Fish counting through underwater fish detection using deep learning techniques

Sundari VEERAPPAN[1], Jansi Rani SELLA VELUSWAMI[2]

[1] Department of Computer Science and Engineering,

Meenakshi Sundararajan Engineering College, Chennai, Tamil Nadu, India

[2]Department of Computer Science and Engineering,

Sri Sivasubramaniya Nadar College of Engineering, Chennai, Tamil Nadu, India

cse.sundari.msec@gmail.com, jansi.research@gmail.com

Abstract: Aquaculture is the practice of reproducing, increasing and yielding aquatic organisms, such as aquatic animals and aquatic plants, in confined water bodies, like ponds, lakes, rivers, oceans, etc. Fishery represents one of the activities of aquaculture. The key function affecting fisheries management is the fishing activity. Operations like locating and counting fish are used to enhance this practice. There is a strong demand for underwater fish identification for multiple uses in sustainable fisheries. Real time monitoring helps to improve fishing activities. Deep Learning Techniques are used to train the computer with the available existing image data, faster GPUs, and algorithms employed to detect, locate and classify various objects within an underwater image or video with high accuracy. A well-liked object detection model, namely YOLO (You Only Look Once), is renowned for its quickness and precision. This paper presents a state-of-the-art version of the YOLO model for detecting and counting fish from underwater images or videos. The primary objective is to develop a system for automatic fish detection using an advanced convolutional neural network YOLOv8 and compare the results with the ones of the YOLOv7 model. It is proved that YOLOv8s performs better than YOLOv7, as YOLOv8s achieves a mAP@0.5 of 0.964, a precision of 0.929 and an IoU of 0.7.

Keywords: Aquaculture, Computer Vision, YOLO, Fish Detection.

1. Introduction

Computer vision has come a long way, but it is still challenging for it to match the precision of human perception. With aided machine learning techniques, computers may now be trained to accurately recognize, locate and classify various objects in an image. For localization and identification, object detection methods can be used to locate and track the objects in real time, as well as to count the objects in a scene. Machine learning and deep learning models have a high impact on the improvement of the object detection process (Zou et al., 2023) and its associated activities. The main target function of object detection in computer vision is locating and identifying various classes of objects in images and videos. It plays a major role in numerous applications like robotics, self-driving automobiles, image retrieval, machine inspection and surveillance. Single-shot detectors and two-stage detectors are the two different categories of object detection algorithms (Diwan et al., 2023). The existence and location of objects are predicted using two passes of the input image using two-shot object detection. A set of predictions or probable locations of objects are generated in the first pass, then these proposals are refined and, in the second pass, final predictions are made. This method has more accuracy than single-shot object detection, but it requires more computational expense. For the above-mentioned real time applications, single-shot object detection is more effective than the two shot detection. The Convolution Neural Network (CNN) (Li et al., 2015) is recognized as the most used method for object detection on various occasions in many research fields. YOLO (Pham et al., 2020) is designed as a single-shot detector. It is implemented to detect the object in an image or video, with the help of a fully convolutional neural network.

For monitoring and processing, the underwater videos and images are collected either statically or dynamically. The quality and visibility of the images are very poor in underwater images, due to the scattering and absorption of light by water, and the underwater suspended particles. Difficulties such as changes in lighting conditions, dark and dirty water, fish camouflage, less resolution, dynamic backgrounds, moving fish shape distortions are playing a major role in underwater images and videos. Identifying and locating the fish are the major tasks under these

https://doi.org/10.33436/v33i4y202306

circumstances. Moreover, humans have the ability to quickly recognize and locate things in an image. The human can perform complex tasks such as identifying various objects, locating their presence and fixing impediments, without much conscious effort. As computer vision and image processing technology has advanced, the use of image processing techniques to enhance the quality of underwater image, in order to meet the needs of the human visual system and machine recognition, has steadily gained attention. One of the most common and difficult issues in computer vision is object detection.

2. Motivation

YOLO is performing well to detect fish in underwater scenarios, when compared with other existing real-time object detection systems. The existing algorithms first use faster Region-based Convolutional Neural Network (R-CNN) (Girshick et al., 2014) and Region Proposal Network (RPN) to identify possible regions of objectiveness. Then, they perform the identification process on each of those regions independently. They also require numerous iterations for the same image, in order to refine identification. Since YOLO uses a fully connected layer, it performs all of its predictions at a single shot and needs only one iteration. Some of the object detection algorithms use classifiers as detectors. But YOLO recommends utilizing an end-to-end fully connected neural network that concurrently predicts the objects surrounded by the bounding boxes and class probabilities.

The first release of YOLO was in 2015. In this context, diverse subsequent iterations of the same concept have been put forth, each building upon and enhancing its predecessor from YOLOv1 to YOLOv8 (Terven et al., 2023). YOLOv2 was introduced in 2016. Anchor boxes, dimension clusters and batch normalization were implemented as added features in YOLOv2. In 2018, YOLOv3 was released with Spatial Pyramid Pooling, huge number of anchors and more efficient backbone network meant to improve its performance. New features such as anchor-free detection head, new loss function and mosaic data augmentation were included in the release of YOLOv4, in 2020. The performance of the model was further enhanced by YOLOv5, which also provided fresh features including automatic export to well-known export formats, integrated experiment tracking, and hyperparameter optimization. Many of Meituan's autonomous delivery robots have been running YOLOv6, which the business opened-source in 2022. Pose estimation on the COCO key points dataset is one of the extra tasks that YOLOv7 implemented. The most recent version of YOLO is known as YOLOv8 by Ultralytics (Jocher et al., 2023). This enhanced model improves the performance, efficiency and accuracy of the object detection. The computer vision tasks such as detection, classification, tracking, segmentation and posture estimation are supported by YOLOv8. Because of its adaptability, YOLOv8 can be used in a variety of contexts and applications. It is also used as a predesigned model to train a custom set of data. The concept behind the predesign model is transfer learning. It is a powerful technique used in Deep Learning. It trains a model with a defined data set. The pretrained model is used to perform different tasks with different data sets. The model uses the information gained from a previous trained task to improve the prediction accuracy of a new task.

The significant objectives of this paper are:

1. Development of a model to detect fish in the given more realistic environment.
2. Finding a better model with better accuracy to make it sustainable in a complex image background.

3. Related work

There are various works in the literature specialized in underwater object detection and some of those closely related to the topic are discussed in this section.

Javaid et al. (2023) presented a framework that provided a solution for the task of video summarization and object detection in underwater videos using YOLOv3. Region proposal networks were used for object detection to hypothesize object locations (Ren et al., 2015).

Li et al. (2015) implemented the object detection model with Spatial Pyramid Pooling net and Fast R-CNN. The running time of the detection networks was reduced and the region proposal computation was considered as a bottleneck of the model. The required accuracy and speed were achieved in the real-time object detection, with the help of deep CNN needed to assist the underwater robot, in order to perform some specific underwater operations (Han et al., 2020).

Wen et al. (2023) developed a YOLOv5 embedded with Coordinate Attention named YOLOv5s-CA network, aimed to improve object detection accuracy when tried to use increased computing power. Jalal et al. (2020) proposed a unified hybrid model in which YOLO deep neural network was combined with optical flow and Gaussian mixture models, in order to identify and classify the fish in unclear underwater videos. The MobileNet was used as a backbone in the YOLOv3 model and the modified model had improved efficiency and accuracy for feature extraction (Cai et al., 2020). The neck module of YOLOv5 was attached with a BoT3 module with the multi-head self-attention (MHSA) mechanism to produce better effects and accuracy, while detecting the objects in the dense environment (Zhang et al., 2023).

Al Muksit et al. (2022) presented YOLOv3 to which Spatial Pyramid Pooling was added, in order to increase the detection capacity of tiny fish in the dynamic environment. Li et al. (2015) proposed an object detection technique by means of fast R-CNN features and trained with generic visual object classes (VOC) and ImageNet dataset, so as to recognize the species of fish in the underwater environments which are more challenging in comparison to others, like land environments, for instance.

4. Methodology

The proposed methodology implies different stages, starting with image acquisition and annotation, followed by preprocessing and finishing with its training by YOLO model, as shown in Figure 1.

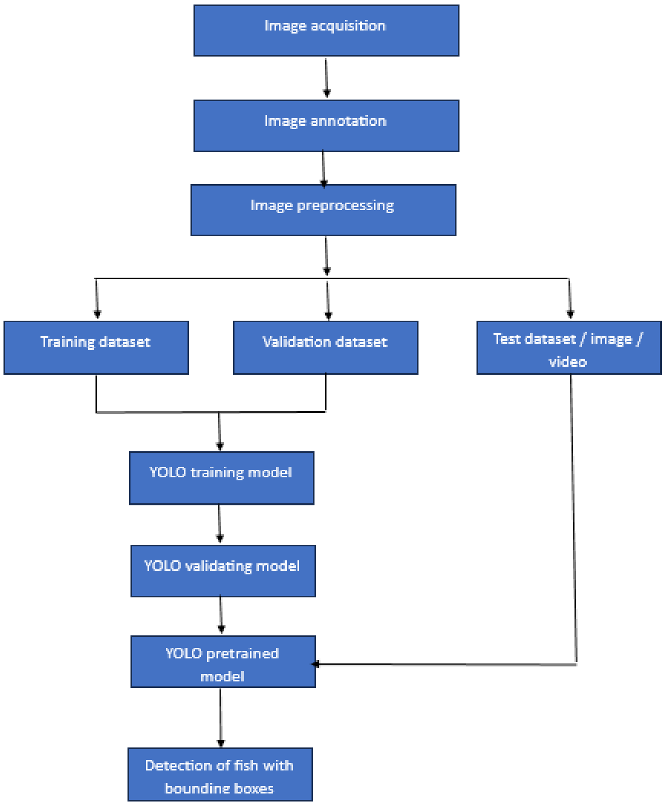


Figure 1. Architecture for fish detection

4.1. Data acquisition and annotation

A custom dataset has been generated from the raw underwater video. The images used in the predefined datasets were taken under controlled underwater settings. Since this is typically not the case in real life, it is much easier to create images where fish are clearly visible and have strong background contrast. Here, the proposed model uses more realistic settings. The video was taken from Central Institute of Brackish Aquaculture (CIBA), Chennai, Tamil Nadu, India, which is an uncontrolled environment. More control over the data used to train the model can also be obtained with a custom dataset. It can be a very useful tool for enhancing the effectiveness and performance of the machine learning models. The images were extracted from videos, then were slightly enhanced and, additionally, uploaded into rob flow, a computer vision framework, for annotation. Both YOLOv8s and YOLOv7 use YOLO format for labeling. The YOLO format is a particular format for marking object bounding boxes (Li et al., 2020) in images for object detection tasks. In this format, each image in the dataset should have a corresponding text file, with the same name as the image file name, which contains the bounding box annotations for that image. The label data must be submitted to YOLOv8s and YOLOv7 in a text (.txt) file format. The object coordinates and class index values are both included in the format that is also normalized to the width and height of the image. The format of each line as well as an object from the matching image are represented by a line in the text file. Label data, otherwise known as ground truth data, is a set of annotations that indicate the presence and location coordinates, class identification number, width and height of the objects in an image. Figure 2a and 2b represent the input image and its corresponding label file, respectively.

Label file format:

<Object-class> <x-coordinate> <y-coordinate> <width> <height>

```
0 0.93515625 0.22109375 0.1296875 0.25
0 0.8515625 0.30859375 0.23125 0.25234375
0 0.7546875 0.421875 0.19453125 0.23984375
0 0.77734375 0.51171875 0.4453125 0.1171875
0 0.1828125 0.51796875 0.365625 0.109375
0 0.571875 0.62734375 0.21796875 0.21328125
0 0.846875 0.63515625 0.30546875 0.08359375
```

Figure 2a. Input image (img1 jpg) **Figure 2b.** Label file (img1.txt)

Both images and labels files are stored separately in two different folders, namely image and label. In order to train a computer vision model on custom data, it is important to split the dataset into a training set, a validation set and a test set. The train dataset has maximum data and is used to train the model how to make predictions. The validation dataset is used to evaluate the accuracy of the trained model and refine it. The common split ratio for train - validation - test dataset is 70–20-10%, but sometimes the ratio may vary, based on the size of the dataset and the task that are going to be performed. Mostly, a higher percentage of the dataset can be used as a training dataset, if the whole dataset is small, whereas a lower percentage can be used as a training dataset, if the dataset is large. It's crucial to divide the dataset into a training, a validation and a testing set, when training a computer vision model on unique data. Totally 500 images were split for training, validating and testing. 350 images were used for training, 100 for validation and 50 for testing.

For the purpose of arranging the parameters of a computer vision model, a special configuration file called "data.yaml" can be created. The model settings are maintained and managed by setting the proper path for the dataset folders such as training, validation and testing and by changing the names of the classes, according to the objects that are going to be detected.

4.2. Image preprocessing

The traditional basic image processing methods such as brightening, sharpening and color correction methods are used to enhance the underwater image. Over the days, algae and fungi, in the form of lichens in green color, can take shape on the walls of the tank, which changes the color of the water. In the original image, the greenish background color dominates the entire image and the appearance of the fish is not up to the level of vision. The original image was slightly enhanced, so that the greenish background was reduced and a little bit enhanced image of the fish (Figure 3b), in comparison with the original image (Figure 3a) could have been seen. Since the used custom dataset had a very poor underwater image quality, the first attempt was to try to slightly reduce the green background.

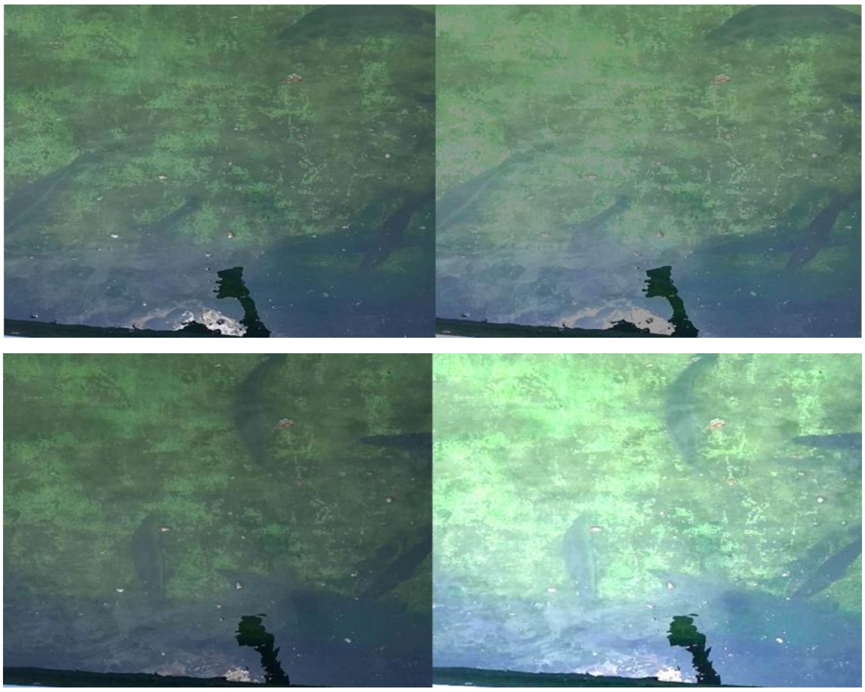


Figure 3a. (a) Original image **Figure 3b.** (b) Enhanced image

Gamma correction is also known as the *Power Law Transform.* First, the image pixel intensities must be scaled from the range *[0, 255]* to the range *[0, 1.0]*. From there, the output gamma corrected image is obtained, by applying the following equation:

$$OMG = IMG^{(1 / GA)} \quad (1)$$

where *IMG* is the input image and *GA* is the present gamma value. The output image *OMG* is then scaled back to the range *[0, 255]*. Gamma values < *1* will shift the image towards the darker end of the spectrum, while gamma values > *1* will make the image appear lighter. A gamma value of *GA=1* will have no effect on the input image.

The above-mentioned preprocessing steps include collecting the dataset, enhancing the images in the dataset, labeling, splitting the dataset into a training, a validation and a testing set and, finally, creating a custom configuration file. Once the preprocessing steps are over, the training of the model can be started, by using the training dataset. Object detection technique identifies the unique or different objects in an image or video, by displaying bounding boxes around those objects, along with class label index and the confidence scores that represent the accuracy of the image in percentage.

4.3. YOLO model

Since the identified objects are surrounded by the bounding boxes, by looking at the image or video only once, the algorithm is named YOLO. The Convolutional Neural Networks are different from other methods by means of their superior performance on image data for feature extraction. The features of the images, like color, edges, shapes, etc., are added one by one, while progressing through the layers of the CNN, from the first layer to the last one. Finally, it detects the objects. The two key characteristics, namely parameter sharing and multiple filters, play a major role in CNN to address the issues in object detection, such as identifying correct classes of various objects and accurately locating their place and position.

The object detection technique divides the image or frame into N x N grid cells. Each grid cell creates a B number of bounding boxes, the probability of an object in the underlying grid, along with class probabilities and classification category. The center of an object must be located inside the grid cell, in order for that grid cell to be able to detect it. Using any appropriate bounding box, this grid cell is responsible for effectively locating that specific object. There is a total of N x N x B bounding boxes. Some threshold value is defined and it removes the bounding boxes with class probability scores below the given threshold automatically. In most of the object detection algorithms, this threshold value is fixed at a standard point of 0.5. This value is changeable in accordance with the dataset and its properties. After eliminating the bounding boxes with lower threshold value, there still remains a high number of bounding boxes present in the image, that are not exactly covering the objects. Non-maximum suppression (NMS), a second criterion for eliminating the less important bounding boxes, also depends on the IoU (Intersection over Union). The precision and effectiveness are increased by the NMS, a post processing procedure. More than one anchor box is generated, even for a single object in an image, during object detection. All these anchor boxes represent the same object and they may overlap or differ in terms of coordinates. NMS is used to select the perfect single bounding box that exactly covers each object and to delete the remaining bounding boxes related to that particular object. First, it selects the bounding box with highest objectiveness score and, then, it suppresses or deletes all the other bounding boxes.

5. Limitations of YOLOv7

Although YOLOv7 is a strong and successful object identification system, it has several drawbacks. Like many object identification systems, YOLOv7 has trouble when detecting tiny things. In congested areas or when things are far from the camera, it may not be able to effectively detect them. Since YOLOv7 is sensitive to changes in lighting or other environmental factors, it is difficult to work with real-world applications that involve changeable lighting effects. YOLOv7 can be computationally demanding, making it challenging to execute in real-time on devices with limited resources. It is very difficult to identify objects that are drastically different in size or shape from the other objects in the scene, because of the struggle to accurately find items of various sizes.

6. Key features of YOLOv8

- The accuracy of the object detection is increased, in comparison to its previous versions, with the help of added features and improvements.
- It offers higher accuracy than other existing object detection algorithms.
- It has a faster inference speed, in comparison to their models.
- It allows the users to select the model with different backbones such as EfficientNet, ResNet, and CSPDarknet, according to what is suitable for their particular task.
- During the training period, it can be able to optimize the learning rate and to balance the loss function, so that the model performance is increased. This method is called adaptive training.
- The architecture of YOLOv8 is very adaptive in nature. Users are able to change the structure of the model by changing its parameters that suit their requirements.
- It supports transfer learning. It allows the creation of pre-trained models that are very convenient to use for custom datasets.

The YOLOv8 architecture expands on previous iterations of the YOLO algorithms. The convolutional neural network used by YOLOv8 is composed of two primary sections: the head and the backbone. The foundation of YOLOv8 is a modified version of the CSPDarknet53 architecture. 53 convolutional layers make up this design, which also uses cross-stage partial connections to enhance information transfer between the various layers. A number of convolutional layers and a string of fully connected layers (Msadaa et al., 2022) make up the head of the YOLOv8 algorithm. These layers play a main role in predicting bounding boxes, objectiveness scores and class probabilities of the objects that are identified and located in the image. With the aid of this mechanism, the model is able to change its attention to various aspects of the image, according to their relevance to the task at hand. The capability of YOLOv8 to carry out multi-scaled object identification is another crucial feature. The model makes use of a Feature Pyramid Network (FPN), a feature extractor implemented with pyramid concept destined to find items in an image that have different sizes and scales, with the help of multiple feature map layers.

7. Results & discussion

In total, 500 images have been taken in 640 x 640 size. Out of 500, 350 images were used for training. 100 images from the others were used for validation and the remaining 50 images were used for testing. The results obtained from YOLOv7 and YOLOv8s are shown in Figure 4a, 4b and 4c. After validation the model created the validated batch files that shown in the Figure 4d and 4e for both YOLOv8s and YOLOv7.

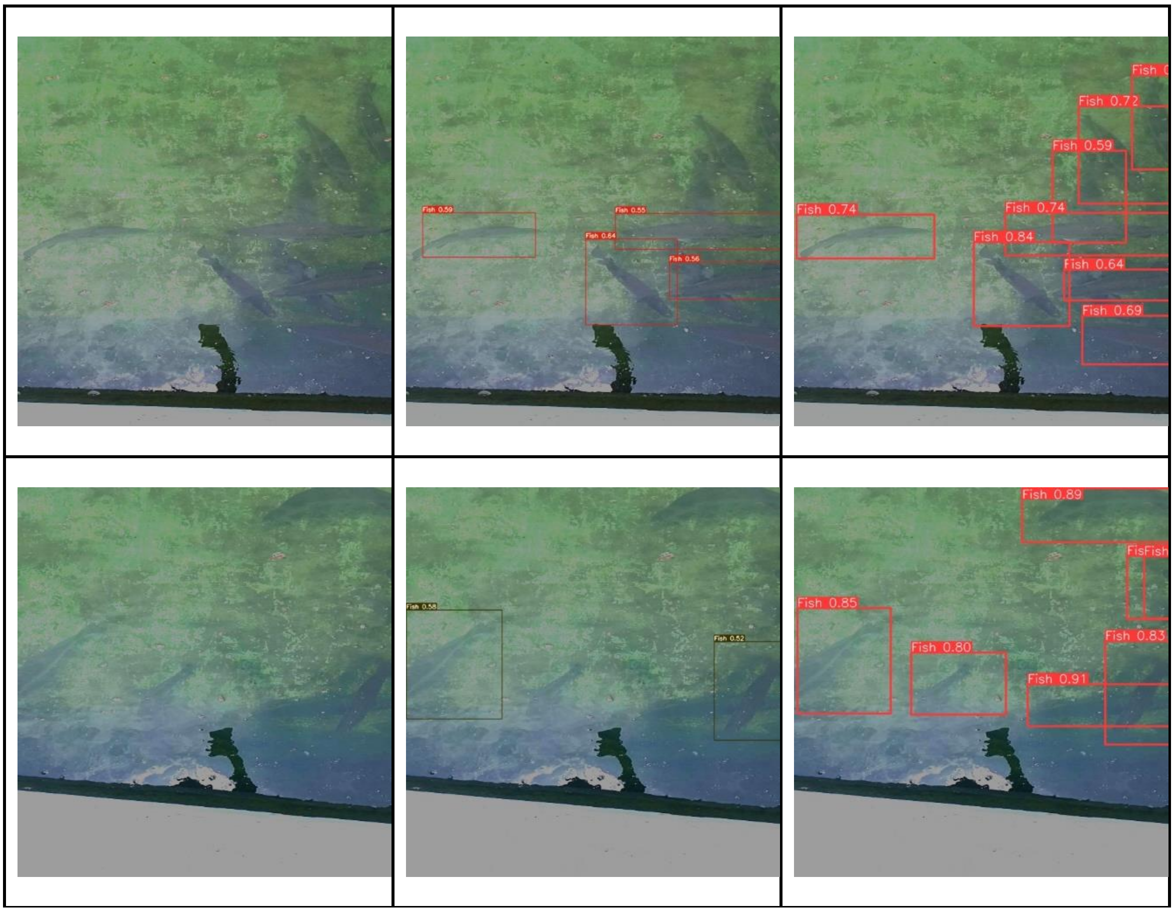


Figure 4a. Input images

Figure 4b. Output images (YOLOv7)

Figure 4c. Output images (YOLOv8s)

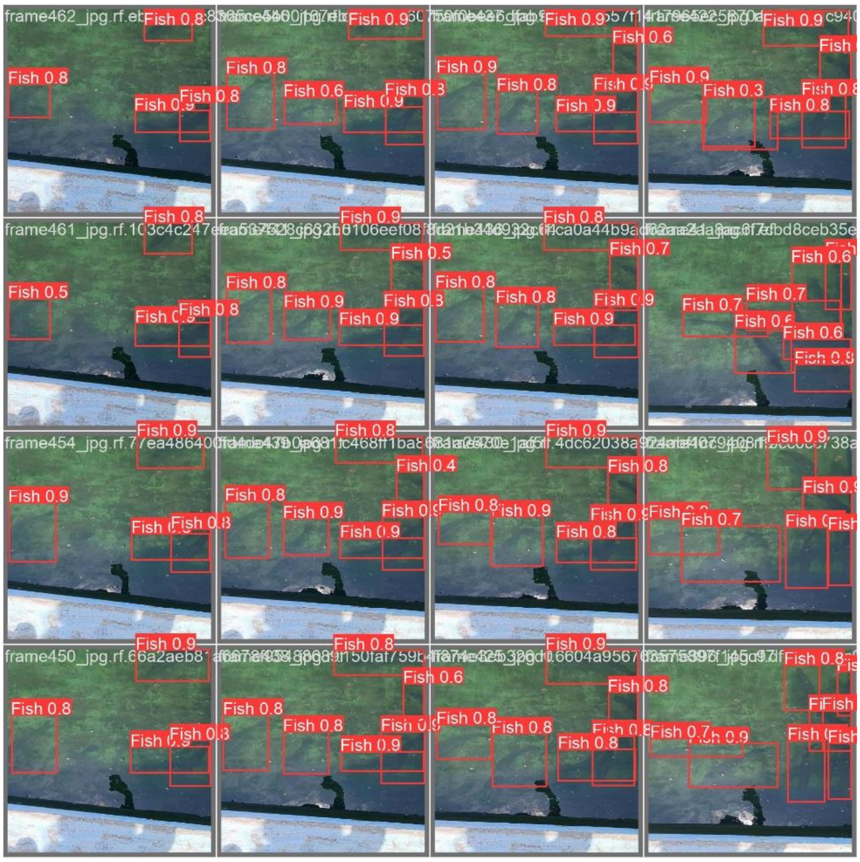


Figure 4d YOLOv8s validated batch file

Figure 4e. YOLOv7 validated batch file

7.1. Evaluation Metrics

For object detection, there are some evaluation metrics used to analyze its performance. Precision, Recall, Intersection over Union (IoU) and Mean Average Precision (mAP) are such metrics used to assess the accuracy.

Intersection over union (IoU)

$$IoU = \frac{AI}{AU} \tag{2}$$

where AI represents Area of Intersection and AU represents Area of Union. The bounding boxes are used to locate the object. Basically, the range of IoU lies between 0 and 1. The perfect bounding box is selected by its IoU value. If the value of IoU is high, then the predicted bounding boxes are close to the actual bounding box, and it can be concluded that the model works well and identifies objects with a high objectiveness score.

Mean Average Precision (mAP)

Precision and recall are the evaluation metrics. The former is used to measure the accuracy of positive predictions, while latter one is used to measure the completeness of positive prediction.

$$Precision = \frac{t_p}{t_p + f_p} \tag{3}$$

$$Recall = \frac{t_p}{t_p + f_n} \tag{4}$$

where t_p is true postive, f_p is false positive, f_n is false negative.

There is a set of bounding box predictions in the image. Calculate the precision values of all the bounding boxes (K) and then take an average of these precision values, which are known as interpolated precision or Average Precision (AP). Now, the Mean Average Precision (mAP) is simply calculated across all the classes.

$$mAP = \frac{\sum_{i=1}^{K} AP_i}{K} \tag{5}$$

The Mean Average Precision is a benchmark evaluation metric used to analyze the performance of the object prediction model.

Confusion Matrix

To depict the actual and expected classification, a confusion matrix is utilized. It is used to describe how well a model performs when tested against a collection of test data. Figures 5a and 5b how the confusion matrix of YOLOv8s and of YOLOv7, respectively. Figure 6 shows the precision and recall of both YOLOv8s and YOLOv7.

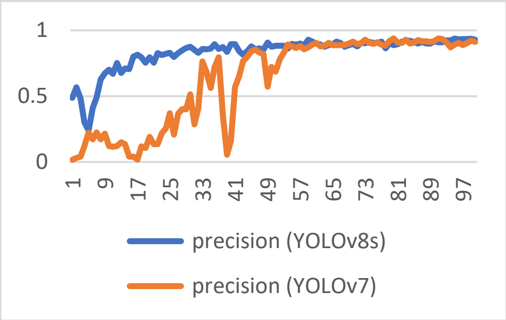
Predicted	Fish (True)	Background (True)
Fish	**0.97**	**1.00**
Background	**0.03**	**0**

Figure 5a. Confusion matrix for YOLOv8s

Predicted	Fish (True)	Background (True)
Fish	**0.95**	**1.00**
Background	**0.05**	**0**

Figure 5b. Confusion matrix for YOLOv7

Precision and Recall Curves



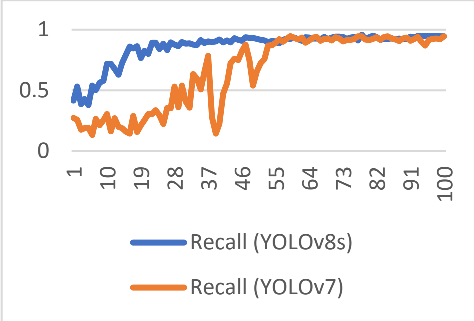


Figure 6. Precision and Recall of YOLOv8s and YOLOv7

7.2. Comparison between YOLOv7 and YOLOv8s

Table 1. Comparison between YOLOv7 and YOLOv8s

YOLO Model	Precision	Recall	mAP@0.5	mAP@0.5-0.95	IoU	Speed
YOLOv8s	0.929	0.945	0.964	0.572	0.7	0.399 hours
YOLOv7	0.911	0.945	0.953	0.452	0.45	0.943 hours

As it can be observed from Table 1, YOLOv8s has a slightly higher detection accuracy than YOLOv7. However, the difference is not particularly considerable. A more precise detection is achieved with the use of YOLOv8s, the suggested method. YOLOv8s achieves a mAP@0.5 of 0.964, a precision of 0.929 and an IoU of 0.7, which are higher values compared to the ones of YOLOv7.

Therefore, in the future, it is desired to further analyze the network compression, in order to decrease network parameters, to speed up detection, and to maintain detection accuracy.

7.3. Comparison of YOLOv8s and existing work

The YOLOv8 results are compared with YOLOv5 model implemented by Wang et al. 2022 work. From Table 2, it is observed that YOLOv5 to YOLOv8 gives precision of 0.84 to 0.929 and recall of 0.75 to 0.945. Hence there is 20% improvement in recall. And hence it is proved that YOLOv8 performs better than the current recent existing work of Wang et al. 2022.

Table 2. Comparison between YOLOv7 and YOLOv8

Methods	Precision	Recall
Wang et al. (2022)	0.84	0.75
YOLOv8s	0.929	0.945

8. Conclusion

The YOLOv8s model was implemented to detect fish in the given more realistic environment. When comparing its detection results with the ones of YOLOv7, it can be noticed that it can provide a better decision. The other models are not particularly good at detecting items, because the underwater image is of extremely poor quality and the objects are not clearly seen. The findings of the qualitative and quantitative evaluations testify the superiority of the proposed method, YOLOv8s achieves a mAP@0.5 of 0.964, a precision of 0.929 and an IoU of 0.7, values which are higher in comparison to those of YOLOv7. The suggested paradigm works well for finding objects under water.

Under the conditions of dense targets, complicated backdrops, and the mimicry of similar species, conventional object-detection algorithms still struggle with the problem of missing and incorrect detection, in the complex and dynamic underwater environment. Further research into more effective image-enhancement techniques for data augmentation is possible, as well as the exploration of better transformer blocks, in order to boost detection precision.

Acknowledgments

The research reported in this paper has been supported by the data collection from Central Institute of Brackish Aquaculture (CIBA), provided by Dr. M. Mahesh, Director and Principal Scientist.

REFERENCES

Al Muksit, A., Hasan, F., Emon, M. F. H. B., Haque, M. R., Anwary, A. R. & Shatabda, S. (2022) YOLO-Fish: A robust fish detection model to detect fish in realistic underwater environment. *Ecological Informatics*. 72, 101847. doi: 10.1016/j.ecoinf.2022.101847.

Cai, K., Miao, X., Wang, W., Pang, H., Liu, Y. & Song, J. (2020) A modified YOLOv3 model for fish detection based on MobileNetv1 as backbone. *Aquacultural Engineering*, 91, 102117. doi: 10.1016/j.aquaeng.2020.102117.

Diwan, T., Anirudh, G. & Tembhurne, J. V. (2023) Object detection using YOLO: Challenges, architectural successors, datasets and applications. *Multimedia Tools and Applications*. 82(6), 9243-9275. doi: 10.1007/s11042-022-13644-y.

Girshick, R., Donahue, J., Darrell, T. & Malik, J. (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, June 23-28 014, Columbus, OH, USA*. pp. 580-587. doi: 10.1109/CVPR.2014.81.

Han, F., Yao, J., Zhu, H. & Wang, C. (2020) Underwater image processing and object detection based on deep CNN method. *Journal of Sensors*. 2020(9), 1-20. doi: 10.1155/2020/6707328.

Jalal, A., Salman, A., Mian, A., Shortis, M. & Shafait, F. (2020) Fish detection and species classification in underwater environments using deep learning with temporal information. *Ecological Informatics*. 57, 101088. doi: 10.1016/j.ecoinf.2020.101088.

Javaid, M., Maqsood, M., Aadil, F., Safdar, J. & Kim, Y. (2023) An Efficient Method for Underwater Video Summarization and Object Detection Using YoLoV3. *Intelligent Automation & Soft Computing*. 35(2), 1295-1310. doi: 10.32604/iasc.2023.028262.

Jocher, G., Chaurasia, A. & Qiu, J. (2023) *YOLO by Ultralytics*. https://github.com/ultralytics /ultralytics/blob/main/CITATION.cff [Accessed 28th February 2023].

Li, X., Shang, M., Qin, H. & Chen, L. (2015) Fast accurate fish detection and recognition of underwater images with fast R-CNN. In: *OCEANS 2015-MTS/IEEE Washington*, October *19-22, October 2015, Washington, DC*. pp. 1-5. doi: 10.23919/OCEANS.2015.7404464.

Li, X., Wang, W., Wu, L., Chen, S., Hu, X., Li, J., Tang, J. & Yang, J. (2020) Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection *Advances in Neural Information Processing Systems*, 33, 21002-21012. doi: https://doi.org/10.48550/arXiv.2006.04388.

Msadaa, I. C. & Grayaa, K. (2022) Covid-19 detection: a Deep Learning Approach based on Wavelet Transform. *Revista Română de Informatică și Automatică [Romanian Journal of Information Technology and Automatic Control]*. 32(1), 87-98. doi: 10.33436/v32i1y202207.

Pham, M. T., Courtrai, L., Friguet, C., Lefèvre, S. & Baussard, A. (2020) YOLO-Fine: One-stage detector of small objects under various backgrounds in remote sensing images. *Remote Sensing*. 12(15), 2501. doi: 10.3390/rs12152501.

Ren, S., He, K., Girshick, R. & Sun, J. (2015) Faster R-CNN: Towards real-time object detection with region proposal networks. In: *NIPS'15: Proceedings of the 28th International Conference on Neural Information Processing Systems, December 7-12, 2015, Montreal Canada*. pp. 91-99.

Terven, J. & Cordova-Esparza, D. (2023) A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond. *arXiv*. [Preprint] https://arxiv.org/abs/2304.00501. [Accessed 8th October 2023].

Wang, H., Zhang, S., Zhao, S., Jiamin Lu, Wang, Y., Daoliang Li & Zhao, R. (2022) Fast detection of cannibalism behavior of juvenile fish based on deep learning. Computers and Electronics in Agriculture. 198 (3-4), 107033. doi: 10.1016/j.compag.2022.107033.

Wen, G., Li, S., Liu, F., Luo, X., Er, M. J., Mahmud, M. & Wu, T. (2023) YOLOv5s-CA: A Modified YOLOv5s Network with Coordinate Attention for Underwater Target Detection. *Sensors*. 23(7), 3367. doi: 10.3390/s23073367.

Zhang, J., Zhang, J., Zhou, K., Zhang, Y., Chen, H. & Yan, X. (2023) An Improved YOLOv5-Based Underwater Object-Detection Framework. *Sensors*. 23(7), 3693. doi: 10.3390/s23073693.

Zou, Z., Chen, K., Shi, Z., Guo, Y. & Ye, J. (2023) Object detection in 20 years: A survey. To be published in *Proceedings of the IEEE*. [Preprint] https://arxiv.org/abs/1905.05055. [Accessed18th January 2023].

Sundari VEERAPPAN is currently working as Assistant Professor, in the Department of Computer Science and Engineering of Meenakshi Sundararajan Engineering College (MSEC). She obtained her Bachelor of Engineering (B.E.) degree from Periyar Maniammai College of Technology for Women, Tanjavur, Bharathidasan University, and her Post Graduate (PG) degree from Easwari Engineering College, Anna University. She has more than 15 years of teaching experience. Her areas of interest include Machine Learning, Deep Learning, and Computer Vision.

Jansi Rani SELLA VELUSWAMI is currently working as Associate Professor in the Department of Computer Science and Engineering of Sri Sivasubramaniya Nadar College of Engineering (SSNCE). She obtained her Bachelor of Engineering (B.E.) degree from Periyar Maniammai College of Technology for Women, Tanjavur, Bharathidasan University, her Post Graduate (P.G.) degree from SASTRA University and her Doctoral (Ph.D.) degree from Anna University, Tamil Nadu, India. She has been nurturing the young minds for more than 18 years. She has published many papers in her area of research. She has more than 19 years of teaching experience. Her areas of interest include Mobile Networks, Deep Learning, Congestion Control, Transmission Control Protocol, Social Network Analysis, Computer Vision, Internet of Things.