https://doi.org/10.24191/jeesr.v18i1.005

CNN-based YOLOv3 Comparison for Underwater Object Detection

Mohamed Syazwan Asyraf, Iza Sazanita Isa, Mohd Ikhmal Fitri Marzuki, Siti Noraini Sulaiman, and Chin Chang Hung

***Abstract*—Object detection that deals with identifying and locating object is one of area that integrate from the advance- ment in machine learning and computer vision. Modern object detection which carried out supervised learning utilizes Convo- lutional Neural Network (CNN) as the backbone of the detection architecture which is significant for underwater object detection as the underwater images are usually low in quality and blurry. Single stage detection such as You Only Look Once (YOLO) is one the famous object detection model that is prominent among researchers due to high performance in accuracy and processing speed. However, YOLO has many versions where the current incremental improvement model of YOLOv3 has been widely used by researchers to solve different types of problem relatedto object detection. Therefore, there is a need to explore thetrade-off relationship between the processing speed and precisionof each YOLO model. In the study, two different open source underwater datasets were used in four different YOLOv3 modelsnamely as YOLOv3-SPP, YOLOv3-Tiny, YOLOv3-Tiny-PRN and the original YOLOv3 in order to study their performance based on metrics evaluation of precision and processing speed (FPS). The result shows that YOLOv3-SPP proved to be the best in terms of precision while YOLOv3-Tiny-PRN lead in terms of execution speed. So, this study shows that YOLOv3 model is highly significant to be implemented and able to accurately detect underwater objects with haze and low-light environment. This study can help researchers and industry in determining the best YOLOv3 model specifically for detection of the underwater images and its application.**

***Index Terms*—Underwater Detection, CNN, YOLOv3, YOLOv3-SPP, YOLOv3-Tiny, YOLOv3-Tiny-PRN**

I. INTRODUCTION

IN the recent few years, deep learning which is one of the sub-sets in Artificial Intelligence (AI) has driven the worldchanging breakthrough in today's technology advancement.

This manuscript is submitted 17 November 2020 and accepted on 16th March 2021. This work was supported by grants from the Ministry of Higher Education (MOHE), Malaysia under the FRGS grant of 600-RMI/FRGS 5/3 (291/2019).

Mohamed Syazwan Asyraf, Iza Sazanita Isa, Mohd Ikhmal Fitri Marzuki, Siti Noraini Sulaiman are with Faculty of Electrical Engineering, Universiti Teknologi MARA Cawangan Pulau Pinang, 13500 Permatang Pauh, Pulau Pinang, Malaysia (email: syazwanasy2@gmail.com).

Chin Chang Hung is with Department of Oceanography, National Sun Yat-sen University, No70. Lianhai Rd., Kaohsiung, 80424, Taiwan.

The acceleration is due to the researchers believe that computer model seems to fit and act more closely with a human brain which apply the concept like neuroscience. Hence, the capability of mimicking neurons connection as in human brain has made deep learning competent enough to carry out data-related task such as classifying object, recognizing something and information matching. Moreover, in classifying and recognizing, an object detection is one of the areas in computer vision that able to take great leaps from the advancement of deep learning. Object detection involve a process to train computer to understand visual data representation and hierarchical features from labelled input images such as colours, shapes, distance, border and many more [1], [2].

In order to produce an efficient object detection model with high precision and low processing time, there were many different networks have been experimented in the algorithm development process. That is included traditional approach such as Histogram of Oriented Gradient (HOG) and Support Vector Machine (SVM) while modern machine learning utilized Convolutional Neural Network (CNN) as based networks. By the way, it has been proved by [2] that object detection which utilize deep learning has gained much popularity compared to traditional method due to its flexibility and supremacy in the accuracy. Besides, the need of domain expertise and human intervention in traditional machine learning has made many researchers believed in deep learning as there were many comparative studies had proven that the performance of deep learning based on CNN surpassed the traditional methods [1], [2], [3], [4].

The main factor for object detection using deep learning with CNN is highly significant was due to the inclusion of classification and object localization. In other way, CNN gives benefit for the image classification approach since it has the ability to learn by assigning weights and biases to various objects in the image. Hence, it able to differentiate from one with the other. Generally, there are two types of modern object detection approaches which are multistage detection and single-stage detection. The pioneer of modern object detection that used multi-stage detection algorithm is Region based Convolutional Neural Network (R-CNN) [5] followed by improvement algorithm which is Fast R-CNN [6] and later is Faster R-CNN [7]. R-CNN operates by selective search method to generate region proposals for object detection. Meanwhile, current approaches such as Fast R-CNN and Faster R-CNN able to achieve good accuracy but have limited ability to achieve

sufficient speed for real time implementation [8]. Currently, YOLO (You Only Look Once) [9] is one of the famous architectures for single-stage detection. This architecture is famous due to its efficacy, fast and accurate [8], [10]. YOLO architecture has three version which is YOLOv1[11], YOLOv2 [12] and the latest is YOLOv3 [9].

Single stage detector that focuses on YOLO architecture has been widely used in many applications for object detection compare to R-CNN architecture. This is due to YOLO's capability to achieve good precision along with short processing time that make YOLO able to be executed for realtime application. Incremental improvement in YOLOv3 has seen other architecture that still based on YOLOv3 such as Tiny-YOLOv3, Tiny-Prn-YOLOv3 and YOLOv3-SPP were developed in order to achieve different objective [13], [14], [15]. With their uniqueness of architecture that make it different from original architecture, all these YOLO-type detection model have their on benefits and function in terms of processing speed and the precision in detecting an object. Therefore, many researchers have implemented these models to solve the problem related to classification and localization [16], [17]. The application of different YOLOv3 architectures may have significant impact in terms of the accuracy and processing speed. Therefore, it is crucial to study the relationship between processing speed and precision of each model. Herein, a comparative study was carried out to analyse the performance of four different architectures of YOLOv3 that focused on the performance comparison of different models. This study also explores the effect of CNN architecture towards detection speed (frame per second) and mean average precision (mAP). Two open-source underwater datasets were used in this study. Both datasets were trained and validated separately which later the model was tested using different set in order to validate the performance in terms of the precision (mAP) and the processing speed (FPS).

II. Literature Review

Object detection is one of the main element in computer vision where it functions to specifically recognize objects and also locate them in an image. Single-stage detection like YOLO treats object detection as regression task by taking input images to set fix number of predictions on grid and learn the class probabilities along with bounding box coordinates. This section is divided into two sections where the first section will reviewed related studies that implemented YOLOv3 as object detection and the other will be the explanation on the YOLOv3 architecture.

A. Related Research and Implementation

Self-driving car is one of the areas that have vast demand on the efficiency of computer vision application. Nugraha et al [16] have conducted a study in contributing to the application of computer vision and deep learning for self-driving car that utilized CNN and road lane detector. The study implemented YOLOv1 as object detector while polynomial regression as road lane guidance. The proposed method had successfully achieved good performance and fast processing time which revealed that there are three main application of adjusting road lane, detect an object and provide steering suggestion. Apart of the lane decision, traffic sign recognition is also an important part as reported by Yusuf et al [15] whose implemented CNN based TINY-YOLO algorithm to detect and classify road sign in only two steps which identify possible sign locations in image and hence classify the signs.

A research by Aleksa et al [17] has implemented YOLO algorithm for real-time detection for traffic participant that include pedestrian for autonomous car application. The Berkley Deep Drive dataset was used to train and validate the YOLOv3 model. The model with challenging dataset have variety of driving conditions such as bright and overcast sky, night, snow and fog had successfully achieved 46.60% of mAP and 25 fps processing time for HD images. Another research that applied a challenging dataset such as severe weather conditions for pedestrian detection was conducted by Tumas et al [13]. The thermal type dataset namely ZUT (Zachodniopomorski Uniwersytet Technologiczny) has successfully trained YOLOv3 model with mAP up to 89.1% and Tiny-YOLOv3 with 66.3%. INRIA is one of the famous pedestrian datasets that had been used by Oltean et al [18] for video surveillance system. YOLOv3 and YOLOv3-Tiny was trained with the dataset and achieved processing speed up to 21 fps and 77 fps respectively.

Unmanned Aerial Vehicle (UAV) is another area that implemented YOLO algorithm as an object detection tool. Luo et al[19] proposed a vehicle detection in UAV images using YOLOv3 algorithm which integrated with K-means++ and Soft-NMS (Non-Max Suppression) and achieved a good result of 97.49% average precision. Meanwhile, Zhang et al [14] proposed YOLO-Lite model with Spatial Pyramid Pooling (SPP) in vision-based detection of power line poles after typhoon striking. The execution through CPU achieved precision of 75.80% with 9 fps.

YOLO also have been implemented by many researchers for application of underwater detection which have more challenging environment especially in murky water and low light surroundings. Paper proposed by Xu et al. [20] utilized YOLOv3 for underwater fish detection for waterpower applications. The datasets used to train and test the model were very challenging with high turbidity, high velocity and murky water as the three datasets were recorded at marine and hydrokinetic energy projects and river hydropower projects. The training and testing of the model shows adequate results for mean average precision (mAP) of 53.92%. Mohamed et al. [21] utilized YOLOv3 for application of fish detection and tracking in fish farms. Pre-processing for the underwater images was executed using Multi-Scale Retinex (MSR) algorithm while optical flow algorithm was used to track fish. The result shows that the model able to track the fish trajectory with the help of YOLO compare to without YOLO.

Literally, most of the application in YOLO models show significant performance in terms of the detection accuracy, mAP and Frame Per Second (FPS). As the evolvement of YOLOv1 until YOLOv3, many improvement have been proposed for variety of object detection. The latest YOLOv3 has been widely used in many application including underwater detection as the approach is significant for low quality images [20], [21].

B. YOLOv3

1) Original YOLOv3: YOLO is a deep neural network algorithm that utilize fully convolutional neural network [9]. Fig.1 roughly shows the process executed in YOLOv3. The algorithm applies a single stage network where it divides an image into grid which later each grid will predicts the bounding boxes, confidences of the boxes and the probabilities for each class. A research proposed by Redmon et al [9] stated that in YOLOv3, a feature extractor called Darknet-53 that is a deeper architecture compare to previous version. Darknet-53, as shown in Fig. 2 contains 53 convolutional layers with stride 2 (down sample the feature maps) and followed by batch normalization layer and Leaky ReLU as activation. As mentioned by [9], YOLOv3 assemble fully convolution layer, so there is no pooling layer is used in the network. After going through feature extractor, the feature will be passed to a classifier section in which the prediction of the detection will be made. The prediction is made by utilizing a 1×1 convolution layer where the size of the prediction map is the same as the size of feature map before. In YOLOv3, each cell can predict 3 bounding boxes in each cell.

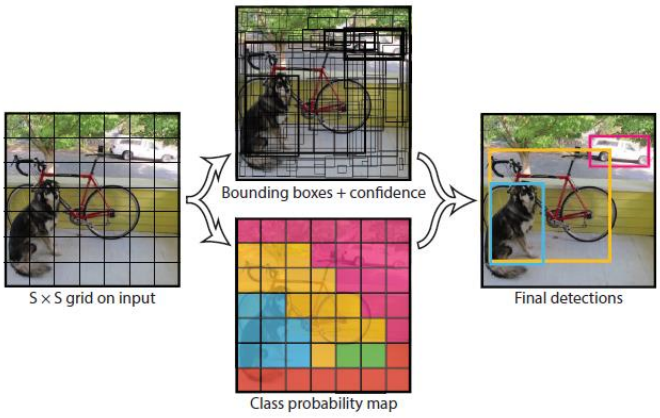


Fig. 1: Detection Process of YOLOv3 [10]

One of the improvements in YOLOv3 compared to previous version is the ability to make prediction across three different scales where on the detection layer, feature maps of different sizes is used to make detection[9]. The input will be down samples by network until first detection layer in which the detection occurred based on the feature maps of a layer with stride 32 (considering input size 416×416). Then, the layer will be unsampled with a factor of 2 and concatenated with feature maps of a previous layer that have same feature map sizes. Another scales detection is done at 16 stride layer. Having the same upsampling procedure and final scale detection is made at stride 8 layer. This improvement help the YOLOv3 model to have the ability to detect small objects [4]. At the output, there is a process to filter the detection since the output detection consist of multiscale detection and many bounding box predictions per cell. There are two process which is the elimination based on objectness score and Non-maximum Suppression (NMS) [22]. The objectness score basically the process to filter the boxes that have scores below the threshold. After applying filtering process based on objectness score, there is still high chance of overlapping boxes. Here comes the second filter which is NMS. When several boxes overlap in detecting the same object, NMS will pick only one box as the best detection [22]. The NMS process started with selecting the box that have the highest score, then

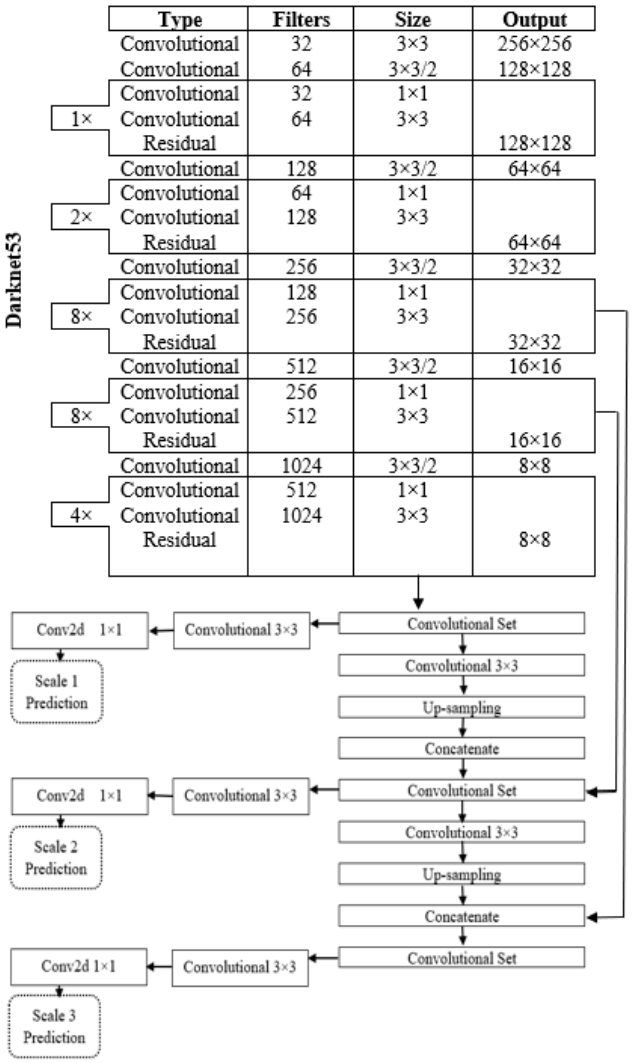


Fig. 2: YOLOv3 Architecture

all the boxes that overlap will be removed if the computation is more than Intersection Over Union (IoU) threshold. The step of NMS will be repeating until no more box left that have lower score than the selected box and finally the YOLO will produce its final output.

2) Tiny-YOLOv3: Tiny-YOLOv3 is the simplified and light version of the original YOLOv3. It operates based on the same principle as original model but with a varied number of parameters in which the depth of convolutional layer is reduced. Tiny-YOLOv3 structure has only seven convolutional layers and small number of 1×1 and 3×3 convolutional layer is used as feature extractor. To reduce the dimensionality size through the network, pooling layer is applied. This network simplification requires this model to occupy less amount of memory and hence, will improve the detection speed [23].

3) YOLOv3-SPP: Spatial pyramid pooling (SPP) is an added layer to the original YOLOv3 model [24]. This layer is placed right after convolutional layer and before the fully connected layer. In YOLO model, the convolutional layer has the capability to process any image without having problem with the image size, but the fully connected layers have the limitation on the image size where it need to be flattened as it only accepts input as 1d vector. This is the reason why the images need to be warped in order for the size to fit the size requirement of the network. SPP layer act to map any input size down to a fixed size output [24]. The feature maps from the last convolutional layer will be divided into a number of different bin scales (Bin scales depends on the image size) and will execute max pooling on each bin which later will be fed to the fully connected layer. The different pooling process with different bin scale are done in parallel.

4) Tiny-YOLOv3-PRN: Normal feature extraction process in deep CNN is passed from one layer to next layer. Partial Residual Network that proposed by [25] applies changes to the normal operation where a single layer can feed the feature into next layer and also directly jump into the 2, 3 or more layers away. Technically, the feature inputs at a lower layer will be available to a node in a higher level [25]. YOLOv3 tiny was modified to add partial residual network (PRN) where it consists of 5 shortcuts from original architecture. The shortcuts layer will add the feature maps from previous layer with the feature map from jump layer. The shortcut layer will be activated using leaky-ReLU type of activation.

C. YOLO Performance Comparison

Generally, most application of YOLO models have been explored on common detection studies instead of underwater object detection. To date, there is none study has been conducted to compare the performance of YOLOv3 models specifically for underwater object detection. The milestone of YOLOv3 in object detection leads to other several improvements and changes in the architecture such as YOLOv3-SPP, Tiny-YOLOv3 and Tiny-YOLOv3-PRN. All these architecture act and perform differently as it was design to fulfill the objectives and hardware used. The major difference we can see is the network's depth. YOLOv3 and YOLOv3-SPP are networks with deeper architecture compare to Tiny-YOLOv3 and Tiny-YOLOv3-PRN. Deeper layer tends to aim for high precision that operates superior in terms of feature extraction while shallow network aims for better performance speed with minimal cost of hardware.

Several papers had presented the comparison result of deeper network versus shallow network [13], [18], [26], [27]. The results indicated that deeper network surpass shallow network in a huge gap of mean average precision (mAP) while in terms of processing speed, lighter network were able to reach high frame per second (FPS). Tumas et al. [13] stated that the layers of the network affected the feature extraction where Tiny model cannot extract more features which lead to lower precision even after same training iteration. Additionally, by modifying and utilizing the network with SPP layer, the models can have better precision due to SPP benefits in allowing for learning multiscale object features more detail [24], [26]. On the other side, modifying the architecture with PRN helps lightweight model to have less parameter and less computation which yield to better performances in terms of frame rate [25]. Literally, it shows that the changes or improvement of YOLOv3 helped to develop more accurate detection model and hence this study is used to compare several YOLOv3 model and hence proposed the best YOLOv3 model for underwater object detection specifically for haze and low-light environment.

III. METHODOLOGY

This section explains the developments of underwater dataset preparation to YOLOv3 architecture based on Deep Convolutional Neural Network (DCNN) detection model. This section also includes the evaluation of all comparison works on YOLOv3 and its respective models' performance. Generally, the overall proposed work in this study is presented as shown in Fig.3.

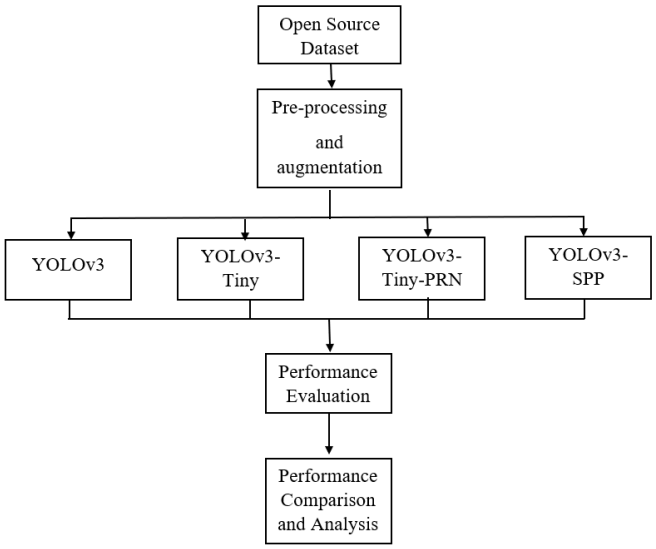


Fig. 3: Different YOLOv3 Architecture Execution

A. Underwater Dataset Acquisition

Training an object detection model in computer vision requires a set of image dataset for training, validating, and testing. Image dataset is a collection of instances of the proposed object to be detected which share the same attributes. In this study, there are two open source underwater datasets that were used to train different type of YOLOv3 architecture.

The first dataset is Google Open Images V6+ Extension dataset [28]. It consists of 600 categories of annotated images. In order to implement the detection for underwater objects, a total of 1420 images of four categories of underwater animals which is squid, skates, jellyfish and shark was downloaded and used to train different YOLOv3 models. Another dataset is taken from The Brackish Dataset [29] that contains six underwater categories such as big fish, jellyfish, crab, shrimp, small fish and starfish. This open source dataset consists of 10,995 annotated files and 14,518 images extracted from recorded videos. Both datasets were separated into 80% for training, 10% for validation and 10% for testing.

Both open source dataset were already annotated by following the YOLO annotation bounding box format. Each image will have its own annotation in .txt file. For YOLO, it has a specific annotation format that consist of 5 components which is object-id, center x, center y, width and height. The objected represent the class number while center x and center y will represent the coordinates of the center points of the bounding boxes. The width and height is the representation of the size of the bounding box.

B. Deep Learning Framework

Neural network framework is used to provide flexible APIs and configuration options for performance optimization where it is designed to facilitate and fasten the training of deep learning models [30]. In this study, the neural network framework that was used is an open source framework called Darknet. Darknet is written in C and CUDA. Using this neural network framework also will allow the execution of the training and detection to be made in Graphical Processing Unit (GPU) which is faster compare using Central Processsing Unit (CPU). All models were executed on a computer with a i7 8th Gen, CUDA based Nvidia GPU GTX1060 Max-Q and a 16GB RAM.

C. YOLO Model Execution

In this research, object detection architecture that based on computer vision was used as a task to recognize an instance of underwater life as an object class and describe the locations of the detected animal in an image using bounding box. Various YOLO architecture such as YOLOv3, Yolov3-Tiny, YOLOv3-Tiny-PRN and YOLOv3-SPP were implemented to compare the performance from two perspective which is the precision of detection and the processing time. The two open source datasets were used to train and validate the different architecture of YOLO models. The real challenges is when one of the downloaded dataset was originally in brackish strait that has degraded visibility. Then, the datasets went through data pre-processing and augmentation. This process was done in configuration file of the YOLO model. Later, all the comparison YOLO models of YOLOv3, Yolov3-Tiny, YOLOv3-Tiny-PRN and YOLOv3-SPP were trained and validated using the open source datasets. Both datasets were separately used to train and validate all the YOLO models. After training, the detection models were tested using test dataset which is not included in training and validation. All the models' performance were evaluated in order to assess their model precision and processing time. Finally, the output from the performance evaluation of all four different YOLO models were compared and analysed to highlight the effect of each detection models.

D. Performance Evaluation

In order to evaluate the performance for each tested YOLO models, the evaluation criterion were measured and calculated based on five common evaluation metrics that are Precision and Recall as shown in Eq. 1 and Eq. 2 respectively. Precision will indicate that out of all predicted instances that belongs to particular class, actually belonged to that particular class. In addition, precision is used to reflect the robustness of detection where high precision returns in truer detected object than false detected in the established model. Meanwhile the recall determines the ability of the model to find all relevant instances in the dataset. High precision indicates a low value of the false positive though generally correlated with a small number of false negatives for recall.

$$\text{Precision} = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Positive\ (FP)} \quad (1)$$

$$\text{Recall} = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Negative\ (FN)} \quad (2)$$

$$\text{F1-score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

$$\text{mAP} = \frac{\sum_{i+1}^{k} API}{k} \quad (4)$$

$$Frame\ Per\ Second, FPS = \frac{Number\ of\ frames}{Total\ Detection\ Time, s} \quad (5)$$

Another evaluation used is F1-measure as shown in Eq. 3 which represents the harmonic mean of precision and recall. Next, the networks were also evaluated with mean average precision (mAP) as denoted in Eq. 4. The mAP for object detection is defined as the average of the AP calculated for all classes involved. Finally, Frame Per Second (FPS) as in Eq.5 is used to express on how fast the model can process the input in one second.

IV. Results and Discussion

In this study, there are four different YOLOv3 architecturebased models were trained and validated and hence finally tested using "never seen dataset" which is the testing dataset. Table I shows the testing result of performance evaluation metrics using both datasets, The Brackish Dataset and Google Open Images Dataset V6+. From the results, it shows that The Brackish Dataset trained model, highest precision result is YOLOv3-SPP while for recall, YOLOv3 original model is the highest and the lowest recall model is YOLOv3-Tiny-PRN.

YOLOv3-Tiny-PRN have high number of false negative that lead to lower performance since recall measures the models performance to return all the positives. As shown in Fig. 5 (a), the model Tiny-YOLOv3-PRN produce false negative result where it detects only one squid and incorrectly rejected another squid. On the other hand, with good performance in F1-Score and mAP, YOLOv3 and YOLOv3-SPP are the most significant model for this dataset. The results also show that both models were able to produce high mAP due to its architecture that has deeper layer and able to extract more features compare to Tiny version. Their robustness indicated that both deeper models have small number of false negative and false positive. Fig. 4 (a) shows that YOLOv3-SPP only detected 2 out of 3 starfish which resulted in false negative while Fig. 5 (b) is the false positive when YOLOv3 incorrectly identified rays and skates as shark.

TABLE I: Performance Evaluation Result

Dataset	Architecture	Precision	Recall	F1-Score	mAP@50 (%)
The Brackish Dataset	YOLOv3	0.96	0.96	0.96	97.56
	YOLOv3-SPP	0.97	0.91	0.94	97.03
	YOLOv3-Tiny	0.80	0.78	0.79	87.18
	YOLOv3-Tiny-PRN	0.81	0.69	0.75	79.16
Google Open Images Dataset V6+	YOLOv3	0.63	0.67	0.65	66.40
	YOLOv3-SPP	0.66	0.73	0.69	74.88
	YOLOv3-Tiny	0.41	0.63	0.50	53.25
	YOLOv3-Tiny-PRN	0.42	0.60	0.50	58.92

TABLE II: Processing Speed Inference

Dataset	Architecture	Frame Per Second(FPS)
The Brackish Dataset	YOLOv3	15.5
	YOLOv3-SPP	15.2
	YOLOv3-Tiny	44.6
	YOLOv3-Tiny-PRN	47.7
Google Open Images Dataset V6+	YOLOv3	17.1
	YOLOv3-SPP	16.9
	YOLOv3-Tiny	99.9
	YOLOv3-Tiny-PRN	102.4

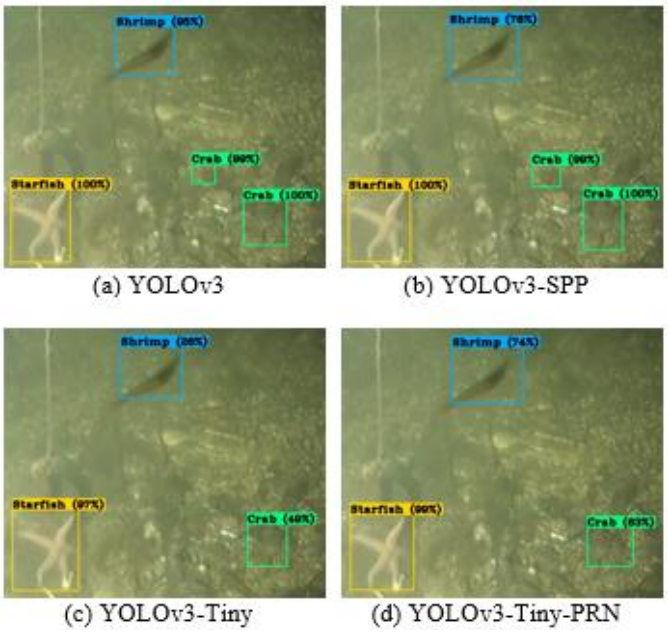
(a) YOLOv3-SPP (False Negative) (b) Tiny-YOLOv3 (False Positive)

Fig. 4: False Negative and False Positive from The Brackish Dataset

(a) Tiny-PRN (False Negative) (b) YOLOv3 (False Positiv

Fig. 5: False Negative and False Positive from Google Open Images V6+ Extension

Next, YOLOv3-SPP proves to be the most significant model for Open Image dataset since having the ability to achieve highest precision and recall. As noted, the high precision represents accuracy of the model's prediction since it returns high value for true positive and lowest return for false positive. Otherwise, both Tiny models produce higher result in recall compare to precision. This indicate that both tiny models has the ability to find the relevant objects in the dataset but poor performance in determining the relevant objects is actually relevant. This can be seen from Fig. 4 (b) as Tiny-YOLOv3 wrongly classified small fish as shrimp which resulted in false positive, hence, reduced the precision. Furthermore, the YOLOv3 with Spatial Pyramid Pooling (SPP) layer gave better detection performance compared to original YOLOv3 for Open Image dataset but almost the same performance (mAP) for The Brackish Dataset. This is due to the different size of the input training images for Open Images dataset compare to The Brackish Dataset that are uniform in size. The benefits of SPP is it can combine different scales features derived to the variable of input scales and hence, improve the execution and backbone of the network [26]. Moreover, the multi-size training and full image representations in SPP network are able to improve the accuracy and decrease overfitting [26], [24]. Table II shows the processing speed performance of every model to correctly classify the input images in test dataset. From the obtained result, the frame per second (FPS) of every model shows that all models are able to be implemented in real-time application. This is the benefits of single stage detector like YOLO where it needs only a single pass through the network and predicts the bounding box of the objects.



(a) YOLOv3 (b) YOLOv3-SPP

(c) YOLOv3-Tiny (d) YOLOv3-Tiny-PRN

Fig. 6: Result from The Brackish dataset trained model

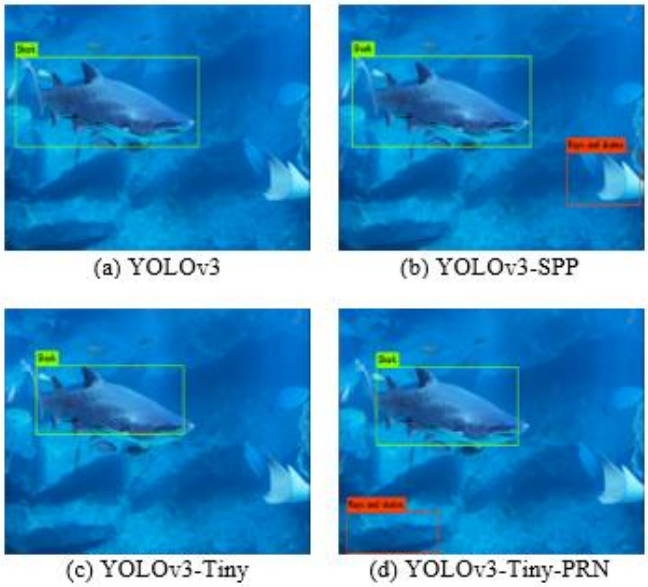


Fig. 7: Result from Google Open Images V6+ Extension trained model

Another thing that can be seen from the result is the difference in FPS performance for YOLOv3-Tiny and YOLOv3-Tiny-PRN as compared to original YOLOv3 and YOLOV3-SPP. Both tiny models were capable to achieve high FPS due to the small size model (shallow network) and hence lead to faster inference speed. Apart from that, the tiny based YOLOv3 architecture with and without PRN also acted differently in terms of FPS. YOLOv3-Tiny-PRN achieved 102.4 FPS compared to YOLOv3-Tiny with 99.9 FPS for Open Image dataset. PRN uses the approach to have a shortcut or skip connection that allow the features to bypass the next layer and jump to the layer in front or after the next layer. This will allow the process and network to have less parameters, less computation and hence increase the speed of execution [25].

In comparison for both datasets, the mAP for The Brackish Dataset trained model were successfully exceed 90% while Open Images Dataset trained model were struggled to achieve even 80%. The reason behind this difference is due to the total number of train images in the dataset. Underwater life in Google Open Images Dataset V6+ was quite lower compared to The Brackish dataset. This proves that for classification and detection in deep learning, the accuracy increases with the increase in training data size [31], [32].

Over and above, YOLOv3 architecture proves the ability to be trained and has good capability in detecting objects in challenging environment such as in The Brackish Dataset. This can be seen in Fig. 6, where all models were showing a good performance in detecting objects in a murky environments which has low visibility. Apart from that, all the models especially the deeper model such as YOLOv3 and YOLOv3-SPP revealed the benefits of the architecture by successfully detecting objects across scales as shown in Fig. 6 and Fig. 7. The ability in detecting objects at longer distance that resulted in small scale objects was a proved to the benefits from the process of up-sample layers that concatenated with the previous layer which helped preserve the fine grained features for small objects [9].

V. Conclusion

This study is proposed to compare different YOLOv3 models by focusing on the architecture configurations for object detection such as original YOLOv3, YOLOv3-SPP, YOLOv3-Tiny, YOLOv3-Tiny-PRN. Two main performance metrics were evaluated that are mAP and FPS to determine the efficiency of the models' ability to detect underwater life. In a nutshell, this study shows a significant evidence that YOLOv3 is applicable to be implemented to detect underwater objects which specific environment such as haze/cloudy and low light conditions. Even though the results shows that different models outperformed another model in terms of their performances, the YOLOv3 model itself own its ability to provide better accuracy and processing time. However, this study is limited to detect and classify between four to six classes of underwater images due to available dataset restriction and machine limitations.

Acknowledgement

The authors wish to thank to members of the Advanced Control System and Computing Research Group (ACSCRG), Advanced Rehabilitation Engineering in Diagnostic and Monitoring Research Group(AREDiM) and Faculty of Electrical Engineering, Universiti Teknologi MARA P. Pinang for providing the assistance and guidance for the field works. The authors are grateful to Research Management Institute (RMI) and Universiti Teknologi MARA (UiTM), Cawangan Pulau Pinang for administrative and financial supports. With special thank as this publication was made possible by grants from the Ministry of Higher Education (MOHE), Malaysia under the FRGS grant of 600-RMI/FRGS 5/3 (291/2019).

References

[1] D. Radovanovic´ and S. Dukanovic, "Image-based plant disease detec- tion: A comparison of deep learning and classical machine learning algorithms," in *2020 24th International Conference on Information Technology (IT)*. IEEE, 2020, pp. 1–4.

[2] J. S. Finizola, J. M. Targino, F. G. S. Teodoro, and C. A. de Moraes Lima, "A comparative study between deep learning and traditional machine learning techniques for facial biometric recognition,"in *Ibero-American Conference on Artificial Intelligence*. Springer, 2018,pp. 217–228.

[3] P. Ding, Y. Zhang, P. Jia, and X.-l. Chang, "A comparison: different dcnn models for intelligent object detection in remote sensing images," *Neural Processing Letters*, vol. 49, no. 3, pp. 1369–1379, 2019.

[4] K. Horak and R. Sablatnig, "Deep learning concepts and datasets for im- age recognition: overview 2019," in *Eleventh International Conference on Digital Image Processing (ICDIP 2019)*, vol. 11179. International Society for Optics and Photonics, 2019, p. 111791S.

[5] R. Girshick, J. Donahue, T. Darrell, J. Malik, and U. C. Berkeley, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[6] R. Girshick, "Fast R-CNN," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 Inter, pp. 1440–1448, 2015.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39,no. 6, pp. 1137–1149, 2017.

[8] A. M. Algorry, A. G. Garc´ıa, and A. G. Wofmann, "Real-time object detection and classification of small and similar figures in image processing," in *2017 International Conference on Computational Scienceand Computational Intelligence (CSCI)*. IEEE, 2017, pp. 516–519.

[9] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[10] B. Benjdira, T. Khursheed, A. Koubaa, A. Ammar, and K. Ouni, "Car detection using unmanned aerial vehicles: Comparison between faster r-cnn and yolov3," in *2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS)*. IEEE, 2019, pp. 1–6.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[12] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[13] P. Tumas, A. Nowosielski, and A. Serackis, "Pedestrian detection in severe weather conditions," *IEEE Access*, vol. 8, pp. 62 775–62 784, 2020.

[14] S. Zhang, B. Chen, R. Wang, J. Wang, L. Zhong, and B. Gao, "Unmanned aerial vehicle (uav) vision-based detection of power line poles by cpu-based deep learning method," in *2019 IEEE 9th Annual International Conference on CYBER Technology in Automation, Control,and Intelligent Systems (CYBER)*. IEEE, 2019, pp. 1630–1634.

[15] Y. Sat, F. Tufan, and A. Muhammed, "CNN based traffic sign recognition for mini autonomous vehicles-annotated.pdf," vol. 1, pp. 85–94, 2019. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-99996-8 16

[16] B. T. Nugraha, S. F. Su, and Fahmizal, "Towards self-driving car using convolutional neural network and road lane detector," *Proceedings ofthe 2nd International Conference on Automation, Cognitive Science, Optics, Micro Electro-Mechanical System, and Information Technology,ICACOMIT 2017*, vol. 2018-January, pp. 65–69, 2017.

[17] A. Ćorović, V. Ilić, S. Durić, M. Marijan, and B. Pavković, "The real-time detection of traffic participants using yolo algorithm," in *2018 26thTelecommunications Forum (TELFOR)*. IEEE, 2018, pp. 1–4.

[18] G. Oltean, L. Ivanciu, and H. Balea, "Pedestrian detection and behaviourcharacterization for video surveillance systems," in *2019 IEEE 25thInternational Symposium for Design and Technology in Electronic Packaging (SIITME)*. IEEE, 2019, pp. 256–259.

[19] X. Luo, X. Tian, H. Zhang, W. Hou, G. Leng, W. Xu, H. Jia, X. He, M. Wang, and J. Zhang, "Fast automatic vehicle detection in uav images using convolutional neural networks," *Remote Sensing*, vol. 12, no. 12, p. 1994, 2020.

[20] W. Xu and S. Matzner, "Underwater fish detection using deep learning for water power applications," in *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 2018, pp. 313–318.

[21] H. E.-D. Mohamed, A. Fadl, O. Anas, Y. Wageeh, N. ElMasry, A. Nabil,and A. Atia, "Msr-yolo: Method to enhance fish detection and tracking in fish farms," *Procedia Computer Science*, vol. 170, pp. 539–546, 2020.

[22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *Proceedings of the IEEE Com- puter Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 779–788, 2016.

[23] V. Molchanov, B. Vishnyakov, Y. Vizilter, O. Vishnyakova, and V. Knyaz, "Pedestrian detection in video surveillance using fully convolutional yolo neural network," in *Automated Visual Inspection and Machine Vision II*, vol. 10334. International Society for Optics and Photonics, 2017, p. 103340Q.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[25] C.-Y. Wang, H.-Y. Mark Liao, P.-Y. Chen, and J.-W. Hsieh, "Enriching variety of layer-wise learning information by gradient combination," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[26] C. Dewi, R.-C. Chen, and S.-K. Tai, "Evaluation of robust spatial pyramid pooling based on convolutional neural network for traffic sign recognition system," *Electronics*, vol. 9, no. 6, p. 889, 2020.

[27] S. Cepni, M. E. Atik, and Z. Duran, "Vehicle detection using different deep learning algorithms from image sequence," *Baltic Journal of Modern Computing*, vol. 8, no. 2, pp. 347–358, 2020.

[28] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, T. Duerig, and V. Ferrari, "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," *IJCV*, 2020.

[29] M. Pedersen, J. B. Haurum, R. Gade, T. B. Moeslund, and N. Madsen, "Detection of Marine Animals in a New Underwater Dataset with Varying Visibility," in *IEEE Conference on Computer Vision and PatternRecognition Workshops*, no. June, 2019, pp. 18–26.

[30] A. Shatnawi, G. Al-Bdour, R. Al-Qurran, and M. Al-Ayyoub, "A comparative study of open source deep learning frameworks," *2018 9th International Conference on Information and Communication Systems, ICICS 2018*, vol. 2018-Janua, pp. 72–77, 2018.

[31] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 843–852.

[32] S. Lei, H. Zhang, K. Wang, and Z. Su, "How training data affect the ac- curacy and robustness of neural networks for image classification," 2019.[Online]. Available: https://openreview.net/forum?id=HklKW