Custom YOLOv8 Model for Crown-of-Thorns Starfish (*Acanthaster planci*) and Long-Spined Sea Urchin (*Diadema antillarum*) Quantification

RAMOS, John Peter M., ECE
College of Engineering
Electronics Engineering Department
Technological University of the Philippines-Manila

GALERA, Adrian Darren S.
College of Engineering
Electronics Engineering Department
Technological University of the Philippines-Manila

MONTEJO, Paulo Jr. M.
College of Engineering
Electronics Engineering Department
Technological University of the Philippines-Manila

PACOMA, Niña Marie B.
College of Engineering
Electronics Engineering Department
Technological University of the Philippines-Manila

ROSETE, Charles L.
College of Engineering
Electronics Engineering Department
Technological University of the Philippines-Manila

SEBASTIAN, Franchesca Ann A.
College of Engineering
Electronics Engineering Department
Technological University of the Philippines-Manila

***Abstract*—Crown-of-Thorns Starfish (COTS) and Long-Spined Sea Urchin populations are significant indicators of coral health. COTS, which feed on coral polyps, are maintained at a population of less than 15 per hectare to avoid rapid and extensive destruction of coral reefs. Long-Spined Sea Urchins are macroalgal glazers maintained at a population of five per sqm to maintain low algal cover. The proposed system is a YOLOv8-based model for automated quantification of Crown-of-Thorns Starfish and Long-Spined Sea Urchins from underwater video inputs. This model was trained with a custom dataset of COTS and long-spined sea urchin images for precise detection and integrated with ByteTrack and Supervision for its counting capabilities. Performance metrics results show a mAP score of 0.989 with an F1 score of 0.97 at a 0.59 confidence interval. The model exhibited an accuracy of 86.16% when counting in 5-ft saltwater. The YOLOv8 model's performance metrics and testing results show that the model proves to be capable of counting crown-of-thorns starfish and long-spined sea urchins in an actual marine environment.**

Keywords— YOLOv8, Crown-of-Thorns Starfish, Long-Spined Sea Urchin

I. INTRODUCTION

You Only Look Once (YOLO) is a deep learning model for real-time object detection applications known for its accurate and efficient detection of objects in real-time scenarios. This model has been widely used in robotics, autonomous driving, and video surveillance [1].

On the other hand, manual techniques have been used for underwater monitoring in the Philippines. SCUBA diving and freediving are two standard methods for ocular surveys to monitor coral reefs. However, these are costly and labor-intensive.

Coral reefs are valuable ecosystems that suffer from threats like coral bleaching caused by certain factors like rising water temperatures, pollution, outbreaks of crown-of-thorns starfish, and reduction of algal glazers [2]. Crown-of-Thorns Starfish (COTS) are marine invertebrates that feed on coral polyps. These are a natural part of the marine ecosystem and are usually found in healthy reefs at a density of 1 to 10 individuals per hectare. However, when the COTS population reaches around 15 individuals per hectare, a COTS outbreak occurs. A COTS outbreak, described as 15 COTS per hectare, can cause rapid and extensive destruction, potentially killing 90% of affected reefs. COTS can consume approximately 5 to 13 square meters of live coral annually [3].

On the other hand, macroalgal glazers, such as the Long-Spined Sea Urchins, help with coral restoration by eliminating seaweed and microalgae, which are the main competitors of corals for space and resources. Research shows that at least five long-spined sea urchins per square meter should be required to maintain a low algal cover [4]. However, there has been a decline in the population because of overfishing, resulting in an increase in seaweed and macroalgae, which negatively affects coral reefs.

In line with these, the proponents proposed a system for monitoring COTS and urchin populations using machine learning, specifically YOLOv8. This object detection model was used to automate species counting by integrating ByteTrack and Supervision libraries. ByteTrack tracks multiple objects in videos by defining their bounding boxes and classes and maintains a unique ID for each detected object. Supervision uses a line-based counter in which an object gets counted in an increment of one for its respective class when it crosses the line.

II. BACKGROUND OF THE PROBLEM

COTS outbreaks have been documented in various regions of the Philippines, including Southern Leyte, Oriental Mindoro, Camarines Sur, and Batangas. Local communities and volunteer divers have engaged in various conservation initiatives, including manual observation and extraction of Crown-of-Thorns starfish (COTS). However, these approaches prove to be time-consuming, hazardous, and labor-intensive.

Despite the critical importance of understanding and managing COTS outbreaks and their ecological effects, there remains a lack of research addressing this issue comprehensively. Recent advancements in technology have created possibilities for machine learning to monitor the condition of coral reefs and identify potential hazards, such as pollution and crown-of-thorns starfish detection. Campos et al. [5] developed a semi-autonomous underwater drone with coral reef monitoring and pollution detection via machine learning, implemented with the object detection algorithms YOLOv4 and Faster-RCNN, to detect COTS and underwater garbage. While this study laid a foundation for integrated COTS and pollution detection, there remains a lack of research specifically focused on quantifying COTS populations, a crucial aspect for devising strategies to identify and control COTS outbreaks and mitigate their impact on coral reefs. To address this gap, the present study introduces a YOLOv8-based automated quantification model for counting crown-of-thorns starfish and long-spined sea urchins from underwater video inputs. This system sought to provide a reliable means of monitoring COTS abundance and distribution trends, aiding in conservation efforts for coral reef ecosystems.

III. OBJECTIVES

This study aims to develop a YOLOv8-based Automated Crown-of-Thorns Starfish (COTS) and Long-Spined Sea Urchin Quantification System. This incorporates the YOLOv8 algorithm, and ByteTrack and Supervision libraries for its object detection and counting capabilities. Specifically, it aims to achieve the following:

1. to develop a customized object detection system based on the YOLOv8 model on a custom dataset of crown-of-thorns starfish and long-spined sea urchin images;
2. to develop a dedicated counting model utilizing ByteTrack and Supervision on the trained custom model for precise quantification; and
3. to test and evaluate the reliability of the counting system using test images in a saltwater environment.

IV. METHODOLOGY

A. *Dataset Acquisition*

Custom datasets of crown-of-thorns starfish and long-spined sea urchins were collected from various open-source databases. The raw images of crown-of-thorns starfish were acquired from a recent study on a coral reef monitoring system [5], snapshots from online videos, and a Kaggle Dataset [6]. The raw images of long-spined sea urchins were acquired from the Sea Animals Image Dataset from Kaggle [7], the Sea urchin image classification dataset [8], and the Global Biodiversity Information Facility repository [9].

Table I presents the number of images used for the dataset of the object detection model. A total of 3935 images containing 5270 annotations were used as a raw dataset.

TABLE I. RAW IMAGES AND ANNOTATIONS

Class	Raw		Train Images	Valid Images
	Image	*Annotation*		
Crown-of-Thorns Starfish	2,264	2,636	1,828	436
Long-Spined Sea Urchin	1,671	2,634	1,327	344
All	3,935	5,270	3,155	780

B. *Augmentation and Preprocessing*

The images were preprocessed by reducing each image to a resolution of 640x640 pixels. These were then divided into training and validation images in an 8:2 ratio, and the training images were augmented through Roboflow, an online computer vision platform [10], which flipped the images horizontally and vertically and rotated them 90 degrees clockwise and counterclockwise.

Table II presents the number of images and annotations produced after augmentation. Overall, 22,325 images, each class having approximately 15,000 annotations, were used as the train images for the object detection model.

TABLE II. AUGMENTED IMAGES AND ANNOTATIONS

Class	Augmented	
	Image	*Annotation*
Crown-of-Thorns Starfish	12,988	15,309
Long-Spined Sea Urchin	9,337	15,617
All	22,325	30,926

C. *Training the Custom YOLOv8 Model*

The code for training and the pre-trained YOLOv8 nano model were cloned from the Ultralytics repository in GitHub [11]. The Albumentations library was also cloned in a GitHub repository for additional augmentation techniques during the model's training process. The Albumentations library applies blur, median blur, converts to grayscale, contrast limited adaptive histogram equalization, random change of brightness and contrast, random gamma, and lowering image quality by compression [12].

The training was set in 100 epochs with a batch size of 16, defined through autobatch, which estimates the optimum batch size based on the system's available memory. After the training, a validation task was performed to evaluate the model's performance.

D. *Implementation of Bytetrack and Roboflow Supervision*

The custom model was integrated with ByteTrack [13] and Supervision [14] to count Crown-of-Thorns Starfish (COTS) and Long-spined Sea Urchins. ByteTrack tracks multiple objects in videos by defining their bounding boxes and classes and maintains a unique ID for each detected object as the video progresses. Supervision uses a line-based counter in which an object gets counted in an increment of one for its respective class when it crosses the line.

Fig. 1 represents the process flow of the counting system. When an object is detected, ByteTrack assigns a unique tracking ID to the object, enabling recognition even if it moves out of the frame. As the object with tracking ID passes through the line zone defined through Supervision, counts are triggered and added to the total count. When an object moves across the line zone from the top to the bottom, it registers an "in" count. If an object moves from the bottom to the top across the line zone, it prompts an "out" count. Fig. 2 shows the interface of the counting system.

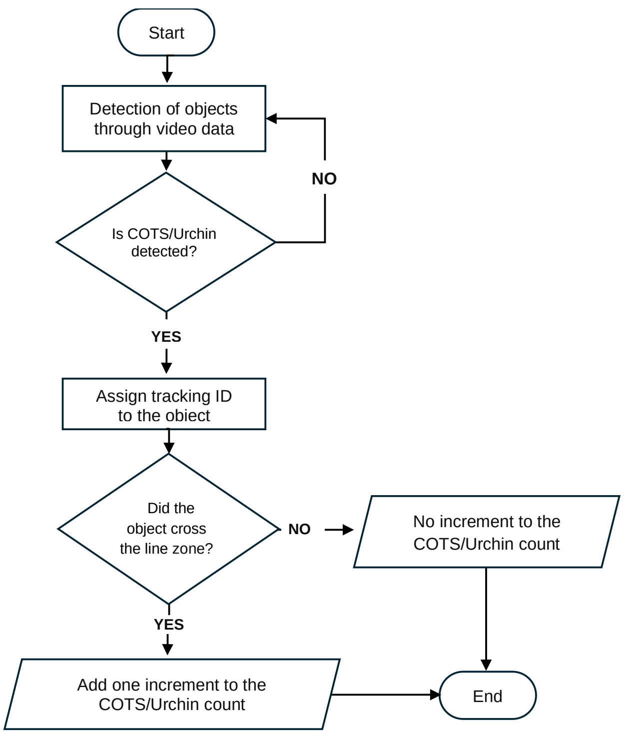


Fig. 1. Counting system flowchart.

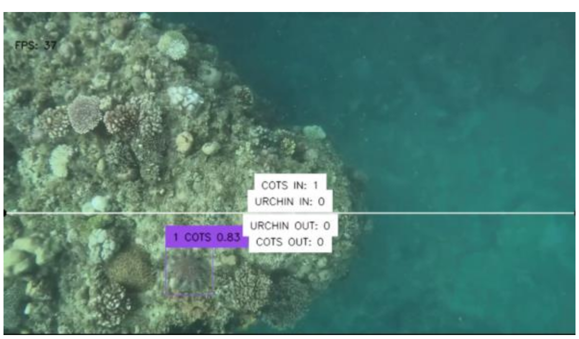


Fig. 2. Interface of the counting system

E. Testing in Saltwater

The testing was done in Balayan Bay, Taal, Batangas, at a depth of 5 ft. The ocean floor was scattered with pictures of crown-of-thorns starfish and long-spined sea urchins. The videos were taken using a Go Pro Hero 7 camera and reduced to a resolution of 480p.

Fig. 3. Setup for testing in saltwater

Four scenarios were considered in the placement of pictures underwater and in how the camera captures the videos: scattered, curved, linear, and curvilinear. The differences between these scenarios are illustrated in Fig. 4.

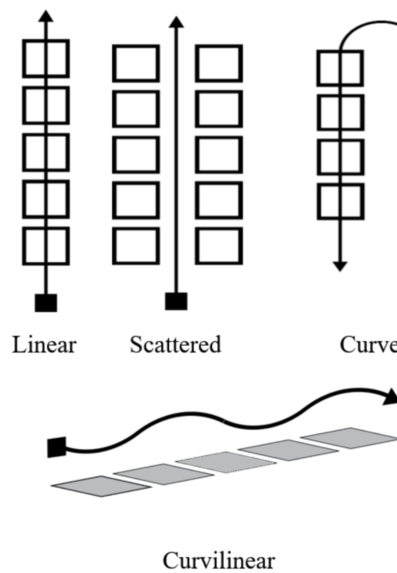


Fig. 4. Scenarios for testing in saltwater

V. RESULTS AND DISCUSSION

A. Evaluation Metrics of YOLOv8 Model

The custom model was trained for approximately 10 hours and 44 minutes, completing 100 epochs. The class-wise YOLO performance metrics trends and validation results are shown in Fig. 5-9. The class-wise YOLO performance metrics include precision, recall, mAP50, and mAP50-95.

These class-wise metrics are used to evaluate the performance of the custom model in overall and specific classes. The precision metric determines the accuracy of the detected objects, while the recall metric determines the capability of the model to detect all instances in each image. The mean average precision (mAP) indicates that the detection of the object is successful and evaluated at various intersection over union (IoU) thresholds, mAP50 and mAP50-95. The mAP50 indicates the mAP calculated at IoU of 0.5 and mAP50-95 at varying IoU thresholds from 0.5 to 0.95. The results of these metrics at each epoch were also recorded in a comma-separated values (CSV) file.

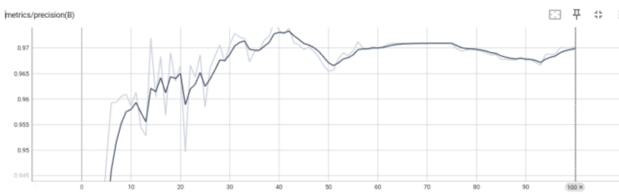


Fig. 5. Precision/epoch

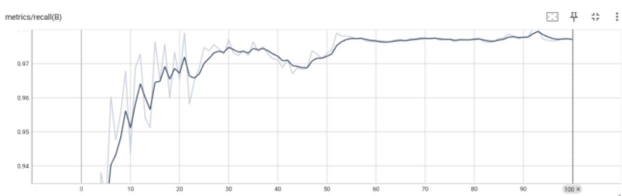


Fig. 6. Recall/epoch

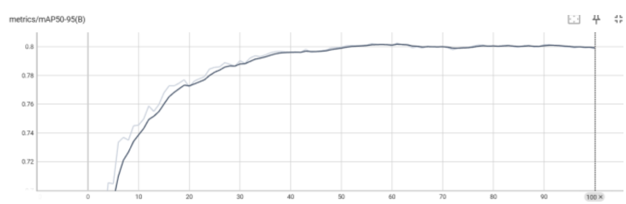


Fig. 7. mAP50/epoch

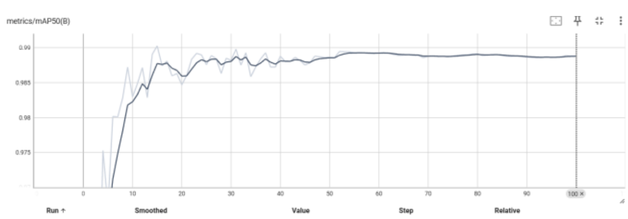
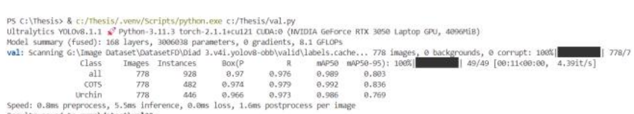


Fig. 8. mAP50-95/epoch



```
PS C:\Thesis> & c:/Thesis/.venv/Scripts/python.exe c:/Thesis/val.py
Ultralytics YOLOv8.1.1 🚀 Python-3.11.3 torch-2.1.1+cu121 CUDA:0 (NVIDIA GeForce RTX 3050 Laptop GPU, 4096MiB)
Model summary (fused): 168 layers, 3006038 parameters, 0 gradients, 8.1 GFLOPs
val: Scanning G:\Image Dataset\DatasetFD\Diad 3.v4i.yolov8-obb\valid\labels.cache... 778 images, 0 backgrounds, 0 corrupt: 100%|██████████| 778/7
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 49/49 [00:11<00:00,  4.39it/s]
                   all        778        928       0.97      0.976      0.989      0.803
                  COTS        778        482      0.974      0.979      0.992      0.836
                Urchin        778        446      0.966      0.973      0.986      0.769
Speed: 0.8ms preprocess, 5.5ms inference, 0.0ms loss, 1.6ms postprocess per image
Results saved to runs\detect\val23
```

Fig. 9. YOLO validation task result of the best weight

After completing the epochs, a validation task was initiated to further evaluate the best model produced. This also provided visual outputs, which include the F1 score curve and confusion matrix. Table III shows the summary of the performance metrics and validation results of the trained custom model.

TABLE III. MODEL'S WEIGHT INFORMATION

Class	Precision	Recall	mAP50	mAP50-95	F1
Crown-of-Thorns Starfish	0.974	0.979	0.992	0.836	0.97
Long-Spined Sea Urchin	0.966	0.973	0.986	0.769	
All	0.97	0.976	0.989	0.803	

The F1-confidence curve of the trained custom model is shown in Fig. 10. The F1 score is the weighted means of precision and recall, recognizing positive detections while minimizing false positives and negatives in an object detection model. The COTS class attained a higher F1 score than the Urchin class. The overall F1 score attained was 0.97 at a 0.59 confidence level.

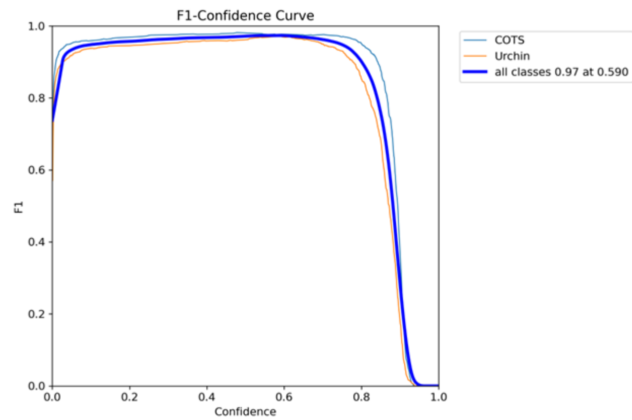


Fig. 10. PF1-confidence curve of the trained custom model

The confusion matrix of the trained custom model is shown in Fig. 11, which shows the count of true positives, true negatives, false positives, and false negatives for each class. A confusion matrix is an object detection metric that maps the counts from predicted and actual values from the model. In the model, the COTS class has higher true positives than the urchin class, represented by the dark shade of blue in the matrix.

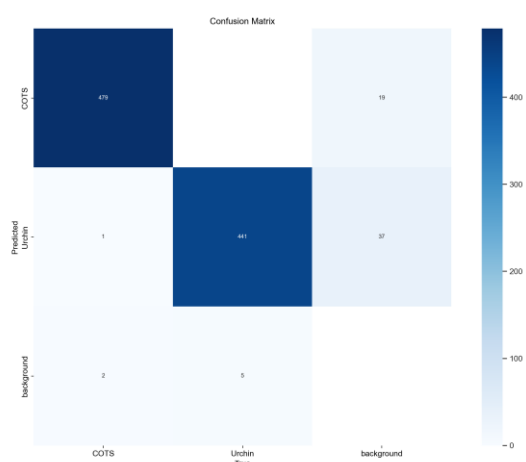


Fig. 11. Confusion matrix of the trained custom model

B. Results from Initial Testing in Saltwater

The YOLOv8 model underwent testing in a 5-foot saltwater for three setups: 30 COTS images, 30 long-spined sea urchin images, and a combination of both classes distributed throughout the saltwater. The testing underwent three trials for each setup and each scenario.

1) Counting of crown-of-thorns starfish images in saltwater

Fig. 12 illustrates counted objects based on the object counting system, and Table IV shows the counting accuracy in three trials of the YOLOv8 Model on videos containing 30 COTS images in saltwater. The model exhibited an average accuracy of 93.61%.

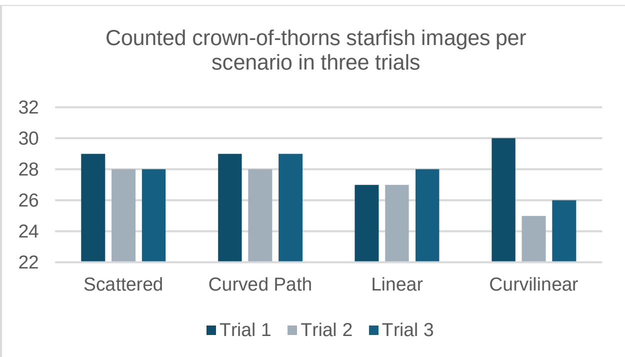


Fig. 12. Counted objects on videos containing dispersed COTS images in three trials

TABLE IV. COUNT ACCURACY OF CROWN-OF-THORNS STARFISH PER SCENARIO IN THREE TRIALS

Scenario	**Accuracy**
Scattered	94.44%
Curved Path	95.56%
Linear	91.11%
Curvilinear	93.33%
Average	93.61%

2) Counting of long-spined sea urchin images in saltwater

Fig. 13 illustrates counted objects based on the object counting system and Table V shows the counting accuracy in three trials of the YOLOv8 Model on videos containing 30 long-spined sea urchin images in saltwater. The model exhibited an average accuracy of 93.89%.

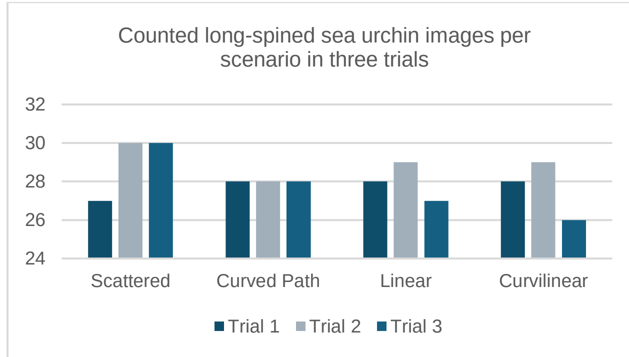


Fig. 13. Counted objects on videos containing dispersed long-spined sea urchin images in three trials

TABLE V. COUNT ACCURACY OF LONG-SPINED SEA URCHIN PER SCENARIO IN THREE TRIALS

Scenario	**Accuracy**
Scattered	96.67%
Curved Path	93.33%
Linear	93.33%
Curvilinear	92.22%
Average	**93.89%**

3) Counting of COTS and long-spined sea urchin images in saltwater

Fig. 14-15 illustrates counted objects based on the object counting system, and Table VI shows the counting accuracy in three trials of the YOLOv8 Model on videos containing the mixed 30 long-spined sea urchin images and 30 crown-of-thorns starfish images in saltwater. The model achieved an accuracy of 76.67% for COTS and 65.28% for urchin. On average, the YOLOv8 Model showed 70.97% accuracy for counting COTS and urchins in images under saltwater.

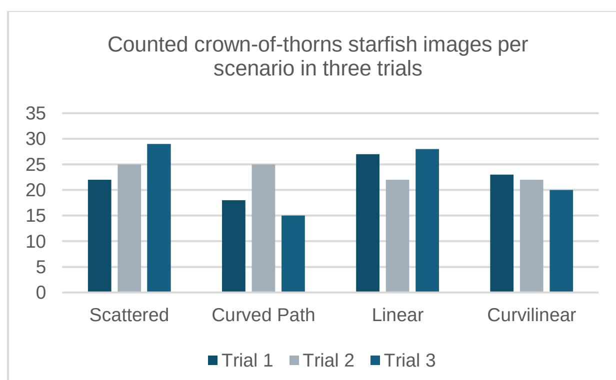


Fig. 14. Counted objects on videos containing dispersed COTS images in three trials

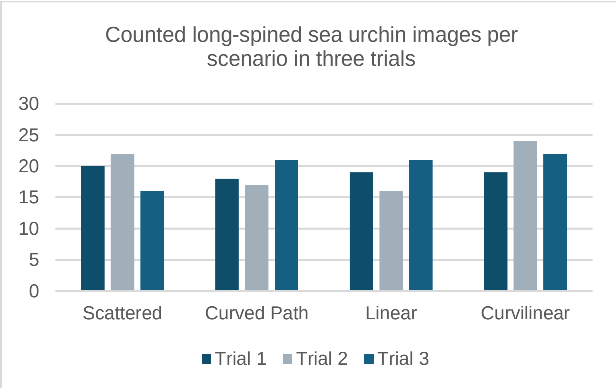


Fig. 15. Counted objects on videos containing dispersed long-spined sea urchin images in three trials

TABLE VI. COUNT ACCURACY OF MIXED CLASSES PER SCENARIO IN THREE TRIALS

Scenario	**Accuracy**	
	COTS	***Urchin***
Scattered	84.44%	64.44%
Curved Path	64.44%	62.22%
Linear	85.56%	62.22%
Curvilinear	72.22%	72.22%
Average per class	**76.67%**	**65.28%**
Average for both classes	**70.97%**	

Overall, the YOLOv8 Model was tested for counting COTS, urchin, and mixed classes, which achieved an accuracy of 93.61%, 93.89%, and 70.97%, respectively. On average, the model exhibited an accuracy of 86.16% for counting in 5-ft saltwater.

TABLE VII. OVERALL COUNT ACCURACY IN THREE SETUPS

Present Class	**Average Accuracy**
COTS only	93.61%
Urchin only	93.89%
Both classes	70.97%
Overall Average Accuracy	**86.16%**

VI. CONCLUSION

In this study, a customized YOLOv8 model was developed from a custom dataset of crown-of-thorns starfish and long-spined sea urchins. A total of 22,325 images, with 30,926 annotations, were used to train the model. Performance metrics results show a mAP score of 0.989 with an F1 score of 0.97 at a 0.59 confidence interval. The model was integrated with ByteTrack and Supervision for its counting abilities. The model was tested for 30 images in three trials in actual saltwater at a depth of 5 feet and various sample placements/scenarios. Overall, the YOLOv8 Model was tested for counting COTS, urchins, and mixed classes, achieving accuracy of 93.61%, 93.89%, and 70.97%, respectively. On average, the model exhibited an accuracy of 86.16% when counting in 5-ft saltwater. The YOLOv8 model's performance metrics and testing results show that the model proves to be capable of counting crown-of-thorns starfish and long-spined sea urchins in an actual marine environment.

VII. REFERENCES

[1] Keylabs. (2023). Under the hood: YOLOv8 architecture explained. https://keylabs.ai/blog/under-the-hood-yolov8-architecture-explained National Ocean Service.

[2] Dempsey, A. (2020, March 17). The devastating effects of a crown-of-Thorns-Starfish outbreak. Living Oceans Foundation. https://www.livingoceansfoundation.org/the-devastating-effects-of-a-crown-of-thorns-starfish-outbreak/

[3] Manuel, O., Williams, S., Weil, E., & Cruz-Motta, J. (2021). Corrigendum to "Experimental evaluation of D. antillarum herbivory effects on benthic community assemblages" [Journal of experimental marine biology and ecology. Volume 541 (2021)]. Journal of Experimental Marine Biology and Ecology, 544, 151612. https://doi.org/10.1016/j.jembe.2021.151612

[4] Goodhue, K. (2021, August 17). Could algae-eating sea urchins save coral reefs? Shorelines. https://sercblog.si.edu/could-algae-eating-sea-urchins-save-coral-reefs/

[5] Campos, F., Dueñas, E., Gecana, K., Lazaro, J., & Marcellana, M. (2022). Layag: A Semi-Autonomous Underwater Drone With Coral Reef Monitoring And Pollution Detection Via Machine Learning. Technological University of the Philippines Manila.

[6] Sibilev, A. (2021). Crown-of-thorns starfish. Kaggle: Your Machine Learning and Data Science Community. https://www.kaggle.com/datasets/antonsibilev/crownofthorns-starfish

[7] Vencer, L. V. (2023). Sea Animals Image Dataset. Retrieved from https://www.kaggle.com/datasets/vencerlanz09/sea-animals-image-dataste.

[8] Images.cv. (n.d.). Sea urchin image classification dataset. Retrieved from https://images.cv/dataset/sea-urchin-image-classification-dataset

[9] GBIF. (2023) GBIF Occurrence Download. Retrieved from https://doi.org/10.15468/dl.hn2z2h

[10] Dwyer, B., Nelson, J., & Solawetz, J. (n.d.). Roboflow (Version 1.0) [Software]. Retrieved from https://roboflow.com

[11] Ultralytics. (2022). Ultralytics YOLOv8. Retrieved from https://github.com/ultralytics/ultralytics

[12] Buslaev A., et al. (2016). Albumentations. Retrieved from https://github.com/albumentations-team/albumentations

[13] Zhang, Y., et al. (n.d.). ByteTrack: Multi-object tracking by associating every detection box. arXiv.org. https://arxiv.org/abs/2110.06864r

[14] Roboflow. (2022). Supervision. Retrieved from https://github.com/roboflow/supervision