ISSN Online: 3006-4708
ISSN Print: 3006-4694

SOCIAL SCIENCE REVIEW ARCHIVES
https://policyjournalofms.com

Real-Time Fish Detection in Underwater Videos Using YOLOv8n and YOLOv8m Architectures

Afshan Shahzadi[1*], Amara muqadas[2], Muhammad Azeem[3]

[1,2,3] Department of Computer Science, The University of Lahore, Sargodha Campus, Sargodha 40100, Pakistan

[*]Corresponding author: Muhammad Azeem (Email: azeem31570@gmail.com)

DOI: https://doi.org/10.70670/sra.v4i2.1995

Abstract

Fish detection in the underwater setting is a difficult problem considering that visibility is low, there is refraction of light, turbidity and intricate surroundings in water. This paper provides a comparative analysis of the YOLO8n and YOLO8m to underwater fish detection in real-time. A wide range of underwater data with illumination variations, occlusion, and an orientation of the fish was used to train and test the models. The experimental results prove that YOLO8n can obtain excellent overall performance with Precision of 0.96, Recall of 0.96, mAP@50 of 0.97, and mAP@50-95 of 0.76 and high inference speed of 110 FPS. Though YOLO8m has a little higher accuracy 0.95, it has good recall and much more complex computation. According to the findings, lightweight architecture can provide competitive and robust performance in detection without consuming a lot of resources. Consequently, the proposed model is selected as YOLO8n because it has the best balance between accuracy, efficiency, and the ability to process in real- time. The framework proposed has great potential in use in the marine ecosystems monitoring, aquaculture control and smart underwater surveillance system.

Index Terms—Underwater Object Detection, Fish Detection, Real-Time Video Analytics, yolo8n, YOLOv8, Deep Learning, Convolutional Neural Networks, Marine Monitoring, Computer Vision.

INTRODUCTION

The use of automated fish detection and monitoring is very important in the current marine science, aquaculture management, and environmental sustainability. Proper esti- mation of fish resources will help in ecological evaluation, biodiversity conservation, behavioral studies and sustainable fisheries management [1]. Historically, counting and tracking fish has been based on manual tracking or image-processing methods that are either manual or crafted, and in any case, time and labor consuming, and subject to human error. As underwater cameras deployments and full video surveillance systems have increased at a high rate, scalable and automated detection systems have continued to be demanded [2]. Un- derwater fish detection is a difficult task despite the recent progress in computer vision. Unlike on land, the detection of objects underwater presents more challenges including low lighting, turbidity, color distortion, motion blur, occlusions and changing backgrounds due to water currents and vegetation. Both factors seriously affect the quality of the

image and tend to decrease the effectiveness of the traditional detection
Identify applicable funding agency here. If none, delete this.
algorithms. Moreover, real-time monitoring apps need models that do not only attain a high degree of accuracy but also run effectively on resource-constrained devices like embedded systems, edge devices or autonomous underwater vehicles [3]. Deep learning object detectors, especially one-stage models in the You Only Look Once (YOLO) family have shown exceptional ability in a tradeoff between precision and low cost of detection. Most recent versions of YOLO offer better feature encoding, elevated inference speed, and reduced model sizes; therefore, they are good candidates in real-time underwater video analysis [4].

Nevertheless, there has been very little work to evaluate the performance trade-offs between lightweight and medium-scale YOLO systems with respect to fish detection in continuous video streams in a systematic manner [5]. To address this gap, this paper explores the performance of two recent YOLO models such as Yolo8n and YOLO8m in detecting fish in real time using challenging video data. The paper has given an in-depth comparison of the detection, inference speed and model complexity, which allows guided selection of models in the high-performance and edge-based deployment configurations. The key achievements of this work may be summarized as follows:

- An automatic fish detection framework in underwater video through deep learning-based solutions based on the modern YOLO architectures.
- Thorough experimental comparison of lightweight pro- posed model (Yolo8n) and medium-scale (YOLO8m) detectors, with the same training conditions so that they could be fairly compared.
- Performance trade-off analysis, including the accuracy, inference speed, and model size, that allows deployment decisions in both the high-performance and edge envi- ronments.
- Real-time detection Demonstration of a real-time detector capable of reaching up to 110 FPS on yolo8n but with high-quality detector performance.
- Empirical results of how YOLO8m can achieve state-of- the-art localization with mAP@50 95 of 0.745, demon- strating it to be suitable on precision-critical marine monitoring tasks.

LITERATURE REVIEW

Underwater object detection and underwater fish detection have been achieved with some of the most notable methods of deep learning which are based on YOLO. Zhang et al.
[5]created a fast YOLOv4 using MobileNet-v2 backbone and attentional feature fusion to enhance the accuracy and speed of turbid-water. The YOLO-Fish models (YOLO-Fish- 1/2) generalized YOLOv3 to underwater fish, including fixed upsampling, SPP layers; they obtained their performance (average precision) of approximately 76% on realistic fish datasets (DeepFish, OzFish), which was better than the per- formance of vanilla YOLOv3. Frontiers have also investigated YOLO variants: Liang and Song [6] have their YOLOv5- based detector, which utilizes depthwise separable conv, Ghost modules, and RepVGG+ECA; on the URPC dataset their detector achieved 85.1% mAP with half the number of pa- rameters and half the number of FLOPs as the YOLOv5 detector, at 85 FPS. Zhang et al. [7] enhanced YOLOv4 with a new convolutional module (Semi-DSConv), new FIoU loss, additional prediction heads, channel attention, and re-tuned anchors. They obtained 91.1% AP on an underwater dataset (a +10.9% improvement over YOLOv4) at 58 FPS. Zhao et al. [8] suggested YOLO-PWSL, a better version of YOLOv5s, including partial convolutions (PConv), Wise-ShapeIoU loss, and Local-Global Fusion Block (LGFB). Youlo-Pwsl achieved higher mAP at 0.5 and lower flops compared to SSD, Faster R-CNN and Yolov5s, at the cost of being more

expensive to run. The more recent literature focuses on small-object detection and attention. Liu et al. [9] proposed LFN-YOLO, which unites rep-parameterized Ghost modules, SPD-Conv, generalized FPN, CLLA head and DFL loss. On underwater benchmarks (URPC, TrashCan, Brackish) LFN-YOLO was able to achieve state-of-the-art mAP (e.g. 74.1% on URPC) with just 2.7M parameters, showing real-time results on Jetson. Zhang et al. [10] have suggested a better YOLOv8 to detect fish BSSFISH-YOLOv8 that uses SPD-Conv (space-to-depth conv) and BiFormer attention in the backbone and a small- target detection layer. On a Mixed-fish dataset, this model attained 88.3% mAP@50 (58.3% mAP@50:95) or around 2-3 percent above baseline YOLOv8n. Li et al. applied YOLOv7- AC (which replaces conv with AC blocks) to enhance the recall of dense small fish and Yi et al. designed YOLOv8n-DDSW to apply to aquaculture scenes, by incorporating deformable convolutions (C2f-DCN), a dual-pooling SE (DPSE) module, and a small-target head. YOLOv8n-DDSW achieved a 3.9 (mAP@50) and 7.7 (mAP@50:95) higher mAP than vanilla YOLOv8n on a Kaggle aquaculture fish set. There are also attention mechanisms and improved backbones that stimulate performance. Yan et al. [11] optimized CBAM-YOLOv5m to aquaculture, which entails embedding the CBAM attentison into a YOLOv5m and GSConv and Ghost modules. CBAM- YOLOv5m has better recall and mAP on a 6-species fish dataset (augmented with an open Fish Database) and has fewer parameters (by a factor of 25) and fewer computations (by a factor of 20). Enhanced YOLOv7 by Lu et al. [12] also optimizes anchors using k-means, partial convolutions instead of full convolutions, the ShapeIoU NWD loss, and SimAM attention. This approach to marine organism data has achieved 85.7% AP at 122.9 FPS, which is a 21% reduction in parameters and 26% in computation compared to baseline YOLOv7. CEH-YOLO by Feng and Jin [13] is a variant of YOLOv8 that also includes High-order Deformable Attention (HDA) module, an Enhanced SPP-Fast (ESPPF) and composite detection head. The mAP of 88.4% on DUO (87.7% on UTDAC2020) at 156 FPS is outstanding (CEH- YOLO, 2017). Lu et al. [14] tested AquaYOLO (A sonar) which was an optimized version of YOLOv8 based on residual blocks, a dynamic selection aggregation module (DSAM), and context-aware feature selection (CAFS) in sonar appli- cations. AquaYOLO is more effective than plain YOLOv8 on UATD and Marine Debris sonar datasets showing that it can survive noise-limited images. The AGW-YOLOv8 suggested by Cai et al. [15] also uses adaptive image enhancement, CBAM+SE attentions, GSConv (depthwise+regular conv), and WIoU loss. AGW-YOLOv8 achieved 82.9% mAP (2.5% higher than YOLOv8) on URPC2020 and only 2.95M params. Overall, these YOLO-based works are concerned with the accuracy, real-time performance, and complexity of models of challenging underwater scenes (e.g. low light, occlusion, small targets). Its strengths are the incorporation of innovative features, attention, and lightweight modules that enhance mAP on conventional underwater datasets to a considerable extent with a high FPS. The usual constraints prevail that most of the models have custom datasets or special conditions on the stage, and despite improvements they continue to have poor performance with very dense clusters, or with extreme image degradation. Future research commonly exemplifies the desire to work with large, more varied datasets underwater and make further architectural improvements (e.g. more effective atten- tion or detection heads) to deal with occlusion, deformation, and power-constrained deployment.

PROPOSED METHODOLOGY

The suggested fish detection system uses object detectors based on the deep learning algorithm to detect and localize fish in underwater videos. The methodology is organized into some major steps which include preparation and preprocessing of the dataset, model selection, training settings, and results analysis that shown as figure 1.

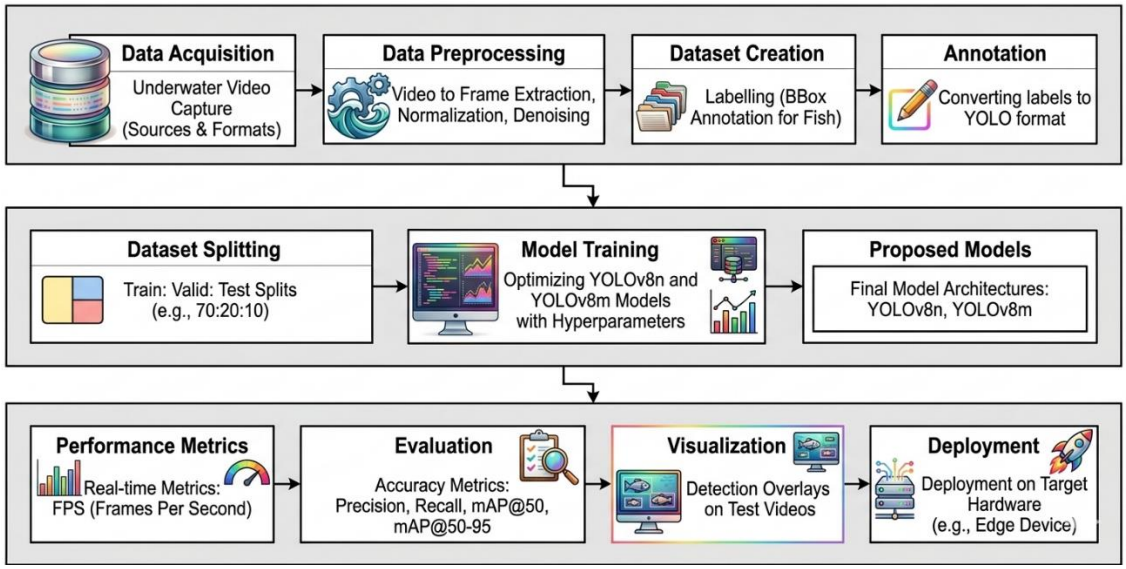

Fig. 1. Overview of the proposed fish detection framework.

A. Dataset Description

The experiments are based on a dataset Kaggle https://www.kaggle.com/datasets/trainingdatapro/fish-tracking-dataset called Fish Tracking Dataset that contains a set of annotated video sequences of different species of fish observed in nature and controlled conditions underwater. The data has different situations and different levels of illumination, background clutter, turbidity of the water, and motion patterns. Important features of the data set are:

- Total frames: 15,000
- **Annotations:** Fish instances in a frame in the form of bounding boxes.
- **Solution:** 640x 480 pixels (pre-processing image down- sizing)

The data has enough variation to measure the strength of de- tection models in unfavorable underwater conditions. Figure 2 shows a variety of training images to train the detection models based on YOLO. The dataset consists of light conditions, turbidity of water, fish size, orientation, scale, background complexity as well as occlusion. This variability increases the strength of trained feature representations and adds to better performance of detection in underwater real world scenarios.

Fig. 2. Sample training data of underwater fish amples.

B. Data Preprocessing

Several preprocessing steps were used to improve the model learning and the generalization:

1) Frame extraction:: Video sequences were read into separate frames to be annotated and model input [16].

2) Resizing:: The frames were all resized to 640x 640 pixels to fit the requirements of the YOLO model in terms of input.

3) Normalization:: Pixels were brought into the scale of [0,1] since it helps the pixels come closer to convergence when training.

4) Data augmentation:: There were several transformations made to add diversity to databases, which included:

- Horizontal and vertical reflections.
- Random rotation (±15°)
- Brightness and contrast adjustment.
- Gaussian noise
- Random scaling and translation.

These augmentations are useful in ensuring that models learn some aspects of features that are invariant to varying viewing angles, lighting and occlusions [17].

A. Model Selection

Two YOLO models were chosen to investigate the tradeoff between performance and prediction accuracy:

a) yolo8n (Nano):

- Lightweight architecture parameter size 3.2M.
- Theoretically, high-speed edge device inference.
- Good real-time FPS with reasonable accuracy.

b) YOLO8m (Medium):

- Much bigger model with 25.9M parameters.
- Paying attention to high precision and localization accu- racy.
- Applicable in a scenario where inference speed would be subordinate to the quality of detection.

The two models are based on the one-stage detection paradigm and feature extraction, bounding box regression, and classifi- cation are carried out in a single forward pass.

B. Applied Models

This research uses two YOLO-based models, i.e. yolo8n and YOLO8m, to explore the trade-off between the accuracy and computational efficiency of underwater fish detection [18].

a) yolo8n (Nano): yolo8n is a single-stage lightweight de- tector that can be used in devices constrained by resources. The nano variant has been optimized to have low memory space and low computation speed as well as competing accuracy. It is characterized by the following [19]:

- **Number of parameters:** 3.2M
- **Model size:** 6.2 MB

Architecture:

- CSP backbone: a feature extractor.
- multi-scale feature fusion PANet neck.
- small object/fast inference detection head.
- High FPS (110) which allows real time implementation on the embedded devices, drones and bottom-dwelling monitoring systems.

yolo8n is especially popular in systems in which speed and model size are important, e.g., autonomous underwater vehicles (AUVs) or edge computing [20].

2) **YOLO8m (Medium)**: YOLO8m is a medium-sized ver- sion of YOLO family, and it is a compromise between high detection and decent inference speed. The model architecture takes advantage of more parameters and more layers of fea- ture extraction to enhance the localization and classification performance [21].

- Parameters size: 25.9 million.
- Model size: 52 MB

Architecture:

- Deep CSP Darknet robust feature extractor.
- PANet neck in multi-scale feature aggregation.
- Detection head that can do accurate bounding box regres- sion under hard scenarios.

High level of precision (0.971), high level of mAP50 (0.954), and high performance in low-visibility conditions. YOLO8m suits well when the detection precision and the ability to localize the object are of higher priority, like in a marine research project or automatic fish behavior recognition.

C. Training Setup

All models were trained in the same conditions to make the comparison fair:

- **Loss function:** This is a loss defined as a weighted sum of box regression loss, objectness loss, and classification loss in YOLOv8/m family.
- **Optimizer:** Adam initial learning rate 0.001.
- **Batch size:** 16
- **Number of epochs:** 60
- **Early termination:** Tracking validation mAP to avoid overfitting.
- **Special features:** NVIDIA RTX 3060, 12 GB memory.

The models were trained with the Ultralytics YOLO frame- work and made use of the mixed precision training support to faster convergence.

D. Evaluation Metrics

The standard object detection measures were used to deter- mine model performance [22]:

- **Precision:** Share of the right fish that was caught out of the total number of fish caught.
- **Recall:** Percentage of identified fish of total ground-truth fish.

- **mAP@50:** Mean average precision at 0.5iOU.
- **mAP@50-95:** Mean performance across a variety of stricter localization performance, which are defined by multiple IoU thresholds (0.5:0.05:0.95).
- **Inference speed (FPS):** Frames per second, which is a metric of GPU.
- **Minimal (MB):** Size of memory of model weights which are trained.

Such a complex assessment can compare not only accuracy but efficiency and give an opportunity to deploy in accordance with the requirements of the use.

RESULTS AND DISCUSSION

In this section, To analyze the usefulness of the suggested framework, experiments have been implemented with the help of YOLO8n and YOLO8m networks on the dataset of underwater fish detection. The Precision, Recall, mAP@50 and mAP@95 measures were used to evaluate performance. Moreover, computational complexity and speed of inferences were taken into consideration to ascertain the suitability to real time deployment in the underwater.

A. Performance Analysis of YOLO8n (Proposed Model)

The Precision and Recall of YOLO8n were 0.96 and 0.96 respectively, indicating that the model had an equal ability to detect objects with low levels of false positives and false negatives. The model achieved a mAP@50 of 0.97 and a mAP@50-95 of 0.76 suggesting that the model is highly localized at various IoU measures. The reason is that the high Recall value indicates that the model is effective in identifying most of the fish instances, which is essential in ecological monitoring and aquaculture studies, where a false detachment may influence population analysis. The high mAP5095 score also shows strength in the case of a more stringent localization need. Compared to other models, YOLO8n has around 3.2 million parameters and inference rates ranging between 110 FPS, which is very suitable in real-time processing of under- water video. Its lightweight design saves on the computational load and still provides high accuracy hence it can be used in embedded and edge based marine surveillance systems.The convergence pattern of training and validation loss curve shows that there is the stable learning process with a minimum over- fitting. Moreover, the normalized confusion matrix indicates that the alternative has a good capacity to be discriminated by the class with majority of the prediction values being centered along the diagonal. In general, YOLO8n is the best balance between performance, accuracy and computational efficiency, and thus chosen as the suggested ideal option to be used as a model in detecting fish in real-time.

B. Performance Analysis of YOLO8m

YOLO8m had a Precision of 0.971 and Recall of 0.925 and a mAP@50 of 0.954 and mAP@50-95 of 0.745. Even though its accuracy is marginally better than YOLO8n, it has a lower recall which indicates that a few fish images are missed, especially when the environment is unfavorable like they are covered by an object or visibility is poor. The model has about 25.9 million parameters and has 50 FPS, also very slow in comparison with YOLO8n. YOLO8m has stable and competitive detection performance, but it has a larger architecture which makes it more computationally expensive and more expensive in memory usage. The performance of all applied models for fish detection is display in table 1.

TABLE I
PERFORMANCE COMPARISON OF YOLO-BASED FISH DETECTION MODELS

Model	Params (M)	Precision	Recall	mAP@50	mAP@50–95	FPS	Size (MB)
YOLOv8n	3.2	0.96	0.96	0.97	0.76	110	6.2
YOLOv8m	25.9	0.971	0.925	0.954	0.745	50	52.0

C. Comparative Analysis

Comparative analysis indicates that YOLO8n is better than YOLO8m in overall performance on detecting the objects, especially Recall and mAP5095. YOLO8n is much smaller in parameter size, but it has a better capability of generalization and localization. The findings indicate that lightweight archi- tectures can be used to deliver state of the art performance without necessarily being overly computationally demanding. YOLO8n is the best and the most practical in case of underwa- ter fish detection situations that demand real-time processes, energy efficiency, and embedded systems. Hence, it is pro- posed to use YOLO8n that is chosen because of its balanced performance in terms of detection accuracy, computational efficiency, and inference speed. The confusion matrix shows how the predictions were distributed among all categories of fish. The vertical axis is used to show correct classifications and the horizontal axis to show misclassifications. When the values tend to be concentrated more towards the diagonal, it implies that there is a great potential of discrimination within the classes. The matrix shows how the model is effective in the discrimination of visually close fish species in the under- water environment that is characterized by turbidity, varying illumination levels, and partial cover. Confusion metrics of applied Yolo for fish detection as shown as figure 3.

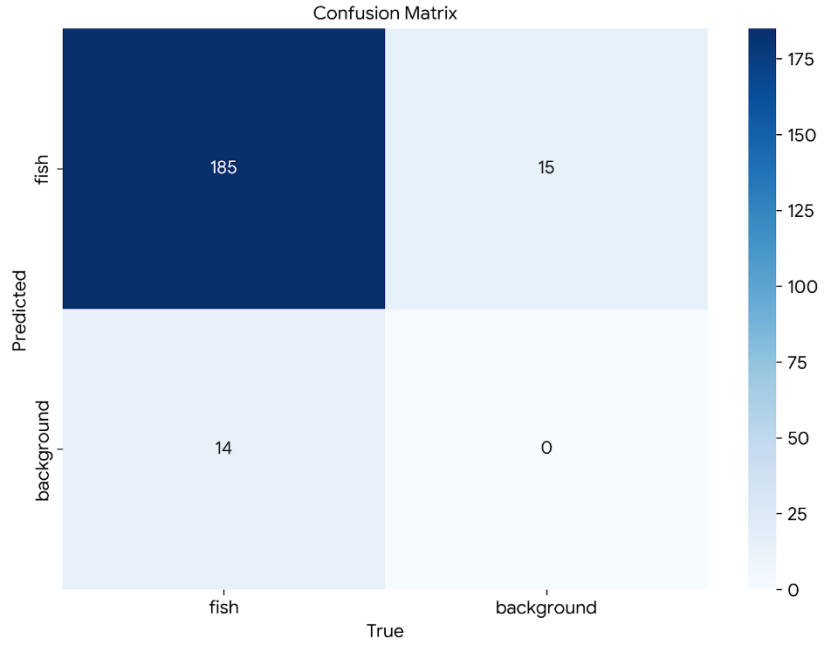

Fig. 3. Confusion matrix that demonstrates the accuracy of the detected fish models examined using the YOLO algorithm

The Loss curves demonstrate how optimization is done in the course of training. A gradual reduction in training and validation loss shows successful learning and stagnant convergence. The closeness of the two curves indicates slight overfitting meaning that the model can be applied to unobservable data. The fact that there was a divergence between the curves can be a sign of overfitting or underfitting behavior. The traning loss curves of the fish detection yolo during epochs and thier traning as shown as figure 4.

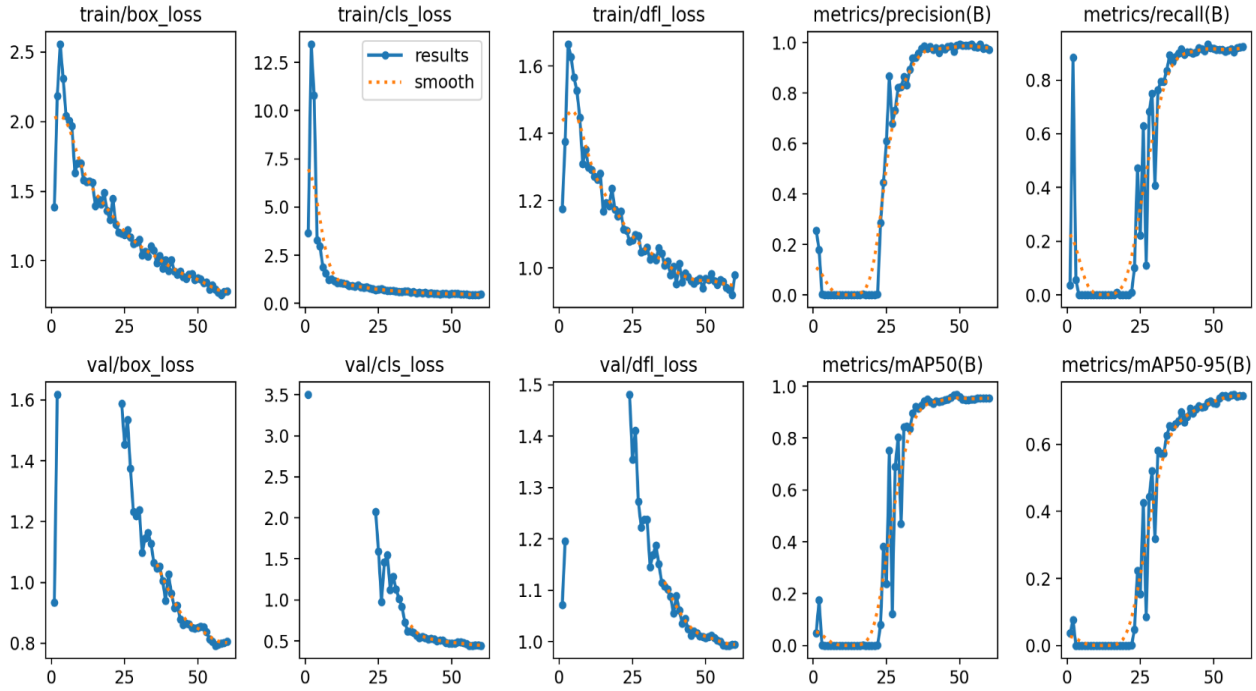

Fig. 4. The training loss curves of the fish detection model based on YOLO during the epochs and their training and validation

In the validation batch, there are images which are not augmented, and this is to guarantee a fair analysis of the trained model. Precision, recall and mAP are the validation metrics that are computed using ground truth annotations. This figure 5 illustrates the data that is employed to test the ability of the model to generalize unknown data.

Fig. 5. Sample and batch and validation of model performance

The figure 6 shows some sample outputs of the trained model, in which identified fish are surrounded by bounding boxes and the corresponding confidence scores. These findings exhibit proper spatial localization of fish and classification of fish in complicated underwater images. The model is suitable in real time use as it is found to identify several fish at once, even in low-visibility or cluttered areas.

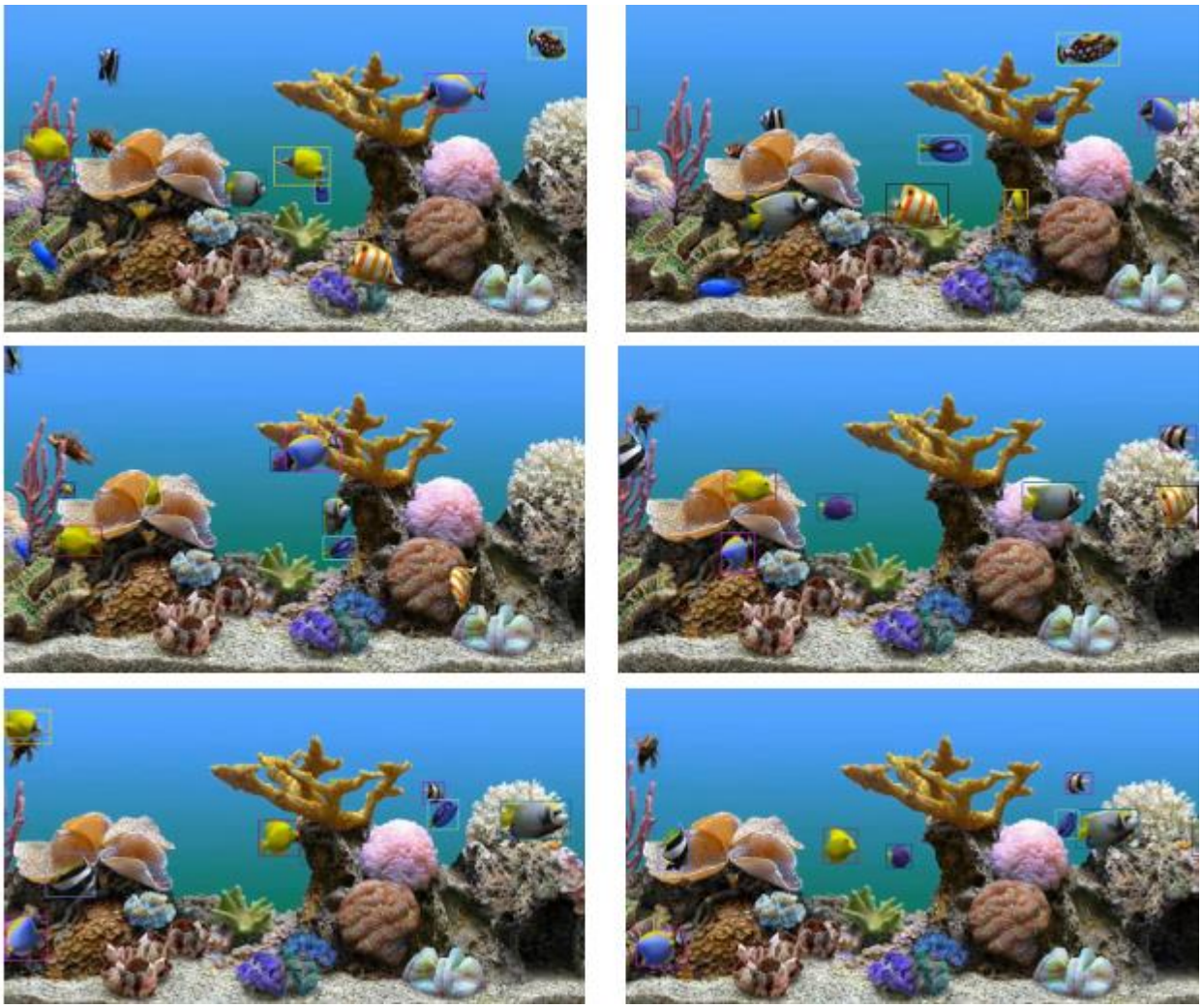

Fig. 6. Outputs of detection with predicted bounding boxes and confidence scores of fish instances

D. Training Stability and Convergence

The stability of training was determined by examining training and validation loss curves with respect to epochs. The suggested YOLO8n model exhibited a smooth and stable convergence and the training and validation loss steadily declined throughout the duration of training. The fact that the two curves are very close shows that there is some low overfitting and a high level of generalization. The mAP progressive curve again affirms the consistent optimization, with slow progression and then a plateau at the subsequent epochs. This action is an indication that the model successfully learns in a discriminative manner in underwater without oscil- lation instability and deviation in training. YOLO8n converged faster than YOLO8m since it had less parameter complexity than YOLO8m, enabling it to update its gradient efficiently and have suitable optimization dynamics. The fact that the performance did not deteriorate substantially in the validation set shows that it is not sensitive to the variability of the data sets, such as varying light conditions, turbidity, and the size of objects. Altogether, the patterns of observed convergence indicate that the proposed YOLO8n architecture has stable training dynamics and yields a high level of detection, which supports its applicability in the context of deploying it to a real-time setting and underwater conditions. Comparison of detection performance between yolo8n and YOLO8m

as shown as figure 7.

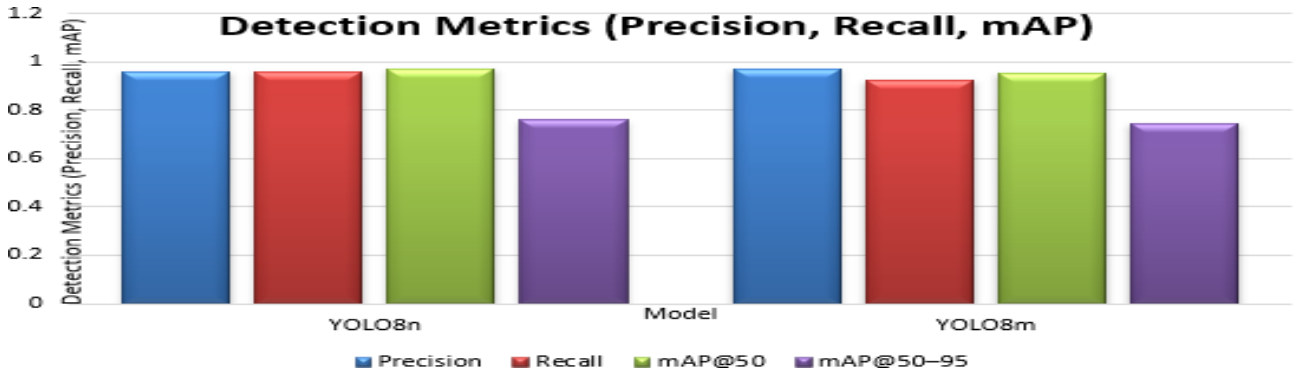

Fig. 7. Comparison of detection performance between yolo8n and YOLO8m

E. Practical Implications

The suggested YOLO8n-based fish detection system has a great practical importance in the context of real-life underwater surveillance system. That is because of its light-weight design, high detection rate and real-time ability to provide inferences, it is applicable in embedded devices, edge computing systems and autonomous underwater vehicles. The system can also facilitate automated monitoring of populations, feeding opti- mization as well as health measurement in aquaculture setting as it can constantly detect fish presence and density. The framework facilitates the manual annotation-free observation of the behavior of marine animals as well as biodiversity anal- ysis in marine ecology studies. High recall in turn guarantees low rates of missed detection and this is vital in conservation applications where high levels of population estimation are required. Besides, the computational efficiency of YOLO8n makes it less energy-intensive, which is appropriate in the battery-powered underwater systems that require long-term working periods. The proposed model has a better trade-off in the resource usage and detection performance than the larger architectures, which makes it more viable in large-scale application in the real time surveillance and environmental monitoring systems. Inference speed comparison of yolo8n and yolo8m as shown in figure 8.

CONCLUSION

This paper examined how well YOLO8-based systems can be used to detect fish in underwater video. Experimental analysis has been fully done on the comparison of YOLO8n and YOLO8m at the level of detecting accuracy, computing complexity and the speed of inferences. Experimental results show that YOLO8n achieves superior overall performance with a Precision of 0.96, Recall of 0.96, mAP@50 of 0.97, and mAP@50–95 of 0.76, while maintaining high inference speed of 110 FPS. Although YOLO8m attains slightly higher precision (0.971) and good Rcall of 0.95, mAP@50 of 0.95, and mAP@50–95 of 0.74. The findings validate that YOLO8n is better than YOLO8m with respect to its overall performance whereby, it has better recall and mAP values and has much less model size and lower computational cost. YOLO8n, though having less parameters, has a high generalization ability and convergence stability. It is very fast in inference and thus it is very apt to be used in real time underwater monitoring systems and edge devices. The results show that lightweight deep learning models can be used to achieve the state of the art detection accuracy even in difficult underwater conditions with no efficiency loss. This renders YOLO8n a convenient as well as scalable solution to marine environment research, automated aquaculture

surveillance, and smart underwater viewing platforms.

FUTURE WORK

Although the framework proposed performs highly, there are a few directions that can be used to improve further underwater

Fig. 8. Inference speed comparison of yolo8n and yolo8m

fish detection. Temporal information integration: Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) or 3D convolutional networks may be used to implement tem- poral information integration to enhance detection and false positives in adverse underwater environment. Multi-species detection: An extension to multi-species detection that may make use of hierarchical or multi-label classification meth- ods. Domain adaptation: Transfer learning and unsupervised domain adaptation to facilitate adaptation of models to new underwater conditions or datasets so that they can generalize to a variety of water conditions.

REFERENCES

[1] A. A. Muksit, F. Hasan, M. F. Hasan Bhuiyan, M. R. Haque, A. R. Anwary, and S. Shatabda, "YOLO-Fish: A robust fish detection model to detect fish in realistic underwater environment," Ecological Informatics, vol. 72, Art. 101847, 2022, doi: 10.1016/j.ecoinf.2022.101847.

[2] M. Vijayalakshmi and A. Sasithradevi, "AquaYOLO: Advanced YOLO- based fish detection for optimized aquaculture pond monitoring," Sci. Rep., vol. 15, p. 6151, 2025, doi: 10.1038/s41598-025-89611-y.

[3] H. Wang, J. Zhang, and H. Cheng, "HRA-YOLO: An effective detection model for underwater fish," Electronics, vol. 13, no. 17, Art. 3547, Sep. 2024, doi: 10.3390/electronics13173547.

[4] C. Yang, J. Xiang, X. Li, and Y. Xie, "FishDet-YOLO: Enhanced underwater fish detection with richer gradient flow and long-range dependency capture," Electronics, vol. 13, no. 18, Art. 3780, Sep. 2024, doi: 10.3390/electronics13183780.

[5] J. Zhang, J. Zhang, K. Zhou, Y. Zhang, H. Chen, and X. Yan, "An improved YOLOv5-based underwater object-detection framework," Sen- sors, vol. 23, no. 7, Art. 3693, 2023, doi:

10.3390/s23073693.
[6] H. Liang and T. Song, “Lightweight marine biological target detection algorithm based on YOLOv5,” Front. Mar. Sci., vol. 10, Art. 1219155, 2023, doi: 10.3389/fmars.2023.1219155
[7] Y. Zhang et al., “Lightweight underwater object detection based on YOLOv4 and multi-scale attentional feature fusion,” Remote Sens., vol. 13, no. 22, p. 4706, 2021, doi: 10.3390/rs13224706.
[8] H. Zhao et al., “BGLE-YOLO: A lightweight model for under- water bio-detection,” Sensors, vol. 25, no. 4, p. 1191, 2025, doi: 10.3390/s25041191.
[9] L. Liu et al., “MarineYOLO: Innovative deep learning method for small target detection in underwater environments,” Alexandria Eng. J., vol. 104, pp. 423–433, 2024, doi: 10.1016/j.aej.2024.07.126.
[10] Z. Zhang et al., “An improved YOLOv8n used for fish detection in natural water environments,” Animals, vol. 14, no. 14, p.2222, 2024, doi: 10.3390/ani14112022.
[11] Z. Yan, L. Hao, J. Yang, and J. Zhou, “Real-Time Underwater Fish Detection and Recognition Based on CBAM-YOLO Network with Lightweight Design,” J. Mar. Sci. Eng., vol. 12, no. 8, p. 1302, Aug. 2024, doi: 10.3390/jmse12081302.
[12] M. Liu et al., “UOD-YOLO: a lightweight real-time model for detecting marine organisms,” Front. Mar. Sci., vol. 12, Art. 1728563, 2025, doi: 10.3389/fmars.2025.1728563.
[13] J. Feng and T. Jin, “CEH-YOLO: A composite enhanced YOLO- based model for underwater object detection,” Ecol. Inform., vol. 82, p. 102758, 2024, doi: 10.1016/j.ecoinf.2024.102758.
[14] M. Liu et al., “LFN-YOLO: precision underwater small object detection via a lightweight reparameterized approach,” Front. Mar. Sci., vol. 11, Art. 1513740, 2025, doi: 10.3389/fmars.2024.1513740.
[15] S. Cai, X. Zhang, and Y. Mo, “A lightweight underwater detector enhanced by attention mechanism, GSConv and WIoU on YOLOv8,” Sci. Rep., vol. 14, Art. 25797, 2024, doi: 10.1038/s41598-024-75809-z.
[16] P. Sarkar et al., “Advanced YOLOv4 for real-time underwater object detection: An application-oriented approach,” Appl. Soft Comput., vol. 185, p. 113837, 2025, doi: 10.1016/j.asoc.2025.113837.
[17] Z. Zhang et al., “YOLO-PWSL-enhanced robotic fish: An integrated object detection system for underwater monitoring,” Appl. Sci., vol. 13, no. 15, p. 7052, 2023, doi: 10.3390/app13157052.
[18] M. Ouis and M. Akhloufi, “YOLO-Based fish detection in underwater environments,” Environ. Sci. Proc., vol. 29, no. 1, p. 44, 2024, doi: 10.3390/ECRS2023-16315.
[19] C. Zhang et al., “Underwater target detection algorithm based on im- proved YOLOv4 with SemiDSConv and FIoU loss function,” Front. Mar. Sci., vol. 10, Art. 1153416, 2023, doi: 10.3389/fmars.2023.1153416.
[20] Y. Lu et al., “AquaYOLO: Enhancing YOLOv8 for accurate underwater object detection for sonar images,” J. Mar. Sci. Eng., vol. 13, no. 1, p. 73, 2025, doi: 10.3390/jmse13010073.
[21] Y. Fu et al., “An improved YOLO-based algorithm for aquaculture object detection,” Appl. Sci., vol. 15, no. 21, p. 11724, 2025, doi: 10.3390/app152111724.
[22] S. Qu et al., “Underwater small target detection under YOLOv8-LA model,” Sci. Rep., vol. 14, Art. 16108, 2024, doi: 10.1038/s41598-024- 66950-w.