Developing an Efficient Underwater Object Detection System using AI

Alla Nithin Reddy[1], Mendu Sravanthi[2], Shaik Dariya Hussain[3], Aluri Aditya Choudary[4] and Brij Kishor Tiwari[*5]
{nithinreddyalla369@gmail.com[1], sravanthimendu24@gmail.com[2], hussain272003@gmail.com[3], adityachoudary96@gmail.com[4], aeromantiwari@gmail.com[5] }

Department of Advanced Computer Science and Engineering, VFSTR Deemed to be University, Vadlamudi, Guntur, 522213, Andhra Pradesh, India[1, 2, 3, 4, 5]

Abstract. Underwater object detection is essential for ocean investigation, ecological monitoring, and security systems. However, challenges such as light absorption, scattering, and limited visibility significantly undermines detection accuracy. In this work, we present a robust deep learning-based approach that leverages advanced YOLO variants such as YOLOv9, YOLOv10, YOLOv11, and YOLOv12 for precise detection of underwater objects identification. For further improvement of detection performance, we integrate image enhancement techniques that mitigate underwater deformations and enhance feature extraction. The models are developed and trained and tested with the aquarium dataset, which provides diverse and realistic underwater images. Experimental results show that our enhanced pipeline significantly increases detection accuracy. This work contributes to the development of reliable underwater object detection systems in challenging aquatic environments.

Keywords: Underwater Object Detection, Deep Learning, YOLO, Image Enhancement, Aquarium Dataset, Real-time Detection.

1 Introduction

In the underwater environments it is difficult to detect objects based due to very poor visibility, refraction of light, and color distortion. Underwater object detection such as fish, jellyfish, and starfish is important in the field of marine science research, marine conservation, and underwater reconnaissance. It is being noted that conventional techniques such as sonar and other possible techniques are typically slow, costly, and less precise. With the advent of deep learning revolution, object detection models such as YOLO (You Only Look Once) have been arrived with high impact to detect objects in real time with high precision.

In this paper, we have investigated the with employing different version of the yolo such as YOLOv9 to YOLOv12 to detect the underwater objects precisely. We utilize the aquarium dataset, consisting of different underwater creatures and realistic images. Additionally, image enhancement and augmentation methods such as brightness correction, flipping, and rotation has been employed to achieve the better image detection performance. We used PyTorch to train our model and evaluation of the object detection accuracy.

ICITSM-Part I 2025, April 28-29, Tiruchengode, India

DOI 10.4108/eai.28-4-2025.2357814

2 Related works

In recent years, underwater object detection and classification have gained significant attention due to their applications in marine surveillance, underwater archaeology, and environmental monitoring. With the advancements in deep learning, especially Convolutional Neural Networks (CNNs) and real- time object detection methodologies such as YOLO, researchers have achieved a remarkable performance improvement in underwater visual tasks. Hao Wang [1] evaluated the YOLOv5 models for underwater object detection and reported a maximum accuracy of 69.3% mAP with YOLOv5x, while YOLOv5s offers 62.7% mAP at 50 FPS, demonstrating high efficiency and accuracy in the challenging environments. Mohamed et. al. [2] discussed object detection and evaluation metrics such as Recall, F1-score, mean Average Precision (mAP), and Frame Per Second (FPS) for various YOLOv3 architectures. It investigated the performance for different models on the Brackish Dataset and Google Open Images. Shakil Ahmed [3] has explored the vision-based underwater object detection using OpenCV and Python and achieved high accuracy with gaussian filtering and canny edge detection and recommended for future work that aims to enhance real-time detection and improve accuracy through additional techniques. Hao Wang [4] studied the enhancement of the underwater object detection using an improved Faster RCNN model with Res2Net101 and Soft-NMS, achieving a mAP@0.5 of 71.7% and marking an 3.3% improvement in detecting marine organisms. Minsung Sung [5] has proposes a CNN-based method for underwater object detection using simulated sonar images. It achieves high detection rates (82.5% true positive, 91.8% true negative) by training on images with randomized degradation, enhancing detection capabilities without real sonar data.

Fenglei Han [6] has investigated the enhancement of the underwater images and detects objects using improved CNN structures, achieving a detection speed of 50 FPS and mAP of 90%. It demonstrates effective application in an underwater robot for real-time object detection. Fenglei Han [7] presents a deep CNN method for real-time detection and classification of marine organisms in underwater videos, achieving over 90% mean Average Precision (mAP) and a detection speed of 58 ms, which is notably better for the underwater robotics applications. Akshita Saini [8] presents a method for enhancing underwater images using contrast stretching, adaptive thresholding and sobel edge detection. Results shows the improved object detection performance compared to the other existing methods and highlighting its effectiveness for clearer object boundaries in underwater environments.

Chia-Chin Wang [9] explores YOLOv3 for underwater object detection, successfully identifying fish species in the aquarium. It addresses challenges like light scattering and color distortion. Wang Hao [10] enhances underwater object detection using an improved YOLOv4 model, incorporating deep separable convolution, k-means clustering for bounding boxes, and multiscale training; resulting in a 62.1% of the F1-score and 81.5% mAP@0.5, which is surpassing the original model's performance. Muwei Jian [11-12] reviews advancements in underwater object detection, discussing various methods and datasets. It highlights challenges like visibility issues and emphasizes the need for diverse datasets and improved techniques to enhance detection of small and camouflaged objects. Fenglei Han [13-15] focuses on enhancing underwater images and detecting objects using improved CNN structures, achieving a detection speed of approximately 50 FPS and mAP of 90%, significantly outperforming existing methods for real-time marine organism classification in underwater robotics.

3 Methodology

In this study we have proposed the system that focuses on detecting underwater creatures using real-time object detection algorithms, specifically various versions of the YOLO architecture. Fig. 1 shows the block diagram of the proposed method. The pipeline includes multiple key stages, from preprocessing underwater images to applying YOLO versions for accurate object detection. The goal is to enhance detection accuracy and efficiency in challenging underwater environments, ensuring real-time processing and robust performance across different YOLO iterations.

3.1 Dataset

The dataset used in this work contains 7 classes of underwater creatures, with provided bounding box annotations for each object in the images. The different classes of the object include the 'fish', 'jellyfish', 'penguin', 'puffin', 'shark', 'starfish', 'stingray'. In this study the total size of the dataset used for training, testing and validation are 638 images.

3.2 Preprocessing

3.2.1 Resizing

All input images are resized to a uniform dimension of 640 × 640 pixels. This standardization ensures consistent input shape for the neural network, which is the essential for efficient GPU memory usage and stable model training.

3.2.2 Normalization

The pixel values of each image are normalized using the mean and standard deviation of ImageNet such as μ = [0.485, 0.456, 0.406] and the σ = [0.229, 0.224, 0.225] for the red, green, and blue channels respectively. Let I be the original image with pixel values in the range [0, 255] and then the normalization is applied channel-wise as:

$$I_{norm} = \frac{I/255 - \mu}{\sigma} \quad (1)$$

Where μ = mean

σ = standard deviation

This transformation first scales the pixel values to the [0, 1] range, then standardizes them by subtracting the mean and dividing by the standard deviation. Such normalization helps in reducing internal covariate shift, which refers to the change in the distribution of layer inputs during training and consequently accelerates model convergence and improves overall training stability.

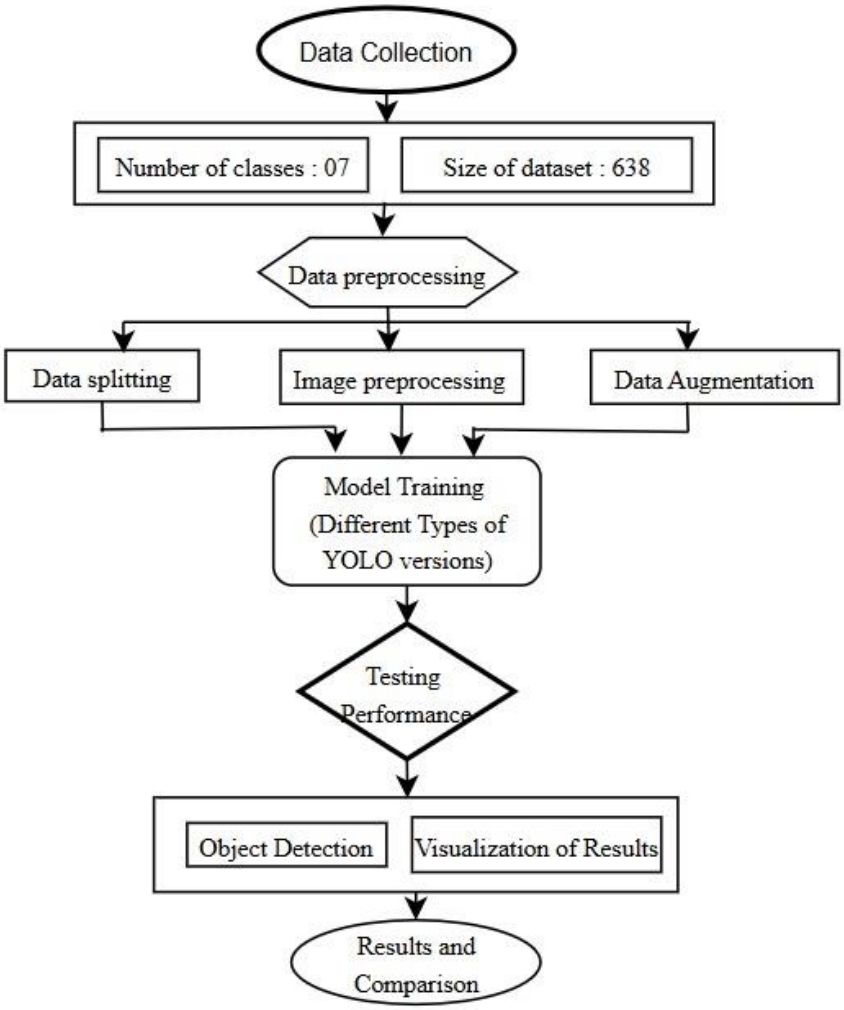

Fig. 1. Block diagram of the proposed method.

3.2.3 Data Augmentation

To improve the generalization and robustness of the model, we can apply the following augmentation techniques such as: a) Horizontal Flip: in this images are flipped horizontally with a probability p = 0.5, helping the model learn orientation- invariant features, b) Random Brightness and Contrast: Random adjustments to the brightness and contrast simulate various lighting conditions found in underwater scenes, c) Shift, Scale, and Rotate: These transformations mimic slight variations in object position and camera angle and d) CLAHE (Contrast Limited Adaptive Histogram Equalization): It enhances the local contrast in areas of low visibility. Let I_{CLAHE} be the output after applying this enhancement to the input I, improving visibility in murky underwater conditions.

3.2.4 Conversion to Tensor

The final step involves in the process is to converting each processed image into a PyTorch-compatible tensor. includes the reordering image dimensions from H × W × C to C × H × W and scaling pixel values from the range [0, 255] to [0, 1]. It makes the images compatible with standard deep learning pipelines.

3.3 Methods of Model Training

YOLO is a real-time object detection algorithm that treats detection as a single regression problem. Unlike the R-CNN based model, YOLO process the entire image in only one forward pass on CNN which results it as an alternative method with fast and efficient.

3.3.1 Basic YOLO Architecture

Basic structure of YOLO is built on a single CNN that directly predicts bounding boxes, class probabilities, and confidence scores from an input image. The architecture consists of: a) Feature Extraction using CNN: A deep CNN processes the input image to extract high-level features, b) Fully Connected Layers for Prediction: The final layer outputs bounding box coordinates, confidence scores, and class probabilities and c) Post-processing (NMS): used for the filtering of the overlapping boxes and finalizes detections. Fig. 2 shows a basic gridded image with bounding box for an underwater object detection.

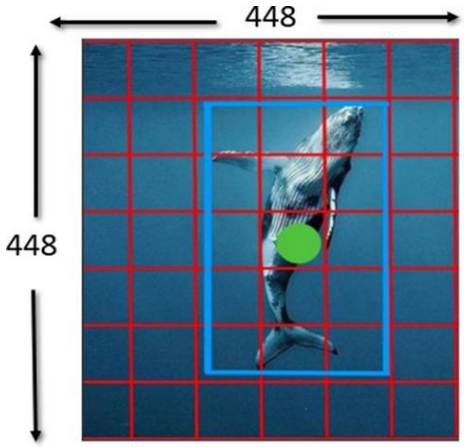

Fig. 2. A gridded image with bounding box for an underwater object detection.

3.3.2 Steps to be Followed for Object Detection using YOLO

Initial step is to resize the given input image as resized to 448 × 448 pixels, and then image has been divided into an S × S grid, in which each grid cell is responsible for predicting an object if the object center falls within the cell. As shown in Fig. 2 the bounding box and inside each image the bounding box for the object detection is represented by (X, Y, W, H). In which (X, Y) is the center point of the bounding box, W is the width, H is the height of the bounding box relative to the entire image. Fig 3 shows the basic representation of the object detection with specific size of the bounding box. In Fig 3, the top-left coordinate of the grid cell (highlighted

in blue color) is denoted as (X_a, Y_a) . Then the center, height and width of the bounding box with respect to the enveloping grid cell can be written as:

$$\Delta X = \frac{X - X_a}{64} \text{ and } \Delta Y = \frac{Y - Y_a}{64} \quad (2)$$

$$\Delta W = \frac{W}{448} \text{ and } \Delta H = \frac{H}{448} \quad (3)$$

Prediction vector of each grid cell can have 5 parameters i.e. $(X_i, Y_i, W_i, H_i, C_i)$, here C_i is the classes of the object in the bounding box. Additionally, 2 bounding box in each grid cell are considered and total 7 different class of the possible object in given dataset. Therefor the total number of parameters in each grid cell can be computed as 17 and final prediction vector representation for each bounding box can be written as:

$$[\Delta X_1, \Delta Y_1, \Delta W_1, \Delta H_1, C_1, \Delta X_2, \Delta Y_2, \Delta W_2, \Delta H_2, C_2, P_1, \dots, P_7] \quad (4)$$

Further we can transform the normalized predictions back to actual image coordinates as for both the bounding boxes can be written as:

$$X_i = 64\,\Delta X_i + X_a \text{ and } Y_i = 64\,\Delta Y_i + Y_a \quad (5)$$

$$W_i = 448\,\Delta W_i \text{ and } H_i = 448\,\Delta H_i \quad (6)$$

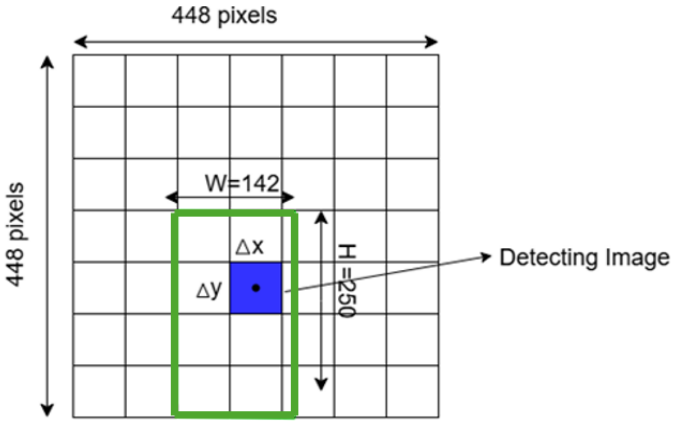

Fig. 3. Representation of the object detection with size of the bounding box.

4 Model Architecture

The YOLO architecture is built upon a Convolutional Neural Network (CNN) and is designed for real-time object detection. Basic structure of the YOLO is shown in Fig 4 from which we can observe that it consists of 24 convolutional layers followed by 2 fully connected layers. The design is inspired by the GoogLeNet architecture, with modifications to suit the needs of object detection.

Convolutional Layers (24 layers): These layers are responsible for extracting spatial and semantic features from the input image. 1×1 convolutions are used to reduce the depth (number

of channels) of feature maps, helping to lower computational cost. A 3×3 convolutions are applied for effective feature ex- traction at local regions.

Fully Connected Layers (2 layers): These layers convert the spatial feature maps into final predictions including bounding boxes and class probabilities. The last convolutional feature map has a size of 7 × 7 × 1024 and this feature map is flattened into a feature vector of size 50176.

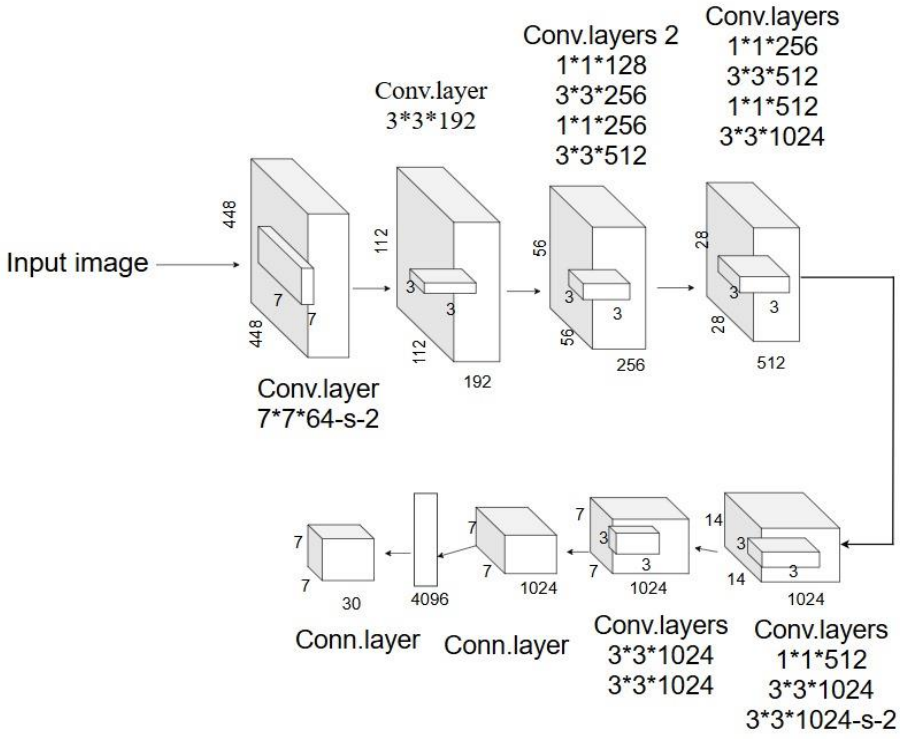

Fig. 4. Basic YOLO Model Architecture.

5 Results and Evaluation

The performance evaluation of underwater object detection system has been conducted using the four different YOLO variants such as YOLOv9, YOLOv10, YOLOv11, and YOLOv12. These models have been trained and tested on the aquarium dataset and assessed based on the important performance metrics of object detection such as precision, recall, mAP@0.5, mAP@0.5:0.95, and validation losses including box loss, classification loss, and DFL loss. Fig. 5(a), Fig. 5(b) and Fig. 5(c) shows the metric performance of precision, mAP@0.5 and classification loss for YOLO9. It illustrates the variation of Precision over the training epochs for the YOLOv9 model and we can observe that after 200 epochs the performance has convergence. This indicates the model's growing ability to correctly identify relevant underwater objects. A steady increase in precision represents that the model is progressively

improving the accuracy of the classifying of the detected underwater objects. From which we can observe that YOLOv9 followed closely, showing comparable results and achieving the lowest box loss (0.668), reflecting its robustness in the localization tasks. Fig. 5(b) illustrates the convergence of mAP at 0.50 (mean average precision at the IoU threshold 0.50) and Fig 5(c) illustrates the convergence of the classification loss. Similarly, Fig. 6(a), Fig. 6(b) and Fig. 6(c) shows the metric performance of precision, mAP@0.5 and classification loss for YOLOv10. Additionally, Fig. 7 shows the performance of YOLOv11 and YOLOv12. From these we can observe that among all the models, YOLOv11 demonstrated the best overall performance with the highest mAP@0.5 (0.967) and mAP@0.5:0.95 (0.833), along with excellent precision (0.955) and recall (0.946). Fig. 8 shows the comparison of different metric performances of the YOLO Versions.

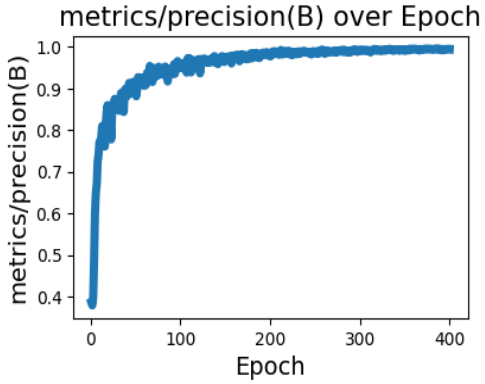

Fig.5 (a) Metric performance of the Precision.

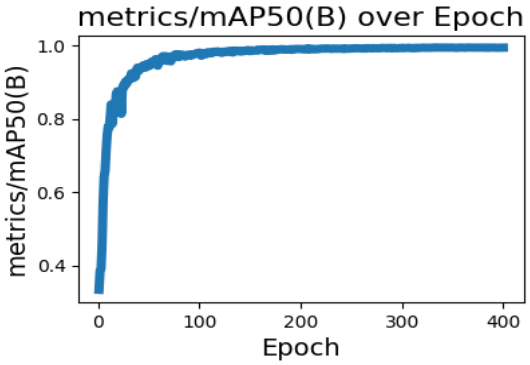

Fig.5 (b) Metric performance of the mAP @ 0.50.

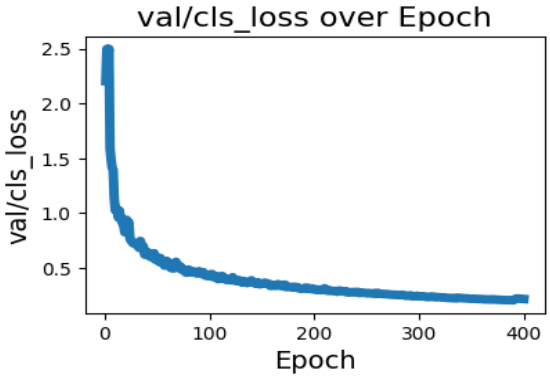

Fig.5 (c) Metric performance of the classification loss.

Fig. 5(a), 5(b), 5(c). Metric performance of different parameters for YOLO version 9.

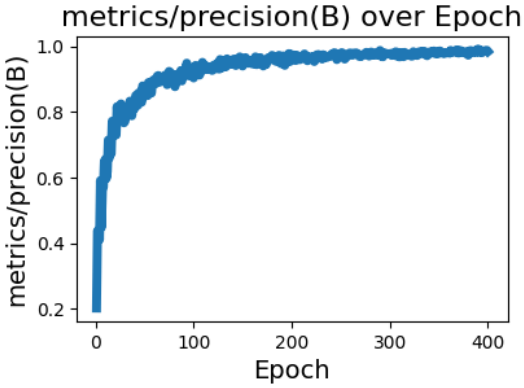

Fig.6 (a) Metric performance of the Precision.

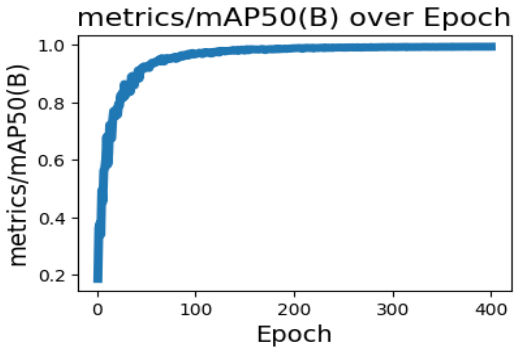

Fig. 6 (b) Metric performance of the mAP @ 0.50.

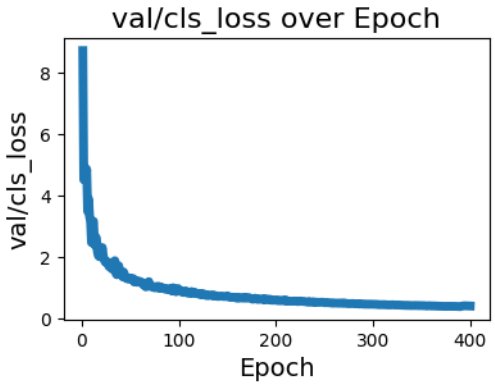

Fig. 6 (c) Metric performance of classification loss.

Fig. 6(a), 6(b), 6(c). Metric performance of different parameters for YOLO version 10.

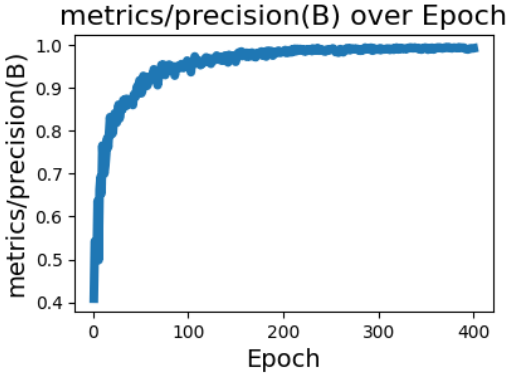

Fig. 7 (a) Metric performance of the Precision.

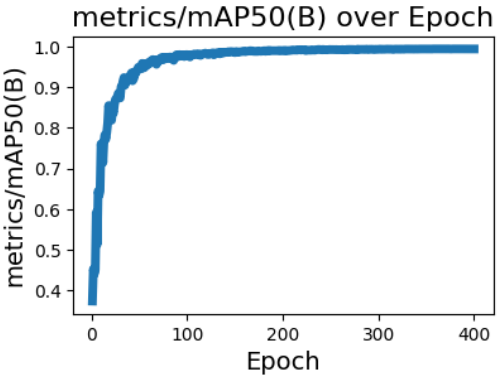

Fig.7 (b) Metric performance of the mAP @ 0.50.

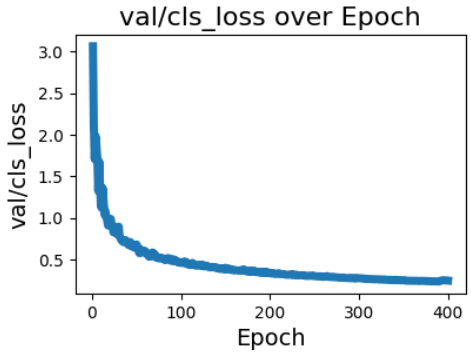

Fig. 7 (c) Metric performance of the classification loss.

Fig. 7(a), 7(b), 7(c). Metric performance of different parameters for YOLO version 11.

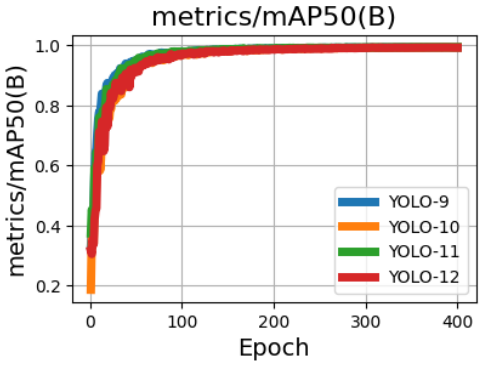

Fig. 8 (a). Comparison of precision of YOLO Versions.

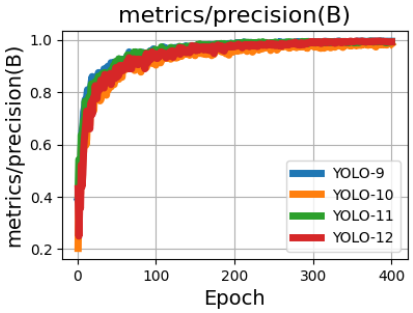

Fig. 8 (b). Comparison of mAP50 of YOLO Versions.

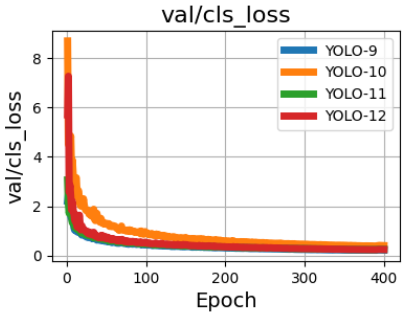

Fig. 8 (c). Comparison of classification loss of different YOLO Versions.

Fig. 8(a), 8(b), 8(c). Comparison of different metric performance of the YOLO Versions.

Table 1 shows the comparison of the performance of YOLO models and it is also observed that the lowest classification loss (0.395), indicating strong capability in accurately detecting and classifying underwater objects. We noticed that YOLOv10 exhibited higher validation losses, particularly box-loss (1.414) and DFL loss (1.784), suggesting reduced stability and potential overfitting during training. YOLOv12 showed moderate performance across all metrics, slightly trailing behind YOLOv9 and YOLOv11. Overall, the experimental results confirm that the integration of image enhancement techniques with state-of-the-art YOLO models significantly improves detection accuracy and reliability in challenging underwater environments. Further, we observe from these results that YOLOv11 and YOLOv9, gives the best performance for real-time underwater object detection applications.

Table 1. Comparison of the performance of YOLO Models.

Metric	YOLOv9	YOLOv10	YOLOv11	YOLOv12
Precision	0.954	0.929	0.955	0.941
Recall	0.944	0.921	0.946	0.933
mAP@0.5	0.965	0.954	0.967	0.955
mAP@0.5:0.95	0.830	0.821	0.833	0.807
Val Box Loss	0.668	1.414	0.670	0.722
Val Cls Loss	0.432	0.830	0.395	0.488
Val DFL Loss	0.886	1.784	0.946	0.934

6 Conclusion

In this work, we developed an efficient underwater object detection system by integrating advanced deep learning models with image enhancement techniques and by leveraging YOLO variants such as YOLOv9, YOLOv10, YOLOv11, and YOLOv12. We have evaluated their performances on the aquarium dataset, which provides realistic underwater object detection with very high precision and can be used in the diverse underwater scenarios. Among the tested models, YOLOv11 and YOLOv9 demonstrated superior detection accuracy, robustness, and

generalization capabilities, making them ideal for deployment in real-time underwater object detection applications. The image enhancement has significantly mitigated the challenges posed by underwater environments such as light distortion and low visibility and thereby improving feature extraction and model performance. This study highlights the potential of deep learning-based solutions for reliable underwater object sensing and sets a foundation for further advancements in marine research, ecological monitoring, and autonomous underwater systems.

References

[1] Wang, H., Sun, S., Wu, X., Li, L., Zhang, H., Li, M., & Ren, P. (2021, September). A yolov5 baseline for underwater object detection. In OCEANS 2021: San Diego–Porto (pp. 1-4). IEEE.

[2] Asyraf, M. S., Isa, I. S., Marzuki, M. I. F., Sulaiman, S. N., & Hung, C. C. (2021). CNN-based YOLOv3 comparison for underwater object detection. Journal of Electrical and Electronic Systems Research (JEESR), 18, 30- 37.

[3] Ahmed, S., Khan, M. F. R., Labib, M. F. A., & Chowdhury, A. E. (2020, February). An observation of vision based underwater object detection and tracking. In 2020 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE) (pp. 117-122). IEEE.

[4] Wang, H., & Xiao, N. (2023). Underwater object detection method based on improved faster RCNN. Applied Sciences, 13(4), 2746.

[5] Sung, M., Lee, M., Kim, J., Song, S., Song, Y. W., & Yu, S. C. (2019, October). Convolutional-neural-network-based underwater object detection using sonar image simulator with randomized degradation. In OCEANS 2019 MTS/IEEE SEATTLE (pp. 1-7). IEEE.

[6] Han, F., Yao, J., Zhu, H., & Wang, C. (2020). Underwater image processing and object detection based on deep CNN method. Journal of Sensors, 2020(1), 6707328.

[7] Han, F., Yao, J., Zhu, H., & Wang, C. (2020). Marine organism detection and classification from underwater vision based on the deep CNN method. Mathematical Problems in Engineering, 2020(1), 3937580.

[8] Saini, A., & Biswas, M. (2019, April). Object detection in underwater image by detecting edges using adaptive thresholding. In 2019 3rd In- ternational Conference on Trends in Electronics and Informatics (ICOEI) (pp. 628-632). IEEE.

[9] Wang, C. C., Samani, H., & Yang, C. Y. (2019, December). Object detection with deep learning for underwater environment. In 2019 4th International Conference on Information Technology Research (ICITR) (pp. 1-6). IEEE.

[10] Hao, W., & Xiao, N. (2021, December). Research on underwater object detection based on improved YOLOv4. In 2021 8th International Con- ference on Information, Cybernetics, and Computational Social Systems (ICCSS) (pp. 166-171). IEEE.

[11] Jian, M., Yang, N., Tao, C., Zhi, H., & Luo, H. (2024). Underwater object detection and datasets: a survey. Intelligent Marine Technology and Systems, 2(1), 9.

[12] Han, F., Yao, J., Zhu, H., & Wang, C. (2020). Underwater image processing and object detection based on deep CNN method. Journal of Sensors, 2020(1), 6707328.

[13] Wulandari, N., Ardiyanto, I., & Adi Nugroho, H. (2022). A Comparison of Deep Learning Approach for Underwater Object Detection. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *6*(2), 252 - 258.

[14] C. -H. Yeh *et al*., "Lightweight Deep Neural Network for Joint Learning of Underwater Object Detection and Color Conversion," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 11, pp. 6129-6143, Nov. 2022, doi: 10.1109/TNNLS.2021.3072414.

[15] Zhao S, Zheng J, Sun S, Zhang L. An Improved YOLO Algorithm for Fast and Accurate Underwater Object Detection. *Symmetry*. 2022; 14(8):1669.