Accredited Ranking SINTA 2
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online on: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 6 No. 2 (2022) 252 - 258 ISSN Media Electronic: 2580-0760

A Comparison of Deep Learning Approach for Underwater Object Detection

Nurcahyani Wulandari[1], Igi Ardiyanto[2], Hanung Adi Nugroho[3]
[1,2,3]Department of Electrical and Information Engineering
Faculty of Engineering, Universitas Gadjah Mada
[1]nurcahyani.wulandari@mail.ugm.ac.id*, [2]igi@ugm.ac.id, [3]adinugroho@ugm.ac.id

Abstract

In recent year, marine ecosystems and fisheries becomes potential resources, therefore, monitoring of these objects will be important to ensure their existence. One of computer vision techniques, it is object detection, utilized to recognize and localize objects in underwater scenery. Many studies have been conducted to investigate various deep learning methods implemented in underwater object detection; however, only a few investigations have been performed to compare mainstream object detection algorithms in these circumstances. This article examines various state-of-the-art deep learning methods applied to underwater object detection, including Faster-RCNN, SSD, RetinaNet, YOLOv3, and YOLOv4. We trained five models on RUIE dataset, then the average detection time used to compare how fast a model can detect object within an image; and mAP also applied to measured detection accuracy. All trained models have costs and benefits; SSD was fast but had poor performance; RetinaNet had consistent performance across different thresholds but the detection speed was slow; YOLOv3 was the fastest and had sufficient performance comparable with RetinaNet; YOLOv4 was good at first but performance dropped as threshold enlargement; also, YOLOv4 needed extra time to detect objects compared to YOLOv3. There are no models that are fully suited for underwater object detection; nonetheless, when the mAP and average detection time of the five models were compared, we determined that YOLOv3 is the best acceptable model among the evaluated underwater object detection models.

Keywords: *Underwater Object Detection, Faster-RCNN, SSD, RetinaNet, YOLOv3, YOLOv4*

1. Introduction

During the COVID-19 pandemic, marine ecosystems and fisheries becomes potential resources in Indonesia. Based on export data released by Central Bureau of Statistics, the demands for fisheries products in March 2020 increased 3.92% compared to the previous year in the same month before the pandemic [1].

Many saltwater species including urchins, sea cucumbers and scallops has seen a tremendous growth in its contribution to state revenue. The total free on board (FOB) value of them exported by Indonesia was approximately 18.9 million U.S. dollars [2]. Therefore, monitoring of these resources will be important to ensure their existence in order to retain aquaculture exports and at the same time maintain natural balance. Many researchers did manual observation to support these monitoring [3]–[5] . But along with various inventions, machines were created to realize automatic surveillance i.e., Autonomous Underwater Vehicle (AUV). This device can follow the fish and capturing them using camera based on real time application. Thus, the fast and accurate method is needed to implement in this system.

Recently, academic participants are figuring out how to apply advanced computer vision technology to support this undersea exploration, such as Bai, et al [6] using HOG for extracting features on zebrafish objects, then perform classification using SVM. However, as data volumes continue to grow and hardware technology advances rapidly, the real-time performance of these classic feature-based algorithms has become comparable to those of deep learning algorithms. There

Accepted: 08-03-2022 | Received in revised: 07-04-2022 | Published: 20-04-2022

is research by Villon, et al [7], who compared HOG + SVM with a deep learning method, selected by CNN, to detect and recognize reef fish. In this research, showed that the efficiency of HOG + SVM is not as good as CNN.

As a result of this excellent CNN performance, the researchers are interested in applying CNN to the underwater object recognition systems. Several studies have been published in the last five years; i.e.: Choi, et al [8] categorized 15 fish species using CNN; the model achieved recall approaching 0.9 and precision greater than 0,8. Rathi, et al [9] also used a CNN model in combination with Otsu's thresholding to classify 12 fish species. In their research, the CNN model obtained an accuracy of 96.29 percent. Cui, et al [10] developed a fish detection system based on 30 CNN layers; this model converged in the 575th epoch with a loss of 0.18. Cueto, et al [11] classified koi fish and resulted accuracy of 84 percent using CNN. Among these investigations, confirms that CNN is capable of accurately classifying underwater objects, as indicated by the high accuracy value.

Additionally, research has been conducted on the application of CNN not only for object recognition but also for identifying the location of multiple objects within an image; i.e.: Xu et al [12] explored R-CNN and feature-based classifier named Haar-cascade. They discovered that R-CNN model has a better performance in Bluefin tuna detection tasks compared with Haar-cascade classifier. The accuracy of Haar-cascade was 53.8% while the R-CNN reached 92.4%. Arvind, et al [13] proposed Mask R-CNN for fish detection, their model generated F1 score of 0.91 and it could detect 16 frames per second. Furthermore, the Fast R-CNN model tested on Image CLEF dataset by Shang, et al [14], it performed 81.4% of mAP and claimed 80 times faster than R-CNN. Mandal, et al [15] proposed the Faster R-CNN for assessing fish abundance on the beaches around Queensland. In this investigation, the performance of the model as measured by mAP was 82.4 percent. While Akdemir, et al [16] implemented SSD to classify three fishes, they discovered that Red Mullet fish had a detection accuracy of 99 percent, Haddock has 89 percent, and Bluefish has 96 percent. Wang, et al [17] investigated the usage of YOLOv3 algorithms for object detection in underwater environment, meanwhile Rosli [18] used YOLOv4 to detect underwater life, such as big fish, jellyfish, crabs, shrimp, small fish and starfish. The model was examined in this investigation and resulted in a mAP of 97.96% and a detection speed of 46.6 frames per second.

Based on these prior studies, the deep learning methods implemented in underwater object detection generated high performance and fast detection, however, there are few studies on comparison among them.

Therefore, this paper investigates the state-of-the-art deep learning methods to recognize and localize underwater object tested on the same dataset in order to suggest the most applicable and swiftest method to be implemented in underwater vehicles. The performance measures by accuracy metrics and time-costs.

Based on our investigation, the SSD, Faster R-CNN, RetinaNet, YOLOv3 and YOLOv4 algorithms are the most commonly used, so in this study, we are focusing on these five deep learning methods.

The section of this work is divided into the following sections: Section 2 discusses the theory, design, and training information for the five deep learning approaches. In this section, the dataset trained on the models and experimental scenario also explained. Then, section 3 highlights the performance and detection time comparative results. Finally, Section 4 contains the conclusion.

2. Research Methods

2.1 Faster R-CNN

Shaoqing Ren designed a faster R-CNN structure in 2016 [19]. Faster R-CNN and RPN are the foundation of first-place winning submissions in the ILSVRC and COCO 2015 competitions in several subject.

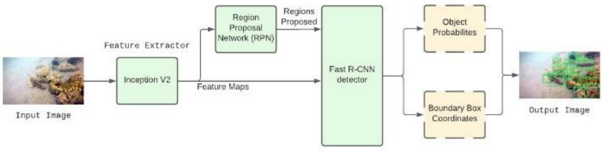

Figure 1. Faster R-CNN Architecture

Faster R-CNN composed of two main modules, namely the network proposal region (RPN) and Fast R-CNN detector, they are combined into a single network by virtue of their shared convolutional features

The Regional Proposal Network (RPN) is intended to predict regional proposals of varying scales and aspect ratios. RPN is a neural network that consists of 3 convolutional layers. One is called the feature extractor, which aims to produce nice features. Our study adopts Inception V2 as the backbone network for feature extractor, as shown in Figure 1. Furthermore, the feature maps which is resulting from the feature extractor, then fed into two sibling fully connected layers, namely and the box-regression layer (reg) and the box-classification layer (cls) The regression layer in charge of making bounding boxes and classification layers helps to identify the containing objects in each anchor, the probability is between 0 and 1, represents foreground and background.

Following that, the RPN module's output instructs the Fast R-CNN to classify any predicted bounding boxes within the image.

DOI: https://doi.org/10.29207/resti.v6i2.3931

2.2 SSD

The SSD (Single Shot Detector) algorithm is a single-stage detection model that allows object localization and classification to be done in a single forward pass of the neural network. SSD algorithm claimed to be faster and easier to train than Faster R-CNN [20]. The fundamental improvement in speed comes from eliminating region proposals and feature resampling stage.

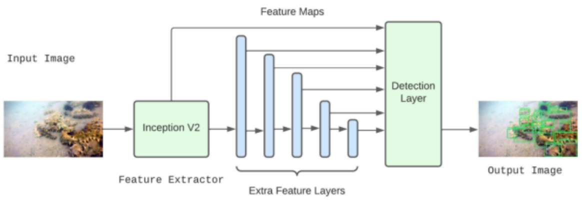

Figure 2. SSD Architecture

Image just fed into the network and then a single network utilized to predict the objects within an image. SSD predicts the offset for default boxes with varying sizes and aspect ratios in several feature layers, and then applies a 3x3 convolution to each feature dimension to provide box and class outputs. Then, all the outputs are combined in the end of the network to apply non maximum suppression.

In this work, we compare SSD and Faster R-CNN performance to discover the difference of single-stage and two-stage detection model. The key to a fair comparison of deep learning algorithms is ensuring that each algorithm is evaluated in the same way on the same data. Therefore, our study utilized the same Inception V2 model as base network.

2.3 RetinaNet

The two-stage approach generates high accuracy on object detection, but it takes enough time to detect the objects. In the other hand, the one-stage detectors have the potential results in terms of speed, but have typically short for accuracy than two-stage detectors. RetinaNet [21] combines the benefits of single-stage and two-stage detectors, including great accuracy, rapid run time, and minimum memory cost. RetinaNet solves the class imbalance issue found in the one-stage detector by using a novel loss function commonly called focal loss. In the case of object detection, class imbalance problem can be represented as follows: while the object of interest containing only few pixels inside an image and most pixels are background, the detector sometimes fails to predict the object. Thus, we need to giving the model a bit toleration to take some risk when making predictions, so even a small piece of the object can be covered. Focal loss works by adding a modulation factor $(1 - pt)^{\gamma}$ to the conventional cross entropy loss in order to reduce the weighting of well-classified cases and rapidly focus the model on challenging examples.

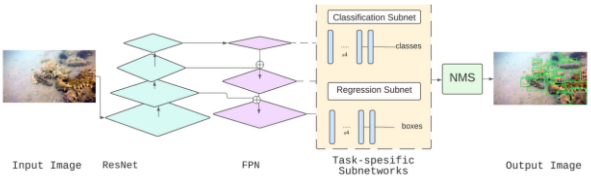

Figure 3. RetinaNet Architecture

As seen in Figure 3, RetinaNet begins with Resnet-101 with FPN as its backbone network, followed by two task-specific subnetworks: the classification subnet and the box regression subnet.

The classification subnet is a fully convolutional network (FCN) that is connected to each FPN level and is responsible for predicting the probability of an object being present at each spatial position.

Moreover, each pyramid level also has a box regression subnet, which is also a tiny FCN. This subnet is responsible for regressing the offset from each anchor box.

2.4 YOLOv3

YOLOv3 is basically a one-stage detector and is the third version of the YOLO detector. Here follows are the modifications made by YOLOv3: Firstly, The Bounding Box Predictions: Using logistic regression, YOLOv3 predicts the objectness score and the bounding box by assigning a single bounding box prior to each ground truth object.

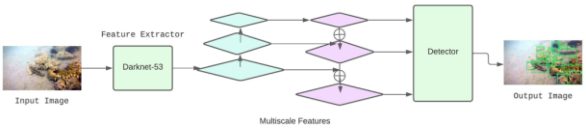

Figure 4. YOLOv3 Architecture

Secondly, The Class Prediction: YOLOv3 employs distinct logistic classifiers for each class, rather than a single softmax layer, and performs training using binary cross-entropy loss. This formulation is thought to be capable of generating multiple label classifications.

Following that, The Predictions Across Scales: YOLOv3 predicts boxes at three different scales. YOLOv3 extracts features from those scales in a similar way to FPN, which implies that feature maps in the earlier network will be concatenated to generate upsampled features. This strategy enables YOLOv3 to acquire more relevant semantic information, which enables it to generate more accurate predictions at various sizes.

Lastly, Finally, The Feature Extractor: YOLOv3 makes use of a new feature extractor officially called Darknet-53, which is named after the network's 53 nodes. It outperforms Darknet-19 but remains more efficient than ResNet-101 or ResNet-152.

YOLOv3 has a short running time, is reported to be three times quicker, and has an accuracy comparable to SSD [22].

2.5 YOLOv4

Alexey Bochkovsky et al [23] created YOLOv4 as an enhancement to YOLOv3. They summarize that object detection consists of the following components:

a. Input: refers to the data that is initially delivered into the system for further processing by succeeding layers of artificial neurons; input may take the form of an Image, Patches, or Image Pyramid;
b. Backbone: refers to the feature extractor, which creates a representation of the input as a feature map;
b. Neck: Neck layers compile feature maps from many stages and are made of numerous bottom-up and top-down paths;
c. Head: refers to object detector, it essentially determines the area in which the item may be found.

On the basic principle of this composition, the YOLOv4 is created by combining CSPDarknet53 as the "Backbone," then the SPP (Spatial Pyramid Pooling) and PAN (Path Aggregation Network) as the "Neck," furthermore the YOLOv3 applies as the "Head."

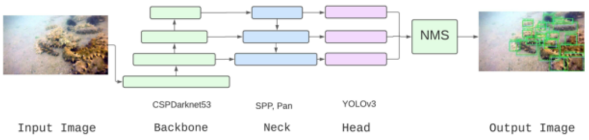

Figure 5. YOLOv4 Architecture

On the MS COCO dataset, YOLOv4 increased the mAP (mean Average Precision) by 10% and the number of FPS (Frames per Second) by 12% over YOLOv3. Additionally, training this neural network on a single GPU has becoming simpler.

2.6 Software and Hardware Environment

Python is the major language utilized in the development of the five models. The models were trained and tested using a PC equipped with an Intel i7-10510U quad-core CPU, 16 GB of RAM, and an NVIDIA GeForce MX230 graphics card.

2.7 Dataset

Due to the limited amount of data from Indonesia's waters, we relied on the dataset supplied by Liu [24]. The data set, popularly called Real-World Underwater Image Enhancement (RUIE), is comprised of 250 hours of video footage and results in an image with a range of differences in illumination, depth, blur level, color.

The entire video was shot in two-time slots, from 8 a.m. to 11 a.m., and 1 p.m. to 4 p.m. in the afternoon, every day between September 21 and 22, 2017. The water depth varies from 5 to 9 meters. They selected over 4,000 images, which 1,800 of them were labeled. The annotations contain marine life objects, including sea urchins, sea cucumbers, and scallops.

Table 1. Number of Objects in Dataset

Labeling Name	Train Set	Test Set
Urchin	7,462	2,520
Sea Cucumber	5,776	1,972
Scallop	5,479	1,884

In this study, we splitted randomly these1,800 labeled images into three portions, they are 1,200 used as train data, 300 images as validation data, and the rest 300 data used in testing process, which have many numbers of objects as we can see in Table 1. All these images just fed into models without any further enhancement step.

2.8 Evaluation Metrics

We employed prominent measures such as precision, recall, average precision (AP), and mean average precision to assess the detection accuracy of five models (mAP).

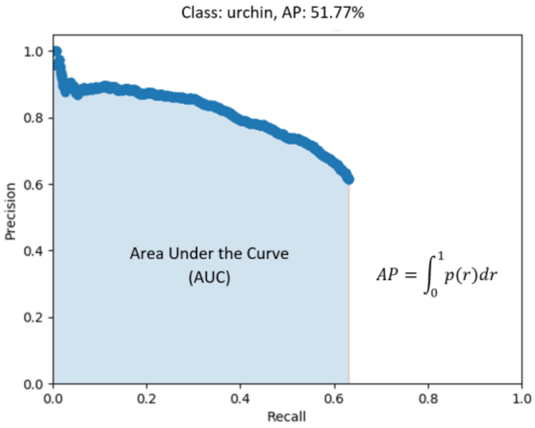

Figure 6. Precision-Recall Curve (PR Curve)

Precision quantifies the model's predictive accuracy, i.e., the percentage of correct positive predictions. However, recall refers to the percentage of true positives that can be accurately detected. As seen in Figure 6, the commonly used definition of Average Precision (AP) is the area under the curve obtained from numerical integration.

Mean Average Precision (mAP) is a statistic that is often employed in issues involving object detection [14], [15], [19], [20], [23]. It is the average precision across all recall values at all IoUs for prediction and ground truth, limited by predefined thresholds [25].

The IOU is defined as the area of overlap between the prediction (B_{pred}) and the ground truth (B_{truth}) divided by the area of union, while the threshold is a predetermined parameter used to identify true positive (TP) and false positive (FP). When the bounding box is bigger than the threshold, it is referred to as TP, and when it is less than the threshold, it is referred to as FP.

3. Results and Discussions

We compared and assessed the performance of five deep learning models for underwater object detection. Then, we examined each algorithm's performance on a few test examples. Among these, we chose sample image to illustrate the algorithm's impact graphically. These are presented in Figure 7. From top to bottom, Faster R-CNN, SSD, RetinaNet, YOLOv3 and YOLOv4 are used to identify underwater objects. Here are the outcomes:

Table 2. Performance of five models

Method	*Mean Average Precision (%)*						*Average Detection Time (s)*
	mAP_{10}	mAP_{20}	mAP_{30}	mAP_{50}	mAP_{60}	mAP_{75}	
Faster R-CNN	58.88	58.26	57.22	51.05	44.04	22.27	1.42
SSD	36.97	36.80	36.55	33.64	28.21	11.10	0.81
RetinaNet	81.72	81.57	81.35	**80.12**	**76.93**	**56.39**	2.23
YOLOv3	83.07	82.96	82.86	77.87	62.72	12.73	**0.50**
YOLOv4	**83.70**	**83.59**	**83.36**	75.08	53.15	9.36	0.65

The PR curves of the objects are shown in Figure 8 at an IoU threshold of 0.5, where the average detection accuracy and detection time are used as metrics to evaluate the performances of five object detection models used to underwater object detection, as listed in Table 2.

Firstly, we compared the performance of one-stage and two-stage detectors in order to determine whether strategy is more appropriate for the RUIE dataset. SSD was used to represent the one-stage detector in this experiment, while Faster-RCNN was used to represent the two-stage detector. The critical factor in conducting a fair comparison is ensuring that each method is assessed on the same way. As a consequence, we used the same Inception V2 model as the base network, batch size equal to 1, and initial learning rate is 0.0001. The models trained on 10,000 epochs and here our investigation: According to figure 7, 7(b) shows the Faster R-CNN result, whereas 7(c) represents the SSD result. The Faster R-CNN can recognize more objects than the SSD, and the confidence score provided by the Faster R-CNN is also greater, even close to 1. As a result, Faster R-CNN outperforms SSD in terms of performance.

Furthermore, according to figure 8, the area under the precision-recall curve for the "green" line is greater than that for the "red" line, which occurs for all objects, which explains why the mAP of Faster R-CNN is greater than that of SSD, implying that Faster R-CNN had a relative advantage in detection accuracy. Although the recall values of SSD and Faster R-CNN are lower than those of other detectors, this indicates a high rate of miss detection. Otherwise, as consequence of going twice, as shown in Table 2, the detection speed of the Faster R-CNN detector is higher compared to the SSD detector, it indicates that Faster R-CNN is slower than SSD.

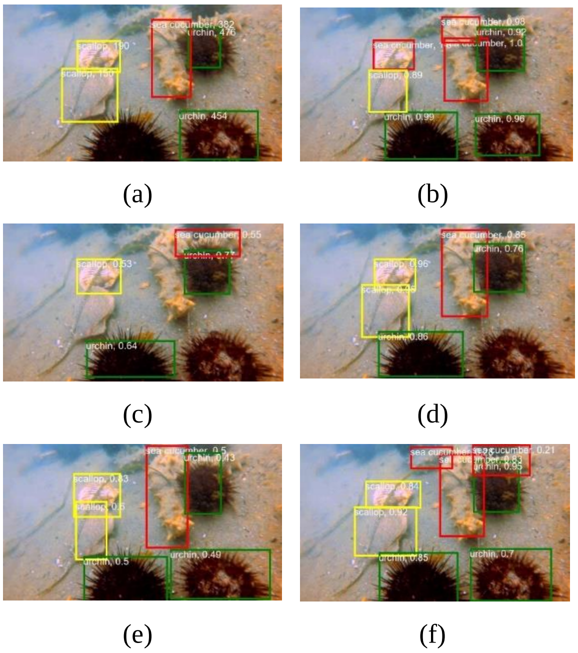

Figure 7. Detection results of the models: (a) Ground Truth, (b) Faster R-CNN, (c) SSD, (d) RetinaNet, (e) YOLOv3, (f) YOLOv4

So, due to the results of first experiment, then we attempted further trial to find models which is having detection speed approximating SSD speed, and has an accuracy close to Faster R-CNN. Several state-of-the-art of one-stage detector, namely RetinaNet, and YOLO were trained. Since Liu [24] using YOLOv3 as detector in their experiment, so this method reconstructed in this research, however the later version of YOLO, it is YOLOv4, also included.

Based on the PR curve and supported with mAP@ [0.1:0.2:0.3:0.5:0.6:0.75] showed on Table 2, we can see that the performance of RetinaNet outperformed both SSD and Faster R-CNN in all class labels. Furthermore, RetinaNet provided highest score on mAP_{75} of all models, it indicates that RetinaNet is stable and consistent across difference confidence thresholds, it generated high quality of detection. But RetinaNet had highest average detection time compared to all models as shown in Table 2.

Whereas YOLOv3 is relatively faster than SSD, also the detection accuracy close to Faster R-CNN and RetinaNet, while IoU threshold under 0.5 (t = 0.1:0.2:0.3). But as threshold increased, the performance of YOLOv3 actually decreased. When a higher IOU threshold t was considered (t = 0.5), the mAP reduced about 5%, even the mAP dropped significantly while threshold escalated, especially in mAP_{75}. Compared to RetinaNet, a YOLOv3 model unsuccessful in maintaining performance consistency. However, while looking at average precision (AP) of

DOI: https://doi.org/10.29207/resti.v6i2.3931

three classes, as seen in Figure 8, with higher recall and precision value means that YOLOv3 has great detection rate for the targets of all objects.

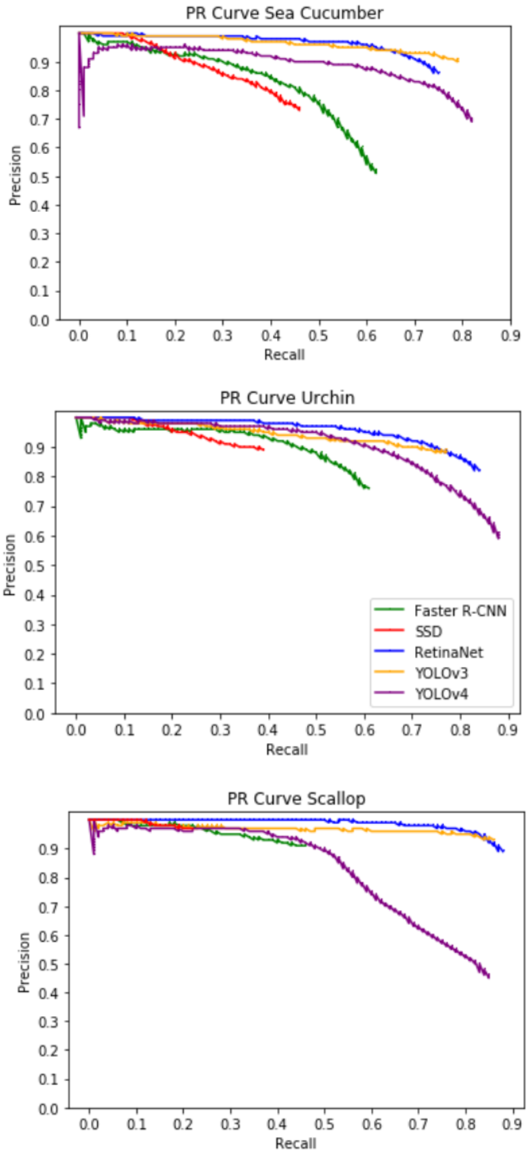

Figure 8. From top to bottom; the PR Curve of Sea Cucumber, Urchin, and Scallop

Moreover, as seen at Table 2, it was found that YOLOv4 defeated YOLOv3 in terms of accuracy; on the contrary, the detection speed was slower. The YOLOv4 had higher mAP_{10} compared to YOLOv3, visualizing them showed that the few predictions from YOLOv4 unsuccessful while predicted using YOLOv3, but the YOLOv4 predictions had low confidence score, as a result of the percentage of false positive increased significantly while IoU threshold raised. Moreover, the PR Curve of YOLOv4 showed higher recall but lower precision value, this means YOLOv4 have a large false detection rate.

4. Conclusion

Five deep learning models are examined in this article for underwater object detection. Each model was developed using data from Liu's study. Popular measures for evaluating model performance include accuracy, recall, average precision (AP), and mean average precision (mAP), as well as average detection time

The two-stage detector was shown to be more accurate than the one-stage detector, this is seen by the fact that mAP_{50} of two-stage detector was 51.05%, whereas the one-stage detector was 33.64%. Meanwhile the one-stage detector had a quicker detection time, with an average difference in detection time of around 0.61 seconds.

The one-stage detectors, namely SSD, RetinaNet, YOLOv3, and YOLOv4 have costs and benefits; SSD was fast but lack of performance, it has the smallest mAP among all models; since RetinaNet was stable and had consistent performance across high thresholds, it generated mAP@0.75 of 56.39% while the others only ranged from 9-22%, however the detection speed of RetinaNet was slow, it needed about 2.23 seconds to detect objects within an image; YOLOv3 was the fastest, it only need 0.5 second in detection and had performance close to RetinaNet at mAP_{50} equals to 77.87%; while YOLOv4 was good in the beginning but the performance dropped along with threshold enlargement, also YOLOv4 needed extra time to detect objects compared to YOLOv3.

There are no models that is perfectly matched implement on AUV device for underwater object detection, however between the five models compared both mAP and average detection time, we concluded that YOLOv3 is the most suitable model among the tested object detection models because YOLOv3 had good performance and fastest detection speed.

We recommend that our findings might serve as consideration for underwater object detection implementation utilize in an AUV for underwater surveying in the future.

References

[1] Badan Statistik Nasional, "Ekspor Ikan Segar/Dingin Hasil Tangkap menurut Negara Tujuan Utama, 2012-2020," 2021. https://www.bps.go.id/statictable/2019/02/25/2024/ekspor-ikan-segar-dingin-hasil-tangkap-menurut-negara-tujuan-utama-2012-2020.html (accessed Jan. 12, 2022).

[2] Direktorat Jenderal Penguatan Daya Saing Produk Kelautan dan Perikanan, *Statistik Impor Hasil Perikanan Tahun 2016-2020*. 2021.

[3] A. A. Sentosa and D. Wijaya, "Potensi invasif ikan zebra Cichlid (Amatitlania nigrofasciata Günther, 1867) di Danau Beratan, Bali ditinjau dari aspek biologinya," *BAWAL Widya Ris. Perikan. Tangkap*, vol. 5, no. 2, pp. 113–121, 2013.

[4] M. Dan, A. Ronny, S. Balai, P. Pemulihan, D. Konservasi, and S. Ikan, "Karakteristik Habitat Ikan Kerapu Di Kepulauan Karimunjawa, Jawatengah the Characteristic Habitat of

Grouper Fish in Karimunjawa Islands, Central Java," *BAWAL Cilalawi*, vol. 7, no. 1, pp. 147–154, 2015.

[5] D. W. H. Tjahjo, S. E. Purnamaningtyas, and A. Suryandari, "Evaluasi Peran Jenis Ikan Dalam Pemanfaatan Sumber Daya Pakan Dan Ruang Di Waduk Ir. H. Djuanda, Jawa Barat," *J. Penelit. Perikan. Indones.*, vol. 15, no. 4, p. 267, 2017, doi: 10.15578/jppi.15.4.2009.267-276.

[6] Y. X. Bai *et al.*, "Automatic multiple zebrafish tracking based on improved HOG features," *Sci. Rep.*, vol. 8, no. 1, pp. 1–14, 2018, doi: 10.1038/s41598-018-29185-0.

[7] S. Villon, M. Chaumont, G. Subsol, S. Villéger, T. Claverie, and D. Mouillot, "Coral Reef Fish Detection and Recognition in Underwater Videos by Supervised Machine Learning: Comparison Between Deep Learning and HOG+SVM Methods," in *Lecture Notes in Computer Science*, vol. 10016 LNCS, 2016, pp. 160–171.

[8] S. Choi, "Fish identification in underwater video with deep convolutional neural network: SNUMedinfo at LifeCLEF fish task 2015," *CEUR Workshop Proc.*, vol. 1391, pp. 1–5, 2015.

[9] D. Rathi, S. Jain, and S. Indu, "Underwater Fish Species Classification using Convolutional Neural Network and Deep Learning," *2017 9th Int. Conf. Adv. Pattern Recognition, ICAPR 2017*, pp. 344–349, 2018, doi: 10.1109/ICAPR.2017.8593044.

[10] S. Cui, Y. Zhou, Y. Wang, and L. Zhai, "Fish Detection Using Deep Learning," *Appl. Comput. Intell. Soft Comput.*, vol. 2020, 2020, doi: 10.1155/2020/3738108.

[11] M. S. Cueto, J. M. B. Diangkinay, K. W. B. Melencion, T. P. Senerado, H. L. P. Taytay, and E. R. E. Tolentino, "Classification of different types of koi fish using convolutional neural network," *Proc. - 5th Int. Conf. Intell. Comput. Control Syst. ICICCS 2021*, no. Iciccs, pp. 1135–1142, 2021, doi: 10.1109/ICICCS51141.2021.9432358.

[12] G. Xu *et al.*, "Detection of Bluefin Tuna by Cascade Classifier and Deep Learning for Monitoring Fish Resources," *2020 Glob. Ocean. 2020 Singapore - U.S. Gulf Coast*, pp. 2020–2023, 2020, doi: 10.1109/IEEECONF38699.2020.9389012.

[13] C. S. Arvind, R. Prajwal, P. N. Bhat, A. Sreedevi, and K. N. Prabhudeva, "Fish Detection and Tracking in Pisciculture Environment using Deep Instance Segmentation," *IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON*, vol. 2019-Octob, pp. 778–783, 2019, doi: 10.1109/TENCON.2019.8929613.

[14] X. Li, M. Shang, H. Qin, and L. Chen, "Fast accurate fish detection and recognition of underwater images with Fast R-CNN," *Ocean. 2015 - MTS/IEEE Washingt.*, pp. 1–5, 2016, doi: 10.23919/oceans.2015.7404464.

[15] R. Mandal, R. M. Connolly, T. A. Schlacher, and B. Stantic, "Assessing fish abundance from underwater video using deep neural networks," *Proc. Int. Jt. Conf. Neural Networks*, vol. 2018-July, pp. 1–6, 2018, doi: 10.1109/IJCNN.2018.8489482.

[16] K. U. Akdemir and E. Alaybeyoglu, "Classification of red mullet, bluefish and haddock caught in the black sea by 'single shot multibox detection,'" *2021 Int. Conf. Innov. Intell. Syst. Appl. INISTA 2021 - Proc.*, pp. 21–24, 2021, doi: 10.1109/INISTA52262.2021.9548488.

[17] C. C. Wang, H. Samani, and C. Y. Yang, "Object Detection with Deep Learning for Underwater Environment," *Proc. 4th Int. Conf. Inf. Technol. Res. Bridg. Digit. Divid. Through Multidiscip. Res. ICITR 2019*, 2019, doi: 10.1109/ICITR49409.2019.9407797.

[18] M. S. A. Bin Rosli, I. S. Isa, M. I. F. Maruzuki, S. N. Sulaiman, and I. Ahmad, "Underwater Animal Detection Using YOLOV4," *Proc. - 2021 11th IEEE Int. Conf. Control Syst. Comput. Eng. ICCSCE 2021*, no. August, pp. 158–163, 2021, doi: 10.1109/ICCSCE52189.2021.9530877.

[19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.

[20] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Lecture Notes in Computer Science*, Dec. 2016, vol. 9905 LNCS, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.

[21] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 2999–3007, doi: 10.1109/ICCV.2017.324.

[22] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *Proc. 4th Int. Conf. Inf. Technol. Res. Bridg. Digit. Divid. Through Multidiscip. Res. ICITR 2019*, Apr. 2018, [Online]. Available: http://arxiv.org/abs/1804.02767.

[23] A. Bochkovskiy, C. Wang, and H. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," Apr. 2020, [Online]. Available: http://arxiv.org/abs/2004.10934.

[24] R. Liu, X. Fan, M. Zhu, M. Hou, and Z. Luo, "Real-World Underwater Enhancement: Challenges, Benchmarks, and Solutions Under Natural Light," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 12, pp. 4861–4875, Dec. 2020, doi: 10.1109/TCSVT.2019.2963772.

[25] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010, doi: 10.1007/s11263-009-0275-4.

DOI: https://doi.org/10.29207/resti.v6i2.3931