Detection of Oil Palm Tree and Loose Fruitlets for Fresh Fruit Bunch's Ready-to-Harvest Prediction via Deep Learning Approach

Marizuana Mat Daud, Zulaikha Kadim, Hon Hock Woon

Abstract— **The oil palm industry is one of the most important industries for Malaysia's economic growth. Worldwide demand for palm oil is anticipated to increase from 51 million tonnes to between 120 and 156 million tonnes over the next 30 years. However, the industry is highly dependent on foreign labor, particularly in harvesting and gathering fresh fruit bunches (FFBs). Workers must manually determine the FFB's ripeness prior to harvest. Thus, labor shortages, which are exacerbated by the recent COVID-19 pandemic, have had a significant impact on industry productivity. Therefore, in this paper, a system is proposed to identify ready-to-harvest FFB automatically by counting loose fruitlets on the ground. Data collection is performed in one of the smallholder plantation areas. The numbers of trees and fruitlets are 2140 and 490, respectively. The system starts with tree detection using YOLOv5 because of its lightweight model with a minimal computational cost. The tracking of the discovered trees will ensure that each tree is identified just once, and loose fruitlets on the ground will thereafter be found. The number of loose fruitlets detected will be used to infer the readiness to harvest the FFB. The results for the fruitlet detection had mAP of 85.45% while those for the palm tree detection had mAP of 97.79%. On the other hand, at counting thresholds of zero and three, respectively, the accuracy of ready-to-harvest classification is 90.48% and 91.34%.**

Index Terms— **Palm oil; fresh fruit bunch; loose fruitlet; image processing, deep learning, harvest-ready prediction**

I. INTRODUCTION

With an anticipated global population of 9.7 billion by 2050, at least 240 million tonnes of palm oil will be demanded to meet the pressure [1]. The global palm oil production of 84% is dominated by Indonesia and Malaysia. According to Fig. 1, Malaysia produced over 20 million tonnes of palm oil annually in 2018, ranking second globally and accounting for 27% of global production [2]. In addition, Malaysia is the world's second-largest exporter of palm oil, sending 17,368,865 tonnes of palm products abroad in 2020 [3] to nations like China, India, the Netherlands, Turkey, and the United States. Malaysia has 5.87 million hectares of oil palm agriculture yielding, but due to the COVID-19 pandemic and the obligation movement control order (MCO), which causes the delay, there was a slight decrease of 0.6% from the previous year in 2019 [4]. Moreover, shortage of labor workers decelerated the production of palm oil crops. Now, agriculturists have been exposed to the artificial intelligence technology and farmers have started to ingress with the machines and tools to meet the demand, which lowers production costs while simultaneously raising productivity.

A revolution in technology that ranges from computer vision to robotics and the fusion of multiple cameras has entirely revolutionized many industries, including palm oil industry to modern agriculture. The main objective of an automated harvesting system is to develop a robust and precise algorithm to perform crop detection, tracking and classification. Throughout its evolution, machine vision has witnessed the advancement of multiple image processing techniques, including edge detection, region growth, shape detection, and color-based preprocessing. These methods have been employed to extract crop features from a diverse range of visual factors. Due to substantial advancements in deep learning, deep convolutional neural networks (CNNs) have attained remarkable accuracy and detection speed, establishing them as the forefront technology for object detection [1].

Many researchers have adopted CNN due to its capability to extract features and learn from the input image automatically through self-learning [1]. Lawal [5] proposed You Only Look Once (YOLOv3) to detect tomatoes for real-time harvesting. He used the same approach to detect muskmelons [6] by comparing YOLOv3, YOLO-Resnet50 and YOLOv4. Yu et al. [7] adopted MaskRCNN to detect strawberry fruits with 100 images that contain 573 ripe fruits. All four research papers achieved mAP of more than 90%. Moreover, there are more similar approaches for different object interests such as fresh fruit bunch (FFB) [1], orange [8], pear [9], blueberry [10], and all sorts of fruits [11]. Currently, FFB color is one of the most important cues that determines the grade and quality of the FFB. It is important to harvest FFB during its optimal maturity stage so as to ensure the quality of the fruits [12-14].

The quality of oil palm FFB is measured by its low free fatty acids (FFA) and high oil extraction rate (OER). By assessing the surface color of the fruit bunches, the Malaysian Palm Oil Board (MPOB) categorized the maturity stages of FFB into four classes; unripe, underripe,

Manuscript received Oct 28, 2022; revised May 22, 2023.
This work was supported in part by MIMOS Berhad.

M. M. Daud is a senior researcher of MIMOS Berhad, 57000 Kuala Lumpur, Malaysia. (corresponding author to provide phone: +60196219189; e-mail: darissa_riz88@yahoo.com).
Z. Kadim is a senior researcher of MIMOS Berhad, 57000 Kuala Lumpur, Malaysia. (e-mail: zulaikha.kadim@mimos.my).
H. H. Woon is a principal reseacher of MIMOS Berhad, 57000 Kuala Lumpur, Malaysia. (e-mail: hockwoon.hon@mimos.my).

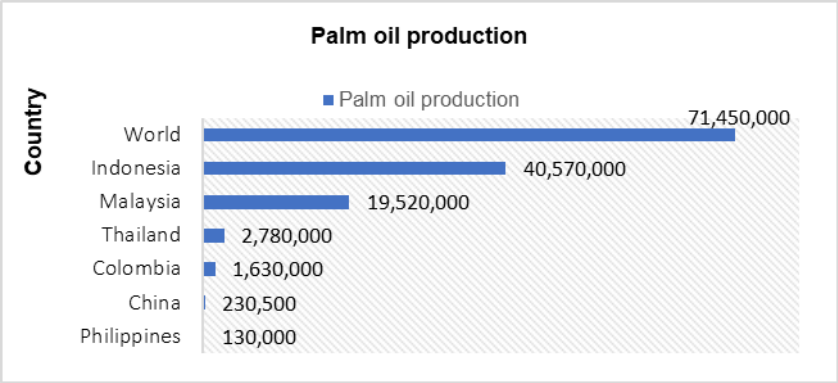


Fig. 1. Palm oil production

optimally ripe, and overripe. The unripe class is characterized by a purplish-black color.

Meanwhile, under-ripe and optimally ripe classes appear to be reddish black and reddish-orange respectively. The overripe class exhibits almost the entire FFB reddish-orange coloration.The bunch is at its most mature stage when it is optimally ripe because it has a high oil content that increases OER, whereas unripe fruits have nil or little oil, which lowers OER [15]. In contrast, because there are already many fruits that have fallen from the bunch, overripe fruits provide less OER than ripe bunches. Therefore, the owners need to know the best time to harvest so as to maximize the production yields.

The majority of the time, the harvest readiness is assessed manually, requiring the workers to visit the trees and gauge the age of the bunch. According to the smallholder's recommendation in [16], ripe bunches can be spotted manually based on the loose fruits that are buried beneath frond butts on the trunk of the palm tree or lying in a weeded circle around it. When harvesting tall palms, it is not advisable for the workers to look at the color of the bunches because, despite appearing red, the bunches may still be immature. In either case, this manual method is time-consuming and heavily reliant on the workers' skill and experience.

Due to this, numerous investigations have been carried out employing sensors as RGB cameras [17–18], hyperspectral imaging [19], near-infrared (NIR) spectroscopy [20], and lidar [21] to automatically determine FFB maturity level. The majority of algorithms that have been created rely on the fruitlets' colors as determined by computer vision and machine learning approaches. Support vector machine (SVM) is utilized as the classifier in [17] to categorize fruitlets into 4 different maturity levels. Color characteristics and a bag of visual words are used as the feature extractors. The method used a color feature and a bag of words, respectively, to obtain classification accuracy of 57% and 70%. Because a cell phone camera was used throughout the procedure, low accuracy may result from the poor quality of the photos that were recorded and used for system training. The color histogram and statistical color feature were employed as the feature extractor in [18], along with other color features. The classifier was an artificial neural network (ANN). The FFB was already harvested and sent to the mill for processing when the proposed technique obtained a classification accuracy of 94%. In [19], reflectance values, rather than color characteristics, were captured using a hyperspectral device at various wavelengths and used as the feature extractor before being sorted into various ripeness levels using ANN as the classifier. Similar to [21], the ripeness level was calculated using the reflectance intensity value recorded by the lidar. The article demonstrated that the reflectance values lacked sufficient distinction to distinguish between the maturity levels.

Although several appearance-based processing techniques produced useful results in identifying the FFB's maturity level, the established techniques are not appropriate for use in a real plantation setting to determine which tree is ripe for harvest. One of the issues is brought on by the inconsistent color of matured FFB, which differs not only between different palms and regions [22], but also depending on when visual data was captured. In addition, the tall tree may result in obstructions to the FFB from nearby bunches and other items like fronds. Therefore, these elements will make it more difficult for approaches based on the FFB's aesthetic look to determine the bunch's maturity level on the tree accurately.

The quantity of loose fruitlets on the ground, on the other hand, was linked in certain studies to the FFB maturity [23–24]. The minimal maturity standard—also known as the amount of loose fruits on the ground—helps establish whether the FFB is at its minimum ripeness standard (MRS). There was standard established by the Thailand Department of agriculture [25], the Indonesian oil palm research institute [26], and the Malaysian palm oil board [27] all set a criterion that declared that the bunch is ready to be harvested if there are more than 10 palm fruitlets under the tree. While it was indicated in [16,28] that the FFB achieves optimum ripeness level if there are at least three loose fruitlets lying on the ground. Although the standard value varies from one company to another; however, the approach is more feasible and accurate as compared to accessing the FFB maturity level by evaluating the bunch on the tree. This is because the fruitlets on the ground can be seen more clearly as opposed to the bunch on the tree. However, no automated system has been suggested in the literature to assess the FFB's level of maturity based on the number of loose fruitlets.

The method was developed using deep-learning model to detect palm tree and loose fruitlets within the ROI of the tree so as to deduce the harvest-ready status whether or not it is time to harvest. The remainder of this paper consists of the following sections. The methodology section outlines the

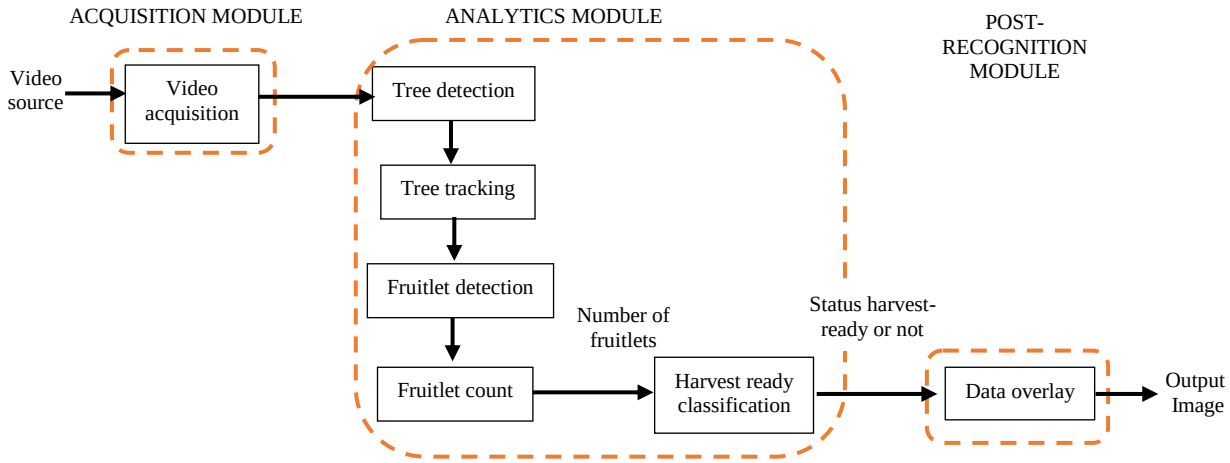


Fig. 2. Overall architecture of the ready-to-harvest prediction method

Fig. 3. Condition of the plantation area for data gathering activity where the road is muddy, pile of dried leaves and tree occluded by leaves.

general approach for gathering data, identifying trees and fruitlets, and making the final classification of harvest-ready status. In the results and discussion section, the results of both YOLOv4 and YOLOv5 performance were demonstrated and compared. Finally, the proposed work was concluded in the conclusion section.

II. Methodology

As shown in Fig. 2, the overall system architecture consists of the three main modules: data collection and acquisition, analytics and post-recognition modules. The input can typically take the form of photos or video input. The analytics module uses the photos that were captured to determine which tree in the image is ready for harvest. In the post-recognition module, the results of the detection and recognition are then superimposed over the output image to show the user which tree is ready for harvest. An oil palm tree is deemed harvest-ready in the first research if at least one or more loose fruitlets are found nearby. Each module's thorough explanation will be covered.

A. Data collection and acquisition module

Data collection and acquisition is a critical step in machine vision as it significantly affects the training process due to the experimental setting and image quality. The data collection process was carried out in a smallholder's plantation located in Sg. Pelek, Selangor, Malaysia. The data-gathering activities were conducted in both morning and afternoon sessions. The plantation area features tall trees, reaching approximately 5 meters in height, with favorable lighting conditions beneath the canopy. Each tree is separated by an average distance of 9 meters, as illustrated in Fig. 3, depicting the environmental state of the plantation. However, the presence of weeds, mounds of dried leaves, and sometimes dried or green leaves hanging from the trees pose significant challenges to data collection. Additionally, the off-road conditions vary depending on the season, with rainy periods leading to muddy roads.

The acquisition module gathers images from a video source or input camera. The specifications of the camera used for data collecting are as follows:

- Camera model: DJI Osmo Pocket
- Camera resolution: 1920x1080
- Frame rate: 60fps
- Camera features: 3-axis gimbal stabilization, portable and lightweight

The camera configuration used during data collection is presented as follows: (a) the camera's distance from the target tree, which was maintained at a range of 3 to 4 meters; (b) the camera's height above the ground, which was set at approximately 1.5 meters; and (c) the camera itself, which was handheld and moved at a walking pace, as illustrated in Fig. 4. The data collection process involved capturing images of tree trunk surfaces, including smooth surfaces, surfaces covered in weeds, and surfaces covered in leaves, as depicted in Fig. 5. Data were collected under three distinct plantation conditions to ensure variability in the dataset.

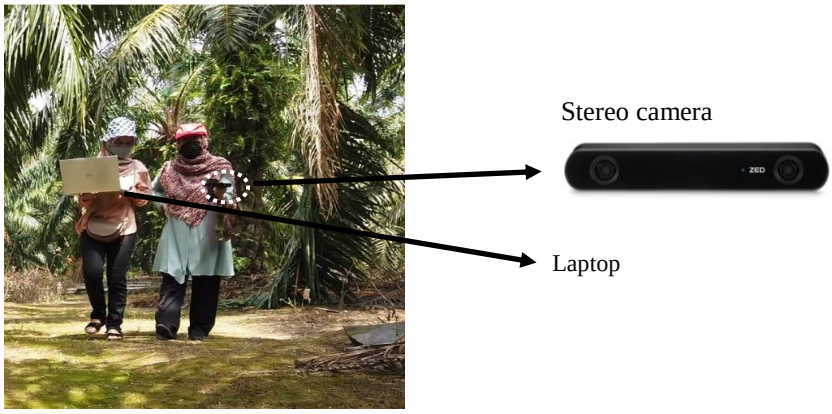


Fig. 4. The handheld camera with laptop to view the online testing output.

Fig. 5. Several types of tree trunk, smooth trunk (left), Trunk with remaining fronds and less weed (center), and Trunk full of leaves (right).

B. Analytics module

The analytics module serves as the primary processing component, responsible for determining whether a tree within the image is ready for harvest. This determination is based on the number of fruitlets detected within the tree's region-of-interest (ROI). The analytics module encompasses five processes, including tree detection, tree tracking, fruitlet detection, fruitlet counting, and harvest-ready classification.

Once the tree is detected, the camera's distance from the tree is estimated to identify trees in close proximity to the camera for further examination. The tree is tracked when the lower midpoint of the detected tree enters the monitoring region to ensure that each tree is classified only once. The area of the image closest to the ground around the tracked tree trunk is cropped and provided to the fruitlet detection module. To ensure that the relative size of the loose fruitlets to the image size is significantly high, this sub-image is used as the input to the fruitlet detection module instead of the entire image. The fruitlets are then tracked for a few frames to establish the final count. Based on the total number of fruitlets detected, each identified tree is classified as either harvest-ready or not. The results from the analytics module are superimposed on the output image to enable user visualization.

Tree and fruitlets detection modules

A training dataset is created in order to establish the detection model. Annotation is then performed manually using the online annotation tool CVAT [34] to indicate the location of trees and fruitlets in each training image.

TABLE I
TOTAL NUMBER OF TREE AND FRUITLET DATASET

Object	Training	Testing	Total
Tree	1498	642	2140
Fruitlet	343	147	490

Annotated data includes the label file for each image in YOLO format. In this project, the model is trained using 1498 images of trees and 343 images of fruitlets, with an average of one tree and more than ten fruitlets per image. The datasets are distributed between training and test data with a ratio of 70:30, as can be seen in Table I. The condition of the plantation environment, which includes dried fronds that are strewn on the ground, green fronds that partially cover the tree trunk, a darker ground area under the tree, and the size of the loose fruitlets, which is relatively small in the image, present significant challenges to the detection of the tree and the loose fruitlets.

In this study, we employed YOLOv4 [29] and YOLOv5 [30] to train a model for detecting oil palm trees and loose fruitlets. Based on prior research, it has been observed that YOLOv5 outperforms previous iterations of YOLO, namely YOLOv3 and YOLOv4, with regards to both accuracy and speed [35-36], despite some controversy surrounding YOLOv5's exceptional advances and lack of official publications [31]. YOLO, which stands for You Only Look Once, is a single convolutional network that uses multiple convolutional networks. YOLO generates predictive vectors for each object that appears in the image. The key concept of YOLO is to compute all features and make predictions for all objects simultaneously. This is achieved by applying a grid cell, typically of size S × S (default is 7×7), to an image. If the center of an object falls within a particular grid cell, that grid cell is responsible for detecting and identifying the object. The network then provides an offset value for the bounding box and class probability for each bounding box by considering "*m*" bounding boxes for each grid cell [32].

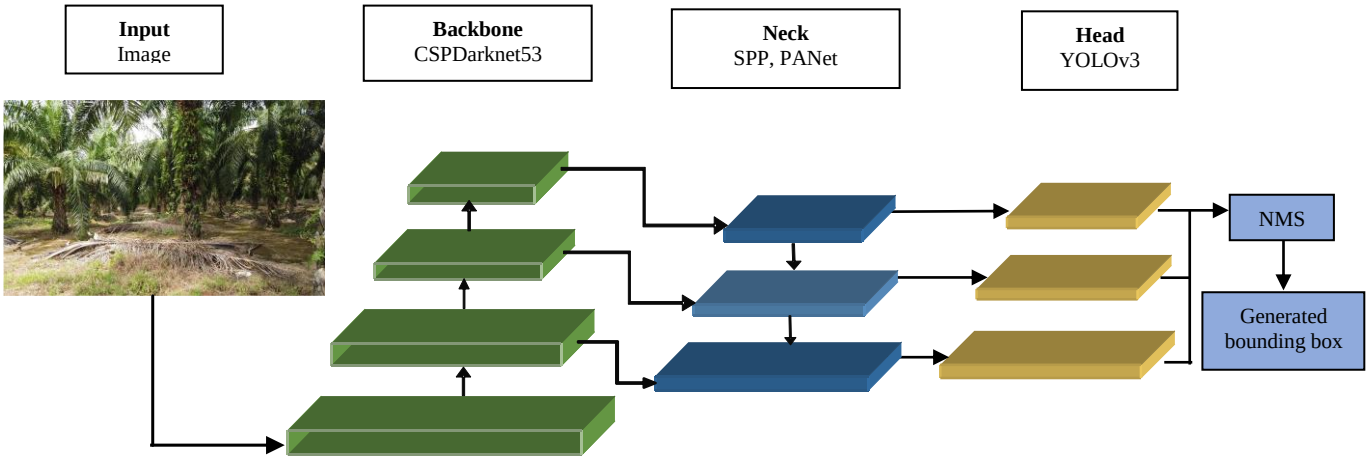


Fig. 6. Overall architecture of YOLOv4

The YOLOv4 architecture exhibits significant improvement over its predecessor, YOLOv3. Bochkovskiy et al. [29] designed a complicated and deeper network using a Dense network to replace the residual block in YOLOv3, resulting in the overall architecture portrayed in Fig. 6. The network's backbone is Darknet-53 from YOLOv3, augmented by the Cross-Stage Partial (CSP) network. CSP divides the input into two parts: (a) the input goes through the convolution of a dense block, and (b) the data moves directly to the subsequent stage of DenseNet without being processed. This approach preserves fine-grained features by moving the process into a deeper layer and reduces the number of network parameters by repeating the features [31]. Spatial pyramid pooling (SPP) is employed in the neck portion of feature aggregation to widen the receptive field and discern the most important features while slowing down. Path Aggregation Network (PANet), which replaces Feature Pyramid Network (FPN) in the previous YOLO version, is another notable modification in YOLOv4. Finally, the network's head portion still uses the original YOLOv3 network.

Glenn Josher [30] introduced YOLOv5 a month after YOLOv4's release. Both YOLOv4 and YOLOv5 share very similar architecture with minor differences. However, YOLOv5 sparked controversy when it performed better than YOLOv4 while being built in Python, which is simpler and easier to install and integrate on devices.The most striking feature of YOLOv5 is its ability to automatically learn the anchor boxes during training and select the appropriate size to fit over the detected object. YOLOv5 can be encapsulated as follows [30]-[31]:

- Backbone: Focus structure, CSP network
- Neck: SPP block, PANet
- Head: YOLOv3 head using GIoU-loss

Moreover, YOLOv5 is a more adaptive learner than YOLOv4 when the object significantly differs in terms of shape or size.

Tree tracking module

The system incorporates tree tracking to prevent redundant tree counting. Given that the distance between trees can range from 5m to 9m, depending on the owner, tree detection consistency is employed as an input to the tree tracking component. A tree is considered to be the same tree if it is consistently detected within the processing window. Moreover, the trees must be at approximately the same distance from the camera as it moves from one tree to the next.

The pseudocode for the process is depicted in Fig. 7, where parameters *f*, *nf*, *label*, *num_tree*, and *tree_label* indicate the number of frames with continuously detected trees, the number of frames with continuously undetected

Algorithm 1: Tree tracking
Input: image sequence & detected tree location
Output: tracking label of the detected tree (*tree_label*)
Initialization: *f-0*, *nf-0*, *label-0*, *num_tree-0*
read current image
if tree detected in current image **then:**
  if tree within the tracking region **then:**
    f++
    nf=0
    if *f*==t_1 **then:**
      //new tree detected
      label++
      tree_label=label
      num_tree++
    if *f*>t_1 **then:**
      //not a new tree
      tree_label=label
  else
    nf++
    if *nf*>t_2 **then:**
      f=0
  go to the next image
else:
  nf++
  if *nf*>t_2 **then:**
    f=0
  go to the next frame
end

Fig. 7. Pseudocode of the tree tracking

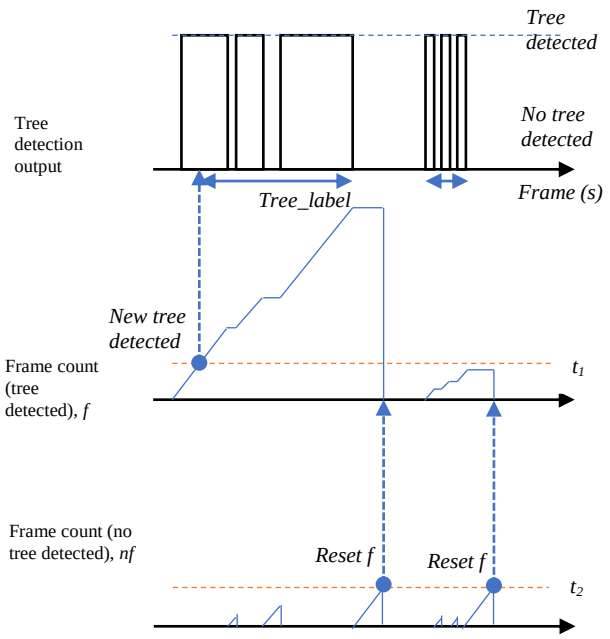


Fig. 8. Overall scenario of tree tracking output

trees, the current tracker label index, the number of trees tracked, and the tracking label of a detected tree in the current image, respectively. In the initial frame, these parameters are all initialized to zero. The input to the algorithm consists of the image sequence and the detected tree location in each image frame. The tracker algorithm distinguishes between new and existing trees and assigns a label to each tree. If a tree is detected within the tracking region, the parameter *f* is incremented (line 4), which denotes the total number of frames with continuous tree detection.

As illustrated in Fig. 8, the value of *f* is used to determine whether the detected tree is new (condition in line 6), previously tracked (condition in line 11), or noise. The value of f is reset when there are continuous frames with no detected tree, which is indicated by the parameter *nf*. Once the value of *nf* reaches the threshold *t2*, only the value of *f* is reset. This accounts for possible detector failure when detecting trees. If the detector does not detect any tree for *t2* frames in a row, then it is concluded that there is no tree in the image. Fig. 8 further demonstrates a sample scenario of the tracking algorithm output. The top graph corresponds to the tree detection output, while the middle graph displays the parameter *f*, representing the accumulated number of frames with continuous tree detection. The threshold *t1* is used to define a new tree to be tracked once tree detection is confirmed. The bottom graph shows the number of frames with continuous undetected trees, *nf*. The value of *nf* is accumulated when there is no tree detected in the current frame and is reset when a tree is detected. If the tree detection model fails to detect a tree in alternate frames, the value of *f* is not reset.

Fruitlets counting & ready to harvest recognition

In this study, the ripeness of an oil palm fruit cluster for harvesting was assessed by counting the number of loose fruits in close proximity to the tree. The counting accuracy directly influences the reliability of the system. Given that

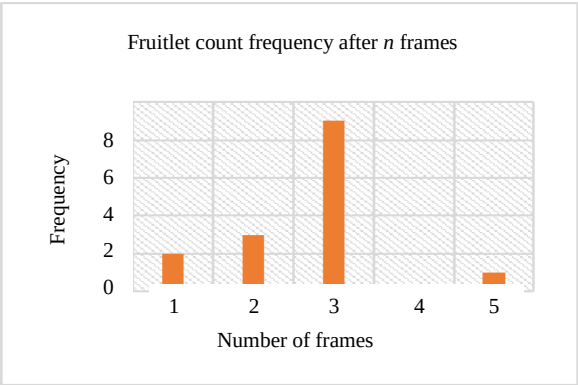


Fig. 9. Example of fruitlet count frequency distribution in n=15 number of frames

some loose fruits may not be visible in earlier frames due to camera movements, a solution was devised. To mitigate this issue, the loose fruit count was calculated after multiple frames of the tree had been tracked. The fruit count in each frame was recorded once the tracking began. After tracking for a predetermined number of frames (*n*), a final count was made based on the highest counting frequency. As illustrated in Fig. 9, for instance, the final fruitlet count was determined to be three after tracking for *n*=15 frames.

C. Evaluation Method

The evaluation of tree and fruitlet detection accuracy is based on the computation of average precision (AP) at a specific intersection over union (IOU) threshold. In this study, AP is computed at a threshold of 50% IOU, denoted as AP@.50. This metric is widely used in assessing the performance of object detectors such as YOLO, Faster R-CNN, and SSD. The average precision is computed by measuring the precision value for recall values ranging from 0 to 1. Precision represents the accuracy of the model's predictions or detections, i.e., the percentage of correct predictions or detections. In contrast, recall measures the ability of the detection model to identify all the true positives. The average precision score ranges from 0.5 (for balanced data) to 1.0 (for a perfect model). The determination of IOU is illustrated in Figure 10, where it is defined as the ratio of the overlapping area between the ground truth (represented by the solid line box) and the predicted or detected object (represented by the dashed line box). IOU values range between 0 and 100%, where higher IOU values indicate better overlap between the ground truth and detection output.

In vision applications that require rigorous performance guarantees, inconsistent behavior becomes a matter for concern [33]. Furthermore, accuracy does not indicate how reliable the detector is. If there are any inaccuracies during detection, there may be further information that we can describe. In our case, consistency is important because our tracker depends on the consistency of object detection performance, as we discussed in *Tree Tracking Module*. The consistency of an object detector on a pair of images (I and J) is referred to as "pairwise consistency". It is determined by using (1) [33] and (2) [33]. G_i is the set of I_i's ground truth, and G_j is the set of I_j's ground truth. $M_{i,j}$, $M_{j,i}$ captures the objects that were inconsistently detected as follows: $M_{i,j}$

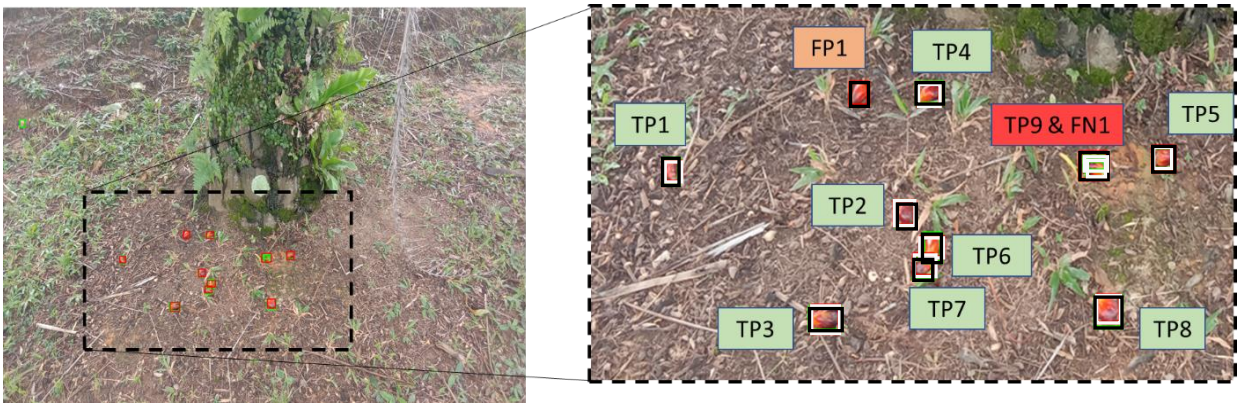


Fig. 11. Sample detection results

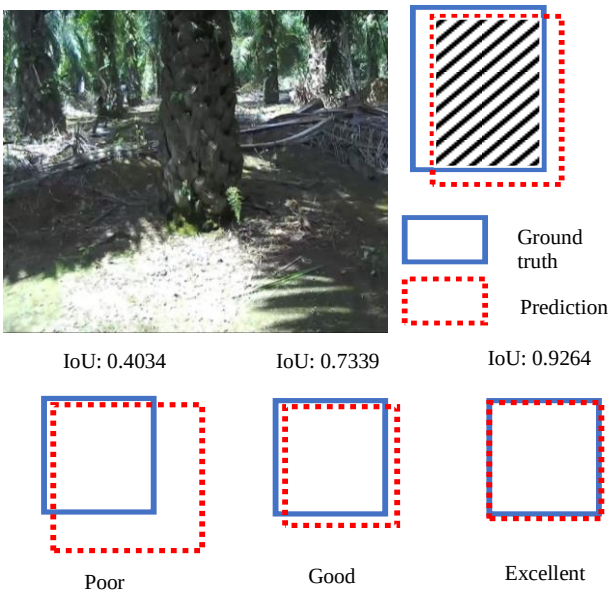


Fig. 10. IoU definition for poor, good and excellent

is the set of ground truth that satisfy the following conditions: (1) the ground truth box is present in both images I_i , I_j (i.e. in $G_i \cap G_j$), (2) the object detector detected the object in frame I_i, and (3) the object detector missed the object in frame I_j. A value of $C=1$ indicates perfect consistency, while a value of $C=0$ indicates complete inconsistency.

$$C_{i,j} = \frac{|G_i \cap G_j| - |M_{i,j}| - |M_{j,i}|}{|G_i \cap G_j|} \quad (1)$$

$$C_v = \frac{1}{N-1}\sum_{i=1}^{N-1} C_{i,i+1} \quad (2)$$

We conducted tests on 1000 images taken from 2 videos that were extracted at a rate of 10 frames per second (FPS). The first video has a duration of approximately 9 minutes and 57 seconds, while the second video lasts for about 2 minutes and 4 seconds.

For the ready-to-harvest prediction component, the accuracy is calculated by comparing the number of correctly recognized harvest-ready trees to the total number of tree samples. Another way to express it is by calculating the ratio of the combined true positives (TP) and true negatives (TN) to the total number of trials, which includes the sum of TP, false positives (FP), false negatives (FN), and TN. To illustrate the TP, TN, FP and FN, a confusion matrix as in Table II is referred to. In the figure, predicted is the ground truth, while true/actual is the system output. From the example in (3) below, accuracy is equal to 70%.

TABLE II
SAMPLE CONFUSION MATRIC FOR ACCURACY CALCULATION

		True/Actual	
		Positive	**Negative**
Predicted	**Positive**	5 (TP)	1 (FP)
	Negative	2 (FN)	2 (TN)

$$\frac{5+2}{5+2+1+2} \times 100 = 70\% \quad (3)$$

For ready-to-harvest classification, the input for the testing is a set of short video clips that contain one palm tree to be analyzed. The clips are annotated as harvest-ready or not based on the number of loose fruitlets on the ground near the tree trunk. When determining readiness, two threshold values—corresponding to at least one or three free fruitlets—are employed.

III. Results and Discussion

A. Tree and Fruitlet Detection Results

The detection results are presented in Fig. 11, where the left image shows the detection area and the right image is a zoomed-in version of the same area. The white and black boxes represent the detection output and ground truth, respectively. The detection output boxes are classified as TP, FP, or FN based on their overlap with the ground truth. In this particular image, there are ten ground truth boxes and ten detection output boxes. Boxes marked as TP1 through TP8 have an overlap of more than 50% with their corresponding ground truth boxes. However, the detection output for boxes marked TP9 and FN1 covers two ground truth boxes, resulting in one being classified as TP and the other as FN. Box FP1 is correctly detected but classified as FP due to the lack of a corresponding ground truth box, highlighting the issue of faulty annotation.

TABLE III
PERFORMANCE OF TREE DETECTION

	Tree					
	YOLOv4			YOLOv5		
Epoch (10^3)	Precision	Recall	mAP	Precision	Recall	mAP
5	0.7633	0.7387	0.7668	0.9956	0.5022	0.9553
10	0.7857	0.8871	0.8651	0.9935	0.5195	0.9705
15	**0.8000**	**0.9032**	**0.8886**	**0.9961**	**0.5180**	**0.9779**
20	0.7807	0.9419	0.8873	0.9954	0.5031	0.9748

TABLE IV
PERFORMANCE OF FRUITLET DETECTION

	Fruitlet					
	YOLOv4			YOLOv5		
Epoch (10^3)	Precision	Recall	mAP	Precision	Recall	mAP
5	0.7710	0.8806	0.8163	0.8441	0.7705	0.7187
10	**0.8397**	**0.8808**	**0.8545**	0.8642	0.7455	0.7216
15	0.8370	0.8676	0.8443	**0.8391**	**0.7899**	**0.7221**
20	0.8363	0.8712	0.8481	0.8210	0.7213	0.7219

Fig. 12. Sample output of false positive, where the frond detected as tree

Detecting trees accurately in oil palm plantations is a challenging task due to several factors. One of the most significant challenges is the presence of dangling fronds, which can be mistaken for trees by the detection model. In our dataset, all oil palm trees have trunks that are completely covered in green grass, making them appear similar in size, shape, and color. As a result, the detection model can receive false alarms. Fig. 12 displays a sample false detection, where the black box represents a falsely detected tree (FP) outside the ROI, while the white bounding box indicates a tree within the ROI being tracked. Once the tree is located and tracked, the loose fruitlet detection model is enabled. For the tracked tree in this image, no fruitlets were found. The algorithm can also detect blurred tree trunks, partially covered trees with many leaves, and trees of varying sizes and heights, as shown in Figs. 13(a) and (b).

The performance of the tree and loose fruitlet detection models, trained on various numbers of epochs on the test datasets, are reported in Table III and Table IV, respectively. Our findings indicate that the YOLOv5 model exhibited superior performance in detecting trees than YOLOv4. Despite displaying lower recall values than YOLOv4, YOLOv5 consistently showed higher precision

(a)

(b)

Fig. 13. Sample output of true positive, even the image is (a) blurred and (b) the tree is partially covered by frond and full of leaves with different heights.

values, indicating fewer false detections. YOLOv5 also demonstrated exceptionally high mean Average Precision (mAP) scores, with a maximum of 97.79% achieved at 15,000 epochs. This may be due to the changes made to the model's box selection process, allowing it to learn the anchor box's size and shape that best matches the dataset. In contrast, YOLOv4 outperformed YOLOv5 in detecting loose fruitlets, with a mAP of 83.97% (epoch = 10x103), precision of 83.97%, and recall of 88.08%. This indicates that the YOLOv4 model is more effective at identifying loose fruitlets in our dataset.

Fig. 14. Consequences of frames show the inconsistent detection

The accuracy of fruitlet detection is challenging due to the small size of the fruitlets, which may be partially or completely obstructed by grass or other contaminants. False negatives may occur as a result. The square bounding box of the fruitlets and YOLOv4's superior performance with small objects may contribute to the observed results. Additionally, the fruitlet's color (orange to red for ripe fruitlets) is significant and can be influenced by the background, which can be brown for the ground or green for the grass. To improve the detection model, high-definition cameras may be utilized to capture crisper and more detailed images of the fruitlets. However, high accuracy does not necessarily ensure consistent object detection across different frames. The performance of object detection consistency was found to be 91.13%. As illustrated in Figure 14, the figure demonstrates cases where object detection was inconsistent, as evidenced by the presence of detected tree in both the first and third images, but it failed to detect a tree in the second image despite its clear presence. In most cases, the tree trunk's color and features were indistinguishable from the background (white circled shape in Fig. 14). Additionally, excessive leaves on the trunk can make it challenging to recognize whether it is a trunk or leaves.

B. Ready-to-harvest Classification Results

Fig. 15 displays sample images of oil palm trees with less than three loose fruitlets in the vicinity of the tree trunk, resulting in a three-point counting threshold that categorizes them as not harvest-ready. In contrast, Fig. 16 shows images of oil palm trees with more than three loose fruitlets on the ground, resulting in a categorization of harvest-ready. Tables V and VI report the ready-to-harvest classification results for thresholds zero and three, respectively, along with the corresponding confusion matrices. For threshold zero, a tree is deemed ready for harvesting if at least one loose fruitlet is detected in its vicinity. The ready-to-harvest classification accuracy for threshold zero is 90.48%, as reported in Table V.

TABLE V
CONFUSION MATRIX FOR HARVEST-READY CLASSIFICATION TESTING (HARVEST-READY THRESHOLD IS SET AS 0)

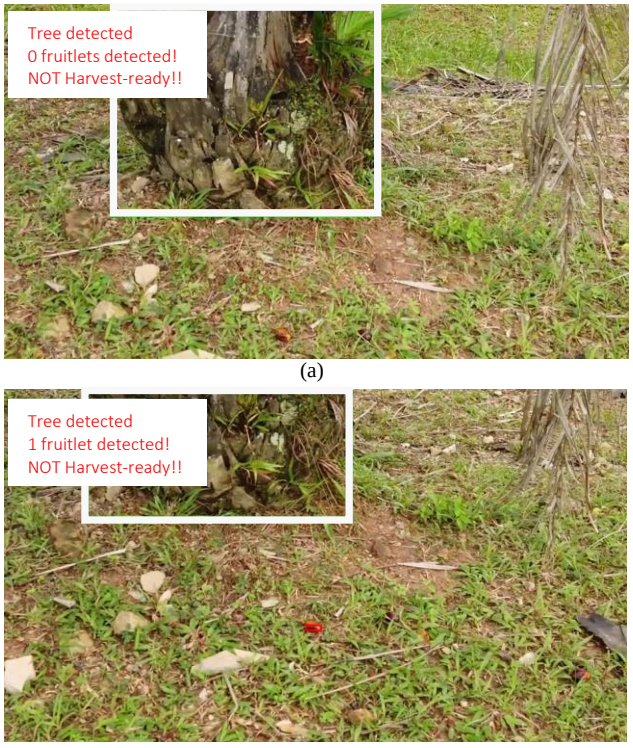
	Predicted: not harvest-ready	**Predicted: harvest-ready**
Actual: not harvest-ready	161	12
Actual: harvest-ready	8	50

TABLE VI
CONFUSION MATRIX FOR HARVEST-READY CLASSIFICATION TESTING (HARVEST-READY THRESHOLD IS SET AS 3)

	Predicted: not harvest-ready	**Predicted: harvest-ready**
Actual: not harvest-ready	161	12
Actual: harvest-ready	8	50

Ready-to-harvest classification accuracy (threshold zero):

$$\frac{TP+TN}{TP+TN+FP+FN}=90.48\% \qquad (4)$$



(a)

(b)

Fig. 15. Sample output detected as not harvest-ready trees

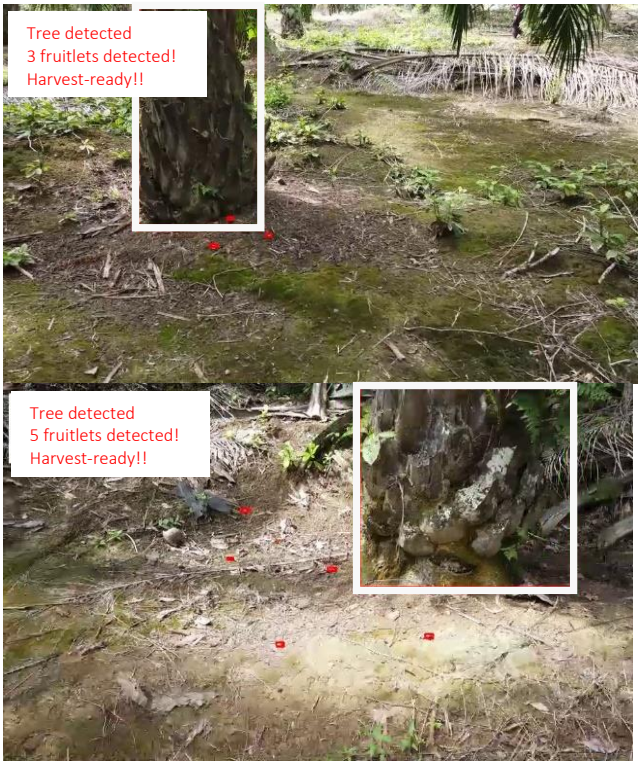


Fig. 16. Sample output detected as harvest-ready trees

On the other hand, for the threshold set at three, there must be at least four loose fruitlets detected on the ground within the proximity of a particular tree before only the tree is considered ready-to-harvest. Based on the matrix in Table VI, the accuracy of the ready-to-harvest classification for threshold three is 91.34% (in (3)).

Ready-to-harvest classification accuracy (threshold three):

$$\frac{TP+TN}{TP+TN+FP+FN}=91.34\% \qquad (5)$$

C. Limitation & Future Work

This study presents a method for predicting the harvest-ready status of fresh fruit bunches (FFBs) based on the count of loose fruitlets on the ground. The approach utilizes the YOLO (You Only Look Once) model to detect palm trees and loose fruitlets within the region of interest (ROI) of the tree, allowing for the harvest-ready status to be deduced. The deep-learning method was chosen due to its high level of accuracy and speed of detection. However, it's important to note that the detection results are dependent on both the trained detection model and the training dataset. When the model is applied to a new environment different from the training datasets, it requires retraining to account for the new knowledge. Additionally, the detection model's effectiveness is contingent on the visibility of at least 80% of the loose fruitlets on the ground within the camera's view and the minimum size of the fruitlet in the image being at least 25x25 pixels from the image resolution of 1920x1080. These limitations notwithstanding, the proposed method offers significant potential for improving the harvest readiness assessment process in oil palm plantations.

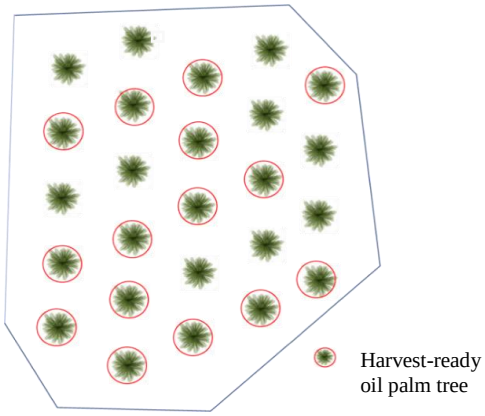


Fig. 17. Sample virtual 2D map of oil palm tree tagging

For future work, the output of the proposed prediction algorithm can be utilized to generate a virtual 2D/3D map that displays the location of ready-to-harvest trees. To assist with harvest resource planning, a tree marking module will be developed by integrating the recognition output into a 3D map that represents the plantation area under investigation. Fig. 17 provides an example of a virtual 2D map of oil palm tree markings based on harvest readiness status. The integration of the proposed virtual map into farm management practices may lead to more efficient harvesting cycles. By identifying and marking the ready-to-harvest trees in the plantation, the resource planning process can be optimized, leading to increased productivity and lower costs. Moreover, during harvesting operations, the virtual map can serve as a valuable tool for guiding workers to the correct trees, further improving the efficiency of the process.

IV. Conclusion

In this study, we propose a novel machine vision-based prediction algorithm for determining the harvest-ready status of fresh fruit bunches (FFBs) in oil palm plantations by counting loose fruitlets on the ground. Our approach involves using YOLOv5 and YOLOv4 models to detect trees and fruitlets, respectively, based on their performance and accuracy. The results of our experiments demonstrate that YOLOv5 outperforms YOLOv4 in tree detection, achieving a mean average precision (mAP) of 97.9%, while YOLOv4 is more accurate in detecting small loose fruitlets, with mAP of 85.45%. Moreover, our object detection consistency findings confirm that the object detection and tracker algorithms are robust. We also discuss the challenges encountered in object detection. We evaluate the accuracy of the ready-to-harvest classification algorithm on 185 video clips, achieving 90.48% and 91.34% accuracy for count thresholds of zero and three, respectively. However, to further improve the prediction and object detection models, we recommend that more training data be collected to account for environmental variability, including extreme lighting conditions and varying tree arrangements in the plantation area. We also suggest the use of high-definition cameras to improve image quality. Overall, our results demonstrate the potential for implementing our proposed

algorithm in oil palm plantations to enhance resource planning and operational efficiency.

REFERENCES

[1] M. H. Junos, A. S. M. Khairuddin, S. Thannirmalai, M. Dahari, "An optimized YOLO-based object detection model for crop harvesting system," IET Image Processing, vol. 15, no. 9, pp 2112-2125, 2021.

[2] R. Hannah, and R. Max, "Forests and Deforestation," Accessed on: Jan. 3, 2022, [Online] Available: https://ourworldindata.org/forests-and-deforestation

[3] Malaysian Palm Oil Council, "Monthly Palm Oil Trade Statistics 2020," Accessed on: Jan. 4, 2022, [Online] Available: https://mpoc.org.my/monthly-palm-oil-trade-statistics-2020/

[4] G. K. A. Parveez, et al., "Oil palm economic performance in Malaysia and R&D progress in 2020," J. Oil Palm Res, vol. 33, no. 2, 2021.

[5] M. O. Lawal, "Tomato detection based on modified YOLOv3 framework," Scientific Reports, vol. 11, no. 1, pp 1-11, 2021.

[6] O. M. Lawal, "YOLO Muskmelon: quest for fruit detection speed and accuracy using deep learning," IEEE Access, vol. 9, pp 15221-15227, 2021.

[7] Y. Yu, K. Zhang, L. Yang, & D. Zhang, "Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN," Computers and Electronics in Agriculture, vol. 163, pp 104846, 2019.

[8] H. Mirhaji, M. Soleymani, A. Asakereh, & S. A. Mehdizadeh, "Fruit detection and load estimation of an orange orchard using the YOLO models through simple approaches in different imaging and illumination conditions," Computers and Electronics in Agriculture, vol. 191, pp 106533, 2021.

[9] A. I. B. Parico, & T. Ahamed, "Real Time Pear Fruit Detection and Counting Using YOLOv4 Models and Deep SORT," Sensors, vol. 21 no.14, pp 4803, 2021.

[10] C. B. MacEachern, T. J. Esau, A. W. Schumann, P. J. Hennessy, Q. U. Zaman, "Deep Learning Artificial Neural Networks for Detection of Fruit Maturity Stage in Wild Blueberries," In 2021 ASABE Annual International Virtual Meeting, American Society of Agricultural and Biological Engineers, pp 1, 2021.

[11] I. Sa, et al., "Deepfruits: A fruit detection system using deep neural networks," Sensors, vol. 16, no. 8, pp 1222, 2016.

[12] D. Cherie, R. Rini & M. Makky, "Determination of the optimum harvest window and quality attributes of oil palm fresh fruit bunch using non-destructive shortwave infrared spectroscopy," AIP Conference Proceedings, vol. 2155, no. 1, pp 020034, 2019.

[13] N. Rajanaidu, A. Ariffin, B. Wood, E. S. Sarjit, "Ripeness Standards and Harvesting Criteria for Oil Palm Bunches," Proceeding of International Oil Palm Conference Agriculture," pp. 224-230, 1988.

[14] Z. B. M. Sharif, et al., "Study on handing process and quality degradation of oil palm fresh fruit bunches (FFB)," In IOP Conference Series: Materials Science and Engineering, vol. 203, no. 1, pp. 012027, 2017.

[15] A. A. Abdul, & Y. A. Tan, "The effects of handling of oil palm FFB on the formation of FFA and the subsequent quality of crude palm oil," In J. Sukaimi (Ed.) Proceedings Palm oil Development Conference, pp 482-485, 1989.

[16] L. S. Woittiez, et al., "Smallholder oil palm handbook," Wageningen University and SNV Netherlands Development Organisation. Accessed on: Jan. 15, 2022, [Online] Available: http://akvopedia.org/wiki/Sustainable_Oil_Palm_Farming

[17] S. Ghazalli, et al, "Image analysis techniques for ripeness detection of palm oil fresh fruit bunches," ELEKTRIKA-Journal of Electrical Engineering, vol. 18, no. 3, pp 57-62, 2019.

[18] M. Alfatni, et al, "Colour feature extraction techniques for real time system of oil palm fresh fruit bunch maturity grading," IOP Conference Series: Earth and Environmental Science, vol. 540, no. 1, pp. 012092, 2020.

[19] O. Bensaeed, et al., "Oil palm fruit grading using a hyperspectral device and machine learning algorithm," IOP Conference Series: Earth and Environmental Science, vol. 20, no. 1, pp 012017, 2014.

[20] D. Silalahi, et al., "Near infrared spectroscopy: a rapid and non-destructive technique to assess the ripeness of oil palm (Elaeis guineensis jacq.) fresh fruit," Journal of Near Infrared Spectroscopy, vol. 24, no. 2, pp 179-190, 2016.

[21] F. Hashim, et al., "A rapid and non-destructive technique in determining the ripeness of oil palm fresh fruit bunch (FFB)," Jurnal Kejuruteraan, vol. 30, no. 1, pp 93-101, 2018.

[22] M. Arokiasamy, "Investigation into the oil content of oil palm fruit bunches," Proceeding of Malaysian Oil Palm Conference, (P.D.Turner, Ed). Incorporated Society of Planter, Kuala Lumpur, pp 136-140 (1969).

[23] D. Kumaradevan, et al., "Optimising the operational parameters of a spherical steriliser for the treatment of oil palm fresh fruit bunch," IOP Conference Series: Materials Science and Engineering, vol. 88, no. 1, pp 012031, 2015.

[24] S. H. Norsani, "Application of ahp for determining the best of palm oil fresh fruit bunch," M.S. thesis, Dept. Manufacturing Eng., Malaysia Uni. Tech., Melaka, Malaysia, 2017.

[25] P. Junkwon, et al., "Potential application of color and hyperspectral images for estimation of weight and ripeness of oil palm (Elaeis guineensis Jacq. var. tenera)," Agriculture Information Research, vol. 18, no. 2, pp 72-81, 2009.

[26] Susilawati & Supijatno, "Palm Oil and Palm Oil Mill Waste Management (Pengolahan kelapa sawit dan pengelolaan limbah pabrik kelapa sawit)," Agrohorti Bulletin, vol. 3, no. 2, 1997.

[27] MPOB, "Oil Palm Grading Manual," second edition. Malaysian Palm Oil Board (MPOB), Malaysia, 2003.

[28] E. A. Ghani, Z. Z. Zakaria, M. B. Basri, "Perusahaan sawit di Malaysia: Satu Panduan," Lembaga Minyak Sawit Malaysia, 2004.

[29] A. Bochkovskiy, C. Y. Wang, H. Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," arXiv preprint arXiv:2004.10934, 2020.

[30] Ultralytics. yolov5. Accessed on: May. 18, 2022, [Online] Available: https://github.com/ultralytics/yolov5

[31] D. Thuan, "Evolution of yolo algorithm and yolov5: the state-of-the-art object detection algorithm" BS.c thesis, Dept. Inf. Tech., Ooulu Uni. Applied Sci., Ooulu, Finland, (2021).

[32] P. Mahto, et al., "Refining Yolov4 for vehicle detection," International Journal of Advanced Research in Engineering and Technology (IJARET), vol. 11, no. 5, 2020.

[33] C. Tung, et al., "Why Accuracy Is Not Enough: The Need for Consistency in Object Detection," IEEE MultiMedia, 2022.

[34] Cvat.ai Corporation, "Open Data Annotation Platform," Accessed on: June. 3, 2021, [Online] Available: https://www.cvat.ai/

[35] M. A. Sahal, "Comparative Analysis of Yolov3, Yolov4 and Yolov5 for Sign Language Detection," IJARIIE, vol. 7, no. 4, pp 2395 – 4396, 2021.

[36] A. Ramya, et. al, "Comparison of YOLOv3, YOLOv4 and YOLOv5 Performance for Detection of Blood Cells," International Research Journal of Engineering and Technology (IRJET), vol. 8, no. 4, pp 4225 – 4229, 2021.