

AUTOMATIC FISH DETECTION FROM DIFFERENT MARINE ENVIRONMENTS VIDEO USING DEEP LEARNING

Ali Loulidi ¹, Rachida Houssa ², Lene Buhl-Mortensen ³, Hakima Zidane ², Hassan Rhinane ¹

¹ Laboratory of Geosciences, Department of Geology, Faculty of Sciences, University Hassan II, Casablanca, Morocco

² National Institute of Fisheries Research (INRH), Casablanca, Morocco

³ Institute of Marine Research, Bergen, Norway

Commission 4, WG 7

KEY WORDS: Fish Detection, Marine Environment, Yolov3, Deep Learning, Computer Vision, Artificial Intelligence.

ABSTRACT

The marine environment provides many ecosystems that support habitats biodiversity. Benthic habitats and fish species associations are investigated using underwater gears to secure and manage these marine ecosystems in a sustainable manner. The current study evaluates the possibility of using deep learning methods in particular the You Only Look Once version 3 algorithm to detect fish in different environments such as; different shading, low light, and high noise within images and by each frame within an underwater video, recorded in the Atlantic Coast of Morocco. The training dataset was collected from Open Images Dataset V6, a total of 1295 Fish images were captured and split into a training set and a test set. An optimization approach was applied to the YOLOv3 algorithm which is data augmentation transformation to provide more learning samples. The mean average precision (mAP) metric was applied to measure the YOLOv3 model's performance. Results of this study revealed with a mAP of 91,3% the proposed method is proved to have the capability of detecting fish species in different natural marine environments also it has the potential to be applied to detect other underwater species and substratum.

1. INTRODUCTION

The marine ecosystem represents significant esteem in terms of biodiversity and economics (Costanza, 1999) through fisheries, mining resources, renewable energies and other products determined from the ocean (Pauly & Zeller, 2017). Seafloor habitat and fish species around the world are beneath danger and endure from over-exploitation (Hutchings, 2000) with significant impact on human wellbeing (Worm et al., 2006). Several studies have been carried out to explore seafloor habitat and biodiversity to conserve and promote a sustainable use of fisheries resources (Lynch et al., 2016). Some researchers have investigated seafloor habitat in open sea (Lene Buhl-Mortensen et al., 2016). Others explored coastal areas as estuarine ecosystem (van Niekerk et al., 2020), and lagoons (Brodie et al., 2020). In addition, mapping strategies have been created to explore and document seafloor habitat such as, the MAREANO Program (Marine Areal Database for Norwegian Coasts and Sea Areas). This program used data from a variety of sampling gears to map the benthic environment and communities from all types of seabed in order to fill knowledge gaps in relation to the implementation of management plans for the Norwegian EEZ (L. Buhl-Mortensen et al., 2015). Among the gears used to explore and map the seabed, the platforms for taking underwater videos. The recent use of these underwater exploration gears has made it possible to fill in the lack of information due to traditional sampling by fishing gears. Data collected by underwater videos provides significant information to understand the benthic ecosystem (Sung et al., 2017). Nevertheless, it is also a challenging task because of the movement speed of fish underwater and their overlapping (Lumauag & Nava, 2019). In addition, the quality and variation in water condition makes this task even more complicated (Sharif et al., 2016). However, Computer vision and deep learning algorithms have shown a great success in image classification (Li et al., 2019), text interpretation (Iqbal &

Qureshi, 2020) and data security (Amanullah et al., 2020), so new possibilities are opening up to automate the workflow. Several computer vision applications have been carried out in the marine environment such as harbor surveillance (Casalino et al., 2009), (Palmieri et al., 2013), and collision avoidance (Caccia, 2006), (Campbell et al., 2012). However, these efficient and contemporary techniques have been used not only identifying marine creatures and objects, but also classifying various marine species (Olsvik et al., 2019). Many deep learning algorithms have been applied in the marine (Wang et al., 2020) and aquaculture environments (M. Sun et al., 2020), such as mask RCNN for the segmentation and measurement scheme of fish morphological features (Yu et al., 2020), U-net for the attenuation of marine seismic interference noise (J. Sun et al., 2020), and YOLO algorithm which shows its efficiency especially in fish detection (Cui et al., 2020), and was applied for marine data to detect various species in the Norwegian fjords and ocean (Stavelin et al., 2021). "You Only Look Once" (YOLO) (Redmon & Farhadi, 2018) is one of the most efficient deep learning algorithms for object detection (Du, 2018), it has shown its good performance for detecting many different objects, such as detecting apples during different growth stages in orchards (Tian et al., 2019), cars (Putra et al., 2018), and human detection in thermal imaging (Ivašić-Kos et al., 2019). Recent studies have investigated the use of yolo in fish detection and they show great performance of this object detection algorithm (Cui et al., 2020). So, in this paper, we present the result of the use of Yolo algorithm to identify fish in a natural environment. Firstly, we prepare fish images and their annotations, and we split our dataset into two section (training and testing). At that point, we utilize the YOLO (Redmon & Farhadi, 2018) model, which is trained by our own dataset to detect fish. Finally, we get fish movements detection by each frame within a video recorded in a natural marine environment.

1.1 Yolov3 Algorithm

YOLOv3 (Redmon & Farhadi, 2018) is a developed version over its predecessors: YOLO v1 and YOLO v2 (named also YOLO9000) (Redmon & Farhadi, 2017). The YOLO network changes the detection issue into a regression problem. It doesn't need a proposal region, and it creates bounding box coordinates and probabilities of each class straightforwardly through regression.

YOLO contains 24 convolutional layers followed by 2 fully connected layers. Some convolutional layers use convolutions of size 11 to diminish depth dimension of the feature maps. A faster form of YOLO, named Fast YOLO, utilizes just 9 convolutional layers yet this effects the accuracy. The general YOLO architecture is represented in Figure 1. The concept is to partition the input image into a $S \times S$ grid, and to make recognitions in each grid cell. Every cell predicts B bounding boxes along the confidence of these boxes. The confidence depends on two important parameters; first one is the probability of the object's existence in a grid cell. Second one is the intersection over union (IoU) of the ground truth boxes (GT) and predicted ones. Formally the confidence is represented as:

$$\text{Confidence} = \text{Pr}(\text{Object}) \times \text{IoU}(\text{GT}, \text{pred}) \quad (1)$$

Where: $\text{Pr}(\text{Object}) \in [0, 1]$.

Five predictions will be done by each grid cell: x , y , w , h , confidence and C class probabilities. (x, y) are the center coordinates of the box, and (w, h) represent the width and height of the box.

YOLOv3 has been improved by the use of the multi-label classification, which is not the same as the common labeling used in the previous versions. It utilizes a logistic classifier to calculate the likeliness of the object being of a particular label. Past versions utilize the softmax function to produce the probabilities structure the scores. For the classification loss, it utilizes the binary cross-entropy loss for each label, rather than the overall mean square error used in the previous versions. In addition to that, YOLOv3 has been evaluated by using different bounding box prediction. It relates the objectness score 1 to the bounding box anchor which covers a ground truth object more than others. It disregards others anchors that overlaps the ground truth object by more than a chosen threshold (0.7 is utilized in the implementation). Thus, YOLOv3 assigns one bounding box anchor for each ground truth object. Moreover, YOLOv3 has been ameliorated by the use of prediction across scales using the concept of feature pyramid networks. YOLOv3 predicts boxes at 3 distinct scales and afterward separates features from those scales. The results of prediction is a 3-d tensor that encodes bounding box, objectness score and prediction over classes. This is why the dimensions of the tensor at the end are changed from past versions to:

$$N \times N \times (3 * (4 + 1 + C)) \quad (2)$$

Where: $N \times N$: is the number of the grid cells of the system
 3: to decode the features extracted from each of the 3 scales
 4 + 1: to decode the bounding boxes offsets + objectness score
 C: is the number of classes we train our model on.

This permits to induce better semantic information from the up-sampled features and finer-grained information from the earlier feature map.

Furthermore, YOLOv3 has been made with another improvement, the new CNN feature extractor named Darknet-53. It is a 53 layered CNN that uses 3×3 and 1×1 convolutional layers. It has demonstrated the advanced accuracy but with less floating-point tasks and better speed. For example, it has less floating-point operations than ResNet-152 but the same performance at a double speed. It utilizes also skip connections network inspired from ResNet.

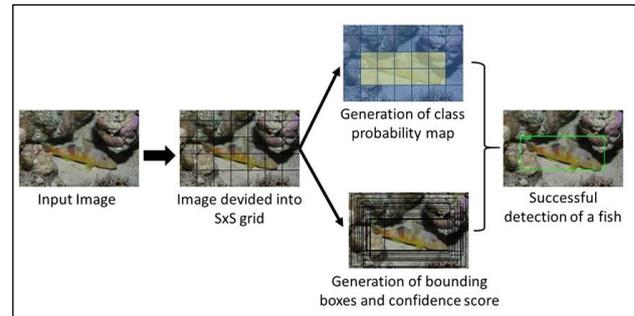


Figure 1. Yolov3 architecture.

2. DATA SET

Dataset of this study is broken down into two sections; the training and testing data and the detection data. The training and testing data used in this study were collected from Open Images Dataset V6 (storage googleapis website). The images were captured in different environments with a 1024×768 -pixel resolution. All the images were taken under other natural and artificial light conditions, including several disturbances: illumination variation, occlusion, and overlap. A total of 1295 Fish images were captured and divided into a training set and a test set. The training set consisted of 70% of the total images, and the remaining 30% of images made up the test set. Figure 2. shows some samples from the dataset under different environments. To assure the validity of our experiment and to test the genericity of the algorithms, we used an underwater video that allowed us to experiment the model's performance in a natural marine environment. Figure 6. shows an example of a frame image from the underwater video that has been recorded in the Moroccan Atlantic coast precisely in a region called Skhirat.

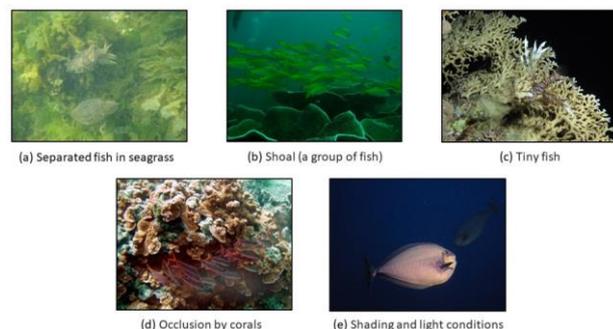


Figure 2. Fish samples with different environment circumstances: (a) separated fish in seagrass, (b) a shoal (a group of fish), (c) tiny fish, (d) occlusion by corals, and (e) shading and light conditions.

2.1 Data Augmentation

The data augmentation technique was used in this study, which help to extend the amount of information by adding adjusted copies of already existing data, and it acts as a regularizer and makes a difference to reduce the overfitting. While training, before input into the model, each image was randomly sampled by one of the following options; the entire original image, horizontal flip, grayscale, blur, noise and cropping. So, each image was horizontally flipped, grayscale was applied to 25% of images, blur was up to 10px, noise was up to 10% of pixels, and for the cropping operation, a patch with the same size as the original image was randomly cropped from image. Some examples of the data augmentation step are illustrated in Figure 3.

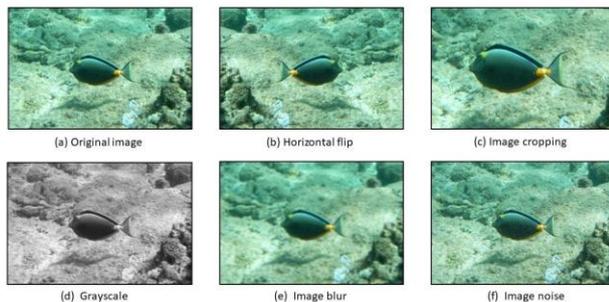


Figure 3. Some examples of the data augmentation step: (a) original image, (b) horizontal flip, (c) image cropping, (d) grayscale, (e) image blur up to 10px, (f) image noise up to 10% of pixels

3. METHODOLOGY

The flowchart of training and detection process of YOLO-fish model is shown in figure 5. The methodology of this study can be broken down into two phases, first one is the training phase; this step consists of three stages. Firstly, data collection, data used in this research is represented by several fish images, then data pre-processing which includes image augmentation and resizing, these images will be split in two sections (the training section which represents about 70% and the testing section which is about 30%). Secondly, data processing, where the training will be started after making the ground truth bounding boxes and the images as inputs to our training model, as a result the predicted bounding boxes plus the confidence scores will be obtained. Thirdly, model's evaluation, so to evaluate our model's performance, the mAP will be calculated based on GT bboxes (Ground truth bounding boxes) and the predicted bboxes. The second phase in methodology is the detection phase, after splitting the video into frames and resizing these unseen frames, we fed the already trained model yolo-fish, in order to predict the bboxes and get the final detection results as a video detection. The experiment was conducted on google collab on a computer that has Intel i7, 64-bit 3.30 GHz CPUs, and a virtual NVIDIA GeForce GTX 1070Ti GPU. The model receives images of 640 × 640 pixels as inputs. The model has been trained for 200 epochs with an initial learning rate of 0.001 and a batch size was set to 64.

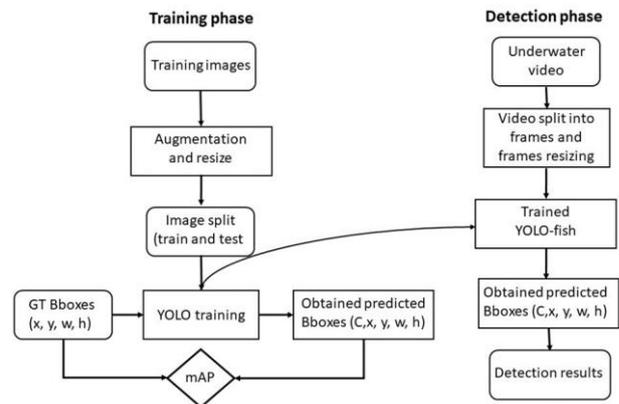


Figure 5. A flowchart of training and detection process of YOLO-Fish model.

A series of experiments were conducted to evaluate the performance of the model. The indexes for evaluation of the trained model are defined as follows:

$$\text{Recall} = \text{TP} / \text{TP} + \text{FN} \quad (3)$$

$$\text{Precision} = \text{TP} / \text{TP} + \text{FP} \quad (4)$$

where TP, FN, and FP are the true positives (correct detection), the false negatives (miss), and the false positives (false detection). The picture in reality may or may not contain any fish but when it contains, the area containing the fish is additionally labelled. In case a specific area contains a fish and the model predicts it accurately, we have a so called True Positive (TP). If the model does not detect any fish in the area and the labelled data confirms that, then this can be referred to as a True Negative (TN). False Positives (FP) tells that fish is identified by the model when none existed within the labelled picture. False Negatives (FN) means that the model failed to distinguish a fish that was there within the picture. To understand this confusion matrix Fig.6 shows all these possibilities.

Another evaluation metric for object detection which is, the Average Precision (AP) was also used in this study. It can show the overall performance of a model under different confidence thresholds, and is defined as follows:

$$\text{AP} = \sum_n (r_{n+1} - r_n) \text{pinterp}(r_{n+1}) \quad (5)$$

With: $\text{pinterp}(r_{n+1}) = \max_{r^{\sim}: r^{\sim} \geq r_{n+1}, p(r^{\sim})}$

Where: $p(r^{\sim})$ is the measured precision at recall r .

		Prediction	
		Fish	No Fish
Ground Truth	Fish	True Positive	False Negative
	No Fish	False Positive	True Negative

Figure 6. Confusion matrix relating TP, TN, FP and FN.

4. EXPERIMENT

Fish training and testing dataset in this study was provided from Open Images Dataset V6 (storage googleapis website). It consists of 1200 images and their annotations. To train and validate the Yolo model we divided dataset into two sections: 70% for training and 30% for testing. Figure 8 shows some samples of the images used in training.

In order to train the model, it is important to provide not only the class of the object but also the bounding box data of the object as correct answer. In this context, we parsed the x, y coordinates, width and height of the fish in the annotation of the image's dataset. In addition, we used data augmentation to extend the amount of data.

The model has better performance as number of epochs increases. However, when number of epochs exceeds 200, the network seems to be overfitted to training data. As a result, Table I shows the hyper-parameters used to train and validate the Yolo model.

Number of iterations	Number of epochs	Learning rate	Batch size	Subdivisions
4000	200	0.001	16	64

Table 1. Hyper-parameters to train the Yolo model.

The video that will be processed by the model in the detection phase, has been provided by the National Institute of Fisheries Research of Casablanca (INRH) and it has been taken underwater by scuba diving. This video is a great real example of an underwater fish movements in different environments.



Figure 6. example of a frame image from the underwater video.

5. RESULTS AND DISCUSSION

The training result is shown in Figure 7. which represent a graph of the relationship between training iterations and average loss.

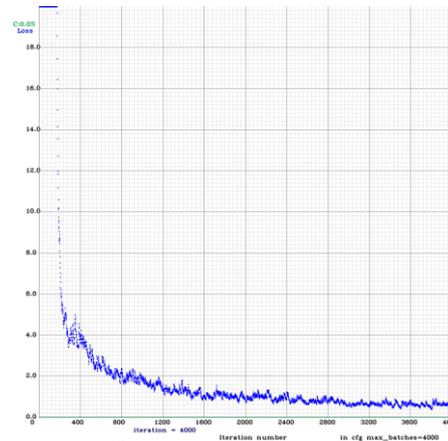


Figure 7. The average loss function of the training.

The graph illustrates the loss while training the neural network and the average loss is reduced to 0.54%. That means the model is affected by training data. A total of 4000 iteration were run and it took 14 hours to complete the training. The model improved swiftly in terms of precision, recall and mean average precision before plateauing after about 1000 iterations, and around 1000 iterations the loss showed a rapid decline.

For each epoch, 64 images are randomly selected and used to train the model. Each image is used multiple times due to the limited number of samples. Figure 8, shows some training samples, and the number zero indicates the fish class.

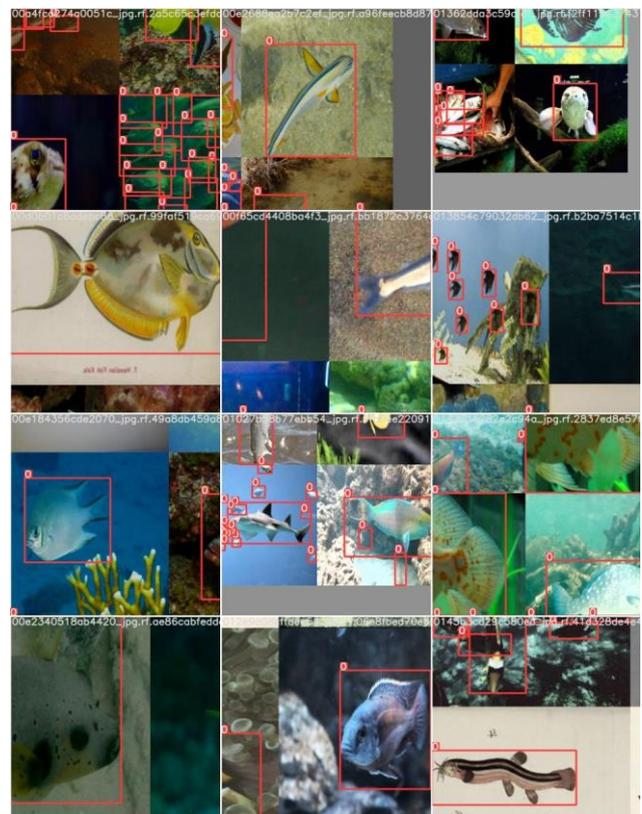


Figure 8. Training samples.

5.1 Image Detection

After training our YOLO model, we tested the network with 30% of the test set, which consists of a new and unseen image. The model recognizes fish in given images in different environments and performs bounding boxes around the detected fishes. As shown in Figure 9 results of fish detection in still images.



Figure 9. Detection results in different environments: (a) detected fish in seaweeds, (b) detected fish in rocky seabed, (c) detection of two separated.

5.2 Video Detection

The detection results within a video and frame images when the proposed method finally detects small and big Fish are shown in Figure 10. The underwater video has been spilt into multiple frames and it is composed of 1032 frames in total, many fish has been detected and classified positive, so Figure 10. will represent some examples of fish detection results within a video frame. It took 532 frames until the fish appeared on the screen and disappeared to the right edge (Figure 10. (a), (b), and (c)), and in the frame image 639, another fish was recognized correctly (Figure 10. (e), (f), and (g)). The detection performance is represented by the cumulative average of the classification performance up to the last frame, even if the model didn't detect some tiny fish in frame 563 (c). Therefore, the proposed method is less likely to yield incorrect classification results. In particular, it has a high classification probability for very slow-moving fish.

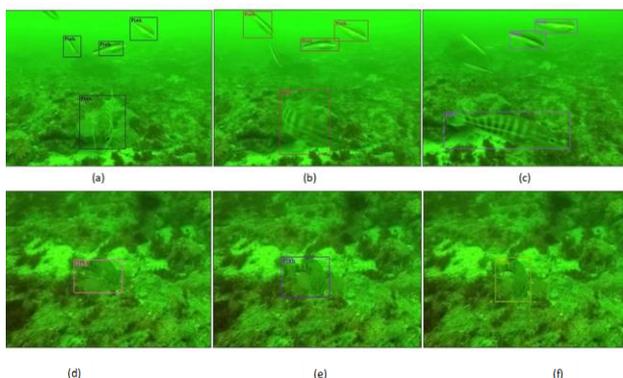


Figure 10. Detection results for video frames: (a) Frame 532, (b) Frame 542, (c) Frame 563, (d) Frame 639, (e) Frame 650, and (f) Frame 657.

In order to evaluate our model, we added some effect and noise to the underwater video to see if the yolo-fish model is accurate in different environments. We have applied the greyscale effect and we lowered the brightness of the original video. As a result, the model recognize correctly fish in different frames, Figure 11. shows the same frames as (Figure 10) so we can compare the detection results between the original video and the edited one. In Frame 639 (d), Frame 650 (e), and Frame 657 (f) the model misclassified the fish, but in the other frame images, ((a) Frame 532, (b) Frame 542, and (c) Frame 563) 50% of fish has been classified positive and misclassified the rest, which is obviously not the same result as the original underwater video. So, the proposed method is 50% accurate even in difficult video recording's circumstance.

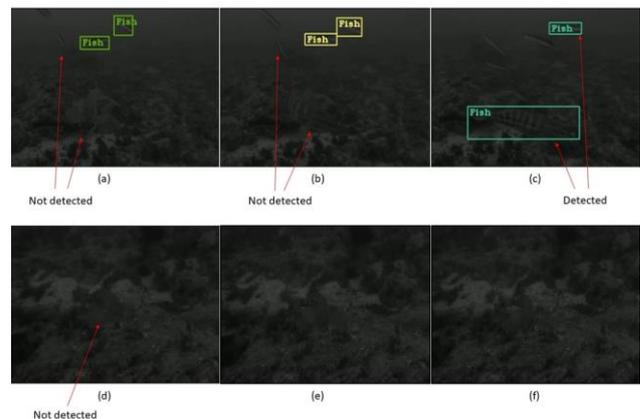


Figure 11. Detection results for video frames in greyscale and low brightness: (a) Frame 532, (b) Frame 542, (c) Frame 563, (d) Frame 639, (e) Frame 650, and (f) Frame 657.

5.3 Model's Evaluation

We used three metrics to measure the performance of the model. First, we measured the accuracy of the classification to see how well the neural network detects the fish. We have prepared 100 images, some of them containing fish and labeled 'positive', and others did not contain fish and labeled 'negative'. We then checked that the network detected the fish in the positive image and did not detect the fish in the negative image. Moreover, in 100 images the total number of predicted bounding boxes was about 371. 7% of them was false positive and 93% was true positive as shown in Figure 12. So as a result, sensitivity was 96% and specificity was 93%.

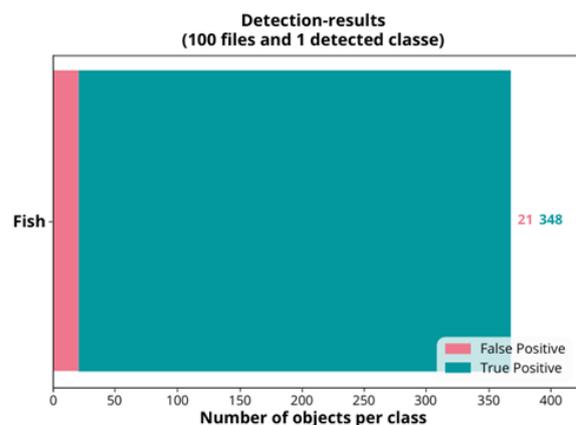


Figure 12. False positive and true positive for 100 images.

Second, the precision–recall curve (P–R curve) Figure 13. This metric was performed to appear the tradeoff between precision and recall for distinctive threshold. The markers indicate the points where recall and precision are obtained when the confidence threshold equals 0.9. High precision reflects a low false positive rate, and high recall relates to a low false negative rate.

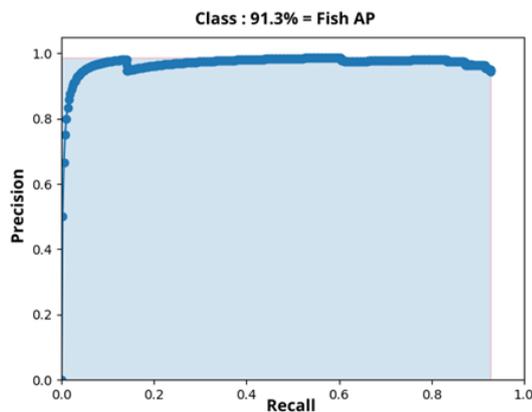


Figure 13. Precision and Recall curve of the model. The markers indicate the points where recall and precision are obtained.

Finally, We evaluated our Yolo-Fish model by calculating the mAP (mean average precision) score; it's a metric for object detection evaluation models, combines both a location and a classification task, and it is calculated by taking the mean AP over our fish class and overall IoU (intersection over union between the ground truth bounding boxes and the predicted ones) thresholds. In our study the mAP was 91,30% as presented in Figure 14.

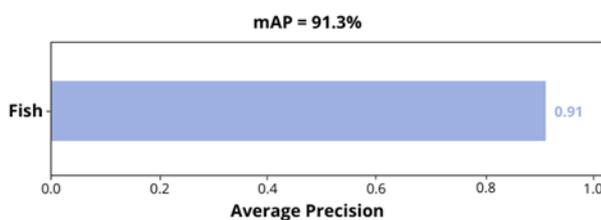


Figure 14. Mean Average Precision.

5.4 Discussion

The marine natural environment is characterized by an important biodiversity made up of species (fauna and flora) of different sizes and shapes. This composition of marine species makes the automatic detection of these marine creatures very difficult, but we can have the greatest potential for improving performance in adjusting the data collection and in improving the data annotation. Our model could detect fish in different natural marine environments, but to improve this detection in a difficult video recording circumstance, it is necessarily to have a wider-angle camera to capture more and more complete fish characteristics, and avoid obstacles in the camera's field of view to increase the resolution of the images.

In the continuity of this work, and to improve our results, we will investigate specifically the difficult recording circumstances where the detection network has failed, as is the case with the detection results for video frames in greyscale and low brightness in Figure 11. Moreover, this work will be expanded to deep learning-based marine species detection with multiple classes. We are going to work on real underwater videos taken for the mapping of benthic bottoms and which allow us to invest in the detection of several objects (species, substrate, etc.)

Through this first experience, we recommend:

1. To continue using high-resolution cameras as it is possible to have a great image or video to identify detailed fish characteristics. Furthermore, the camera should deliver usable images for all light and climate conditions that occurred during the period of the marine information collection project.
2. Increase the number of observations (number of videos) to take advantage of the strategy of increasing the sample size.
3. To slightly modify the architecture of this model to improve the detection rate accuracy.

Finally, we will also study how to adapt our experiment to the new version of YOLO (i.e., YOLOv5), which has recently appeared.

6. CONCLUSION

In this paper, the Yolo-V3 based fish detection method was proposed. We adopted architecture of YOLO for object detection, and trained the network using custom fish images. In addition, we trained the network using non-fish organisms and various type of seabed to enhance specificity of the network. As a result, we could detect fish precisely in different natural environments. To detect object using neural network, the network is trained using various images of target object. For fish detection, due to the body fish protective coloration, there are many cases that seabed was misclassified as fish. Therefore, training the network using data augmentation was helpful to enhance the accuracy. The network can be improved to be applied for multiple marine class objects for marine videos analysis and image information extraction. In addition, our model classifies all fish as a 'positive' class regardless of species. However, it uses multiple images captured in different environments, so if we collect more images by species, we would be able to detect, observe and classify fish species.

REFERENCES

- Amanullah, M. A., Habeeb, R. A. A., Nasaruddin, F. H., Gani, A., Ahmed, E., Nainar, A. S. M., Akim, N. M., & Imran, M. (2020). Deep learning and big data technologies for IoT security. *Computer Communications*, 151(December 2019), 495–517. <https://doi.org/10.1016/j.comcom.2020.01.016>
- Ayaz, A., Acarli, D., Altinagac, U., Ozekinci, U., Kara, A., & Ozen, O. (2006). Ghost fishing by monofilament and multifilament gillnets in Izmir Bay, Turkey. *Fisheries Research*, 79(3), 267–271. <https://doi.org/10.1016/j.fishres.2006.03.029>
- Beyan, C., & Fisher, R. B. (2012). A filtering mechanism for normal fish trajectories. *Proceedings - International Conference on Pattern Recognition*, 2286–2289.
- Brodie, G., Brodie, J., Maata, M., Peter, M., Otiawa, T., & Devlin, M. J. (2020). Seagrass habitat in Tarawa Lagoon,

- Kiribati: Service benefits and links to national priority issues. *Marine Pollution Bulletin*, 155(March), 111099. <https://doi.org/10.1016/j.marpolbul.2020.111099>
- Buhl-Mortensen, L., Buhl-Mortensen, P., Dolan, M. J. F., & Gonzalez-Mirelis, G. (2015). Habitat mapping as a tool for conservation and sustainable use of marine resources: Some perspectives from the MAREANO Programme, Norway. *Journal of Sea Research*, 100, 46–61. <https://doi.org/10.1016/j.seares.2014.10.014>
- Buhl-Mortensen, Lene, Tandberg, A. H. S., Buhl-Mortensen, P., & Gates, A. R. (2016). Behaviour and habitat of *Neohela monstrosa* (Boeck, 1861) (Amphipoda: Corophiida) in Norwegian Sea deep water. *Journal of Natural History*, 50(5–6), 323–337. <https://doi.org/10.1080/00222933.2015.1062152>
- Caccia, M. (2006). Autonomous surface craft: Prototypes and basic research issues. 14th Mediterranean Conference on Control and Automation, MED'06. <https://doi.org/10.1109/MED.2006.328786>
- 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(August), 102117. <https://doi.org/10.1016/j.aquaeng.2020.102117>
- Campbell, S., Naeem, W., & Irwin, G. W. (2012). A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres. *Annual Reviews in Control*, 36(2), 267–283. <https://doi.org/10.1016/j.arcontrol.2012.09.008>
- Casalino, G., Turetta, A., & Simetti, E. (2009). A three-layered architecture for real time path planning and obstacle avoidance for surveillance USVs operating in harbour fields. *OCEANS '09 IEEE Bremen: Balancing Technology with Future Needs*, November 2017. <https://doi.org/10.1109/OCEANSE.2009.5278104>
- Costanza, R. (1999). PII- S0921-8009(99)00079-8.pdf. 31, 199–213.
- Cui, S., Zhou, Y., Wang, Y., & Zhai, L. (2020). Fish Detection Using Deep Learning. *Applied Computational Intelligence and Soft Computing*, 2020. <https://doi.org/10.1155/2020/3738108>
- Du, J. (2018). Understanding of Object Detection Based on CNN Family and YOLO. *Journal of Physics: Conference Series*, 1004(1). <https://doi.org/10.1088/1742-6596/1004/1/012029>
- Gebretsadik, T., & Gebretsadik Wana, T. (2016). A Review on the Causes for the Loss of Major Fishes and Prospects for Future Research in Ethiopia Survey on honeybee pests and predators in Sidama and Gedeo zones of Southern Ethiopia with emphasis on control practices View project A Review on the Causes . July. www.iiste.org
- Holmer, M., Black, K., Duarte, C. M., Marbà, N., & Karakassis, I. (2008). Aquaculture in the ecosystem. In *Aquaculture in the Ecosystem* (Issue December). <https://doi.org/10.1007/978-1-4020-6810-2>
- Hutchings, J. A. (2000). Collapse and recovery of marine fishes. *Nature* 406, 882–885. *Nature*, 406(6798), 882–885. http://www.ncbi.nlm.nih.gov/sites/entrez?Db=pubmed&DbFrom=pubmed&Cmd=Link&LinkName=pubmed_pubmed&LinkReadableName=RelatedArticles&IdsFromResult=10972288&ordinalpos=3&itool=EntrezSystem2.PEntrez.Pubmed.Pubmed_ResultsPanel.Pubmed_RVDocSum%0Ahttp://www.ncbi
- Iqbal, T., & Qureshi, S. (2020). The survey: Text generation models in deep learning. *Journal of King Saud University - Computer and Information Sciences*, xxxx, 1–14. <https://doi.org/10.1016/j.jksuci.2020.04.001>
- Ivašić-Kos, M., Krišto, M., & Pobar, M. (2019). Human detection in thermal imaging using YOLO. *ACM International Conference Proceeding Series, Part F1482*, 20–24. <https://doi.org/10.1145/3323933.3324076>
- Kramer, D. L. (1987). Dissolved oxygen and fish behavior. *Environmental Biology of Fishes*, 18(2), 81–92. <https://doi.org/10.1007/BF00002597>
- Li, S., Song, W., Fang, L., Chen, Y., Ghamisi, P., & Benediktsson, J. A. (2019). Deep learning for hyperspectral image classification: An overview. *IEEE Transactions on Geoscience and Remote Sensing*, 57(9), 6690–6709. <https://doi.org/10.1109/TGRS.2019.2907932>
- Lumauag, R., & Nava, M. (2019). Fish tracking and counting using image processing. 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management, HNICEM 2018, 1–4. <https://doi.org/10.1109/HNICEM.2018.8666369>
- Lynch, A. J., Cooke, S. J., Deines, A. M., Bower, S. D., Bunnell, D. B., Cowx, I. G., Nguyen, V. M., Nohner, J., Phouthavong, K., Riley, B., Rogers, M. W., Taylor, W. W., Woelmer, W., Youn, S. J., & Beard, T. D. (2016). The social, economic, and environmental importance of inland fish and fisheries. *Environmental Reviews*, 24(2), 115–121. <https://doi.org/10.1139/er-2015-0064>
- Mohamed, H. E. D., Fadl, A., Anas, O., Wageeh, Y., Elmasry, N., Nabil, A., & Atia, A. (2020). MSR-YOLO: Method to Enhance Fish Detection and Tracking in Fish Farms. *Procedia Computer Science*, 170(2019), 539–546. <https://doi.org/10.1016/j.procs.2020.03.123>
- Neves, D., Sobral, P., & Pereira, T. (2015). Marine litter in bottom trawls off the Portuguese coast. *Marine Pollution Bulletin*, 99(1–2), 301–304. <https://doi.org/10.1016/j.marpolbul.2015.07.044>
- Olsvik, E., Trinh, C. M. D., Knausgård, K. M., Wiklund, A., Sjørdalen, T. K., Kleiven, A. R., Jiao, L., & Goodwin, M. (2019). Biometric fish classification of temperate species using convolutional neural network with squeeze-and-excitation. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11606 LNAI, 89–101. https://doi.org/10.1007/978-3-030-22999-3_9
- Palmieri, F. A. N., Castaldo, F., & Marino, G. (2013). Harbour surveillance with cameras calibrated with AIS data. *IEEE Aerospace Conference Proceedings*, December. <https://doi.org/10.1109/AERO.2013.6496907>

- Papadakis, V. M., Papadakis, I. E., Lamprianidou, F., Glaropoulos, A., & Kentouri, M. (2012). A computer-vision system and methodology for the analysis of fish behavior. *Aquacultural Engineering*, 46(1), 53–59. <https://doi.org/10.1016/j.aquaeng.2011.11.002>
- Pauly, D., & Zeller, D. (2017). Comments on FAOs State of World Fisheries and Aquaculture (SOFIA 2016). *Marine Policy*, 77(August 2016), 176–181. <https://doi.org/10.1016/j.marpol.2017.01.006>
- Putra, M. H., Yussof, Z. M., Lim, K. C., & Salim, S. I. (2018). Convolutional neural network for person and car detection using YOLO framework. *Journal of Telecommunication, Electronic and Computer Engineering*, 10(1–7), 67–71.
- Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. *ArXiv*.
- Robertson, D. R., & Smith-Vaniz, W. F. (2008). Rotenone: An essential but demonized tool for assessing marine fish diversity. *BioScience*, 58(2), 165–170. <https://doi.org/10.1641/B580211>
- Sharif, M. H., Galip, F., Guler, A., & Uyaver, S. (2016). A simple approach to count and track underwater fishes from videos. 2015 18th International Conference on Computer and Information Technology, ICCIT 2015, 347–352. <https://doi.org/10.1109/ICCITechn.2015.7488094>
- Stavelin, H., Rasheed, A., San, O., & Hestnes, A. J. (2021). Applying object detection to marine data and exploring explainability of a fully convolutional neural network using principal component analysis. *Ecological Informatics*, 62(September 2020), 101269. <https://doi.org/10.1016/j.ecoinf.2021.101269>
- Sun, J., Slang, S., Elboth, T., Larsen Greiner, T., McDonald, S., & Gelius, L. J. (2020). Attenuation of marine seismic interference noise employing a customized U-Net. *Geophysical Prospecting*, 68(3), 845–871. <https://doi.org/10.1111/1365-2478.12893>
- Sun, M., Yang, X., & Xie, Y. (2020). Deep Learning in Aquaculture : A Review. *Journal of Computers*, 31(1), 294–319. <https://doi.org/10.3966/199115992020023101028>
- Sung, M., Yu, S. C., & Girdhar, Y. (2017). Vision based real-time fish detection using convolutional neural network. *OCEANS 2017 - Aberdeen*, 2017-Octob, 1–6. <https://doi.org/10.1109/OCEANSE.2017.8084889>
- Taylor, P., Thomas, M. V., & Haas, R. C. (2011). North American Journal of Fisheries Management Capture of Lake Sturgeon with Setlines in the St . Clair River , Michigan Capture of Lake Sturgeon with. 5947(October 2013). [https://doi.org/10.1577/1548-8675\(1999\)019](https://doi.org/10.1577/1548-8675(1999)019)
- Tian, Y., Yang, G., Wang, Z., Wang, H., Li, E., & Liang, Z. (2019). Apple detection during different growth stages in orchards using the improved YOLO-V3 model. *Computers and Electronics in Agriculture*, 157(January), 417–426. <https://doi.org/10.1016/j.compag.2019.01.012>
- van Niekerk, L., Adams, J. B., James, N. C., Lamberth, S. J., MacKay, C. F., Turpie, J. K., Rajkaran, A., Weerts, S. P., & Whitfield, A. K. (2020). An Estuary Ecosystem Classification that encompasses biogeography and a high diversity of types in support of protection and management. *African Journal of Aquatic Science*, 45(1–2), 199–216. <https://doi.org/10.2989/16085914.2019.1685934>
- Wang, N., Wang, Y., & Er, M. J. (2020). Review on deep learning techniques for marine object recognition: Architectures and algorithms. *Control Engineering Practice*, December 2019, 104458. <https://doi.org/10.1016/j.conengprac.2020.104458>
- Worm, B., Barbier, E. B., Beaumont, N., Duffy, J. E., Folke, C., Halpern, B. S., Jackson, J. B. C., Lotze, H. K., Micheli, F., Palumbi, S. R., Sala, E., Selkoe, K. A., Stachowicz, J. J., & Watson, R. (2006). Impacts of biodiversity loss on ocean ecosystem services. *Science*, 314(5800), 787–790. <https://doi.org/10.1126/science.1132294>
- Yu, C., Fan, X., Hu, Z., Xia, X., Zhao, Y., Li, R., & Bai, Y. (2020). Segmentation and measurement scheme for fish morphological features based on Mask R-CNN. *Information Processing in Agriculture*, 7(4), 523–534. <https://doi.org/10.1016/j.inpa.2020.01.002>