predict regions with high confidence implies that it can effectively distinguish coral from other marine elements when the image quality is sufficient. Future research could focus on enhancing image quality through advanced preprocessing methods or expanding the diversity and quantity of training data to improve the model's ability to manage fluctuations in underwater visibility.

YOLOv9-seg Strengths

1. High-Performance Metrics: The YOLOv9e-seg model has a mean Average Precision (mAP) of 0.883 at an IoU threshold of 0.5, indicating great accuracy in recognizing and segmenting objects such as corals, seagrass, and seaweed. This is a considerable improvement over previous models like YOLOv5-seg and YOLOv8-seg, which had lower mAP values in similar tasks. The YOLOv9e-seg's improved performance in this marine dataset exhibits its capacity to discriminate between complicated forms and objects in an underwater environment with variable visibility and shapes. Furthermore, the F1 score of 0.84 at a confidence level of 0.534 indicates that YOLOv9e-seg well balances precision and recall, which is critical in environmental monitoring. Its great capacity to recognize various object types, even when their characteristics overlap, demonstrates the superior feature extraction processes.
2. Architectural Improvements: The YOLOv9e-seg model includes ProtoNet, which increases segmentation quality by refining the output of the segmentation head. This innovation enables more accurate boundary identification and separation of objects, which is crucial in underwater situations where objects such as seaweed and seagrass may visually overlap. Furthermore, the additional fully connected layers in YOLOv9e-seg provide it an advantage over previous models such as YOLOv5-seg. These deeper layers enable more accurate feature extraction and understanding of difficult situations, such as detecting microscopic and intricate aquatic things. As a result, the model is particularly adept at distinguishing minute changes in texture and appearance between corals, seaweed, and other marine items.
3. Handling of Complex Scenes: YOLOv9e-seg has increased its capacity to handle complicated situations, making it more resilient in different maritime habitats with varying occlusions and lighting conditions. This versatility is particularly useful for underwater image processing, where vision is limited and objects to be recognized are frequently partially hidden.

YOLOv9-seg Weaknesses

1. Class Imbalance Sensitivity: The YOLOv9e-seg model performs poorly on underrepresented classes, such as certain marine life types that were not regularly found in the training dataset. While the model excels at dominant classes such as corals and seagrass, the imbalance causes biases, causing the algorithm to underperform in underrepresented categories. Although class imbalances are a prevalent feature of many models, some alternatives, such as YOLOv8-seg, have improved their techniques for compensating for class imbalances. To reduce biases across classes, YOLOv8-seg, for instance, can apply data augmentation approaches more effectively. To provide more equal representation across classes, YOLOv9e-seg could profit from comparable training techniques like class weighting or oversampling.

2. Computational Complexity: YOLOv9e-seg has 60.5 million parameters, which contributes to its high computational complexity. Its larger size increases the model's memory and processing power requirements, which may be a barrier for real-time applications or deployment in settings with limited resources. In less resource-intensive jobs, for example, YOLOv5-seg and YOLOv8-seg offer better trade-offs between accuracy and efficiency. For example, the largest model in YOLOv5-seg has 88.8 million parameters, whereas the greatest model in YOLOv8-seg has only 70 million. YOLOv9e-seg's higher complexity may also result in slower inference times, which would make it less appropriate for real-time processing applications. However, because they are explicitly designed to be fast, models such as YOLOv8-seg are better suited for real-time detection situations where quick decision-making is essential.
3. Applicability in Limited Hardware Settings: YOLOv9e-seg is less suited for embedded systems or low-power devices like drones or underwater robots that may require real-time processing with restricted hardware capabilities due to its enormous size and high computing demands. YOLOv5-nano and YOLOv8-nano, two of the YOLO family's smallest versions, on the other hand, provide a better balance between efficiency and performance in these kinds of situations, albeit at a somewhat lower accuracy level.

V. Conclusion

The results of this research demonstrate that the YOLOv9e-instance segmentation model is a powerful tool for detecting and classifying marine objects such as corals, seagrass, and seaweed. The model's performance, as evidenced by key metrics, reflects its capability to predict and segment these classes accurately. The mean Average Precision (mAP) of 0.883 at an Intersection over Union (IoU) threshold of 0.5 indicates that the model maintains high precision and recall across different classes, making it highly reliable for marine object detection tasks.

The F1-Score, which reached 0.84 at a confidence threshold of 0.534, suggests that the model effectively balances precision and recall, particularly for classes like seagrass and seaweed. This balance is crucial for environmental monitoring, where the accurate detection of different marine objects can have significant implications for conservation efforts and ecological studies. The smooth decline in training and validation losses, along with the steady improvement in precision and recall metrics, confirms that the chosen hyperparameters—such as a learning rate of 0.01, a batch size of 32, and a weight decay of 0.001—were well-tuned to optimize the model's performance.

However, the study also identifies certain limitations, particularly in the model's ability to accurately detect marine life, which exhibited lower precision and recall compared to other classes. This discrepancy can be largely attributed to class imbalances within the dataset, where the number of marine life instances was significantly lower than that of corals and other classes. Such imbalances can lead to a bias in the model, causing it to perform better in the more represented classes while struggling with the underrepresented ones.

To address these challenges, the research underscores the importance of early detection of class imbalances. It suggests implementing strategies like class weight adjustments during training or data augmentation for underrepresented classes. These strategies can help achieve a more balanced model that performs well across all classes, including those with fewer instances. Moreover, the analysis of the confusion matrix and F1-Confidence Curves provides valuable insights into the types of errors the model makes, such as false positives and false negatives, which are critical for further refining the model.

In conclusion, while the YOLOv9e-instance segmentation model has proven effective for marine object detection, particularly for corals, seagrass, and seaweed, there is room for improvement in its performance on marine life. Future research should focus on enhancing the model's ability to handle class imbalances and exploring additional data collection and annotation efforts to provide a more comprehensive dataset. By addressing these issues, the model can become even more reliable and effective in supporting environmental monitoring and conservation initiatives in marine ecosystems.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Ken D. Gorro contributed to the conceptualization, methodology, model training, hyperparameter tuning, project administration, and manuscript preparation. Anthony S. Ilano provided supervision, validation, and critical review of the manuscript. Lawrence P. Roble was responsible for data collection, preprocessing, hyperparameter tuning, and annotation for training and evaluation datasets. Rue Nicole R. Santillan conducted performance evaluation, while Joseph C. Pepito focused on the visualization of segmentation results. Elmo B. Ranolo contributed to hyperparameter tuning and drafting the discussion sections. Kim D. Gorro coordinated fieldwork, conducted environmental assessments, and prepared supplementary materials. Apple Jane M. Gorro undertook the literature review, data curation, and administrative support. Moustafa F. Ali offered technical advice on machine learning approaches and guidance on instance segmentation techniques. Archival J. Sebial handled the development of code infrastructure, integration of software tools, and deployment framework.

Finally, Jezreel N. Buot managed editing and proofreading of the manuscript, coordination among authors, and resource management. All authors have read and approved the final manuscript.

FUNDING

This project is funded by Cebu Technological University—Carmen Campus.

ACKNOWLEDGMENT

We extend our profound gratitude to Cebu Technological University Carmen Campus for their unwavering support throughout the course of this research endeavor.

REFERENCES

[1] L. Edwards, "Catami classification scheme for scoring marine biota and substrata in underwater imagery," *CATAMI Classification Scheme for Scoring Marine Biota and Substrata in Underwater Imagery*, pp. 1–17, 2013.

[2] O. Beijbom, P. J. Edmunds, D. I. Kline, B. G. Mitchell, and D. Kriegman, "Automated annotation of coral reef survey images," in *Proc. 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, RI, USA, 2012, pp. 1170–1177.

[3] B. J. Boom, J. He, S. Palazzo, P. X. Huang, C. Beyan, H.-M. Chou, F.-P. Lin, C. Spampinato, and R. B. Fisher, "A research tool for long-term and continuous analysis of fish assemblage in coral reefs using underwater camera footage," *Ecological Informatics*, vol. 23, pp. 83–97, 2014.

[4] A. Shihavuddin, N. Gracias, R. Garcia, J. Escartif, and R. B. Pedersen, "Automated classification and thematic mapping of bacterial mats in the North Sea," in *Proc. OCEANS-Bergen, 2013 MTS/IEEE*, Bergen, Norway, 2013, pp. 1–8.

[5] A. Gómez-Ríos, S. Tabik, J. Luengo, A. S. M. Shihavuddin, B. Krawczyk, and F. Herrera, "Towards highly accurate coral texture images classification using deep convolutional neural networks and data augmentation," *Expert Systems with Applications*, vol. 118, pp. 315–328, 2019.

[6] J. Borbon, J. Javier, J. Llamado, E. Dadios, and R. K. Billones, "Coral health identification using image classification and convolutional neural networks," in *Proc. 2021 IEEE 13th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, Manila, Philippines, 2021, pp. 1–6.

[7] O. Beijbom *et al.*, "Towards automated annotation of benthic survey images: Variability of human experts and operational modes of automation," *PLOS One*, vol. 10, no. 7, pp. 1–22, 2015.

[8] A. Friedman. (2017). *Squidle*. [Online]. Available: https://squidle.org/content/snippetabout.html

[9] P. J. Edmunds and B. Riegl, "Urgent need for coral demography in a world where corals are disappearing," *Marine Ecology Progress Series*, 2020.

[10] A. King. (2018). *DeepSegment*. [Online]. Available: https://andrewking.io/portfolio/deepsegments

[11] Alonso, M. Yuval, G. Eyal, T. Treibitz, and A. C. Murillo, "Coralseg: Learning coral segmentation from sparse annotations," *Journal of Field Robotics*, vol. 36, no. 8, pp. 1456–1477, 2019.

[12] M. Yuval, I. Alonso, G. Eyal, D. Tchernov, Y. Loya, A. C. Murillo, and T. Treibitz, "Repeatable semantic reef-mapping through photogrammetry and label-augmentation," *Remote Sensing*, vol. 13, no. 4, 2021.

[13] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman, "Geodesic star convexity for interactive image segmentation," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, CA, USA, 2010.

[14] J. N. Blanchet. (2016). *CoralMe*. [Online]. Available: https://github.com/jnblanchet/CoralMe

[15] Z. Liang, Y. Wang, X. Ding, Z. Mi, and X. Fu, "Single underwater image enhancement by attenuation map guided color correction and detail preserved dehazing," *Neurocomputing*, vol. 425, pp. 160–172, 2021.

[16] J. C. Zhou, D. H. Zhang, and W. S. Zhang, "Classical and state-of-the-art approaches for underwater image defogging: A comprehensive survey," *Frontiers of Information Technology and Electronic Engineering*, vol. 21, pp. 1745–1769, 2020.

[17] M. Jian, X. Liu, H. Luo, X. Lu, and J. Dong, "Underwater image processing and analysis: A review," *Signal Processing: Image Communication*, vol. 91, p. 116088, 2021.

[18] Z. J., Z. Liu, W. Zhang, and W. Zhang, "Underwater image restoration based on secondary guided transmission map," *Multimedia Tools and Applications*, vol. 80, pp. 7771–7788, 2021.

[19] Y. Wang, W. Song, G. Fortino, L. Qi, W. Zhang, and A. Liotta, "An experimental-based review of image enhancement and image restoration methods for underwater imaging," *IEEE Access*, vol. 7, pp. 140233–140251, 2019.

[20] R. Wang, Y. Wang, J. Zhang, and X. Fu, "Review on underwater image restoration and enhancement algorithms," in *Proc. the 7th International Conference on Internet Multimedia Computing and Service (ICIMCS '15)*, Association for Computing Machinery (ACM), New York, NY, USA, 2015, p. 6.

[21] H. Mliki, S. Dammak, and E. Fendri, "An improved multi-scale face detection using convolutional neural network," *Signal, Image and Video Processing*, vol. 14, pp. 1345–1353, 2020.

[22] J.-N. Blanchet, S. Déry, J.-A. Landry, and K. Osborne, "Automated annotation of corals in natural scene images using multiple texture representations," *PeerJ Preprints*, e2026v2, 2016.

[23] O. Beijbom, "Automated annotation of coral reef survey images," in *Proc. 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, RI, USA, 2012.

[24] E. Tusa, A. Reynolds, D. M. Lane, N. M. Robertson, H. Villegas, and A. Bosnjak, "Implementation of a fast coral detector using a supervised machine learning and Gabor wavelet feature descriptors," in *Proc. 2014 IEEE Sensor Systems for a Changing Ocean (SSCO)*, Brest, France, 2014, pp. 1–6.

[25] M. H. Shakoor and R. Boostani, "A novel advanced local binary pattern for image-based coral reef classification," *Multimedia Tools and Applications*, vol. 77, pp. 2561–2591, 2018.

[26] A. S. M. Shihavuddin, N. Gracias, R. Garcia, A. C. R. Gleason, and B. Gintert, "Image-based coral reef classification and thematic mapping," *Remote Sensing*, vol. 5, no. 4, pp. 1809–1841, 2013.

[27] Z.-J. Duan, S. B. Li, J. J. Hu, J. Yang, and Z. Wang, "Review of deep learning based object detection methods and their mainstream frameworks," *Laser & Optoelectronics Progress*, vol. 57, no. 12, 120005, 2020.

[28] R. Cheng, "A survey: Comparison between convolutional neural network and YOLO in image identification," *Journal of Physics: Conference Series*, vol. 1453, p. 012139, 2020.

[29] M. Pedersen, J. Bruslund Haurum, R. Gade, and T. B. Moeslund, "Detection of marine animals in a new underwater dataset with varying visibility," in *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Long Beach, CA, USA, Jun. 16–17, 2019, pp. 18–26.

[30] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 1137–1149, 2016.

[31] S. A. Siddiqui, A. Salman, M. I. Malik, F. Shafait, A. Mian, M. R. Shortis, and E. S. Harvey, "Automatic fish species classification in underwater videos: Exploiting pre-trained deep neural network models to compensate for limited labeled data," *ICES Journal of Marine Science*, vol. 75, pp. 374–389, 2018.

[32] Z. Cai and N. Vasconcelos, "Cascade R-CNN: High quality object detection and instance segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, pp. 1483–1498, 2021.

[33] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. the IEEE*

Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, Jun. 27–30, 2016, pp. 779–788.

[34] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, Jul. 21–26, 2017, pp. 7263–7271.

[35] A. Farhadi and J. Redmon, "YOLOv3: An incremental improvement," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, Springer, Berlin/Heidelberg, Germany, 2018, pp. 1804–02767.

[36] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: Challenges, architectural successors, datasets, and applications," *Multimedia Tools and Applications*, pp. 1–33, 2022.

[37] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. the European Conference on Computer Vision (ECCV)*, Springer, Berlin/Heidelberg, Germany, 2016, pp. 21–37.

[38] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, Oct. 22–29, 2017, pp. 2980–2988.

[39] X. Zhang, F. Wan, C. Liu, R. Ji, and Q. Ye, "Freeanchor: Learning to match anchors for visual object detection," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, Vancouver, BC, Canada, Dec. 8–14, 2019, pp. 147–155.

[40] C. Zhu, Y. He, and M. Savvides, "Feature selective anchor-free module for single-shot object detection," in *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, Jun. 15–20, 2019, pp. 840–849.

[41] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. the IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea, Oct. 27–Nov. 2, 2019, pp. 9627–9636.

[42] H. Yang, P. Liu, and Y. Hu, "Research on underwater object recognition based on YOLOv3," *Microsystems Technology*, vol. 27, pp. 1837–1844, 2021.

[43] Y. Song, B. He, and P. Liu, "Real-time object detection for AUVs using self-cascaded convolutional neural networks," *IEEE Journal of Oceanic Engineering*, vol. 46, pp. 56–67, 2019.

[44] I. Iqbal, K. Walayat, M. U. Kakar, and J. Ma, "Automated identification of human gastrointestinal tract abnormalities based on deep convolutional neural network with endoscopic images," *Intelligent Systems with Applications*, vol. 16, 200149, 2022. https://doi.org/10.1016/j.iswa.2022.200149

[45] S. Ashwini, J. R. Arunkumar, and R. T. Prabu *et al.*, "Diagnosis and multi-classification of lung diseases in CXR images using optimized deep convolutional neural network," *Soft Computing*, vol. 28, pp. 6219–6233, 2024. https://doi.org/10.1007/s00500-023-09480-3.

[46] R. Wang, Y. Lu, S. Huang, J. Liu, and M. Wang, "A forewarning method for falling hazard from hole based on instance segmentation and regional invasion detection," *Lecture Notes in Operations Research*, pp. 157–174, 2022. http://dx.doi.org/10.1007/978-981-19-5256-2_14.

[47] E. Casas, L. Ramos, E. Bendek, and F. Rivas-Echeverría, "Assessing the effectiveness of YOLO architectures for smoke and wildfire detection," *IEEE Access*, vol. 11, pp. 96554–96583, 2023. http://dx.doi.org/10.1109/ACCESS.2023.3312217

[48] W. Liu, K. Quijano, and M. M. Crawford, "YOLOv5-tassel: Detecting tassels in RGB UAV imagery with improved YOLOv5 based on transfer learning," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 8085–8094, 2022. http://dx.doi.org/10.1109/JSTARS.2022.3206399

[49] Ultralytics. "Ultralytics YOLOv8". [Online]. Available: https://docs.ultralytics.com/models/yolov8/

[50] S. Rath. (2023). Getting started with YOLOv5 instance segmentation. *LearnOpenCV*. [Online]. Available: https://learnopencv.com/yolov5-instance-segmentation/.

[51] J. Gallagher. (2024). How to train YOLOv9 on a custom dataset. *Roboflow*. [Online]. Available: https://blog.roboflow.com/train-yolov9-model/

[52] F. Mattioli, M. R. Munawar, G. Jocher, and Q. Burhan. (2024). YOLOv9: A leap forward in object detection technology. *Ultralytics*. [Online]. Available: https://docs.ultralytics.com/models/yolov9/

[53] S. Liu, J. Jia, S. Fidler, and R. Urtasun, "SGN: Sequential grouping networks for instance segmentation," in *Proc. the IEEE/CVF International Conference on Computer Vision (ICCV)*, Venice, Italy, 2017.

[54] E. Ranolo, K. Gorro, A. Ilano, H. Pineda, C. Sintos, and A. J. Gorro, "Underwater and coastal seaweeds detection for fluorescence seaweed photos and videos using YOLOv3 and YOLOv5," in *Proc. the 2nd International Conference for Innovation in Technology (INOCON)*, 2023, pp. 1–6.

[55] K. Gorro, A. Ilano, E. Ranolo, H. Pineda, C. Sintos, and A. J. Gorro, "Coral detection in fluorescence images and videos using YOLOv3 and YOLOv5," in *Proc. the 2023 International Conference on Business Analytics for Technology and Security (ICBATS)*, Mar. 2023, pp. 1–6.