Global Scientific and Academic Research Journal of Multidisciplinary Studies
ISSN: 2583-4088 (Online)
Frequency: Monthly
Published By GSAR Publishers
Journal Homepage Link- https://gsarpublishers.com/journals-gsarjms-home/

Underwater Marine Species Detection Using YOLOv8

By

Moayed D. Daneshyari[1], Kanika Warman[2]

[1,2]Department of Computer Science, California State University, East Bay 25800 Carlos Bee Blvd, Hayward, CA 94542, USA

Article History

Received: 01/12/2024
Accepted: 05/12/2024
Published: 07/12/2024

Vol – 3 Issue –12

PP: - 35-44

Abstract

As our oceans play a crucial role in maintaining ecological balance, the need for efficient methods to monitor and study marine life becomes paramount. This paper introduces a novel approach to underwater marine species identification using YOLOv8, a state-of-the-art object detection model. Leveraging a diverse dataset of underwater imagery, this study addresses the unique challenges posed by the underwater environment, such as variable lighting conditions and complex backgrounds. We detail the methodology, encompassing dataset preparation, model configuration, and training parameters specific to underwater species identification. Our experimental results showcase the effectiveness of YOLOv8 in accurately detecting and classifying various marine species. This study marks a pioneering effort in deploying YOLOv8 for underwater marine species identification and lays the groundwork for future advancements. Additionally, the model's architecture is designed with scalability in mind, paving the way for transfer learning to incorporate more parameters and expand the range of identified species, thus enhancing the model's capabilities for broader applications in marine research and conservation.

I. Introduction

The exploration and conservation of marine life are integral to understanding and preserving the delicate balance of our oceans. This study introduces a pioneering application of YOLOv8, implemented through the Ultralytics library, for the purpose of underwater marine species identification. YOLOv8, renowned for its efficiency and accuracy in object detection, is adapted to address the distinctive challenges presented by the underwater environment. The identification of marine objects, particularly underwater species, has traditionally relied on manual methods, visual observation, and limited automated techniques. Current approaches often involve time-consuming and labor-intensive processes, such as human divers visually cataloging species or utilizing basic image recognition algorithms. However, these methods are prone to inaccuracies and lack the precision required for comprehensive marine species monitoring. The inherent challenges of underwater environments, such as varying lighting conditions and complex backgrounds, further exacerbate the limitations of existing methodologies.

In response to these challenges, the application of computer vision, particularly advanced object detection models like YOLOv8, emerges as a transformative solution. By leveraging deep learning and sophisticated algorithms, computer vision offers the potential to enhance the accuracy and efficiency of marine object identification. This study focuses on harnessing the power of YOLOv8 to address the shortcomings of current methods, providing a more accurate and precise approach to underwater marine species identification.

Real-time object detection remains challenging due to variances in object spatial sizes and aspect ratios, inference speed, and noise. This is especially true for our use case, as marine species can change location, scale, rotation, and trajectory very quickly. This conveys the necessity for fast inference speed and thorough model evaluation between low variance classes, object sizes, rotations, backgrounds, and aspect ratios.

The model is trained on a meticulously curated dataset featuring five distinct marine species: fish, eel, jellyfish, lobster, and lionfish. Each of these classes represents a critical component of marine ecosystems, and the accurate identification of these species is essential for marine biologists, ecologists, and conservationists.

This paper provides an in-depth exploration of the methodology employed, encompassing dataset preparation, model configuration, and training specifics, as well as the experimental results demonstrating the model's efficacy in accurately identifying the specified underwater species.

*Corresponding Author: **Moayed D. Daneshyari**

Additionally, this work sets the foundation for future research by highlighting the model's adaptability for transfer learning, enabling the incorporation of additional parameters and the expansion of the classification scope to contribute to a more comprehensive understanding of underwater biodiversity.

II. LITERATURE OVERVIEW

The literature review aims to contextualize the current state of knowledge in the field of underwater marine species identification, object detection, and computer vision. In examining existing studies, we find that the application of object detection techniques for underwater marine species identification has demonstrated promising outcomes. Studies utilizing various models and techniques, have successfully identified marine species. However, common challenges persist, indicating the need for more robust and adaptable models.

Underwater object detection and marine monitoring have seen significant progress in recent studies, each offering distinct insights into challenges posed by the underwater environment. In the realm of underwater robotic vision, MARS (Multi-Scale Adaptive Robotics Vision) emerges as a notable innovation [1]. MARS integrates Residual Attention YOLOv3 with Domain-Adaptive Multi-Scale Attention (DAMSA), enhancing detection accuracy and adaptability across diverse underwater scenarios. Notably, MARS achieves a mean Average Precision (mAP) of 58.57% on the original dataset, demonstrating proficiency in detecting critical underwater objects. Its robust performance is further highlighted by an mAP of 36.16% on the augmented dataset, showcasing adaptability in varying conditions.

Another significant contribution comes from a study focusing on temperate fish detection and classification. This research employs a two-step deep learning approach, utilizing the YOLO object detection technique and a Convolutional Neural Network (CNN) with Squeeze-and-Excitation (SE) architecture [2]. The solution attains state-of-the-art accuracy of 99.27% on pre-training and exhibits viability in post-training, with accuracy reaching 83.68% and 87.74% with and without image augmentation, respectively.

In addressing challenges related to underwater target detection, an improved YOLOv7 network (YOLOv7-AC) is proposed [3]. This network introduces innovations such as the ACmixBlock module, ResNet-ACmix module, and a Global Attention Mechanism (GAM), resulting in enhanced feature extraction and network reasoning speed. The improved YOLOv7 network outperforms the original YOLOv7 model and other popular underwater target detection methods, achieving a mean average precision (mAP) of 89.6% and 97.4% on the URPC dataset and Brackish dataset, respectively.

Furthermore, a comprehensive review and analysis shed light on the challenges and advancements in underwater object detection. Traditional methods often fall short in accuracy and generalization capabilities, necessitating the exploration of deep learning-based techniques. The study emphasizes the importance of addressing challenges unique to underwater environments, providing insights for future research efforts [4]. Additionally, a study focusing on object detection in underwater environments assesses conventional state-of-the-art (SOTA) algorithms, highlighting the need for "ad-hoc" architectures tailored to the underwater setting [5]. The evaluation includes considerations of pretraining, two-stage detectors, and generalization capability, providing valuable guidance for future underwater object detection research.

Moreover, the adaptation of YOLO or similar object detection models to specific domains, such as underwater environments, is an area that requires more exploration. Studies employing YOLO-based approaches, including Underwater target detection algorithm based on improved YOLOv4 with SemiDSConv and FIoU loss function [6], showcase the efficiency and real-time capabilities of these models. These studies collectively contribute to the evolving landscape of underwater object detection, showcasing innovations and insights that pave the way for further advancements in this challenging domain.

In tandem with these advancements, our research makes distinctive contributions to the field of underwater object detection. Unlike previous studies that primarily focus on specific marine species or environmental conditions, our approach aims for a more holistic understanding by encompassing a diverse set of marine objects and environmental scenarios. This broader scope allows for a more comprehensive evaluation of the model's performance in real-world applications, where variability in species and environmental conditions is the norm.

Our work leverages the advanced YOLOv8 model, known for its superior accuracy and speed. YOLOv8 achieves a remarkable 50.2 mAP score at 1.83 milliseconds on the COCO dataset and A100 TensorRT [7]. Our approach capitalizes on key features of YOLOv8, including mosaic data augmentation, anchor-free detection, a C2f module, a decoupled head, and a modified loss function [8]. This not only enhances the precision of object detection but also provides a more efficient and adaptable solution compared to existing models, making it particularly well-suited for real-time applications in underwater image analysis and target detection.

The dataset used in our study is meticulously curated to include a comprehensive representation of specific marine objects, ensuring that the model's training and evaluation encompass a wide spectrum of conditions. This approach positions our research as a valuable addition to the current body of literature, providing insights that extend beyond specific species or environments.

In summary, our research not only builds upon the achievements of prior studies but also introduces novel methodologies and considerations in current underwater object detection research. By offering a more holistic and adaptable approach, our work contributes to advancing the state-of-the-art in underwater computer vision, providing valuable insights for marine robotics, biology research, and environmental monitoring.

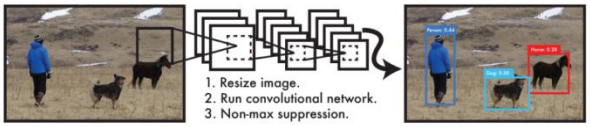
*Corresponding Author: **Moayed D. Daneshyari**

III. Model architecture

In this section, we present a detailed exploration of our methodology, focusing on the YOLOv8 model—a leading object detection framework recognized for its accuracy and speed. Our methodology incorporates key YOLOv8 features, its architecture, explaining why it's the optimal choice for our research objectives. Additionally, we highlight adaptations and innovations tailored for efficient underwater image analysis and target detection.

YOLO: A Comprehensive Overview

The object-detection algorithm "You Only Look Once" (YOLO) made its debut in 2015 through the research paper titled "You Only Look Once: Unified, Real-Time Object Detection" by Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi [8]. This groundbreaking algorithm marked a significant leap forward in real-time object detection, outperforming its predecessors and introducing a unified framework that revolutionized the field of computer vision.



YOLO operates as a single-shot algorithm, classifying an object directly in a single pass. It accomplishes this by utilizing only one neural network to predict bounding boxes and class probabilities, employing a full image as input. Over time, the YOLO model family has seen continuous evolution, with various research teams releasing different versions. The latest iteration, YOLOv8, represents the most recent advancement in this lineage. The subsequent section provides a concise overview of the historical versions of YOLO and their respective enhancements.

This sets itself apart from traditional Convolutional Neural Network (CNN) models by adopting a unique strategy for object detection. Unlike many models employing a two-stage process that includes region proposals and subsequent classification, YOLO accomplishes object detection in a single forward pass through the neural network.

For instance, while both Faster R-CNN and YOLO leverage CNN as their core and share the common objective of enhancing the division of region proposals based on CNN, their frameworks exhibit notable differences.

This distinguishing feature becomes apparent when we contrast YOLO with the R-CNN model [9].

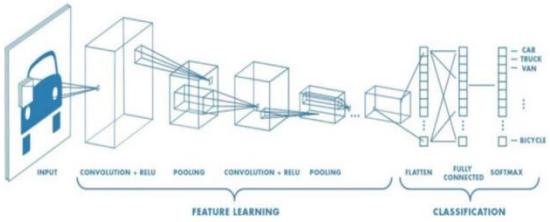


Figure 1: Architecture of CNN

R-CNN utilizes region proposal methods to initially generate potential bounding boxes in an image and then applies a classifier to these proposed boxes. It conducts detection on multiple region proposals, leading to predictions being made multiple times for various regions of an image.

This involves intricate pipelines that are slow and difficult to optimize, primarily due to the necessity of training each individual component separately.

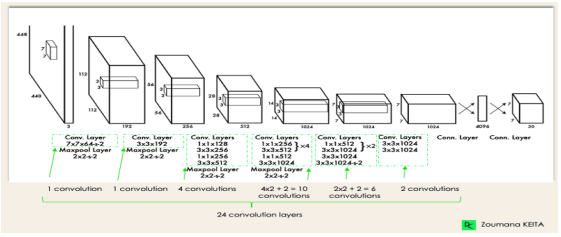


Figure 2: Yolo architecture [11]

In contrast, YOLO takes a different approach by dividing the input image into a grid and directly predicting bounding boxes and class probabilities, ensuring computational efficiency. The YOLO architecture resembles a fully connected convolutional neural network (FCNN), where the image passes through the FCNN once, and the output provides the prediction. This unified model involves a single convolutional network that simultaneously predicts multiple bounding boxes and class probabilities for those boxes. Training on full images and optimizing detection performance directly, YOLO's unified model offers several advantages over traditional object detection methods. The incorporation of anchor boxes in YOLO further enhances the precision of bounding box predictions, allowing effective handling of objects with diverse sizes and aspect ratios.

The real-time processing capabilities of YOLO make it well-suited for applications like video analysis and autonomous vehicles, where low latency is crucial. In comparison to other models, which performs convolution on a region of interest, YOLO achieves detection and classification simultaneously. YOLO demonstrates fewer background errors, making it more efficient than Faster R-CNN in certain scenarios. The end-to-end training and real-time speed of YOLO contribute to its high average precision. While there are several models like Faster R-CNN which also end-to-end training, the process involves more steps compared to YOLO. YOLO strikes a balance between accuracy and speed, positioning it as a practical choice for real-time applications.

The YOLO (You Only Look Once) series of models has become famous in the computer vision world. YOLO's fame is attributable to its considerable accuracy while maintaining a small model size. YOLO models can be trained on a single GPU, which makes it accessible to a wide range of developers. Machine learning practitioners can deploy it for low cost on edge hardware or in the cloud.

Brief History of YOLO

YOLO, introduced in 2015 by Joseph Redmond, has undergone significant evolution within the computer vision

*Corresponding Author: **Moayed D. Daneshyari**

community [10]. In its early days (versions 1-4), YOLO was maintained in C code within a custom deep learning framework named Darknet, crafted by Redmond. The journey of YOLO has witnessed iterative improvements, with each version building upon the successes of its predecessors.

YOLOv1 (2015): The inaugural YOLO model, utilized a single convolutional neural network (CNN) for object detection, distinguishing itself with speed but lagging in accuracy compared to two-stage models at the time.

YOLOv2 (2016): This model introduced anchor boxes to enhance detection accuracy and incorporated the Up-sample layer to improve the resolution of the output feature map.

YOLOv3 (2018): YOLOv3 aimed at improving accuracy and speed, adopting the Darknet-53 architecture, a variant of ResNet tailored for object detection. Enhanced anchor boxes, Feature Pyramid Networks (FPN), GHM loss function, and broader object size and aspect ratio coverage contributed to increased accuracy and stability.

YOLOv4 (2020): - Released in 2020, this model brought notable improvements, including a new backbone network, enhanced training processes, and increased model capacity. It introduced Cross mini-Batch Normalization to enhance training stability.

Figure 3: Timeline of Yolo models [8]

YOLOv5 (2020): An open-source project by Ultralytics, Yolov5 utilized the EfficientDet architecture based on EfficientNet, achieving improved object detection performance. It emerged as the world's state-of-the-art repository for object detection in 2020.

YOLOv6 (2021): YOLOv6 focused on efficiency and memory footprint reduction, employing the SPP-Net (Spatial Pyramid Pooling Network) architecture for handling objects of various sizes and aspect ratios.

YOLOv7 (2022): YOLOv7 introduced the ResNeXt CNN architecture, a multi-scale training strategy, and the Focal Loss technique to address class imbalance in object detection tasks.
YOLOv8 (2023): YOLOv8, the latest version as of 2023, signifies a leap forward in real-time object detection capabilities, offering state-of-the-art accuracy and speed. Represents continuous advancements driven by research and innovation in the computer vision community.

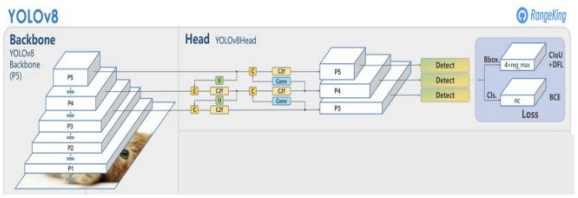
The evolution from YOLOv1 to YOLOv8 showcases continuous research and innovation, pushing the boundaries of object detection in computer vision. This collective effort has enabled real-time object detection systems to operate with unparalleled efficiency and accuracy, contributing to the ongoing progress in the field.

YOLOv8 Architecture

The architecture of YOLOv8 refines the groundwork laid by its predecessors, introducing crucial enhancements to fortify object detection capabilities. Comprising two main components—the backbone and the head—YOLOv8 leverages a modified version of the CSPDarknet53 architecture as its backbone. This architecture, equipped with 53 convolutional layers, incorporates cross-stage partial connections, enhancing information flow between layers for more effective feature extraction.

The head of YOLOv8 plays a pivotal role in the architecture, consisting of multiple convolutional layers followed by fully connected layers. This segment is instrumental in predicting essential information, including bounding boxes, objectness scores, and class probabilities for identified objects.

One of the distinctive features of YOLOv8 is its integration of a self-attention mechanism within the head. This mechanism empowers the model to dynamically focus on different regions of the image, adjusting the importance of various features based on their relevance to the detection task. This adaptive mechanism contributes to the model's overall flexibility.



Another notable strength of YOLOv8 lies in its proficiency in multi-scaled object detection. Leveraging a feature pyramid network, the model demonstrates the capability to detect objects of diverse sizes and scales within an image. This network, consisting of multiple layers tailored to identify objects at various scales, enables comprehensive detection of both large and small objects, enhancing the model's versatility in handling a wide range of scenarios.

YOLOv8 has also introduced anchor-free model architecture. In contrast to previous YOLO versions, YOLOv8 predicts the center of an object directly instead of calculating the offset from a predefined anchor box. Anchor boxes are fixed bounding boxes chosen based on the sizes of objects in the training dataset, and they serve as starting points for boundary box predictions. The advantage of anchor-free detection lies in its flexibility and efficiency, eliminating the need for manually specified anchor boxes. This not only simplifies the model architecture but also speeds up the post-processing step, Non-Maximum Suppression (NMS), which refines candidate detections after inference.

YOLOv8 introduces a C2f module in its backbone instead of the C3 module present in previous versions. The key difference lies in the concatenation of outputs from bottlehead

*Corresponding Author: **Moayed D. Daneshyari**

modules. The C2f module concatenates the outputs of all bottlehead modules, while in C3, only the output of the last bottlehead module is used. Bottlehead modules consist of residual blocks that reduce computational costs, thereby accelerating the training process and improving gradient flow in deep learning networks.

YOLOv8 employs online image augmentation during training, presenting the model with slightly different variations of images at each epoch. One specific augmentation technique, mosaic augmentation, involves stitching four images together. This forces the model to learn objects in new locations, partial occlusion, and against different surrounding pixels. YOLOv8, however, introduces a change by stopping the mosaic augmentation in the last ten training epochs to optimize performance.

In YOLOv8, the head of the network no longer performs classification and regression tasks simultaneously. Instead, these tasks are decoupled, leading to improved model performance. This modification allows the model to independently focus on each task, enhancing its ability to make accurate predictions.

The decoupled head introduces the possibility of loss misalignment, where the model may localize one object while classifying another [15]. To address this, YOLOv8 incorporates a task alignment score. This score, based on positive and negative samples, multiplies the classification score with the Intersection over Union (IoU) score. The IoU score measures the accuracy of a bounding box prediction. Using the alignment score, the model selects top-k positive samples and computes classification loss using Binary Cross-Entropy (BCE) and regression loss using Complete IoU (CIoU) and Distributional Focal Loss (DFL). The BCE loss measures the difference between actual and predicted labels, CIoU loss considers the predicted bounding box's relation to the ground truth in terms of center point and aspect ratio, and DFL optimizes the distribution of bounding box boundaries by focusing more on misclassified samples.

Rationale behind using YOLOv8

YOLOv8 is designed with a focus on extensibility, making it compatible with all previous versions of YOLO. This unique feature allows users to seamlessly switch between different YOLO versions, facilitating easy comparison of performance. For users with existing YOLO models, YOLOv8 provides an opportunity to leverage the latest YOLO technology while maintaining compatibility with previous versions.

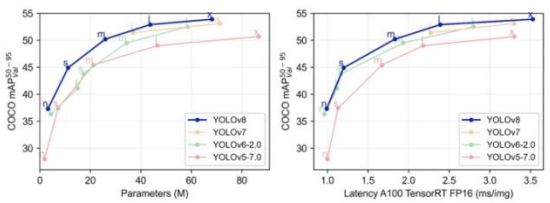


Figure 4: Evaluation metrics of yolov8 for coco dataset

The Ultralytics team has consistently benchmarked YOLOv8 against the COCO dataset, showcasing impressive results compared to its predecessors [13]. These benchmarking results underline the effectiveness of YOLOv8 in delivering a balance between performance, speed, and computational cost, making it a compelling choice for various computer vision applications, including object detection tasks on diverse datasets.

Model	size (pixels)	mAP^{val} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

Figure 5: Benchmarking results for yolov8 on coco dataset

When evaluating object detection performance on the COCO dataset across different YOLO lineages and model sizes, YOLOv8 demonstrates compelling results:

- YOLOv8m Model: Achieves an mAP of 50.2% on the COCO dataset.
- YOLOv8x Model: Achieves a higher mAP of 53.9%, despite having more than double the number of parameters compared to YOLOv8m.

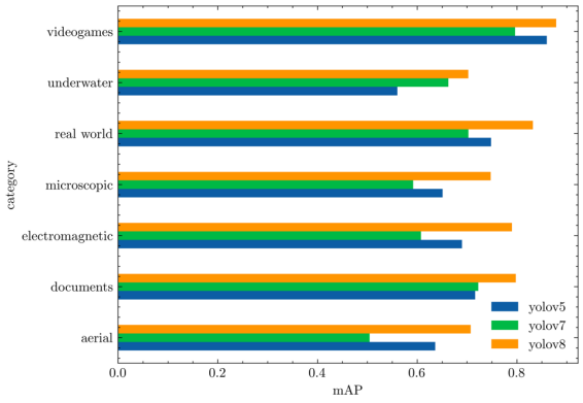


Figure 6: Bar plot shows the average mAP@.50 for each RF100 category

IV. experiment

This section details the comprehensive experimental approach undertaken to evaluate the performance and capabilities of the proposed YOLOv8-based model for underwater marine species identification and object detection. The experiments are designed to validate the effectiveness of the YOLOv8 architecture, assess the impact of key features, and provide insights into its real-world applicability. In the subsequent subsections, we delve into the dataset used, training procedures, evaluation metrics, and present the results, offering a thorough analysis of the model's performance under various conditions. This empirical exploration serves as a critical component in substantiating the claims and contributions put forth in this research.

*Corresponding Author: **Moayed D. Daneshyari**

1. Datasets

This section delves into the composition of the dataset utilized for training and validating the YOLOv8 model, emphasizing a diverse representation of marine species with a primary focus on eel, fish, lobster, lionfish, and jellyfish classes.

The foundation of the dataset was laid with the "Sea Animals Image Dataset" which is also available on Kaggle. This initial dataset included 496 images each for eel and fish (including lionfish), along with 500 images for lobster and 846 images for jellyfish.

To enrich the model's adaptability and extend its capabilities, the dataset underwent a substantial expansion. Additional images were curated from diverse online repositories such as Fathomnet and images.cv. Manual annotation using Roboflow ensured precise labeling for each image [14]. Leveraging Roboflow's capabilities, image augmentation techniques were applied, encompassing adjustments for noise, brightness, saturation, and exposure. Preprocessing steps involved auto-orientation and resizing to a standardized 640x640 pixels.

The final dataset, a result of meticulous expansion and annotation efforts, comprised a total of 6853 images out of which 5644 were training images, 799 validation images, and 410 test images. This comprehensive dataset, serves as a robust foundation for training and evaluating the YOLOv8 model's performance in real-world scenarios.

2. Model Training

The process of training the YOLOv8 model spanned over a period of approximately 3 to 4 months, encompassing various stages from data collection and preprocessing to model selection, training, and evaluation. Ultralytics Yolov8 framework was used for the model training. It simplifies the process and provides invaluable tools for both novice and experienced practitioners, offering a seamless integration of state-of-the-art algorithms, thus expediting the development of highly accurate and efficient object detection models tailored to specific domains. Additionally, Ultralytics YOLOv8 framework provides automated scripts for evaluating metrics and model performance, streamlining the process of assessing the effectiveness of trained models.

The model underwent an iterative learning process with four distinct versions trained using transfer learning. Each iteration involved meticulous adjustments based on the insights gained from the previous results. Adaptations were made to data augmentation techniques, the use of specific weights, alterations in learning rates and epochs, and the transition to GPU utilization. Throughout these iterations, a noteworthy observation emerged—the dataset validation played a pivotal role in influencing the results. It became evident that the quality of augmentation and the intensity of dataset preparation significantly impacted the model's performance. As a result, a key learning from these iterations was the importance of tailoring the dataset to mimic underwater surroundings, accounting for low light conditions and incorporating additional background noise. This involved adapting images from open sources with a specific focus on these environmental considerations.

The foundation of this approach rested on transfer learning, commencing with the initial training utilizing pre-trained weights from YOLOv8 on the COCO dataset. This preliminary iteration, executed on a CPU for 100 epochs, achieved a mean average precision (mAP) of 52% using yolov8n.pt. Subsequent iterations involved continual refinements to both the model and the dataset. Transitioning to the YOLOv8s.pt model and augmenting the dataset with an additional 1370 training images and 394 test images, the training duration extended to 150 epochs, resulting in a notable improvement.

Recognizing the need for accelerated training and deeper image analysis, the training environment shifted to Google Colab, harnessing its T4 GPU capabilities. The dataset underwent expansion, incorporating more images across multiple classes. In Version 3 of the model, a dual strategy was employed, incorporating both pre-trained weights and transfer learning from the preceding iteration. In this iteration, the dataset, consisting of 2932 training images, 418 validation images, and 209 test images, underwent augmentation. Two variations of the model were trained: yolov8m.pt and transfer learning from v2. The yolov8m.pt, trained on 25.9 million parameters on the COCO dataset, was chosen for its superior mean average precision (mAP) compared to yolov8n.pt and yolov8s.pt. The motivation behind training two versions, v3.1 and v3.2, was to assess how the pre-trained weights influenced the learning dynamics of the new model. Given the similarity in datasets between the current and previous models, capturing parameters and achieving faster learning times was facilitated.

Building upon the iterative improvements from version 3, version 4 underwent further enhancements, incorporating changes in the dataset guided by insights gleaned from both versions of model 3. Emphasis was placed on addressing the performance issues observed in specific images in the preceding models. These challenging images were strategically integrated into the training dataset to refine the model's ability to handle complex scenarios.

In this iteration, the training environment transitioned to Kaggle, leveraging the advantages of GPU T4 x2, 30GB storage, and unrestricted training time. Two variations were trained: version 4.1 with the traditional yolov8 weights, while version 4.2, employing transfer learning from the previous version, The results from this underscore the continued refinement of the model, addressing specific challenges identified during the training of earlier versions.

The training process for Version 5 involved two distinct models, each contributing valuable insights into the model's performance and potential areas for improvement. In Version 5.1, which was trained on yolov8n.pt, several enhancements were introduced compared to previous datasets. The image count was increased, and preprocessing and augmentation efforts were intensified. Notably, the class selection was refined to include only five classes, omitting the 'school of

*Corresponding Author: **Moayed D. Daneshyari**

fish' class. Additional lobster images were incorporated, and image lighting was adjusted to simulate underwater conditions. Augmentation techniques, such as noise, saturation, brightness, exposure adjustments, and flips, were applied. The dataset was structured with 5644 training images, 799 validation images, and 410 testing images.

Post-training evaluations, revealed promising results, indicating high precision levels and substantial recall rates across various classes. Upon further inspection of the dataset, the results appeared positive, showcasing higher recall and accuracy than initially evaluated by the in-built model parameters. This observation provides valuable insights for future adjustments and enhancements, highlighting the dynamic nature of model training in object detection tasks.

In conclusion, the iterative training process underscored the effectiveness of transfer learning and dataset augmentation in enhancing model performance. Significant strides were made in accuracy and precision, with version 5.1 showcasing notable achievements.

3. Evaluation Metrics

The iterative training process provided valuable insights into the model's learning and performance evaluation. Initial trainings on the CPU, utilizing pre-trained weights from YOLOv8 such as yolov8n.pt and yolov8s.pt, indicated a positive trajectory. The preliminary iteration V1, executed for 100 epochs, achieved a mean average precision (mAP) of 52% using yolov8n.pt. Subsequent iterations involved continuous refinements to both the model and the dataset.

For V2, Transition to the YOLOv8s.pt model was made, and the dataset was augmented \with an additional 1370 training images and 394 test images. This training was completed in 150 epochs, resulting in a significant improvement. The mAP increased to 66% during training and 68% during testing, accompanied by a precision of 71%.

Version 3.1, leveraging the best.pt from the previous version, achieved notable enhancements, exhibiting a mAP of 68% and a precision of 73%. This version demonstrated significant progress, boasting an 88% recall during testing and improved accuracy metrics.

On the other hand, version 3.2, trained using the yolov8m.pt model, achieved comparable results with a mAP of 60% during training and 71% during testing. Both iterations contributed valuable insights for subsequent model enhancements.

Version 4.1 achieved a mean average precision (mAP) of 70% and a precision of 72%, while version 4.2, employing transfer learning from the previous version, reached a mAP of 71% and a precision of 76.5%. These results underscore the continued refinement of the model, addressing specific challenges identified during the training of earlier versions.

In the training of Version 5, two distinct models provided insights into the model's performance. Version 5.1, trained on yolov8m.pt, showcased enhancements compared to previous datasets. The image count was increased, preprocessing and augmentation were intensified, and the class selection was refined to five classes. With 5644 training images, 799 validation images, and 410 testing images, the model achieved commendable metrics with a 200-epoch approach using yolov8m.pt: 82.9% precision, 75.4% recall, and a mAP of 81.7%. During predictions on Kaggle, the model successfully detected objects in 371 out of 410 test dataset images, achieving an 81% mAP and higher recall.

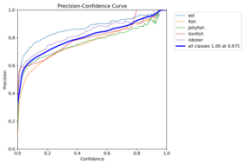


Figure 7: Precision Curve

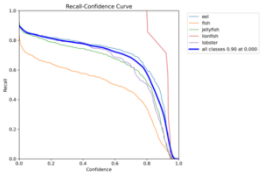


Figure 8: Recall Curve

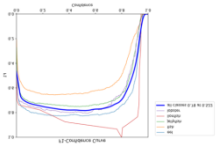


Figure 9: F1 Confidence Curve

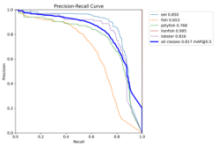


Figure 10: PR Curve

Figure 11: Evaluation metrics from V5

In Version 5.2, trained on weights from 3.1.1, the model underwent additional refinement with a 200-epoch training cycle on Kaggle using the best.pt model. It achieved a precision of 76.6%, demonstrating similar mAP on training and 78% mAP on testing. Insights from evaluation metrics include a precision of 0.798, indicating accurate positive predictions, and a recall of 0.64, suggesting room for improvement. The mAP values at different IoU thresholds highlight the trade-off between precision and recall.

Several visualizations are presented from the training V5 to provide a comprehensive understanding of the model's performance. The Precision-Recall Curve (P_curve) graphically illustrates the trade-off between precision and recall, offering insights into the model's ability to accurately detect objects across varying confidence thresholds. The PR_Curve, an extension of the Precision-Recall Curve with thresholds, allows a detailed examination of precision and recall at specific confidence levels. The Recall Curve (R_curve) focuses on the model's ability to capture true positive instances across different confidence thresholds, shedding light on its sensitivity to detecting objects. Additionally, the F1 Score Curve (f1_curve), showcasing the harmonic mean of precision and recall, provides a holistic view of the model's overall performance. Furthermore, the Normalized Confusion Matrix offers a detailed breakdown of classification results, highlighting the distribution of true positives, true negatives, false positives, and false negatives across different classes.

Some visual inspection of model predictions is shared below which provides an illustrative snapshot, showcasing the model's proficiency in identifying objects within a given image. The bounding boxes accurately delineate the detected

*Corresponding Author: **Moayed D. Daneshyari**

objects, exemplifying the precision achieved by the model in localizing and classifying underwater targets.

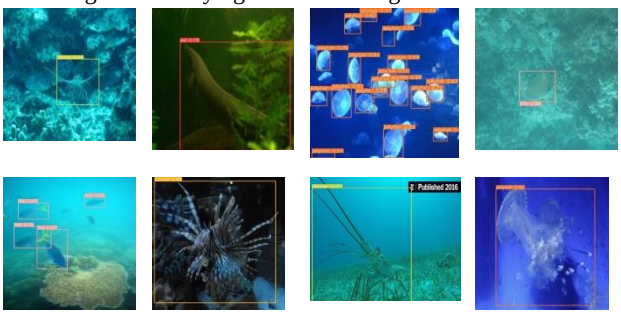


Figure 12: Visual Inspection of Model Predictions on Test Dataset

A comparative visual analysis across various iterations highlights the evolution of the model's performance. Images from earlier iterations reveal instances where the model may have encountered challenges, such as misidentifications or absence of predictions. The below comparison depicts how the model has improved from V3 to V4 by decreasing the absence of predictions for the class fish. In the following comparison, we can see how the 'eel' was not detected in earlier versions mainly due to background lighting and noise. However, the predictions improved with a precision of 73% in V5.

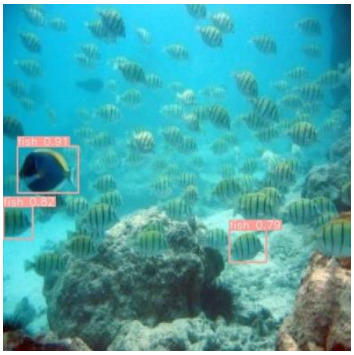


Figure 12: Prediction from V3.2

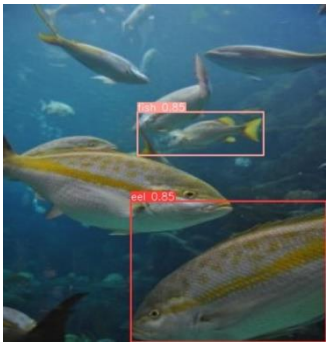


Figure 13: Prediction from V4.1

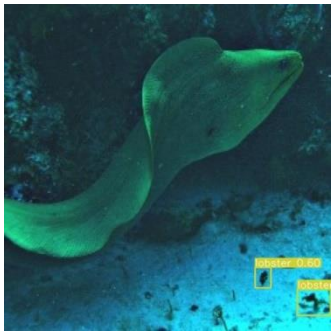


Figure 14: Missed 'eel' prediction in V4

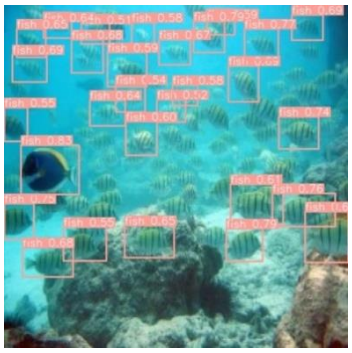


Figure 15: Prediction from V4.1

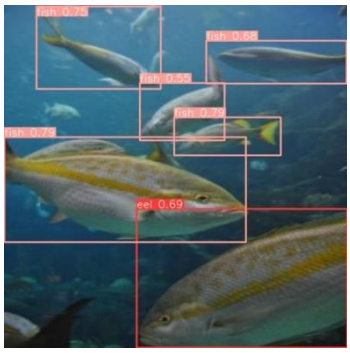


Figure 16: Prediction from V5.1

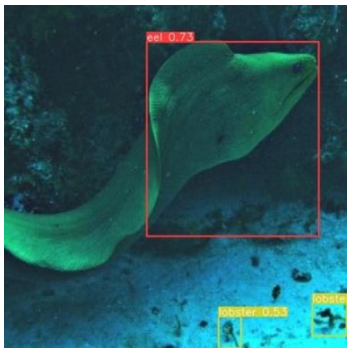


Figure 17: Correctly detected 'Eel' in V5

Also, the latest iterations exhibit remarkable improvements, demonstrating corrected box placements and more accurate predictions. The focus was also maintained on the camouflaged images where the marine species resembles the background color and gets lost in the vision. This iterative comparison underscores the model's learning trajectory and the incremental enhancements achieved over successive training iterations. The below

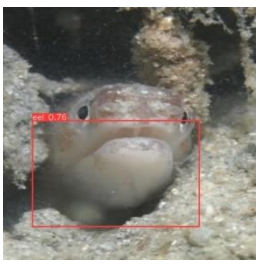


Figure 18: Irregularity in the anchor box

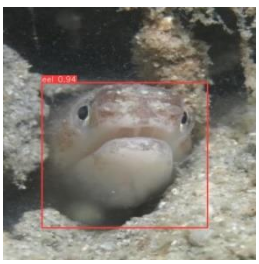


Figure 19: Corrected box placements

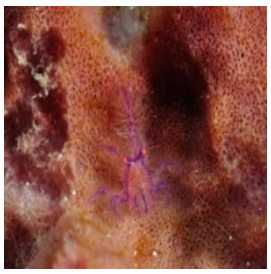


Figure 20: camouflaged lobster not detected in earlier versions

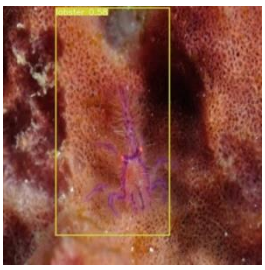


Figure 21: Camouflaged lobster detected in V5

After the successful training of the iterative YOLOv8 model, rigorous testing was conducted, including evaluations on brackish water datasets. The model demonstrated remarkable adaptability, achieving an impressive 88% accuracy in detecting marine objects under brackish conditions. Notably, the recall rate reached an outstanding 97% [this is calculated on the classes on which the model is trained], underscoring the

*Corresponding Author: **Moayed D. Daneshyari**

model's ability to effectively identify and recall instances in challenging environments. This achievement signifies the robustness and practical applicability of the trained model across diverse marine scenarios, emphasizing its potential for real-world deployment.

Figure 23: Testing on Brackish water dataset

V. Conclusion and future directions

In our pursuit of developing a robust object detection model tailored for underwater environments, our research has yielded promising results with favorable metrics. However, as is customary in cutting-edge technology, there remains an ongoing quest for improvement. The evaluation process not only validated our model's capabilities but also illuminated crucial insights for future directions and enhancements.

Our focus on achieving high accuracy in object detection under challenging conditions, such as low lighting and scenarios involving multiple instances of an object, lays a solid foundation. Proactive measures, including dataset adjustments, rebalancing, and augmented augmentation, are poised to further elevate the model's performance across diverse scenarios. It is noteworthy that the scarcity of high-quality underwater datasets and images remains a significant challenge in the development of target detection in underwater environments. Hence, future research efforts will aim to curate a large and diverse set of underwater datasets, employing image enhancement techniques to enhance the overall quality of underwater images crucial for detecting underwater targets.
The iterative training process positions us strategically to address evolving challenges in object detection. Future research endeavors will concentrate on enhancing the model's adaptability to diverse lighting conditions and meticulously refining its competence in identifying and categorizing multiple instances of objects. This commitment seeks to solidify the model's performance in real-world scenarios, reinforcing its utility across a spectrum of applications.

Looking ahead, our exploration extends beyond the confines of the current YOLOv8 architecture. The incorporation of advanced algorithms or techniques holds considerable promise. Future endeavors will focus on innovative approaches to data augmentation and preprocessing, tailored meticulously for underwater settings. This refinement aims to unlock additional potential, fortifying the model's robustness in intricate and challenging conditions. The extensibility of our model for transfer learning, coupled with advanced techniques like model ensembling, opens avenues for heightened accuracy and recall in object detection tasks.

In conclusion, our research not only advanced the understanding of object detection in underwater environments but also set the stage for continuous improvement. The identified areas for enhancement present exciting opportunities for future research, reaffirming our commitment to pushing the boundaries of computer vision and contributing to impactful technological advancements.

REFERENCES

1. Saoud, Lyes Saad, Lakmal Seneviratne, and Irfan Hussain. "MARS: Multi-Scale Adaptive Robotics Vision for Underwater Object Detection and Domain Generalization." 23 Dec. 2023, arXiv:2312.15275.
2. Knausgård, Kristian Muri; Wiklund, Arne; Sørdalen, Tonje Knutsen; Halvorsen, Kim Tallaksen; Kleiven, Alf Ring; Jiao, Lei; Goodwin, Morten. (2022). "Temperate fish detection and classification: a deep learning based approach." Applied Intelligence, 52(6), 6988-7001.
3. Kaiyue Liu, Qi Sun, Daming Sun, Mengduo Yang, Nizhuan Wang, “Underwater target detection based on improved YOLOv7”, 14 Feb 2023, arXiv:2302.06939
4. Andre Jesus, Claudio Zito, Claudio Tortorici, Eloy Roura, Giulia De Masi, “Underwater Object Classification and Detection: first results and open challenges”. 4 Jan 2022, arXiv:2201.00977.
5. Xu, S., Zhang, M., Song, W., Mei, H., He, Q., Liotta, A., "A systematic review and analysis of deep learning-based underwater object detection," Neurocomputing, Volume 527, 2023, Pages 204-232, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2023.01.056.
6. Chengpengfei, Z., Guoyin, Z., Heng, L., Hui, L., Jie, T., Xiaojun, X., "Underwater target detection algorithm based on improved YOLOv4 with SemiDSConv and FIoU loss function," Frontiers in Marine Science, Volume 10, 2023, Article 1153416, ISSN 2296-7745, https://doi.org/10.3389/fmars.2023.1153416.
7. Mehra, A., "Understanding YOLOv8 Architecture, Applications & Features," Blog Post, Apr 16, 2023, [https://www.labellerr.com/blog/understanding-yolov8-architecture-applications-features/]
8. Encord, "YOLOv8 for Object Detection Explained [Practical Example]," Blog Post, Mar 22, 2023 [https://medium.com/cord-tech/yolov8-for-object-detection-explained-practical-example-23920f77f66a].
9. Joiya, F., "OBJECT DETECTION: YOLO VS FASTER R-CNN," IRJMETS (International Journal), Volume 04, Issue 09, September 2022,

*Corresponding Author: **Moayed D. Daneshyari**

DOI: https://www.doi.org/10.56726/IRJMETS30226

10. Redmon, J., Divvala, S., Girshick, R., Farhadi, A., "You Only Look Once: Unified, Real-Time Object Detection," University of Washington, Allen Institute for AI, Facebook AI Research, May 9, 2016, arXiv:1506.02640v5
11. Keita, Z., "YOLO Object Detection Explained," Data Science Blog, September 2022, [https://www.datacamp.com/blog/yolo-object-detection-explained]
12. Sharma, P., "A Practical Guide to Object Detection using the Popular YOLO Framework – Part III (with Python codes)," Blog Post, 26 Aug 2021, [https://www.analyticsvidhya.com/blog/2018/12/practical-guide-object-detection-yolo-framewor-python/].
13. Ultralytics Team, "Introducing Ultralytics YOLOv8," Blog Post, Mar 8, 2023 [https://www.ultralytics.com/blog/introducing-ultralytics-yolov8].
14. Jacob Solawetz, Francesco, "What is YOLOv8? The Ultimate Guide," Blog Post, Jan 11, 2023 [https://blog.roboflow.com/whats-new-in-yolov8/]
15. Gaudenz Boesch, "YOLOv8 Guide," [https://viso.ai/deep-learning/yolov8-guide/].

*Corresponding Author: **Moayed D. Daneshyari**