Received: 15 July 2020 | Revised: 2 December 2020 | Accepted: 16 February 2021

IET Image Processing

DOI: 10.1049/ipr2.12181

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH PAPER

An optimized YOLO-based object detection model for crop harvesting system

Mohamad Haniff Junos[1] | **Anis Salwa Mohd Khairuddin**[1] | **Subbiah Thannirmalai**[2] | **Mahidzal Dahari**[1]

[1] Department of Electrical Engineering, Faculty of Engineering, University of Malaya, Kuala Lumpur, Malaysia

[2] Advanced Technologies and Robotics, Sime Darby Technology Centre Sdn Bhd, Selangor, Malaysia

Correspondence
Anis Salwa Mohd Khairuddin, Department of Electrical Engineering, Faculty of Engineering, University of Malaya, Kuala Lumpur, Malaysia
Email: anissalwa@um.edu.my

Funding information
RU Grant-Faculty Programme by Faculty of Engineering, University of Malaya, Grant/Award Number: GPF042A-2019; Industry-Driven Innovation, Grant/Award Number: (IDIG)-PPSI-2020-CLUSTER-SD01

Abstract

The adoption of automated crop harvesting system based on machine vision may improve productivity and optimize the operational cost. The scope of this study is to obtain visual information at the plantation which is crucial in developing an intelligent automated crop harvesting system. This paper aims to develop an automatic detection system with high accuracy performance, low computational cost and lightweight model. Considering the advantages of YOLOv3 tiny, an optimized YOLOv3 tiny network namely YOLO-P is proposed to detect and localize three objects at palm oil plantation which include fresh fruit bunch, grabber and palm tree under various environment conditions. The proposed YOLO-P model incorporated lightweight backbone based on densely connected neural network, multi-scale detection architecture and optimized anchor box size. The experimental results demonstrated that the proposed YOLO-P model achieved good mean average precision and F1 score of 98.68% and 0.97 respectively. Besides, the proposed model performed faster training process and generated lightweight model of 76 MB. The proposed model was also tested to identify fresh fruit bunch of various maturities with accuracy of 98.91%. The comprehensive experimental results show that the proposed YOLO-P model can effectively perform robust and accurate detection at the palm oil plantation.

1 | INTRODUCTION

Currently, Malaysia has around 4.49 million hectares under oil palm cultivation yielding about 17.73 million tonnes and 2.13 tonnes of palm oil and palm oil kernel that jointly contributed to ≈60.6% of global supply [1]. Rapid growth of world population and economic development has increased the demand for palm oil. It is estimated at least 240 million tonnes of palm oil will be required to meet global demand by 2050 [2]. Consequently, the production of palm oil can be increased by improving the harvesting process since it is a very laborious operation. Large human workforce with skill and experience is required to execute the harvesting operation.

Generally, the fresh fruit bunch (FFB) harvesting process includes cutting, collecting and transporting the harvested FFB to the processing site [3]. Over the years, numerous methods have been implemented to optimize FFB harvesting operation. Traditional cutting technique relies on manual tools such as chisel and sickle [4] while mechanized harvesting technique such as E-cutter [5], Cantas7 [6] and mechanical machine [7] were introduced to make the harvesting process more efficient and productive. Currently, mechanical tractor with grabber (MTG) is extensively employed for in-field FFB collection which contributed to efficient evacuation operation [8]. However, the usage of MTG is still limited especially in a big scale oil palm plantation. Therefore, automated harvesting system based on machine vision is a practical solution to reduce high dependency on human workforce, reduce production cost as well as improving crop harvesting productivity [9].

Many researchers have made an attempt to develop robust algorithms for precise crop detection and classification as this is a crucial aspect for an automated harvesting system [9]. In

recent years, the performance of crop detection systems has been improved remarkably, but they are still far from practical application. The obvious challenges in developing such crop detection system are the uncertain and complex environment of the orchards which include naturally inconsistent illumination, indistinguishable backgrounds, highly overlap and occlusion condition and low resolutions. Over the years, machine learning techniques have been adopted in various crop detection systems. Support vector machine (SVM) classifier has been adopted for fruit recognition [10] and detection of citrus fruits [11]. Besides, colour and texture analysis method was implemented with SVM classifier in several studies such as in green fruit counting [12], mangoes counting [13], fruit counting on coffee branches [14] and disease detection and classification for soybean [15]. On the other hand, colour, shape and texture information were used to extract the feature of peach and classified using artificial neural network (ANN) [16]. Meanwhile, classifier and regression trees (CART) was applied for tomato detection [17]. In another study, fuzzy classification method based on colour features was adopted to estimate apple ripeness [18]. The above-mentioned researches adopted hand-crafted features approach and machine learning classifier for crop detection. However, the problem of detection in highly challenging conditions remains unsolved. Thus, state of the art deep learning technique is employed in an effort to improve the accuracy of detection system for precision agriculture.

With the development of deep learning technology in machine vision applications, deep convolutional neural networks (CNNs) have achieved the state of the art results for object detection in term of accuracy and detection speed [19]. The main advantage of CNN is its ability to automatically extract features from the input image through self-learning. In a study conducted by [20], blob detectors based on fully connected CNNs was used for extraction of candidate regions, segmentation of object areas and calculate the number of fruits utilizing a subsequent CNN counting algorithm. Ref. [21] employed a fully connected CNN for automatic weed detection using images with the condition where many of the leaves were blocked. SVM and CNN method was proposed by [22] to automatically extract the features of apple blossoms in a complicated natural background. Image segmentation methods based on deep learning have shown excellent results in segmentation of crop area which provide significant contributions for crop detection and localization. Nonetheless, these methods unable to precisely segment the regions of each target in heavily overlapped condition.

In general, there are two types of object detection methods based on deep learning which include candidate region-based model and regression-based model. Two stage detection model is based on region proposals, where it is generated in the first stage. Later, features are extracted from these proposals for bounding box and classification regression [23]. Faster R-CNN is the state of the art for two stage object detection method and is extensively used for crop detection based on image analysis. Ref. [24] proposed the DeepFruits approach which explored the use of the Faster R-CNN network to accurately detect capsicum and rockmelon where impressive result was obtained. A similar method was implemented for detection of three different types of fruits; mango, apple and almond [25]. The adoption of transfer learning and data augmentation has increased the detection accuracy on apple and mango. In other study, Faster R-CNN was combined with a novel multi-sensor framework and a multiple viewpoint approach where good result was achieved for yield estimation of mango [26]. However, despite of high localization and recognition accuracy, two stage detection method suffer from slow detection speed and not applicable for real time application.

On the contrary, single stage detectors address object detection as a simple regression problem that takes the entire image as input and simultaneously generates class probabilities and multiple bounding boxes [27]. This has made the model much faster than the two stage object detectors. Various applications on enhanced version of you only look once (YOLO) model have been proposed such as YOLO-L for vehicle license plate detection [28], YOLOv3-MobileNet for detection of electronic component [29], TF-YOLO for detection of multiple object from aerial images [30], YOLO-CA for car accident detection [31] and YOLO-UA for traffic flow monitoring [32]. These proposed techniques modified the network model to solve object detection problems in respective applications. Recently, efforts have been devoted to implement YOLO based model for crop detection. A novel YOLO architecture was developed based on the feature of YOLOv3 and YOLOv2 tiny for detection of mangoes with high accuracy and speed [33]. Besides, densely connected neural network (DenseNet) was utilized to enhance the performance of the state of art YOLOv3 model for apple detection at different growth stages [34]. The proposed YOLOv3-dense model integrated original Darknet53 backbone used in YOLOv3 model with shallow DenseNet architecture that contains two dense blocks with eight dense layers in the feature extractor. The DenseNet architecture was adopted to solve problems related to the detection of one class objects at different growth stage. Hence, the developed model might not be feasible for multi-class problems. The modified YOLOv3 method proposed in [33] and [34] has significantly increased the detection performance, however longer computation time is required to train the model due to the network complexity. Besides, large model size is generated which is not applicable for embedded system.

On the other hand, lighter version of YOLOv3 known as YOLOv3 tiny can basically satisfy real time object detection based on constrained environments in which the memory, storage and processing power are limited [35]. The model can be trained faster leading to low computation cost. These advantages are crucial for the development of automatic crops detection system based on machine vision.

Considering the advantages of YOLOv3 tiny, a modified YOLOv3 tiny network namely YOLO-P is proposed to solve object detection problem focusing on oil palm plantation for precision agriculture. This paper aims to develop an automatic detection system with high accuracy performance, low computational cost and lightweight model size. The contributions of this work are twofold. First, a modified YOLOv3 tiny model namely YOLO-P is developed by replacing the

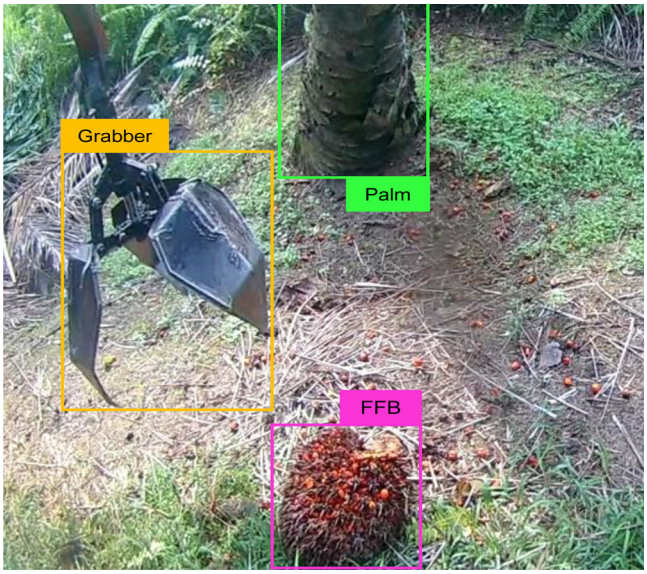

FIGURE 1 Example of image in palm dataset

original backbone with deeper DenseNet architecture. DenseNet incorporated feature reuse architecture that helps to solve information loss and thus improve the classification accuracy. This feature extractor is connected to multi scale detection where four detection scales are utilized in the network to improve detection on small object. Moreover, K-means clustering technique is utilized to determine appropriate anchor box size corresponding to the palm dataset. Meanwhile, the second contribution of this work is the employment of the proposed YOLO-P model to identify FFB of different maturity.

The remainder of the paper is organized as follows: Section 2 introduces the palm oil dataset with detail explanation on data acquisition and preparation. Section 3 describes the proposed YOLO-P model which employs DenseNet201 architecture as feature extractor, multi-scale detection and clustering of anchor boxes. The experimental results are presented and discussed in Section 4. Finally, the conclusion and future works are explained in Section 5.

2 | IMAGE DATABASE

2.1 | Image acquisition

The image data were collected by using 12 MP EKEN H9R Ultra HD camera during harvesting season in Sime Darby oil palm plantation located in Selangor, Malaysia. The collection periods included 9 AM, 1 PM and 5 PM and during sunny and cloudy weather conditions. The camera was attached at the left and right side of the mechanical tractor that is used to collect the FFBs. Initially, videos of the harvesting process are recorded as the tractor moves. The camera is pointed at one specific angle in order to capture the existence of three different classes of objects which are FFB, grabber and palm tree. Later, the recorded videos are converted to data images. Figure 1 shows the example of image in palm dataset.

In total, 5000 images were selected to develop palm dataset. In order to ensure the richness and robustness of the dataset, different scenes and complex conditions were chosen which include different illumination, single and multiple objects, occluded and overlapped environment and background images that contain no object.

2.2 | Data preparation

The training and validation set images were rescaled to 1000 × 750 pixels for annotation process. The images were manually annotated where bounding boxes were drawn, and the categories were classified for each object in the images. An open source annotation tool called LableImg was used for the process. The images were annotated into three classes namely bunch, grabber, and palm. In each image, every visible object was labelled with a bounding box representing the location of the object. Images that contain inadequate or unclear pixel area were not labelled. For occlusion condition, an object with occlusion area greater than 90% and the object at the edge of the image with less than 10% area were not labelled.

The palm dataset was split into three different groups which are training, validation and testing dataset. For training process, the 5000 images were divided into 70% as train data and 30% as validation data. Additional 350 images were used for testing purpose.

3 | METHODOLOGIES

YOLO is a single stage detection method proposed by Redmon, J. et.al [36]. It tackles the object detection problem into a single regression problem in which region detection and classification occur simultaneously in its network. Currently, YOLOv3 is the state of art for single stage object detection [37]. YOLOv3 can basically achieve its real-time performance on a standard computer with graphics processing unit (GPU). However, in the small-scale embedded devices, the algorithm operates slowly. On the contrary, YOLOv3 tiny network has the ability to satisfy real-time requirements in limited hardware resource. YOLOv3 tiny is a simplified version of YOLOv3 with relatively lighter model size. Besides, YOLOv3 tiny implements only two scale of detection on feature maps of two different sizes at two distinct places in the network. Previous works show that YOLOv3 tiny can significantly improves the detection speed due to the extremely less convolution layers and fewer filters in the layers [30, 38, 39]. However, the shallow architecture leads to reduction in its detection accuracy.

3.1 | The proposed novel YOLO-P network model

An optimized YOLO model based on YOLOv3 tiny network is proposed with the aim to enhance the detection performance

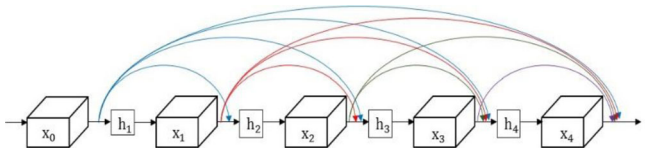

FIGURE 2 DenseNet architecture

while retaining the detection speed in order to satisfy the real-time performance of the system. In the proposed model, the network structure and depth of YOLOv3 tiny model is modified to form a novel model named YOLO-P where P is the abbreviation for palm. The proposed method is composed of the following components: (i) densely connected neural network (DenseNet), (ii) multi-scale target detection, and (iii) anchor box optimization by using K-means clustering algorithm.

3.1.1 | Densely connected neural network as feature extractor

In a deeper convolutional neural network, the information transmission from input layer to output layer becomes big causing the feature information to vanish. To alleviate this problem, DenseNet was proposed to ensure maximum and strong gradient flow by simply connecting each layer directly to every other layer in a feed-forward architecture [40]. As a result, layer, l obtains all the feature maps from the previous layers as shown in Figure 2.

$$x_l = h_l[x_0, x_1, \ldots, x_{l-1}] \tag{1}$$

In Equation (1), $[x_0, x_1, \ldots, x_{l-1}]$ represents the concatenation of the feature maps generated in layer $[0, \ldots, l-1]$. h_l refers to the function that process the spliced feature maps where it provides non-linear transformation of $x_0, x_1, \ldots, x_{l-1}$ layers. If h_l produce k number of feature maps and it can be generalized as follow:

$$k_l = k_0 + k \tag{2}$$

k refers to the growth rate that controls the amount of information added to the next layer.

In general, DenseNet utilizes the potential of the network through feature reuse, thus tends to have more diversified features and richer patterns. DenseNet has smaller parameter and better computational efficiency than ResNet. Considering the superior advantages of DenseNet network, DenseNet architecture with 201 configurations is used as the feature extractor for the proposed YOLO-P model. The modified network architecture of YOLO-P model is shown in Figure 3.

DenseNets are divided into dense blocks, transition block and dense layer. In the dense block, D_n, the dimensions of the feature maps within a block remains constant, however the number of filters increases between them and therefore increases the volume of the network. From Figure 3, the number on top of each dense block illustrates the feature maps dimension. Since feature maps were concatenated, the dimension of this channel expands at every layer. It can be noted that at every new volume, the first addition number for calculation of the feature maps matches the dimension of the feature maps from previous volume. This shows that new information is concatenated to the previous volume which explains the concept of feature reused. In addition, growth rate of 32 was used in the network. The final size of feature maps in each dense block is determined by using Equation (3).

$$k_l = k_0 + k \times l_{\text{th}} \tag{3}$$

l_{th} is the number of dense layer in that particular dense block.

Furthermore, transition block, T_n, is the layer between the dense blocks. The transition block performed 1×1 convolutional and 2×2 average pooling with a stride of two. This resulted in reduction of the number of feature maps and the size of volume by half. In the dense layer, h_n, BN-ReLU-Con (1×1) with 128 filters and BN-ReLU-Conv (3×3) with 32 filters is applied to decrease the feature map size. This layer is the combination of convolution batch normalization and rectified linear unit activation function. The input volume of these operations was then concatenated to add new information in the network. The same network procedures were applied in each of the dense blocks. The proposed DenseNet architecture incorporated four dense blocks that contain 6, 12, 48 and 32 dense layers.

3.1.2 | Multi-scale target detection

To further enhance the performance of the proposed YOLO-P network especially in detecting small objects, multi scale prediction based on Feature Pyramid Network (FPN) was adopted. Additional prediction scales were added to the network to obtain the location information of small target with fine-grained features. Therefore, prediction can be made on the finest level. From the DenseNet201 network, the generated feature maps are transferred to the FPN.

The anchor boxes on early stage feature maps contain a smaller receptive field for object detection at a smaller scale whereas the anchor boxes on later stage feature maps contain a larger receptive field for object detection at a larger scale. The convolution layers at early stage of neural network contain weak object information that is only made up of high-level features from the input image. Thus, features from different layers of early and later stage convolution network are combined. It connects the multiple feature maps with the same feature scale. This will significantly enhance the ability to detect objects at different sizes. Besides, the dimension of feature maps from layers at multiple stages are different, therefore, an up-sampling operation is applied to combine them effectively. In the network, the feature in each scale is up-sampled by two times. As a result, the deep features and characteristics of the hidden layer can be extracted by the full connection layer. Later, the combined feature maps are subjected to 3×3 and 1×1 convolutional layer for the

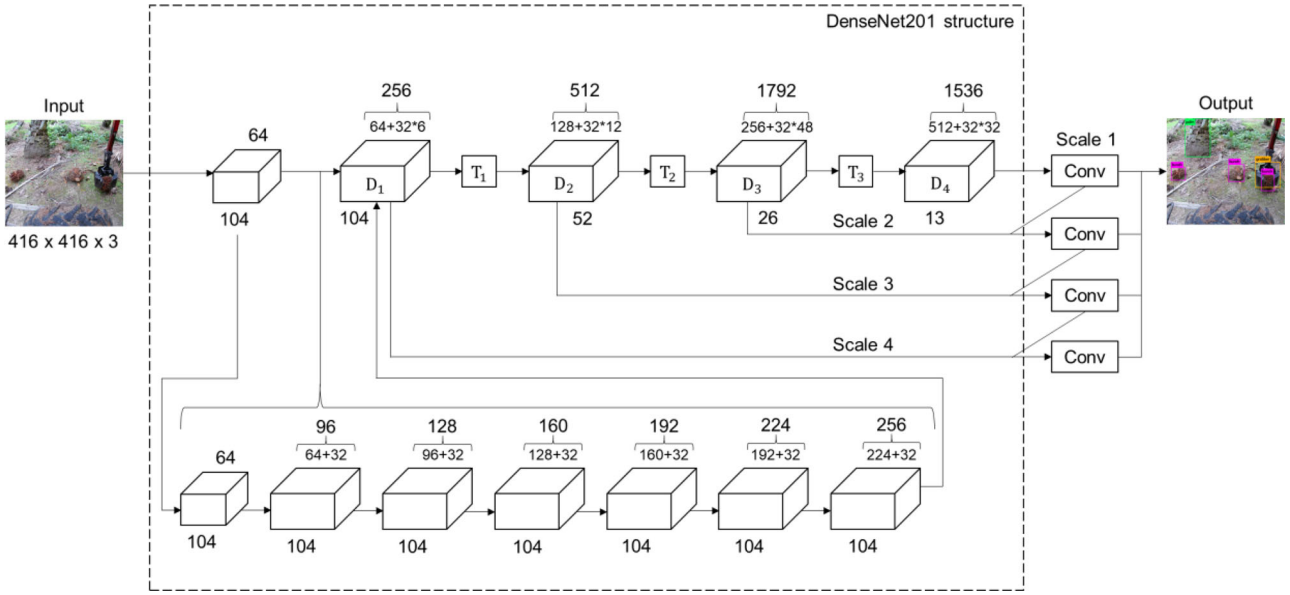

FIGURE 3 YOLO-P with DenseNet201 structure diagram

purpose of fusing the features from the earlier stage layer. A batch normalization layers was applied to obtain the final feature map. The process of up-sampled layers that were concatenated from previous layers helps to conserve the fine-grained features that are greatly important in object detection. These procedures were applied on FPN before each prediction layers.

In the proposed YOLO-P network, four scales are utilized for target detection in multiple sizes. Larger scale detection was performed in 13 × 13 size map while medium, small and smallest scale target was detected in 26 × 26, 52 × 52 and 104 × 104 size map. The detail network parameter of the YOLO-P model is shown in Figure 4.

3.1.3 | Optimization of anchor box

Anchor boxes are a set of initial candidate boxes with a fixed height and width. The fixed parameters provided by the anchors may not be suitable for specific dataset. Consequently, the initialization of these anchor boxes will influence the detection model performance in term of accuracy and speed. Thus, it is crucial to assign appropriate anchor boxes suitable for the palm dataset. Instead of manually mapping the coordinates of the anchor boxes, YOLOv3 implemented K-means clustering algorithm on the dataset to calculate the optimal size and number of anchor boxes automatically. The clusters developed by K-means indicate the distribution of the samples in the dataset, which help the network to learn easily and achieve better predictions. Average IoU was adopted as the objective function to analyse the clustered box. Average IoU is defined in Equation (4).

$$Avg\ IoU = \frac{\sum_{i=1}^{k} \sum_{j=1}^{n_k} IoU\ (box,\ centroid)}{n} \quad (4)$$

	Type	Filters	Size/stride	Output
	Convolutional	64	7x7/2	208x208x64
	Max pooling		2x2/2	104x104x64
x6	Convolutional	128	1x1/1	
	Convolutional	32	3x3/1	104x104x256
	Convolutional	128	1x1/1	104x104x128
	Max pooling		2x2/2	52x52x128
x12	Convolutional	128	1x1/1	
	Convolutional	32	3x3/1	52x52x512
	Convolutional	256	1x1/1	52x52x256
	Max pooling		2x2/2	26x26x256
x48	Convolutional	128	1x1/1	
	Convolutional	32	3x3/1	26x26x1792
	Convolutional	512	1x1/1	26x26x512
	Max pooling		2x2/2	13x13x512
x32	Convolutional	128	1x1/1	
	Convolutional	32	3x3/1	13x13x1536
	Convolutional	1000	1x1/1	13x13x1000
	Convolutional	256	1x1/1	13x13 x256
	Convolutional	512	3x3/1	13x13x512
	Convolutional	24	1x1/1	13x13x24
	YOLO			
	Route			13x13x256
	Convolutional	128	1x1/1	13x13x128
	Up-sampling		2x	26x26x128
	Route			26x26x384
	Convolutional	256	3x3/1	26x26x256
	Convolutional	24	1x1/1	26x2624
	YOLO			
	Route			26x26x384
	Convolutional	64	1x1/1	26x26x64
	Up-sampling		2x	52x52x64
	Route			52x52x192
	Convolutional	128	3x3/1	52x52x128
	Convolutional	24	1x1/1	52x52x24
	YOLO			
	Route			52x52x192
	Convolutional	32	1x1/1	52x52x32
	Up-sampling		2x	104x104x32
	Route			104x104x160
	Convolutional	64	3x3/1	104x104x64
	Convolutional	24	1x1/1	104x104x24
	YOLO			

Scale 1 Scale 2 Scale 3 Scale 4

FIGURE 4 Network structure of YOLO-P

TABLE 1 Distribution of anchor boxes at four prediction scales

Scale	Dimension	Clusters
1	104 × 104	(21, 42), (33, 54), (43, 63)
2	52 × 52	(33, 85), (50, 90), (35, 140)
3	26 × 26	(52, 136), (66, 111), (49, 200)
4	13 × 13	(67, 167), (90, 136), (93, 245)

Box refers to the ground truth of the target and centroid refers to the centre of the cluster. k represents the total number of samples while n_k denotes the numbers of samples in the kth cluster center. n is referred to the numbers of clusters. IoU is the intersection over union of the clusters and the sample. The IoU ratio is defined in Equation (5).

$$IoU\ (box,\ centroid) = \frac{B_{gt} \cap B_c}{B_{gt} \cup B_c} \tag{5}$$

B_{gt} represents the ground truth bounding box and B_c represents the clustered bounding box.

As proposed in earlier section, additional prediction scales were added into the YOLO-P algorithm. Bigger number of prior boxes leads to substantial overlap between bounding boxes and anchor boxes, but the increase in number of anchor boxes will linearly increase the number of convolution filters in prediction filters. This will produce a larger network size with increased in training time. Therefore, considering the number of detection layers and average IoU, twelve anchor boxes were assigned which consist of three sizes of boxes in each prediction layer. The width and height of each anchor boxes allocated for each of the scale prediction is shown in Table 1.

4 | EXPERIMENTS AND DISCUSSION

In this study, several experiments were carried out to validate the reliability of the proposed YOLO-P network to detect three classes of objects namely, FFB, grabber and palm tree at oil palm plantation. Firstly, the evaluation metrics are discussed. Then, the experimental results are compared with other detection models. Finally, analysis of influencing factors on the proposed model is demonstrated. The experiments were conducted through Windows 10 64-bits operating system, equipped with Intel core i7-4790 CPU @ 3.6G Hz processor with installed memory of 16GB RAM and NVIDIA GeForce GTX 750 Ti, graphic card having 2GB of GDDR5 memory type. On top of that, Google's Colab with Tesla K80 GPU accelerator was used for training, validation and testing purposes.

4.1 | Evaluation metrics

The performance of the proposed YOLO-P network was evaluated using several metrics. Average precision (AP), average IoU, F1-score and detection time were applied to demonstrate the detection performance of the network. AP describes the area under the precision-recall curve at different detection threshold whereas mean average precision (mAP) determines the mean accuracy for n class of object. The equation of AP and mAP is shown in Equations (6) and (7) respectively.

$$AP = \int_0^1 Precision\,(recall)\ \ dRecall \tag{6}$$

$$mAP = \frac{\sum_{i=1}^{n} AP_i}{n} \tag{7}$$

The precision and recall are described in Equations (8) and (9). For binary classification problems, four types of indicator are considered; true positive (TP), false positive (FP), false negative (FN) and true negative (TN), which are based on the combinations of the predicted and true class of the learner. TP denotes the number of successfully detected target by the detection model. FN refers to the number of undetected target while FP indicates the number of falsely detected target. F1-score is defined as the harmonic mean that considers both precision and recall. The description of the F1 score is shown in Equation (10). IoU is used to examine the overlapping area between predicted bounding box by the proposed network and ground truth bounding box while average IoU is the mean value of IoU measured over the number of data images. The equation of average IoU is describe in Equation (5). Besides, the detection time is measured in order to evaluate the time required for the model to perform detection per image.

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

$$F_1 = \frac{2 \times P \times R}{P + R} \tag{10}$$

In addition, other metrics including floating point operations (FLOPS), training run time and trained model size are investigated to study the computation performance of the model. FLOPS is one of the significant ways to estimate the amount of calculation in CNN model which consider the amount of calculation in convolution layer and fully connected layer [41]. Training run time is the time required to complete network training at specific training steps while model size is a storage space measured in megabyte which refers to the size of the trained model.

4.2 | Experimental results

The input images in the network were set to 416 × 416 pixels. The batch size and subdivision of 64 and 32 was used due to limitation of GPU memory. The models were trained for 20 000 training steps in order to obtain converged result with the lowest

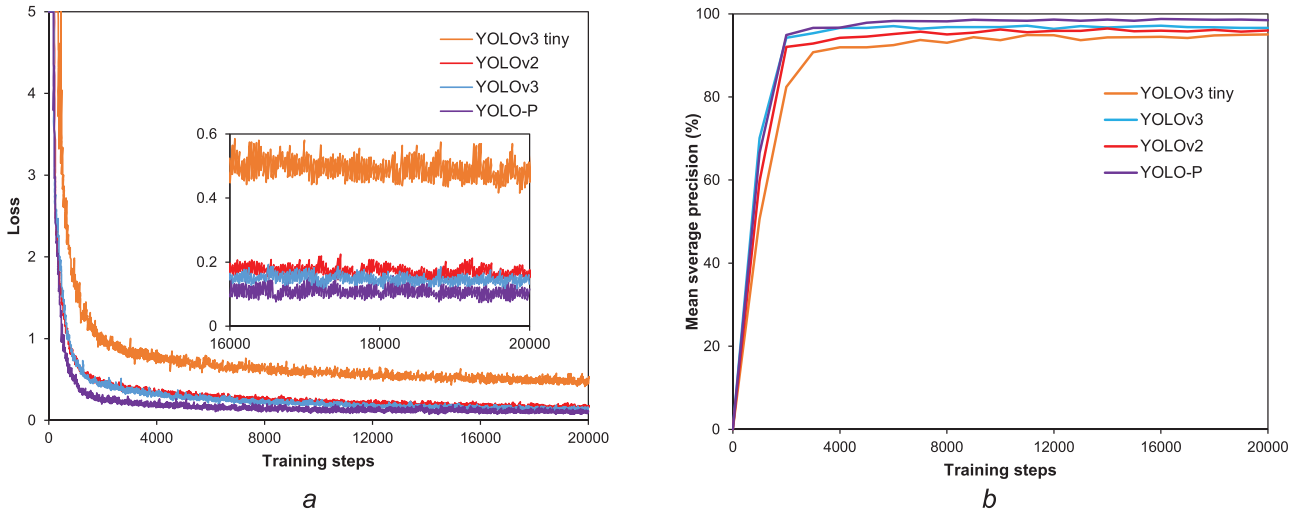

FIGURE 5 Validation of training performance. (a) Comparison of loss curve. (b) Comparison of mAP curve

TABLE 2 Detection performance of several detection models

Method	Average IoU (%)	Precision (%)	Recall (%)	F1-score	AP (%)			mAP_{50} (%)	mAP_{75} (%)	$mAP_{50:95}$ (%)
					Bunch	Grabber	Palm			
YOLOv2	67.77	84.80	96.12	0.90	94.69	97.85	96.93	96.49	70.19	58.97
YOLOv3 tiny	69.89	90.55	90.93	0.90	93.30	97.42	94.01	94.91	65.27	57.34
YOLOv3-dense [34]	75.88	91.91	96.32	0.94	96.92	98.17	96.51	97.20	78.40	64.52
YOLOv3	75.48	93.00	96.54	0.94	96.29	97.56	97.63	97.17	79.77	66.04
YOLO-P	83.34	96.60	98.12	0.97	98.57	98.97	98.51	98.68	90.78	73.25

average loss. To further optimize the training parameters, the initial learning rate was set to 0.001while the momentum was set to 0.9. In addition, weight decay of 0.0005 was used to avoid model overfitting.

4.2.1 | Performance comparison with different algorithms

In this section, the proposed YOLO-P model is compared with several YOLO based detection models to demonstrate the effectiveness of the model in term of detection accuracy as well as computation and detection speed. The loss function during training process of the four YOLO models is shown in Figure 5(a). Based on the results, the YOLO-P, YOLOv3 and YOLOv2 converged faster and produced smaller loss than YOLOv3 tiny. These three models started to converge at around 2000 training steps while YOLOv3 tiny at around 3000 training steps. The final average loss of YOLOv3, YOLOv2 and YOLOv3 tiny were approximately 0.139, 0.172 and 0.443, respectively. YOLO-P showed significant improvement where the average loss was ≈0.104 which is a reduction of 0.339 from the original YOLOv3 tiny. Moreover, the training model is further validated to determine its mAP. The validation of the training model is shown in Figure 5(b). Results show that the mAP obtained for each algorithm correspond to the loss function where YOLO-P achieved the highest mAP and closely followed by YOLOv3, YOLOv2 and YOLOv3 tiny relatively.

Comparison of detection performance for several YOLO based models are shown in Table 2. The evaluation metrics are measured at default value of IoU at 0.5. YOLOv3-dense model which was proposed in [34] is evaluated by using the palm dataset. It can be observed that the proposed YOLO-P has the highest accuracy performance over the other models. YOLO-P achieved mAP of 98.68% which is higher than the state-of-art YOLOv3 (97.17%), YOLOv3-dense (97.20%), YOLOv2 (96.49%) and YOLOv3 tiny (94.91%). This exhibits the effectiveness of the feature reuse in obtaining richer information from every single layer of the network. The obvious advantage of the proposed YOLO-P model is the adoption of four-scales detection that aid the extraction of useful multiscale deep features through various receptive field sizes. On the other hand, YOLOv3 and YOLOv3-dense incorporated three-scales detection layer, hence there were slight decreased in mAP values. YOLOv3 tiny adopted two-scales detection layer while YOLOv2 does not incorporated multiscale detection layer

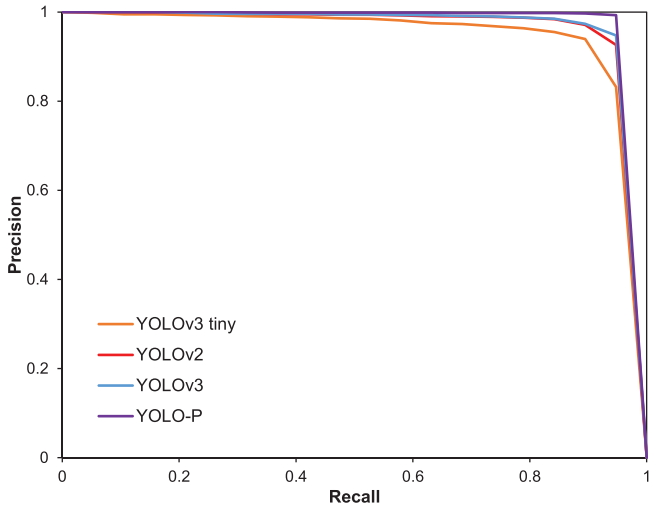

FIGURE 6 Comparison of overall precision-recall curve

which resulted in low mAP value due to the problem of small object detection. As IoU is increased to above 0.75, the percentage of mAP value for YOLO-P is reduced by 8%. This is the lowest percentage of reduction as compared to the other models; YOLOv3 (17.9%), YOLOv3-dense (19.3%), YOLOv3 tiny (31.2%) and YOLOv2 (27.3%). In the case of mAP at average IoU from 0.5 to 0.95 with 0.05 interval, YOLO-P outperformed the

other models with 73.25%. These results imply that the proposed model can produce decent boxes for objects at a higher IoU threshold. As the IoU threshold increased, the performance of the other YOLO models dropped significantly which shows the difficulty to perfectly align the bounding boxes with the ground truth.

In term of average IoU, the proposed YOLO-P achieved the highest percentage of 83.34%. This indicates that the bounding boxes created by the detection model able to overlap with the ground truth at a higher percentage. In addition, YOLO-P achieved F1-score of 0.97 which is the highest among the rest of the models. This demonstrates that the overall precision and recall performance of the proposed model is better than the other models. YOLO-P has achieved the highest precision and recall value than the other four models indicating that the number of undetected and falsely detected objects is minimum. YOLOv3 and YOLOv3-dense model obtained considerably good recall value, nonetheless, the precision value is low due to higher number of falsely detected objects. Figure 6 demonstrates the precision-recall curves of the detection models. The average precision of each class; bunch, grabber and palm are further analysed in order to compare the performance of the models in more detail. It can be found that YOLO-P achieved better average precision for all classes compared to other models.

The performance of these models is also evaluated in terms of time required to accomplish training, the number of billion floating point operations (BFLOPS) and the memory size of the trained networks. The comparative results of the computational performance are presented in Table 3. It is notable that the proposed model achieved better computational performance than state-of-art YOLOv3 and YOLOv3-dense model proposed in [34]. The proposed model generated about half number of BFLOPs as compared to YOLOv3 and YOLOv3-dense [34] models during training process. This index indicates how well the proposed model operates in computability limited device. BFLOPS describes the number of operations required to run a model that affects the time it takes to train a model on a given hardware. It is worth to note that the proposed model was trained ≈1.57 and ≈1.53 times faster than YOLOv3 and YOLOv3-dense. In addition, the proposed YOLO-P model size is remarkably small which requires storage space of only 76 MB. The YOLO-P model size is smaller by 67.7%, 66.7% and 60.6% compared to YOLOv3 (235 MB), YOLOv3-dense (228 MB) and YOLOv2 (193 MB), respectively. DenseNet structure has the obvious advantage of reducing the number of parameters in the network which resulted in smaller trained model. The adoption of DenseNet201 in YOLO-P as feature extractor and addition of two detection layer has greatly increased the number of layers in the network resulted in increasing number of BFLOPS by approximately six times from the original YOLOv3 tiny. However, the trained model size of YOLO-P increased by only 2.3 times compared to YOLOv3 tiny model. On the other hand, the adoption of DenseNet structure in [34] has resulted to a significantly higher BFLOPS (64.35) and model size (228 MB). This shows that the adoption of DenseNet in YOLO-P model is more effective as compared to the YOLOv3-dense in [34]. As reported in Table 3, YOLOv2 runtime is slightly faster compared to the proposed YOLO-P model because YOLOv2 utilized 24 convolutional layers in the network. Nevertheless, the use of Darknet53 and Darknet19 as feature extractor in YOLOv3 and YOLOv2 has generated higher number of parameters in the networks, hence producing a bigger model size. YOLOv3-dense as proposed in [34] integrated two dense blocks with four dense layers each while retaining the original Darknet53 which resulted in slight decreased in model size as compared to YOLOv3. Besides, YOLOv3 tiny incorporated only 13 convolutional layers in the network, thereby it requires the shortest time for training and generated the smallest model size among the others. Nonetheless, the mAP of this model is low especially at higher IoU threshold as presented in Table 2. Overall, the obtained model size and computing time for YOLO-P demonstrated significant improvement over the state of art

TABLE 3 Comparison of computation performance for various detection models

Method	BFLOPS	Runtime (h)	Model Size (MB)
YOLOv2	29.342	17.5	193
YOLOv3 tiny	5.451	8.5	33
YOLOv3-dense [34]	64.352	36	228
YOLOv3	65.304	37	235
YOLO-P	33.662	23.5	76

a b

FIGURE 7 Visual detection results for the proposed YOLO-P model

TABLE 4 Comparison of mAP and detection speed with other methods

Methods	mAP (%)	Speed (ms/img)
Faster R-CNN ResNet101	94.90	202.5
SSD-MobileNet	74.10	25.0
YOLOv2	96.49	23.6
YOLOv3 tiny	94.80	22.2
YOLOv3-dense [34]	96.82	29.3
YOLOv3	97.17	28.5
YOLO-P	98.68	29.7

YOLOv3 model thus making it very practical for the low-end applications in small or mobile devices with limited storage memory devices.

In term of average detection time, the YOLO models are compared with other detection methods namely, single shot detector (SSD) [42] with MobileNet and the state-of-art for two stage detection method, Faster R-CNN with ResNet101 as shown in Table 4. The detection time of YOLO-P was comparable to YOLOv3 and YOLOv3-dense although it is 1.2 and 0.4 ms slower due to the higher number of layers in the algorithm. This is considered as sufficient for real time application. On the other hand, YOLOv3 tiny method achieved the fastest detection speed followed by YOLOv2 model. This result is obtained as both models utilized a shallow structure. Conversely, the obtained mAP is less accurate as compared to the other models. It can be inferred that there is a trade-off between mAP and detection speed for YOLO based detection models. Table 4 shows the obvious advantages of one stage detection methods (SSD and YOLO models) where the detection speed was faster than faster R- CNN method. Faster R-CNN detects object in an image in two separate tasks namely generation of region proposal and

region classification. The computations are carried out repeatedly per region, resulting in an increase in the computational load with the number of regions proposed by the region proposal network. Hence, longer time is required for detection per image which is not suitable for real time application. The detection speed of SSD-MobileNet was faster than YOLOv3 and YOLO-P. However, the mAP is the lowest among the other methods due to limitations in detecting small object. Therefore, Faster R-CNN with ResNet101 and SSD-MobileNet model is not suitable for FFB detection in harvesting application.

A total of 350 test images were used to validate the effectiveness of the proposed model. In total, the test images consist of 967 ground truth objects where the number of objects according to their class is: 446 of bunches, 160 of grabbers and 362 of palms, respectively. From the results, it can be observed that YOLO-P have the highest precision, recall, and F1-score compared to the other models. This is due to the lower number of false positive and false negative detected by the model. The detection results on test images are shown in Table 5.

The comparison of visual detection in several scenarios is highlighted in Figures 7–10. In Figure 7 it can be observed that all of the visible objects were successfully detected by YOLO-P. Besides, YOLO-P was able to detect occluded objects better than other detection models as shown in Figure 7(b). For YOLOv3 model, almost all of the visible objects were detected. However, one of the occluded bunches in Figure 8(b) was not detected. In contrast, YOLOv2 produced more falsely detected bounding box leading to low precision value as reported in Table 5. In Figure 9(a), extra bunch was falsely detected. YOLOv3 tiny achieved low recall value model due to more objects were misidentified as shown in Figure 10(a) where one of the bunches was misidentified due to different illumination condition. In Figures 9(b) and 10(b), both YOLOv2 and YOLOv3 tiny models were not able to detect the occluded bunch. Overall, the proposed model achieved good performance in term of detection accuracy as well as computational and detection speed. Therefore, the reliability and robustness of the proposed model is further analysed in the following experiments.

4.3 | Analysis of influencing factors

4.3.1 | Model robustness towards augmented test images

In order to analyse the robustness of the proposed model, the test images were augmented to replicate actual conditions in palm oil plantation such as different brightness, Gaussian blurring and motion blur. In order to simulate the situation under different illumination intensities, two types of brightness transformation were used. The brightness of the test images was

TABLE 5 Detection results on test images

Parameter	Class	Test images	Models				
			YOLO-P	YOLOv3	YOLOv3-dense [34]	YOLOv3 tiny	YOLOV2
True positive	Bunch	446	424	413	414	387	410
	Grabber	160	160	158	159	159	160
	Palm	361	351	352	351	328	351
False positive	Bunch	0	14	34	36	34	69
	Grabber	0	4	8	8	8	28
	Palm	0	21	53	55	75	75
False negative	Bunch	0	22	33	32	59	36
	Grabber	0	0	2	1	1	0
	Palm	0	10	9	10	33	10
Precision (%)		1	96.0	90.7	90.3	88.2	84.3
Recall (%)		1	96.7	95.4	95.6	90.4	95.2
F1-score (%)		1	96.3	93.0	92.9	89.3	89.4

FIGURE 8 Visual detection results for YOLOv3 model

increased by numeric scalar of +25 for brighter condition and reduced by numeric scalar of −25 for darker condition. In addition, images captured at the palm oil plantation may be influenced by the camera movement and incorrect focus during image acquisition process. Hence, Gaussian filter with standard deviation of 1.5 was applied to simulate blur images on the test images. Moreover, numeric scalar of 10 that specify the length of motion and zero-degree angle of motion was applied to simulate motion blur images.

The detection results of the augmented test images are shown in Table 6. Experimental results show that the detection for brighter and blur test images have higher precision value compared to detection based on original test images. However, greater number of objects were misclassified leading to lower

FIGURE 9 Visual detection results for YOLOv2 model

a b

FIGURE 10 Visual detection results for YOLOv3 tiny model

TABLE 6 Detection results on augmented test images

		Test images				
Parameter	**Class**	**Original**	**Brighter**	**Darker**	**Blur**	**Motion blur**
True positive	Bunch	424	418	420	389	390
	Grabber	160	160	160	159	153
	Palm	351	348	348	337	309
False positive	Bunch	14	12	20	11	16
	Grabber	4	3	2	2	0
	Palm	21	19	21	19	19
False negative	Bunch	22	28	26	57	56
	Grabber	0	0	0	1	8
	Palm	10	13	13	24	52
Precision (%)		96.0	96.5	95.8	96.5	96.1
Recall (%)		96.7	96.0	96.2	91.7	88.3
F1-score (%)		96.3	96.2	96.0	94.0	92.0

recall value. Besides, detection of bunch and palm on the darker test images showed an increment in false positive and false negative, thus, lowering the percentage of precision and recall as compared to results based on original test images. Overall, the F1-score is slightly reduced for all augmented test images. The lowest reduction in percentage of F1-score is 4.47% which is on the motion blurred test images. The results demonstrated the robustness of the proposed YOLO-P model toward complex environment, which is a crucial aspect in developing an accurate object detection system for automated harvesting technology.

4.3.2 | Robustness towards different dataset size

In order to investigate the effect of training data size on the detection performance, six different dataset size were randomly chosen which include 50, 100, 500, 1000, 2500 and 5000 number of images respectively. From the precision-recall curve shown in Figure 11, it is worth to note that, the detection performance

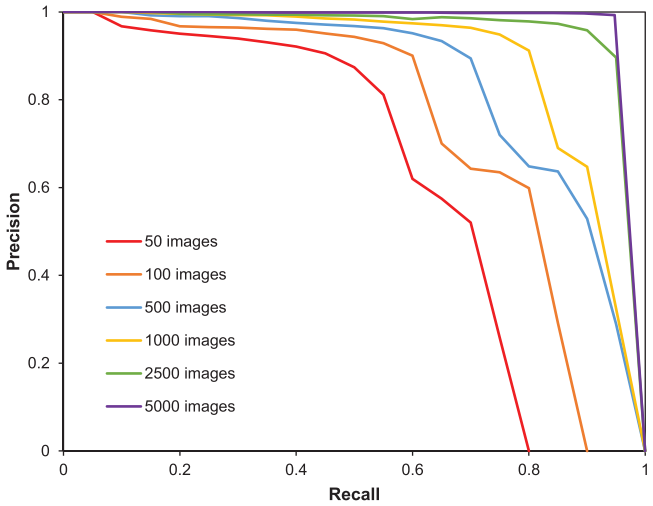

FIGURE 11 Precision-recall curves of the proposed model trained with different size of datasets

TABLE 7 Detection performance of the proposed YOLO-P trained with different size of dataset

Number of images	mAP (%)	IoU	F1- score
50	60.62	63.19	0.66
100	72.99	69.70	0.76
500	82.54	70.62	0.83
1000	88.97	72.21	0.88
2500	95.76	73.65	0.93
5000	98.68	83.84	0.97

of the model gradually improved corresponding to the expansion of dataset size. Beyond the size of 2500 number images, the impact of dataset size on the detection performance was reduced. The detection performance of the proposed YOLO-P is shown in Table 7. Hence, it can be concluded that 5000 images used as training images in this research is sufficient for the detection model.

4.3.3 | Detection performance under occlusion condition

Object detection in palm oil plantation is challenging due to the presence of complex environment such as occlusion. During the process of collecting FFB, occlusion may occur under various conditions where the objects are occluded by fronds, grabber or wheel of the tractor. On the other hand, there may be condition where the FFBs are overlapped with one another. This would have certain impact on the object detection system. Figure 12 presents the example of detection results for objects under occlusion condition by using the proposed YOLO-P model. Results show that the objects which are heavily occluded by palm frond and grabber were successfully detected by using the proposed YOLO-P model. In Figure 12(a,b), both grabber and palm tree were covered by the frond while some of the FFBs were covered by the grabber in Figure 12(b). Besides, the proposed model able to detect FFB under overlapped condition as shown in Figure 12(c). The results of this experiment exhibited the superiority of the YOLO-P model for detection under occlusion conditions with high classification accuracy.

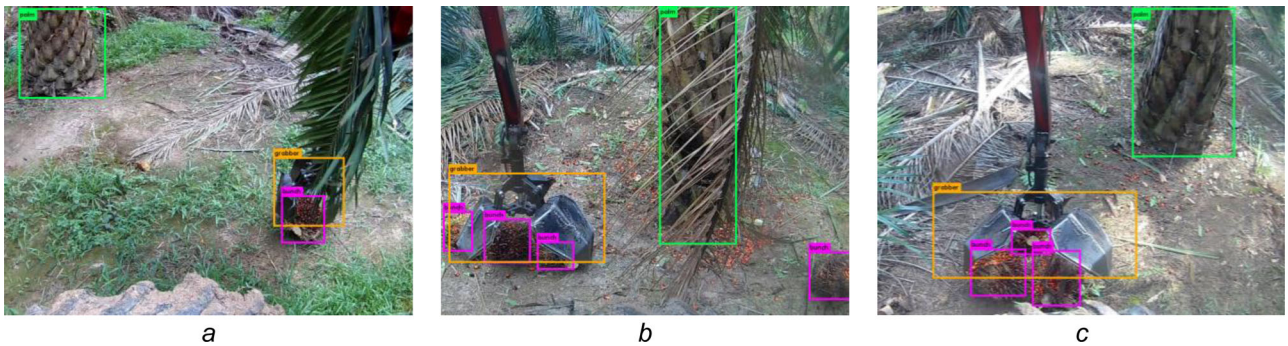

FIGURE 12 Detection of the proposed model in occluded condition

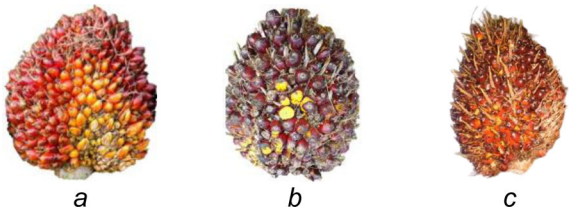

FIGURE 13 Fresh fruit bunches categories (a) ripe, (b) underripe and (c) overripe

4.3.4 | FFB ripeness classification

In order to further demonstrate the efficiency of the proposed YOLO-P model, another database was developed to classify FFB at different maturity during harvesting season. A total of 504 images were used as training images and 216 images were used as test images. The classification is divided into three categories of FFB ripeness which are ripe, overripe and underripe. The FFB ripeness categories are shown in Figure 13. FFB ripeness affects the oil extraction rate (OER) where ripe fruit produces higher OER which results to higher quantity of oil as compared to underripe and overripe fruit. Hence, classification of FFB ripeness is an important task in order to produce optimum oil production.

The detection results for several YOLO based models are shown in Table 8. It is notable that YOLO-P outperformed the other models with mAP of 98.96% followed by YOLOv3-dense (98.32%), YOLOv3 (98.17%), YOLOv2 (97.15%) and YOLOv3 tiny (96.52%). This result is in line with the result obtained in Table 2 demonstrating the advantages of using feature reuse architecture, multiple detection scale as well as optimization of anchor box. The proposed YOLO-P model obtained the highest precision and recall value which resulted in the highest F1-score compared to the other models. This indicates that the proposed model obtained the lowest false positive and false negative value. In term of average IoU, YOLO-P model achieved the highest percentage of 83.69%.

TABLE 8 Detection performances for FFB ripeness dataset

Method	Average IoU (%)	Precision (%)	Recall (%)	F1-score	AP (%)			
					Ripe	Underripe	Overripe	mAP (%)
YOLOv2	75.73	88.08	96.65	0.92	98.58	96.82	96.06	97.15
YOLOv3 tiny	76.33	93.53	93.25	0.93	96.66	93.56	99.33	96.52
YOLOv3-dense [34]	82.47	95.75	96.93	0.96	98.35	97.53	99.08	98.32
YOLOv3	82.88	95.46	96.93	0.96	97.32	97.97	99.21	98.17
YOLO-P	83.69	96.07	97.54	0.97	98.65	98.18	99.91	98.91

Furthermore, YOLO-P model has the ability to classify the three categories of FFB ripeness with high accuracy and the AP obtained are the highest among the other models. The AP for ripe is 98.65%, underripe is 98.18% and overripe is 99.91%. These results exhibited the effectiveness of the proposed YOLO-P model for classification of different FFB ripeness which is crucial for crop harvesting system based on machine vision.

5 | CONCLUSION

A novel detection model based on computer vision was proposed to detect images at palm oil plantation. The proposed YOLO-P model is inspired by YOLOv3 tiny network considering its advantages of fast computation time with good detection performance. In order to improve the accuracy of detection, several improvements were proposed which include the implementation of DenseNet with 201 configurations to act as the backbone of the network, utilization of four scales detection and clustering to determine appropriate anchor size. The proposed model was compared with other YOLO based models. The experimental results show that, the proposed model achieved satisfactory detection performance with mAP of 98.68%, F1 score of 0.97 and average IoU of 83.34% which was the highest compared to the rest of the models. In term of detection time, it was slightly lower than the other YOLO based models and SSD model but is still considered faster compared to Faster R-CNN models. Furthermore, YOLO-P model demonstrated significant improvement in computational time and trained model size. The proposed model outperformed YOLOv3 by 1.57 times faster in term of computational time and developed a smaller model size with 76 MB. These advantages may significantly reduce the cost for hardware implementation. In addition, satisfactory result was obtained for classification of different FFB maturity with mAP of 98.91%. In conclusion, the results have proven that YOLO-P model is feasible for object detection in palm oil plantation that is applicable for automatic crop harvesting system.

For future work, the proposed model will be used for object detection in video to meet real time application. Besides, counting algorithm will be employed for yield prediction.

ACKNOWLEDGEMENTS

The research is funded by RU Grant-Faculty Programme by Faculty of Engineering, University of Malaya with project number GPF042A-2019 and Industry-Driven Innovation Grant (IDIG) by University of Malaya with project number PPSI-2020-CLUSTERSD01.

REFERENCES

1. Kushairi, A., et al.: Oil palm economic performance in Malaysia and R&D progress in 2017. J. Oil Palm Res. 30(2), 163–195 (2018)
2. Corley, R.H.V.: How much palm oil do we need? Environ. Sci. Policy 12(2), 134–139 (2009)
3. Ng, Y.G., et al.: Ergonomics observation: Harvesting tasks at oil palm plantation. J. Occup. Health 55(5), 405–414 (2013)
4. Kassim, M.S.M., et al.: Oil palm fresh fruit bunches (FFB) growth determination system to support harvesting operation. J. Food Agric. Environ. 10(2), 620–625 (2012)
5. Azhar, F., et al.: Initial progress and possible improvement of E-Cutter linear actuator development. In: *2012 IEEE International Conference on Power and Energy*, Kota Kinabalu, Malaysia, pp. 940–945 (2012)
6. Jelani, A.R., Hitam, A., Jamak, J., et al.: High reach oil palm motorized cutter (cantas7). MPOB Inf. Ser. (349), 1–2 (2007)
7. Abd Rahim, S., Mohd Ramdhan, K., Mohd Solah, D.: Innovation and technologies for oil palm mechanization. In: *Further Advances in Oil Research, 2000–2010* Malaysian Palm Oil Board, pp. 570–597 (2011)
8. Shuib, A.R., Khalid, M.R., Deraman, M.S.: Enhancing field mechanization in oil palm management. Oil Palm Bull. 61, 1–10 (2010)
9. Mairon, R., Edan, Y.: Computer vision for fruit harvesting robots - State of the art and challenges ahead. Int. J. Comput. Vis. Robot. 3, 4–34 (2012)
10. Song, Y., et al.: Automatic fruit recognition and counting from multiple images. Biosyst. Eng. 118(1), 203–215 (2014)
11. Sengupta, S., Suk, W.: Identification and determination of the number of immature green citrus fruit in a canopy under different ambient light conditions. Biosyst. Eng. 117, 51–61 (2014)
12. Maldonado, W., Barbosa, J.C.: Automatic green fruit counting in orange trees using digital images. Comput. Electron. Agric. 127, 572–581 (2016)
13. Qureshi, W.S., et al.: Machine vision for counting fruit on mango tree canopies. Precis. Agric. 18(2), 224–244 (2016)
14. Ramos, P.J., et al.: Automatic fruit count on coffee branches using computer vision. Comput. Electron. Agric. 137, 9–22 (2017)
15. Kaur, S., Pandey, S., Goel, S.: Semi-automatic leaf disease detection and classification system for soybean culture. IET Image Proc. 12(6), 1038–1048 (2018)
16. Kurtulmus, F., Lee, W.S., Vardar, A.: Immature peach detection in colour images acquired in natural illumination conditions using statistical classifiers and neural network. Precis. Agric. 15(1), 57–79 (2014)
17. Yamamoto, K., et al.: On plant detection of intact tomato fruits using image analysis and machine learning methods. Sensors 14, 12191–12206 (2014)

18. Hamza, R., Chtourou, M.: Design of fuzzy inference system for apple ripeness estimation using gradient method. IET Image Proc. 14(3), 561–569 (2020)
19. Koirala, A., et al.: Deep learning – Method overview and review of use for fruit detection and yield estimation. Comput. Electron. Agric. 162, 219–234 (2019)
20. Chen, S.W., et al.: Counting apples and oranges with deep learning : A data driven approach. IEEE Rob. Autom. Lett. 2, 781–788 (2017)
21. Dyrmann, M., Jørgensen, R.N., Midtiby, H.S.: RoboWeedSupport - Detection of weed locations in leaf occluded cereal crops using a fully convolutional neural network. Adv. Anim. Precis. Agric. 8, 842–847 (2017)
22. Dias, P.A., Tabb, A., Medeiros, H.: Apple flower detection using deep convolutional networks. Comput. Ind. 99, 17–28 (2018)
23. Ren, S., et al.: Faster R-CNN : Towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. 36(1137–1149), 1–14 (2017)
24. Sa, I., et al.: Deepfruits: A fruit detection system using deep neural networks. Sensors (Switzerland) 16(8), 1222 (2016)
25. Bargoti, S., Underwood, J.: Deep fruit detection in orchards. In: *IEEE International Conference on Robotics and Automation*, Singapore, pp. 3626–3633 (2017)
26. Madeleine, S., Bargoti, S., Underwood, J.: Image based mango fruit detection, localisation and yield estimation using multiple view geometry. Sensors 16(11), 1915 (2016)
27. Le, T.T., Lin, C.Y., Piedad, E.J.: Deep learning for noninvasive classification of clustered horticultural crops - A case for banana fruit tiers. Postharvest Biol. Technol. 156, 110922 (2019).
28. Min, W., et al.: New approach to vehicle license plate location based on new model YOLO-L and plate pre-identification. IET Image Proc. 13(7), 1041–1049 (2019)
29. Huang, R., et al.: A rapid recognition method for electronic components based on the improved YOLO-V3 networks. Electronics 8(8), 825 (2019)
30. He, W., et al.: TF-YOLO : An improved incremental network for real-time object detection. Appl. Sci. 9(16), 3225 (2019)
31. Tian, D., et al.: An automatic car accident detection method based on cooperative vehicle infrastructure systems. IEEE Access 7, 127453–127463 (2019)
32. Cao, C., et al.: Investigation of a promoted You Only Look Once algorithm and its application in traffic flow monitoring. Appl. Sci. 9(17), 3619 (2019)
33. Koirala, A., et al.: Deep learning for real-time fruit detection and orchard fruit load estimation: Benchmarking of "MangoYOLO". Precis. Agric. 20, 1107–1135 (2019)
34. Tian, Y., et al.: Apple detection during different growth stages in orchards using the improved YOLO-V3 model. Comput. Electron. Agric. 157, 417–426 (2019)
35. Li, T., Ma, Y., Endoh, T.: A systematic study of tiny YOLO3 inference: Toward compact brainware processor with less memory and logic gate. IEEE Access 8, 142931–142955 (2020)
36. Redmon, J., et al.: You Only Look Once : Unified, real-time object detection. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, USA, pp. 779–788 (2016)
37. Redmon, J., Farhadi, A.: YOLOv3 : An incremental improvement. In: *IEEE Conference on Computer Vision and Pattern Recognition*, Utah, USA (2018)
38. Zhang, P., Zhong, Y., Li, X.: SlimYOLOv3: Narrower, faster and better for real-time UAV applications. In: *Conference on Computer Vision and Pattern Recognition*, CA, USA, pp. 37–45 (2019)
39. Liu, Z., Wang, S.: Broken corn detection based on an adjusted YOLO with focal loss. IEEE Access 7, 68281–68289 (2019)
40. Huang, G., et al.: Densely connected convolutional networks. In: *IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, USA, pp. 2261-2269 (2017)
41. Han, S., et al.: Learning both weights and connections for efficient neural networks. arXiv:1506.02626, 1–9 (2015)
42. Liu, W., et al.: SSD : Single shot multiBox detector. In: Leibe B., Matas J., Sebe N., Welling M. (eds) Topics in Artificial Intelligence, Computer Vision–European Conference on Computer Vision, 2016. Lecture Notes in Computer Science, vol. 9905. pp. 21–37.Springer, Cham (2016)

How to cite this article: Junos, M.H., et al.: An optimized YOLO-based object detection model for crop harvesting system. IET Image Process. 15, 2112–2125 (2021). https://doi.org/10.1049/ipr2.12181