International Journal of Leading Research Publication (IJLRP)

E-ISSN: 2582-8010 ● Website: www.ijlrp.com ● Email: editor@ijlrp.com

Diseased Fish Detection in Aquatic System

Alekhya Vadlamani[1], Binoy Kurikaparambil Revi[2], Sithara Samudrala[3]

[1,2,3]University of Tennessee, Knoxville

Abstract
We present a study and design of a trained model using the YOLOv8 model and custom dataset to identify the diseased fish in an aquatic system. This is the first project to apply the YOLOv8 model to this domain.

Index Terms: Identification, Detection, Fish, Classification

Introduction

Whether for public, commercial, or recreational purposes, cultivating aquatic organisms in recirculating aquaculture sys- tems (RAS), such as home aquaria, can be setback by a single mortality if the remains are not removed promptly. For example, if a deceased fish is not removed from the aquarium, in the matter of days, its remains decompose, deplete oxygen levels in the water, and release ammonia into the environment that will harm all of the other organisms, potentially leading to their deaths [1]. On top of this, if the diseased fish had passed away from bacterial, viral, parasitic, or fungal disease, the un- touched remains will continue to brew the disease and spread to the other organisms with the system, often resulting in mass mortality [1]. The threat of fish diseases and mortalities to the welfare of recirculating aquaculture systems amplifies when considering intensive aquaculture, where there are high densities of aquatic organisms or fish, typically for farming and human consumption. Globally, fish consumption increases every year at a rate of 3.1 percent between 1961 to 2017, which is almost double the rate of the annual world population growth rate [2]. In 2018, aquaculture, including intensive aquaculture, accounted for about 46 percent of total fish production, which was 82 million tons of fish [2]. Economic loss from disease in fish aquaculture is estimated to be between $1.05 to $9.58 billion dollars per year [3]. Aquaculture, particularly for fish, continues to grow as a crucial pillar of human society, yet still struggles with the problem of efficiently detecting diseased fish from recirculating aquaculture systems (RAS) in a timely manner.

Traditionally, the problem of detecting diseased fish has been approached with tons of manual labor to survey the fish for mortality and to closely monitor the fish and the environment, which comes at a high price of time and money [4]. To increase the efficiency of this approach, traditionally, fish mortality and other metrics to examine water quality are studied in depth through historical patterns and are used to gauge and predict potential harm to the fish and the recirculat- ing aquaculture system as a whole [5]. As artificial intelligence develops and comes to play, many researchers have turned to many deep learning techniques to alleviate this problem. The newest state of the art model of the You Only Look Once series, known as YOLOv8, was released on January 10th, 2023 and has yet to be tested and evaluated in this domain of detecting diseased fish in recirculating aquaculture systems [6], [7]. This project aims to evaluate the efficacy and efficiency of the YOLOv8 models as a potential strategy to detect diseased fish to prevent fish mortality in recirculating aquaculture systems.

Previous Work

Many of the previous generations of the YOLO series of models are heavily adapted in aquaculture research, whether it is detecting diseases in fish, monitoring abnormal behavior in dishes, or population counting. Early generation of the YOLO models imposed constraints on bounding box predictions, which limits the number of nearby objects that the YOLO model could predict [8]. For the purpose of YOLO model application, early generations of YOLO models struggle with detecting smaller objects and especially struggle with detecting smaller objects that collect in groups, which is very inconve- nient when applying this to fish detection [8]. Still, YOLO models tends to be relied upon in this domain since YOLO models have better accuracy in comparison to other single- stage detection models and low detection latency, which allow for fish classification or detection in real-time, but can have lower accuracy in comparison to its two-stage target detection real-time counterparts such as Faster R-CNN (Region-based Convolutional Neural Network) or DPM (Deformable Parts Models) [8], [9].

As a result, in previous works, not only were early gen- erations of YOLO models tested for fish detection, but these models were also extended or enhanced to compensate for the limitations of YOLO models. For example, in 2020, a study [4] aimed to improve accuracy of the YOLOv3 model by taking steps to improve another shortcoming of the YOLO family of models - the need for good resolution of images. Obtaining crisp images can be hard in real-world applications, especially in intensive aquaculture where there are an overwhelming amount of fish per volume and perhaps unclear, dirty water, so this study tackled another issue of applying YOLO to fish detection [4]. Image enhancement with the Multi-Scale Retinex (MSR) algorithm was performed in the preprocessing phase, which was then fed into YOLOv3 for fish detection [4]. In addition to image enhancement, the researcher also coupled the YOLOv3 with the optical flow algorithm to track the movements of detected fish as well [4]. There were no architectural changes to the YOLOv3, but overall the accuracy of YOLOv3 was still improved with image enhancement.

With the release of YOLOv4 and YOLOv5, for the detection of diseased fish, Zhao et al. [9] compared YOLOv3, YOLOv4, and YOLOv5 with the MobileNetV3-YOLO4 and two-staged Faster R-CNN as well as the researchers' proposed YOLOv4 model with infrastructural changes known as Deformable-MobileNetV3-YOLOv4 (DM-YOLOv4). The proposed en- hanced model or the DM-YOLOv4 adds deformable con- volutions at the beginning of the network to enhance the detection speed and uses the feature extraction backbone network of MobileNetV3 instead of YOLOv4's version for feature extraction to decrease the number of parameters [9]. Overall, YOLOv5 has better precision, average precision, and recall than the DM-YOLOv4. DM-YOLOv4 had a leading position in accuracy, quickest processing speed of all models, and smallest model calculation and size [9]. DM-YOLOv4 had an inference speed of 64 frames per second, which is much greater than 35 frames per second in the YOLOv5, so essentially DM-YOLOv4 is a lightweight take of an already lightweight YOLO model [9].

Though YOLOv5 is a step-up from the YOLOv4, like YOLOv4, YOLOv5 grapples with obtaining clean feature extraction in detecting diseased fish [10]. Wang et al. [10] chose the YOLOv5m model of the four types of YOLOv5, replaced the CSPNET component of the YOLOv5 with a C3 component to simplify the model, replaced the convolutional kernels with a convolutional kernel group to improve feature extraction, and added an attention mechanism to the model with a convolutional block attention module (CBAM) as a YOLOv5 enhancement that the researchers named DFYOLO [10]. DFYOLO is only able to detect fish and diseased fish, but does so at a higher inference speed of 93 frames per second and higher precision than YOLOv5m [10].

As even more generations of YOLO models are released, accuracy and inference speed have massively improved that Ranjan et al. [5] developed a "MortCam" completely on the YOLOv7 model without any added updates or enhancements to the model (MortCam) to monitor mortality. The custom YOLOv7 models were implemented with Roboflow and Ult- alytics and performed well and similarly with varied lighting experiments and image quality [5]. The model that trained on both ambient lighting and supplement lighting ended up being the most robust, which goes to show that additional preprocessing algorithmic enhancements to YOLOv7 may not improve the model performance by much [5].

Technical Approach

The problem is that there is a lack of an efficient method to detect diseased fish in recirculating aquatic systems, so that diseased fish can be removed from their environment in a timely manner before their deaths to prevent catastrophic mass mortality of fish in the system. Despite the fact that deep learning techniques, such as YOLO, are currently starting to be adopted to detect diseased fish in aquaculture, the newest, latest, and most improved edition of the YOLO family of models or YOLOv8 has not been evaluated in this domain of fish mortality in aquaculture. The goal of this project is to purely assess the generation of YOLOv8 models on their efficiency and efficacy to detect diseased fish in recirculating aquatic systems as a proof of concept.

The YOLO family of models is a state-of-the-art deep learning technique to segment, detect, pose, or classify objects in an image. This project extends application on this family of models for diseased fish because YOLO uses a single stage approach for object detection, which achieves low latency and has capability for real-time applications, a characteristic important to this domain due to the timely constraints of aquaculture mortality [5]. The first version of YOLO was released in 2015, and since then, generations of YOLO models have been released with many architectural changes leading to massive improvement [7].

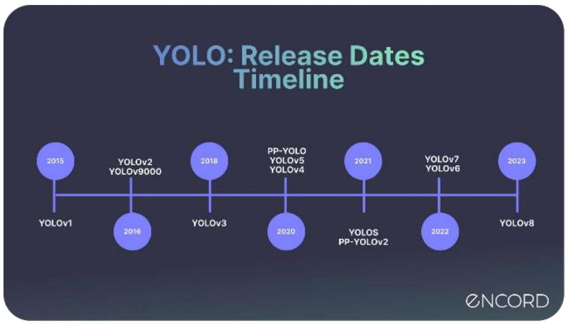


Fig. 1. YOLO: Release Dates Timeline [11]

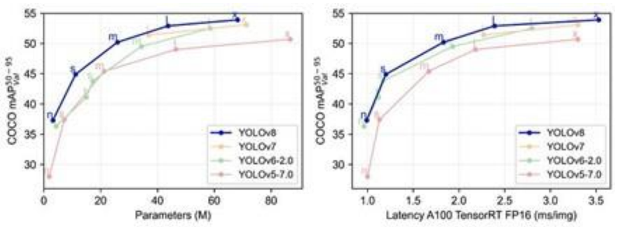


Fig. 2. YOLOv8 Architecture [7]

The YOLOv8 employs a fully convolutional network with five main components: the backbone, the neck, the head, the loss function, and the post-processing.

The YOLOv8 backbone follows a structure based on the CSPDarknet53 [12]. The CSPDarknet53 convolution network uses the Darknet53 backbone approach to object detection and couples that with the CSPNet strategy [12]. The Cross- Stage Partial Network or the CSPNet strategy splits the base layer into two sections and then merges the sections with a cross-stage hierarchy, which increases gradient flow, reduces

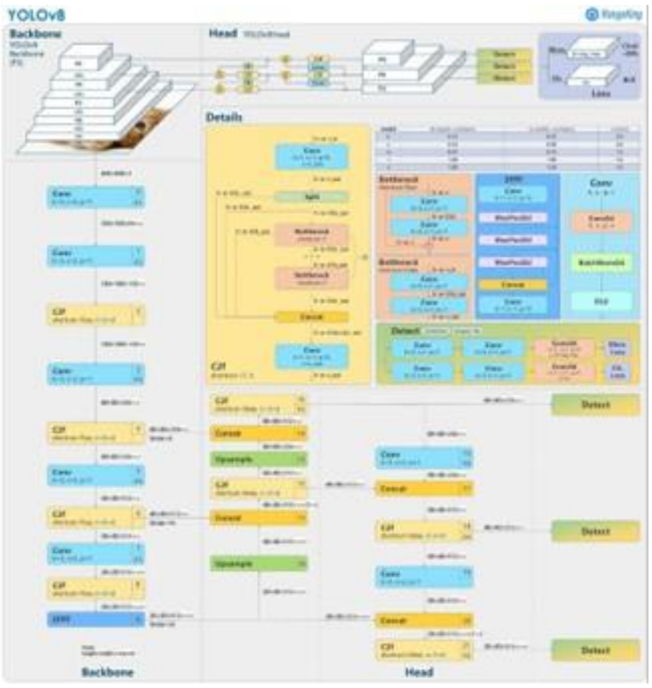
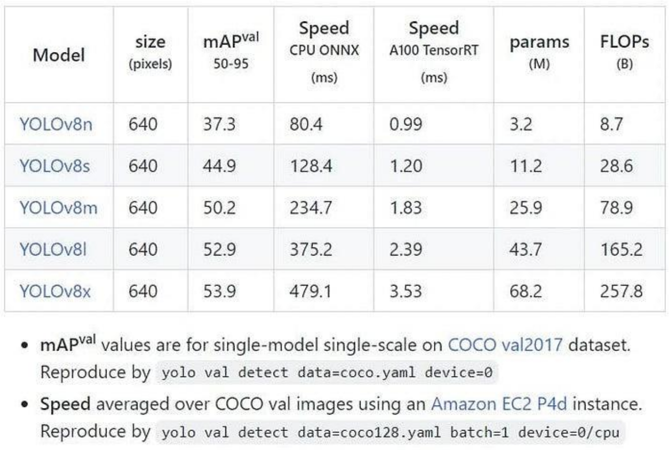


Fig. 3. Comparison of YOLO Models [7]

computational cost, and sustains good accuracy [7], [12]. The YOLOv8 backbone consists of 53 convolutional layers with the purpose of feature extraction.
The YOLOv8 neck is the portion of the network that joins the backbone with the head, which consists of the spatial pyramid pooling (SPP) module [7]. The SPP multi-layered network helps to detect both large and small objects by using multiple pooling sizes to extract features on different scales [12].
The YOLOv8 head includes many convolutional layers. The head features a self-attention mechanism, which allows the model to prioritize and to focus on relevant features. The objective of the YOLOv8 head is to predict the bounding boxes, objectness scores, and probabilities for the class of each object [7].
The YOLOv8 loss function takes objectness loss, distribu- tion focal loss, classification loss, and bounding box regression loss into account [7], [12].
The post-processing actions done by YOLOv8 involve ap- plying a non-maximum suppression to choose the most proba- ble bounding box, to increase the accuracy of the predictions, and to remove irrelevant bounding boxes [7], [12].
For the purpose of this project, only the YOLOv8 detection models are evaluated, since the YOLOv8 detection models alone were able to achieve the desired results without the need or help of segmentation, classification, or pose YOLOv8 models. Such choice of technical approach was based on documentation and light preliminary testing of the detection, classification, and pose YOLOv8 models to gauge potential results and their capabilities.
There are 5 YOLOv8 detection models: YOLOv8n,
YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x. The dif- ferences among the models are shown in

the table below.



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

- **mAP^{val}** values are for single-model single-scale on COCO val2017 dataset.
  Reproduce by `yolo val detect data=coco.yaml device=0`
- **Speed** averaged over COCO val images using an Amazon EC2 P4d instance.
  Reproduce by `yolo val detect data=coco128.yaml batch=1 device=0/cpu`

Fig. 4. Comparison of YOLOv8 Detection Models [7]

All are trained and then analyzed for which one would be best suited for this domain.
This project used a supervised learning method on data that was created from scratch. That being said, the data or the videos taken from the Knoxville Aquarium had to go through a hefty pre-processing process, in which its implementation is described in the next section. Essentially, the pre-processing method involved selecting frames from all taken video, an- notating the images with “healthy” or diseased” labels for the model, and converting these images into a compiled text format for the model to read in.

Dataset and Implementation

A. Creating the Dataset Implementation

Capturing Videos*:* All videos were taken at the Knoxville Aquarium in April of 2022. Based on the available fish species and potential diseases, the scope of fishes was limited to guppies, which have two types of tail colors - red and white - that are similar to the differences in coloration found in fish with white tailed disease. Five videos were taken outside of the tanks with iPhone 13 Pro in three different aquariums of guppies. All aquariums have similar lighting and are recirculating aquaculture systems.

Fig. 5. Aquarium of Knoxville

Selecting Frames: Frames from each video were man- ually selected using VLC player. Frames were selected based on clarity and uniqueness. Images selected had a variety of angles of fish and a variety of backgrounds. Approximately, 26 frames were selected for every 10 seconds.

Annotating Images: The images were uploaded to a Computer Vision Annotation Tool (CVAT.ai). With CVAT.ai, boxes were drawn around each fish and labeled as either “healthy” or ”diseased.” White-tailed fish were considered “diseased,” and the red-tailed fish were considered “healthy.”

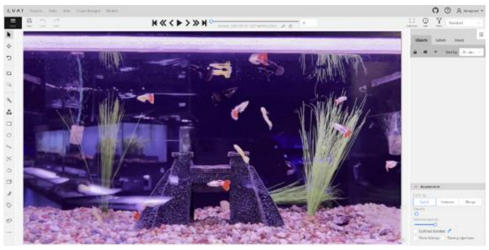


Fig. 6. CVAT Tool : Before Annotation

Exporting Dataset in YOLO 1.1 format*:* CVAT.ai has an export feature that generates to label text files in YOLO 1.1 format which is the correct format for YOLOv8. Each image corresponds to a single text file and contain the information on classification and bounding box in predefined format.

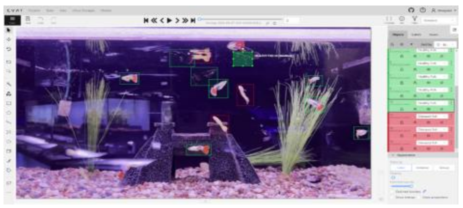
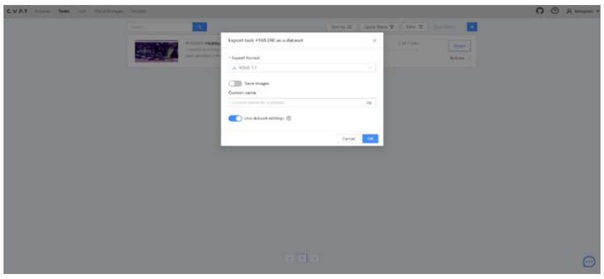


Fig. 7. CVAT Tool : After Annotation



Export task #165390 as a dataset

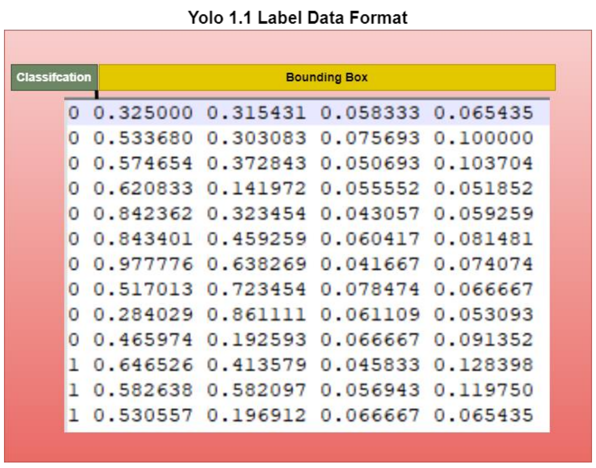
Export format: YOLO 1.1

Save images

Custom name

Use default settings

Fig. 8. CVAT Tool: Export Option



```
0 0.325000 0.315431 0.058333 0.065435
0 0.533680 0.303083 0.075693 0.100000
0 0.574654 0.372843 0.050693 0.103704
0 0.620833 0.141972 0.055552 0.051852
0 0.842362 0.323454 0.043057 0.059259
0 0.843401 0.459259 0.060417 0.081481
0 0.977776 0.638269 0.041667 0.074074
0 0.517013 0.723454 0.078474 0.066667
0 0.284029 0.861111 0.061109 0.053093
0 0.465974 0.192593 0.066667 0.091352
1 0.646526 0.413579 0.045833 0.128398
1 0.582638 0.582097 0.056943 0.119750
1 0.530557 0.196912 0.066667 0.065435
```

Fig. 9. YOLO 1.1 Label Output Format

B. Implementation

Overall, there were 77 images in the dataset total that include 67 images and 67 label text files for training and validation. To analyze the specific YOLOv8 detection models, YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x were all trained with 17 images and 17 label text files. The results of the training were used to choose the best of the five, and the best of the five models underwent further training, validation, and testing.

For the chosen model:

- Total Training Images: 50
- Total Training Label Text Files: 50
- Total Validation Images: 17
- Total Training Label Text Files: 17
- Total Images for Testing: 10

Google Colab pro was used to build, train, validate, and test the YOLO V8 on the Tesla T4 GPU.

In preliminary testing all models were trained with 100 epochs. All training was done with a batch size of 64 at a learning rate of 0.001. All models have an input size of 608x608, 80 classes, confidence threshold of 0.25, NMS threshold of 0.45, IOU threshold of 0.5, 64 filters, 53 layers, and a LeakyReLU activation function. Training time took 0.727 hours per 100 epochs. The chosen model out of this was trained on 50, 100 and 150 epochs.

Experiments and Results

A. YOLOv8 Detection Model Analysis

After training the detection models YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x for 100 epochs with 17 data, the results of the training are shown below.

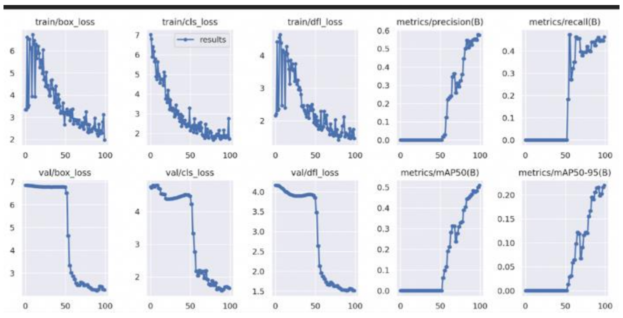


Fig. 10. Training Results : YOLOv8n

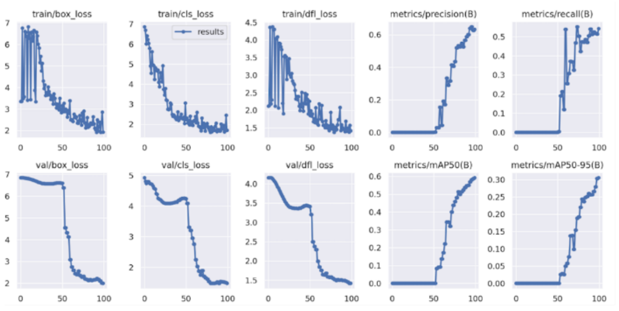


Fig. 11. Training Results : YOLOv8s

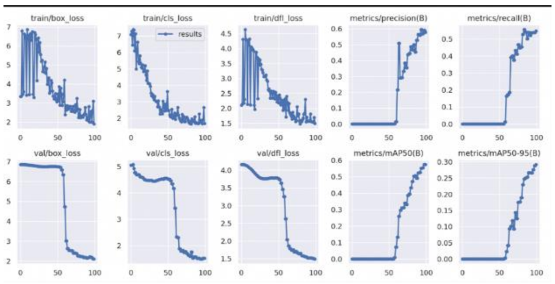


Fig. 12. Training Results : YOLOv8m

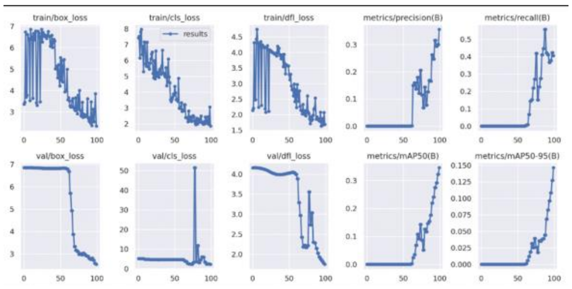


Fig. 13. Training Results : YOLOv8l

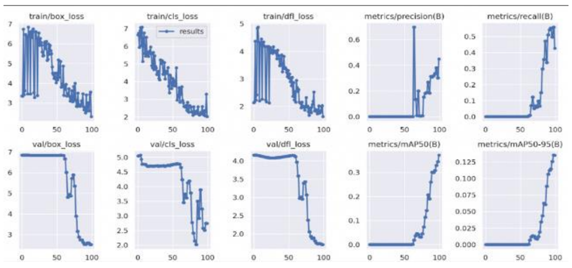


Fig. 14. Training Results : YOLOv8x

After analyzing all of the five models after training, at the 100th epoch, YOLOv8n, YOLOv8s, and YOLOv8m had the lowest box loss and classification loss at under 2. At the 100th epoch, YOLOv8s, and YOLOv8m had the lowest distribution focal loss at around 1.5. The precision, average precision, and recall of the YOLOv8m had the highest values of all of the models. From the analysis of the training results of all the 5 models, it is evident from the loss and precision curves that the classification and distribution focal loss of YOLOv8m model are better than all other models. YOLOv8m was chosen for further training and validation.

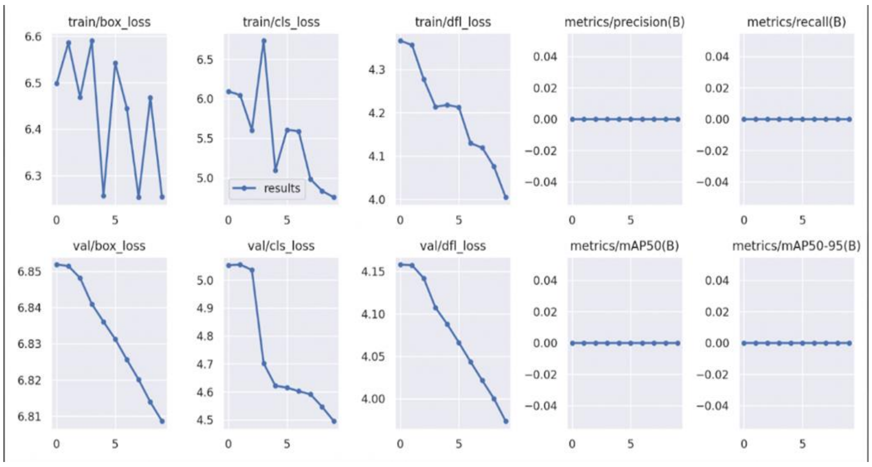


Fig. 15. Training Results with complete dataset : YOLOv8m

When comparing the training results for 50, 100, 150 epochs, there was no significant improvements in the loss, precision and recall. After 100 epochs, loss and precision curves flatten out. Hence, only the YOLOv8m model that was trained with 100 epochs underwent testing for obtaining optimal results.

B. YOLOv8m Training, Validation and Testing

Training and Validation: The YOLOv8m model was trained for 100 epochs, and the results are captured from the runtime memory to the shared drive for analysis and shown below.

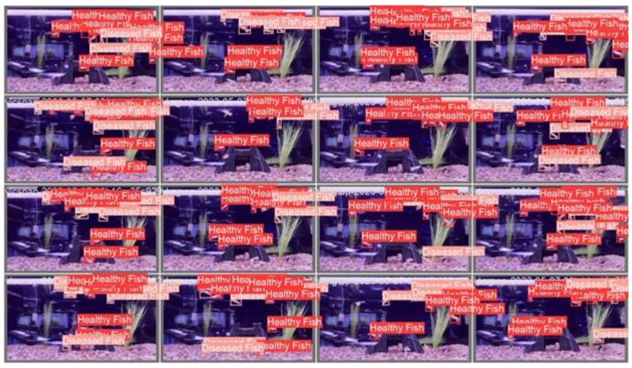


Fig. 16. YOLOv8m Validation Result

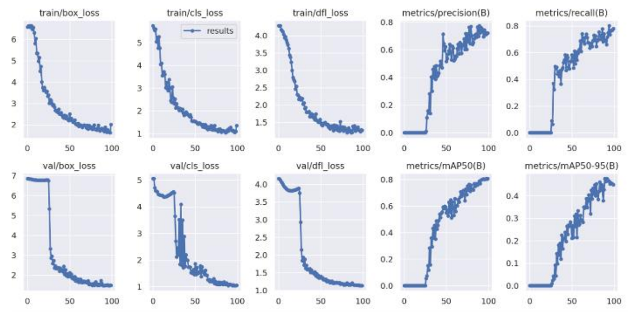


Fig. 17. YOLOv8m : Loss and Precision Curves

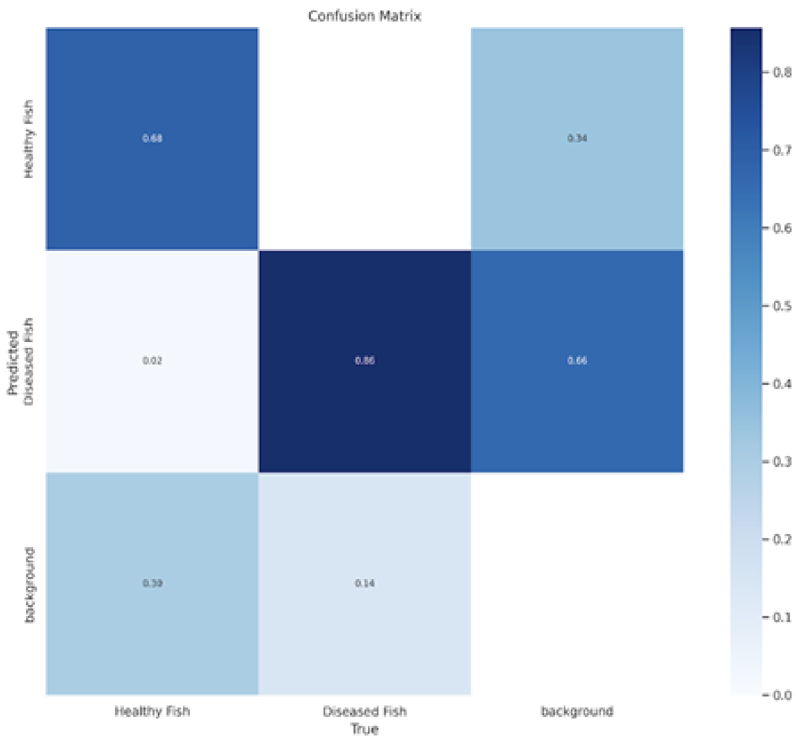


Fig. 18. YOLOv8m : Confusion Matrix

Testing: With the YOLOv8m trained model, testing was done on 10 test images.

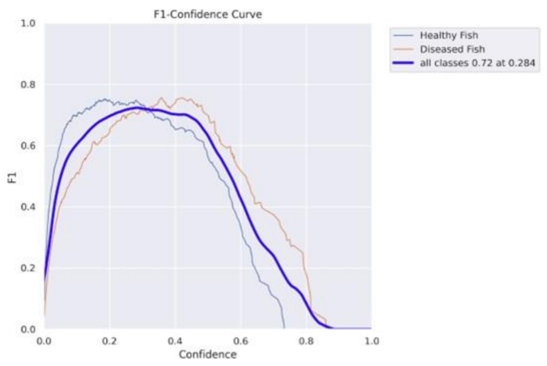


Fig. 19. YOLOv8m : Confidence Curve

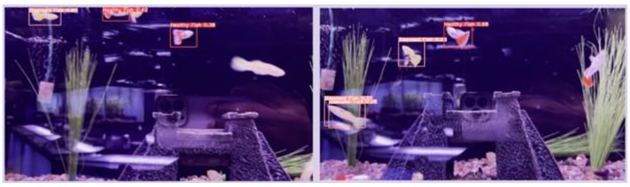


Fig. 20. YOLOv8m : Test 1 and 2

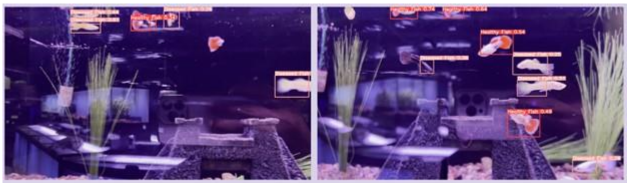


Fig. 21. YOLOv8m : Test 3 and 4

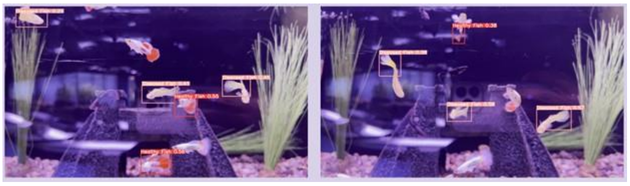


Fig. 22. YOLOv8m : Test 5 and 6

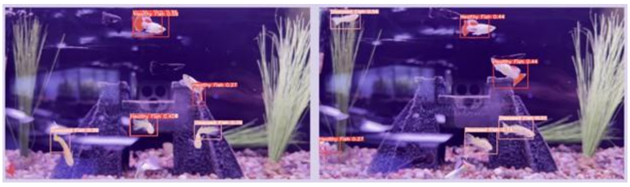


Fig. 23. YOLOv8m : Test 7 and 8

Conclusion

The Yolov8m model can be considered as a good proto- type for the fish disease detection. This project validated the YOLOv8m model as a great lightweight candidate for further application in this domain due to YOLOv8m's good accuracy and precision. Future work would include training on a wide range of fishes and their diseases that can be detected based on the skin coloration. YOLOv8 models' capability of running the trained model on the system command line allows for model integration to any cloud-based web application as an enhancement to use in real world. Another fantastic next step would be to integate the model to a web application. The lim- itations of manually creating the custom training dataset may be overcome by AI based tool for labeling which can enhance the training. Adding more pre-processing improvements will be a good choice to pipeline the data.

A. Lessons learned

- The quality and variety of the data created for a particular application is critical in the performance and robustness of the application.
- It is also important to test different models and chose the model with right parameters and architecture to suit the application.

References

1. K. Mills, “How to Sterilize Fish Tank after Fish Died,” National Park Aquarium, 2023. [Online]. Available: https://www.nationalparkaquarium.org/how-to-sterilize-fish-tank-after- fish-died/: :text=Indeed
2. Food and Agriculture Organization of the United Nations, The State of World Fisheries and Aquaculture: Sustainability in Action, 2020. [E- book] Available: https://www.fao.org/3/ca9229en/ca9229en.pdf
3. M. Tavares-Dias and M. L. Martins, “An overall estimation of losses caused by diseases in the Brazilian fish farms.” National Library of Medicine, July 2017. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5660038/.
4. H. El-Din Mohamed, A. Fadi, O. Anas, Y. Wageeh, N. ElMasry, Nabil, and A. Atia, “MSR-YOLO: Method to Enhance Fish Detection and Tracking in Fish Farms,” Procedia Computer Science, vol. 170, p. 539-546, 2020. [Abstract]. Available: https://reader.elsevier.com/reader/sd/pii/S1877050920305615?token=2B 18F93138A757FA5F87E4C860A5E27BAAAED943E3E58AB2DBFD5 778E11512354C280BE5385136F7DE21A5229D23C4C3origin Region=us-east-1originCreation=20230514192451. [Accessed Apr. 25, 2023].
5. R. Ranjan, K. Sharrer, S. Tsukuda, and C. Good, “MortCam: An Artificial Intelligence-aided fish mortality detection and alert system for recirculating aquaculture,” Aquacultural Engineering, vol. 102, 2023. Available: https://www.sciencedirect.com/science/article/pii/S0144860923000286. [Accessed Apr. 27, 2023].
6. F. J. Solawetz, “What is YOLOv8? The Ultimate Guide,” Roboflow, Jan. 11, 2023. [Online]. Available: https://blog.roboflow.com/whats-new-in-yolov8/. [Accessed Apr. 28, 2023].
7. G. Jocher, “YOLOv8,” docs.ultralytics.com, May 1, 2023. [Online]. Available: https://docs.ultralytics.com/models/yolov8/. [Accessed May 2, 2023].
8. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” Computer Vision Foundation, 2016. [Accessed May 4, 2023].
9. S. Zhao, S. Zhang, J. Lu, et. al., “A lightweight dead fish detection method based on deformable convolution and YOLOv4,” Computer and Electronics in Agriculture, vol. 198, July 2022. Available: A lightweight dead fish detection method based on deformable convolution and YOLOV4 - ScienceDirect. [Accessed May 5, 2023].
10. Z. Wang, H. Liu, G. Zhang, X. Yang, L. Wen, and W. Zhao, “Diseased Fish Detection in the Underwater Environment Using an Improved YOLOv5 Network for Intensive Aquaculture,” Fishes, vol. 8 issue 3, March 21, 2023. Available: https://www.mdpi.com/2410-3888/8/3/169. [Accessed May 7, 2023].
11. Encord, “YOLOv8 for Object Detection Explained [Practical Example],” medium.com, March 22. 2023. [Online]. Available: https://medium.com/cord-tech/yolov8-for-object-detection-explained-practical-example-23920f77f66a. [Accessed May 8, 2023].
12. Q. B. Phan and T. Nguyen, ”A Novel Approach for PV Cell Fault Detection using YOLOv8 and Particle Swarm Optimization”. TechRxiv, Apr. 26, 2023 [Online]. Available: https://doi.org/10.36227/techrxiv.22680484.v1 [Accessed May 10, 2023].