IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 4 June 2023, accepted 9 June 2023, date of publication 13 June 2023, date of current version 20 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3285537

RESEARCH ARTICLE

Real-Time Oil Palm Fruit Grading System Using Smartphone and Modified YOLOv4

SUHARJITO[1], (Member, IEEE), MUHAMMAD ASROL[1], DITDIT NUGERAHA UTAMA[2], FRANZ ADETA JUNIOR[3], AND MARIMIN[4], (Member, IEEE)

[1]Industrial Engineering Department, BINUS Graduate Program-Master of Industrial Engineering, Bina Nusantara University, Jakarta 11480, Indonesia
[2]Computer Science Department, BINUS Graduate Program-Master of Computer Science, Bina Nusantara University, Jakarta 10480, Indonesia
[3]Cyber Security Program, School of Computer Science, Bina Nusantara University, Jakarta 11480, Indonesia
[4]Agro-Industrial Technology Department, Faculty of Agricultural Engineering and Technology, Bogor Agricultural University, Bogor, West Java 16680, Indonesia

Corresponding author: Suharjito (suharjito@binus.edu)

This work was supported in part by the Ministry of Education, Culture, Research, and Technology in Indonesia; and in part by the Directorate General of Higher Education under Contract 155/E5/PG.02.00.PT/2022.

ABSTRACT The classification of the ripeness degree of oil palm fruit has attracted the attention of numerous researchers. However, there are still many challenges due to constraints in the dataset, methodologies used, and variations in the use of data categories. Detecting oil palm fruit bunches accurately is crucial, given their complex shape and characteristics, particularly when different ripeness categories are present in a pile of oil palm. Most studies utilize oil palm images or the color spectrum of oil palm fruit to classify the level of ripeness. However, these methods are not real-time and lack efficiency. This study proposes a real-time model for determining the ripeness degree of oil palm using a smartphone and video data as input, incorporating modifications to the object detection approach. The research process involves collecting videos of palm oil piles using smartphones in the grading area of the palm oil industry. The videos are then pre-processed and labelled for the object detection and classification process. A detection and classification model is developed using the YOLOv4 approach with several performance improvements, enabling implementation on smartphones. The best-performing model is tested for detecting and classifying the ripeness of fresh fruit bunches using an android-based smartphone. The testing results, based on the mAP value, demonstrate that the YOLOv4 model with 16 quantization performs 12% better than YOLOv4 Tiny. Based on the test results at the grading location, this model can efficiently detect fruit bunches that do not meet the quality standards.

INDEX TERMS Hyperparameter tunning, oil palm ripeness, real-time detection, modified YOLOv4.

I. INTRODUCTION

The demand for palm oil continues to increase due to its usage in various oleochemical industries and its cost-effectiveness compared to other vegetable oils [1]. In Indonesia, oil palm is extensively cultivated as one of the major vegetable oil-producing plants. This plant thrives in the country's natural environment and has significant potential to enhance social welfare and economic development. Consequently, oil palm plantations in Indonesia have been rapidly expanding, making the country the world's leading palm oil producer, with an annual production of 45.6 million tons [2]. However, with this increased production, it becomes imperative to focus on improving quality to generate higher added value.

The quality of crude palm oil is greatly influenced by the content of free fatty acids (FFA) present in oil palm. Achieving the appropriate level of ripeness during the harvesting process plays a crucial role in determining the FFA content [3]. Therefore, it is essential to classify the ripeness level during harvest and evaluate the degree of ripeness in oil palm fresh fruit bunches (FFB) at the palm oil mill to maintain oil quality. However, the current practice of manual grading by human graders has yielded inconsistent results. To address this issue, this study proposes an automation process using a smartphone approach and object detection method.

The associate editor coordinating the review of this manuscript and approving it for publication was Jiachen Yang.

Differences in perception often occur in classification, especially in fruit and plant ripeness classifications. Therefore, there is significant research focusing on the development of the Deep Learning approach using the Convolutional Neural Network (CNN) method. CNN is an artificial neural network model that can learn and be trained iteratively to achieve a high level of accuracy in detection and classification [4]. Currently, research utilizing CNN benefits from easy access to large datasets, enabling the achievement of exceptional accuracy and reliable classification outcomes. CNN has been widely employed to detect various fruit and plant ripeness levels [5], identify species [6], and classify fruit types [7]. However, most of these studies have primarily used images rather than video data, relying on object detection for real-time classification and detection.

A study on the ripeness classification of oil palm fruit using a computer vision approach has been conducted. For example, [8] captured images of fresh oil palm fruit bunches and extracted color information to compare with machine vision results. Another approach utilized an Artificial Neural Network by [9], where two types of models were employed: one using all the features and the other using selected features obtained through Principal Component Analysis. In addition, [10] employed image histogram processing to convert RGB (Red, Green, Blue) colors in an image into a single gray element, and compared it to a database using mean, skewness, kurtosis, and entropy approaches for feature extraction. Although these studies were able to predict the ripeness level of oil palm, real-time sorting based on ripeness levels is not feasible as the models can only recognize still images as input.

In addition, a study conducted by [11] successfully employed various methods to classify the ripeness of oil palm fresh fruit bunches (FFB) using deep learning approaches. The study demonstrated that transfer learning from a pre-trained model (AlexNet) yielded better results compared to other classification methods. Another study [12] utilized deep learning and visual attention techniques for oil palm fruit ripeness classification. The research focused on developing models with pre-trained CNN architectures, specifically AlexNet, Squeeze and Excitation-Densely Connected Convolutional Network (SE-DenseNet), RestAtt DenseNet, and Sigmoid DenseNet, to categorize the degree of ripeness in oil palm fresh fruit bunches. Similarly, a study by [13] proposed a model for identifying the ripeness of oil palm fruit using the Convolutional Neural Network (CNN) method, specifically AlexNet. This research aimed to implement a system using a Low-Cost Processor by developing software in Mathlab, which would later be converted into Python Programming compatible with the Tinker Board.

With the recent advancements in deep learning technology, several lightweight deep learning models have been developed that do not require high computational resources but still provide accurate object classification. Examples of such models include MobileNet [14], MobileNetV2 [15], EfficientNet [16], and MNasNet [17], which are designed for resource-efficient mobile devices using automated neural architecture search. In the context of classifying the ripeness level of oil palm FFB, research has been proposed to utilize mobile devices and lightweight deep-learning models [18]. By quantifying the classification model and deploying it to a mobile device, this research successfully applied ripeness-level classification using low-cost computing resources. However, the focus of the research is mainly on augmentation techniques rather than the model's real-world applicability. The data used in the research consists of still images, which are suitable for classification purposes. Since the research does not employ object detection methods, the model is unable to detect the ripeness level of multiple objects in each video frame or perform real-time analysis. This limits the efficiency in determining the ripeness of oil palm. In contrast, our research aims to develop a method for detecting and classifying the ripeness of oil palm using a smartphone and an object detection approach, providing reliable speed, high accuracy, and low costs for users in the palm oil industry.

To enable quick and real-time object detection, video cameras are commonly used as input devices. Deep learning techniques, such as the faster R-CNN (Region-based Convolutional Neural Network) method [19] and the SSD (Single Shot Detector) method [20] have been applied to real-time detection of protective helmet usage in surveillance cameras using video input. Object detection methods in the field of computer vision can generally be categorized into one-stage and two-stage methods. One-stage methods include YOLO (You Only Look Once) [21] and SSD [22], while two-stage methods include Fast/Faster R-CNN [23] and R-FCN (Region-based Fully Convolutional Network) [24]. Single-stage methods are known for their processing speed and accuracy, making them suitable for applications requiring fast and lightweight computation. The SSD-Mobilenet-v1 [25], a single-stage object detector, performs well in object detection tasks. Based on this, our research aims to develop a model for object detection of oil palm FFB and subsequently classify their ripeness in a fast and real-time manner using a one-stage detection approach and deep learning techniques. This model can be implemented in mobile applications for practical use.

Previous research has explored the detection of tomato ripeness using YOLOv4 [26] with the DarkNet53 backbone. This approach proved to be effective in detecting ripeness based on color changes in the fruit skin, outperforming YOLOv3 and R-CNN. Other studies focused on automatic calculations of pear ripeness using RGB data in mobile applications, utilizing various YOLOv4 [27] and YOLOv4-Tiny [28] models. Additionally, the automatic detection of cherry ripeness using circular bounding boxes was conducted using various YOLOv3 and YOLOv4 models [29]. In the context of apple ripeness detection implemented in a picking robot, YOLOv4 with different backbones was employed [30]. These

studies demonstrate the successful use of YOLOv4 in fruit ripeness detection.

Furthermore, research has proposed the use of YOLO for detecting the ripeness of oil palm fruit. One study collected fruit data through photographs captured by an unmanned aerial vehicle (UAV) that fell from an oil palm tree [31]. Another study focused on smartphone-based plant pest detection using YOLO, comparing it with other deep learning methods such as Faster R-CNNs and SSDs [32]. YOLOv4 demonstrated superior computational speed and compact size, making it well-suited for real-time detection, albeit with a slight decrease in accuracy. To address accuracy concerns, high-quality datasets were used in these studies. However, most of these studies employed single-object images for training and testing. In contrast, this study utilizes image data with multiple objects obtained from videos recorded directly at the palm oil mill's grading section. Real-time detection faces challenges posed by dynamic environmental factors such as lighting, shadows, and obstructed objects, hence the use of video data. Videos consist of connected image frames, resembling real-time interactions. The video dataset of oil palm FFB stacks from the grading area of a palm oil mill served as the dataset for this study [33].

According to previous research, the focus on applying real-time detection in a cost-effective and reliable manner has been lacking in most studies on the detection and classification of oil palm fruit ripeness. This study aims to address this gap by leveraging mobile phones to enhance flexibility and efficiency in ripeness detection. The technology developed in this study can be utilized not only in palm oil mills but also in oil palm fields at an affordable cost. The contributions of this study are as follows: (1) Utilizing a complex dataset consisting of videos capturing oil palm FFB piles in outdoor settings, collected from smartphones. The dataset encompasses six categories of oil palm fruit ripeness: empty bunches, unripe, underripe, ripe, overripe, and abnormal. (2) Developing a ripeness level detection application implemented on a smartphone, capable of classifying fresh palm fruit bunches into six quality classes. (3) Employing the YOLOv4 approach for ripeness level detection, incorporating various model improvements such as data augmentation, hyperparameter tuning, and model quantization. These enhancements enable the model to be deployed on efficient smartphones. (4) Training the oil palm fruit ripeness detection model using a combination of various ripeness levels, ensuring its applicability in real-world conditions during the FFB grading process prior to entering the palm oil mill.

II. MATERIALS AND METHOD

A. DATASET

In this study, video data was collected using a smartphone during the grading process at a palm oil mill in Central Kalimantan Province, Indonesia [33]. The data collected is in the form of videos measuring 1280 × 720 pixels in .mp4 format. The video of the piles of oil palm FFB is classified into six categories of ripeness of oil palm FFB, namely unripe, underripe, overripe, abnormal, and empty bunches. The oil palm FFB ripeness category is also adjusted to the conditions in the field and confirmation from practitioners and experts in the palm oil mill grading section. The videos collected comprised 56 multi-category FFB videos and 45 single-category FFB videos. Examples of collected data sets can be explained in Figures 1 and Figure 2. It can be seen in Figure 1 that each image has a single category of ripeness. This image was generated from videos in a single category using frame extraction. The duration of each video is 10–15 seconds. Based on frame extraction, each video can generate 30 images (Figure 1). On the other hand, Figure 2 shows that each image has an FFB pile for more than one category. This dataset was generated based on videos of multi-category oil palm FFB with a duration of 10–15 seconds. Using frame extraction, a dataset of image frames can be generated, as shown in Figure 2.

B. RESEARCH STAGES

This research stage can be explained in Figure 3. The research is divided into 6 stages: the data collection stage, the pre-processing data stage, the ripeness detection stage, the classification model development stage, the model conversion stage to mobile applications, the application development stage on smartphones, and the application evaluation stage by users. In the early stages, a dataset was collected using a smartphone for each stack of FFB by circling it for 10–15 seconds to get images of each blunt FFB from various angles. The video data that has been collected is pre-processed. The pre-processing stage consists of image framing, labeling, augmentation, and resolution resizing. The dataset has been pre-processed and divided into three parts: training data, validation data, and testing data, with a ratio of 7:2:1. For the creation of deep learning models, this data separation is required. In building the deep learning model, hyperparameter tuning is carried out to get the best model for detecting and classifying FFB piles. The deep learning model for FFB ripeness grading was developed through the YOLOv4 Family. In developing the model, several aspects will be considered, including training models, validating models, tuning models, and performance models. The best-performing model is then implemented in a mobile application. To be able to implement the best model, quantification of the model must first be carried out using TensorFlow Lite. After the model can be implemented on a smartphone, the last step is to test the model at the palm oil mill grading location by the user to get the model performance according to the conditions in the field.

C. DATA PRE-PROCESSING

At the pre-processing data stage, data in the form of video is converted into image frames using VLC Player with a recording ratio of 10. For videos with 30 frames per second (fps), a recording ratio of 10 means that it will take pic-

FIGURE 1. Dataset of the piles of oil palm FFB in single class category per image.

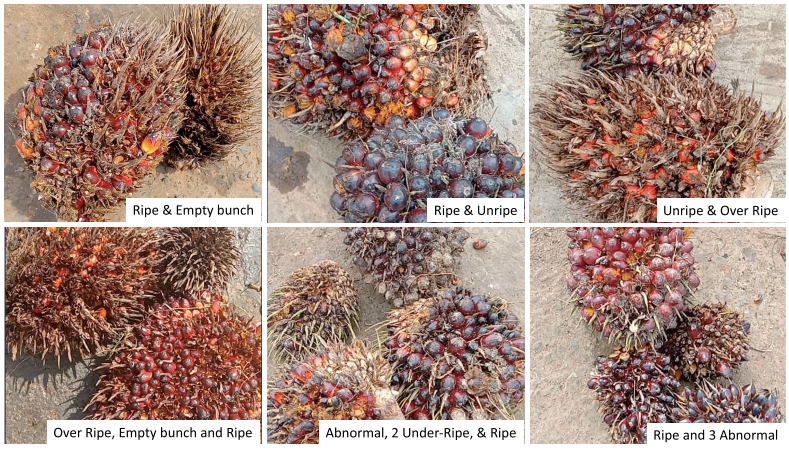

FIGURE 2. Dataset of multi categories per image of oil palm FFB piles.

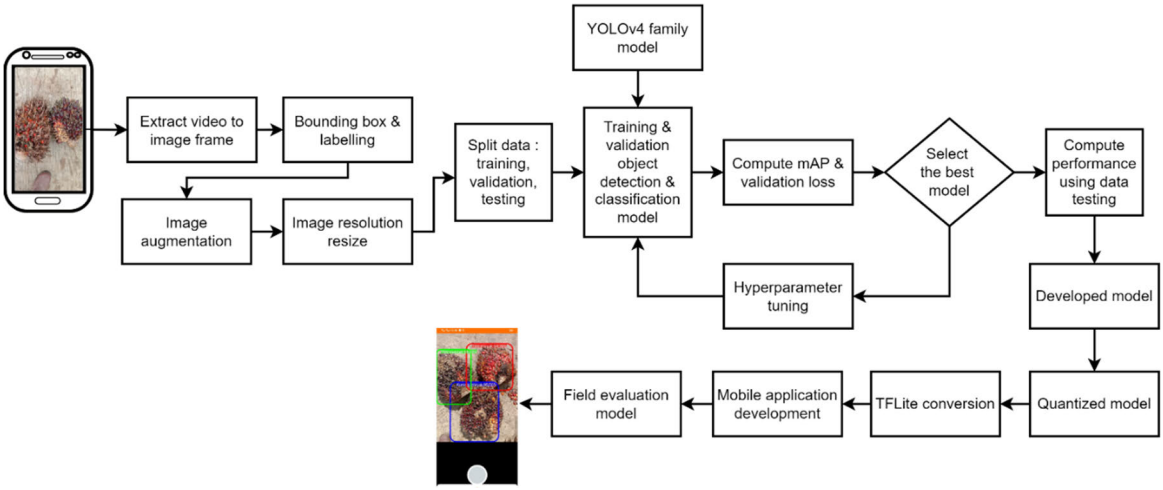

FIGURE 3. The research stages diagram.

tures every 10/30 or 0.33 seconds. If the video duration is 15 seconds, approximately 45 images will be produced. The images are then uploaded to supervise.ly to be annotated and downloaded in image + txt file format. After that, the data in the form of a file is then uploaded to the Roboflow application to be used as a training dataset.

The data in the form of annotated image files are then divided into training, testing, and validation datasets for developing a grading model for the ripeness level of oil palm FFB. Figure 4 shows an illustration of the processing of the oil palm video dataset that will be used as the input model. The process begins with preparing video data, extracting the

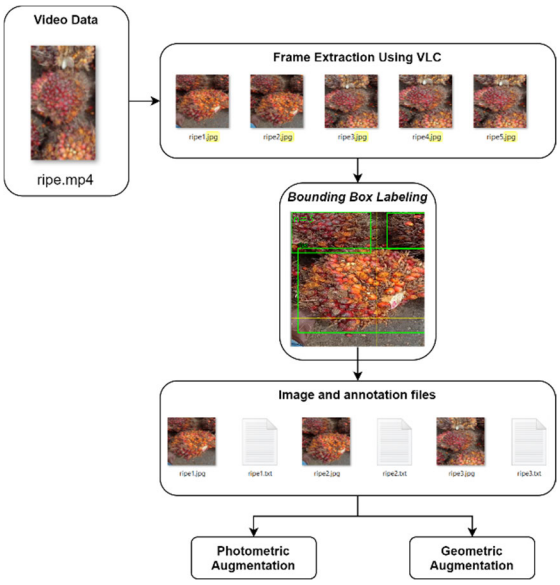

FIGURE 4. The illustration of data video data pre-processing stages.

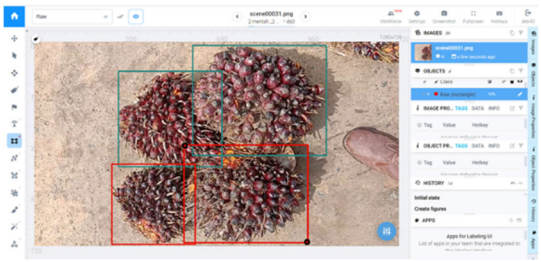

FIGURE 5. An Illustration of labelling process and object bounding box using RoboFlow.

image frame to produce a sequential image. The next process is to provide a bounding box to mark the boundaries of the objects, namely oil palm FFB, and annotate them according to their ripeness level class. After that, an annotation file is obtained, which contains the coordinates of the FFB bounding box and its ripeness level class. Each image will have a file with the .txt extension to store the bounding box annotation results and the oil palm ripeness level in the image.

The dataset is labeled by creating a bounding box for each image of the pile of oil palm fruit bunches using the RoboFlow application to conform to the annotation format of the YOLO model. This labeling process is carried out by making a bounding box for each image of oil palm bunches and labeling it according to a predetermined level of ripeness as shown in Figure 5. The bounding box used is in the form of a square with 4 coordinate points and has an annotation in the form of the class name of the FFB ripeness level of each object contained in the bounding box. Images that have been labeled and bound have an annotation containing a class id and bounding box point coordinates. Then the data augmentation process is done by adding variations to the dataset. The types of augmentation used are photometric and geometric. This type of data augmentation using photometric and geometric transformations is a method commonly used in the agricultural sector [34]. The geometric augmentation uses a rotation of 45° and 90° and a translation with a ratio of 0.5. Then, the photometric augmentation uses random brightness with a range of −40% to 60% and Gaussian blur with a variance value of 7% to 9%.

D. ARCHITECTURE OF THE PROPOSED MODEL

At the development stage, the detection and classification model with YOLOv4 was carried out based on 2 different architectures, namely YOLOv4-CSPDarknet53 [35] and YOLOv4-Tiny [36], with some modifications in the hyperparameters used. As shown in Figure 6, there are 3 main parts in the YOLOv4 model, namely the Backbone layer, Neck layer, and Head layer. The Backbone layer between the YOLOv4-CSPDarknet53 and YOLOv4-Tiny series can be distinguished by the number of CSP blocks used [35]. YOLOv4-CSPDarknet53 has 5 CSP blocks, which can be called CSPDarknet53, and in YOLOv4-Tiny series, there are 3 CSP blocks, which can be referred to as CSPDarknet53-Tiny. By concatenating input features from the base layer with input features that have undergone convolutional processing, the CSPNet architecture [36] and Darknet53 [37] combined to create CSPDarknet53, which can solve the vanishing gradient problem. On the YOLO architecture (Head layer), CBL is made up of Convolution-Batch Normalization-LeakyReLu, which has the dual purposes of normalizing the output and preventing the neuron's output from turning 0 due to dying ReLU, which would cause the neuron to cease learning. Convolutional-Batch Normalization-Max Pooling is the block symbol for CBM. This block is responsible for capturing global features in the input data. CSP allows the architecture to get a wider variety of features to improve model performance. Moreover, unlike the YOLOv4-Tiny series does not have an SPP layer, the YOLOv4-CSPDarknet features one for pooling using spatial bins with three distinct dimensional scales. SPP can shorten repeated convolution calculations and increase the detector's accuracy [38]. At the neck, all types of architectures use PANet to perform feature aggregation. PANet can capture different feature perspectives and can perform aggregation from each level of the feature piece [39]. The head section uses YOLOv3 as a detector to detect images from 3 different scales. YOLOv4-CSPDarknet53 uses a scale of $52 \times 52 \times 3$, $26 \times 26 \times 33$, and $13 \times 13 \times 33$. Meanwhile, YOLOv4-Tiny uses a scale of $13 \times 13 \times 33$ and $26 \times 26 \times 33$. Both YOLOv4-CSPDarknet53 and YOLOv4-Tiny employ the same number of layers.

E. TRAINING MODEL

The model training process uses the framework from DarkNet and is carried out in the Google Colaboratory. Nvidia Tesla P100 using 54.8 GB of RAM and CUDA 11.2 are the hardware specifications utilized.

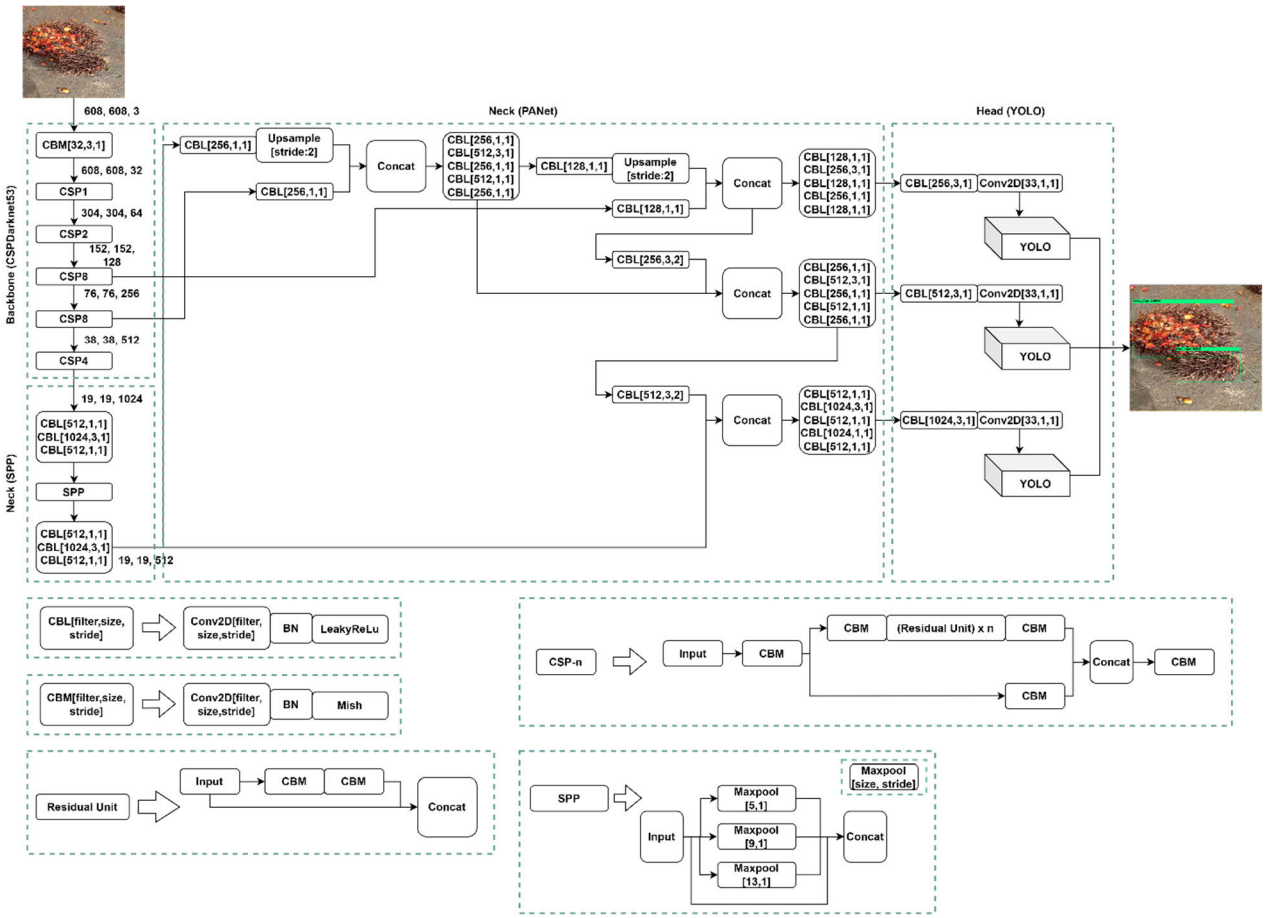

FIGURE 6. The architecture of the proposed model.

TABLE 1. Hyperparameter tuning of YOLOv4 model.

	YOLOv4-320	YOLOv4-416	YOLOv4-512	YOLOv4-608	YOLOv4-CSP-512	YOLOv4-CSP-640
Height	320	416	512	608	512	640
Width	320	416	512	608	512	640
Subdivisions	16	16	16	32	16	16

The design employed has a maximum bath size of 12,000, which was determined by using equation (1).

$$Max\ Batches = total\ class.2000 \tag{1}$$

The learning rate scheduler is optimized using steps with a scale of .1, which results in the original learning rate being multiplied by 0.1 at the 9600th and 10800th stages. The momentum used is 0.949 on YOLOv4-CSPDarknet53 and YOLOv4 series. The decay used is 0.0005 on all architectures. The batch size used is 64. The main distinction is the value of the input resolution employed. A model with the best detectability can be obtained by experimenting with different resolutions. The used learning rate is 0.001. All of the hyperparameters used are the result of optimization, particularly in scenarios with varying input resolution values. Because transfer learning has been proven to boost accuracy in comparison to training without it [40], the model was trained using this technique using an initial weight that had been developed using a dataset from MS-COCO (Microsoft Common Objects in Context).

F. OPTIMIZATION MODEL

In this study, YOLOv4 optimizes the model using the Bag of Freebies (Bof) and the Bag of Specials techniques (BoS). BoS is a method to considerably increase the accuracy of object identification by just minimally raising the inference cost, whereas BoF is a technique to alter the training strategy to increase overall model performance without incurring additional costs [41]. In this work, BoF is employed in place of the conventional bounding box regression as in Equations (2) and (3).

$$\text{Loss} = 1 - \text{IoU} + \text{R}(\text{B}, B^{gt}) \tag{2}$$

$$\text{IoU} = \frac{B \cap B^{gt}}{B \cup B^{gt}} \tag{3}$$

where R(B, B^{gt}) represents the form of the penalty between the predicted bounding box (B) and the actual bounding box (B^{gt}). The Complete IoU (CIoU) bounding box regression equation, which has superior aspect of consistency ratio [42] and takes less iterations to achieve a convergent point, is the bounding box regression equation employed in this study. The following equation ensures that the utilized aspect ratio is consistent.

$$R_{CIoU} = \frac{\rho^2(\mathrm{B}, \mathrm{B}^{gt})}{C^2} + \alpha v \tag{4}$$

$$v = \frac{4}{\pi^2}(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h})^2 \tag{5}$$

$$\alpha = \frac{\mathrm{v}}{(1 - IoU) + v} \tag{6}$$

The Euclidean distance from the bounding box is ρ, the positive trade-off parameter is α, the diagonal length of the two bounding boxes is C, and the consistency of the aspect ratio is v. The loss equation from the bounding box can be created using the equation, which is defined by equation (7).

$$L_{CIoU} = 1 - \mathrm{IoU} + \frac{\rho^2(\mathrm{B}, B^{gt})}{C^2} + \alpha v \tag{7}$$

To improve the accuracy of object detection, CIoU is used across all YOLOv4 designs used in this study. Next, BoS is applied to the neck and backbone of YOLOv4. The CSP technique is applied to the backbone, and SPP and PANet are applied to the neck. Mish and Leaky ReLu are the activation functions used in the YOLOv4-CSPDarknet53 and YOLOv4-Tiny series, respectively. The mish activation function is a function that can address exploding and disappearing gradient issues. Moreover, Mish can create models to improve accuracy and generalization [43]. By designating a negative value as a very small number close to "0," Leaky ReLu can prevent malfunctions in neurons with a value of "0." Leaky ReLu is not as effective at dealing with exploding and disappearing gradients as the Mish activation function. The Mish activation function equation is as follows:

$$f(x) = x.\tanh(softplus(x)) \tag{8}$$

$$Softplus(x) = \log(1 + \exp(x)) \tag{9}$$

G. EVALUATION MODEL

Based on the training results and model validation, a model evaluation is performed to determine the best model configuration. The mean average precision (mAP), F1-score, and IoU were used for the evaluation. Since the F1-score can be used to assess how well accuracy and recall are interacting as a whole because it is a harmonic mixture of both metrics. The ratio of correctly identified data to all previously gathered positive data is a measure used to quantify precision. Every positive prediction is compared to every positive result in the data to determine recall. TP is a true positive, indicating that the forecast was accurate. False positives (FPs) occur when a prediction is made that does not match the actual situation, while false negatives (FNs) occur when a value that ought to be positive is revealed to be negative by the prediction findings. For evaluation, the following formula is used:

$$mAP = \frac{1}{total\ class} \Sigma_{i=1}^{total\ class} \frac{TP_i}{TP_i + FP_i} \tag{10}$$

$$F1\ Score = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{11}$$

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

H. MOBILE APPLICATION DEVELOPMENT

Mobile application development is carried out using an object-oriented approach and is carried out in the Java language in Android Studio. As seen in Figure 7, the system contains several main classes, namely: ModelChooseActivity to select a model; DetectorActivity for handling connection with models, performing model inference on camera input, and implementing various tools from other packages. The system also contains various help classes from imported packages, such as YoloV4Classifier in tflite class, which is an implementation of the TensorFlow Lite Object Detection API in Java. AutoFitTextureView, OverlayView, RecognitionScoreView, and ResultsView are all part of the custom view. AutoFitTextureView's purpose is to adjust the aspect ratio of whatever device is being used. OverlayView renders views on other classes in order to detect them. RecognitionScoreView is used to display existing detection results by specifying a color and text size; MultiBoxTracker in tracking class is to handle the display of bounding boxes; The env class contains BorderedText, which encapsulates code that performs the rendering function of writing on a canvas (especially in bounding boxes). ImagesUtils is used to convert YUV420SP to ARGB8888 format. Logger is a function for monitoring by adding a prefix. Size is useful as a measure of the size of the camera object independently such as bitmap size, rotation manipulation and so on. Meanwhile, utils is a multi-purpose class such as mapping memory from the tflite model into applications, transforming matrices for cropping and rotation and for storing detection results.

The TensorFlow Lite Object Detection API is a tool component of the TensorFlow Lite Support Library that provides numerous functions like streamlining image pre-processing and processing model output, making the interpreter simpler to use. The API resides in the YoloV4 Classifier class, which accepts various main parameters such as the model's name, input size, is Quantized, is Tiny, to handle the various types of models implemented in the application. Pre-processing will entail scaling the video frame that the camera captured to match the input model's dimensions. In addition, the outcomes will be transformed into RGB format, which the interpreter will input to make inferences. When the interpreter is loaded, the application will issue a list of six class confidence scores, which indicate how likely it is that an image belongs to a class. Six categories—empty, unripe, underripe, overripe,

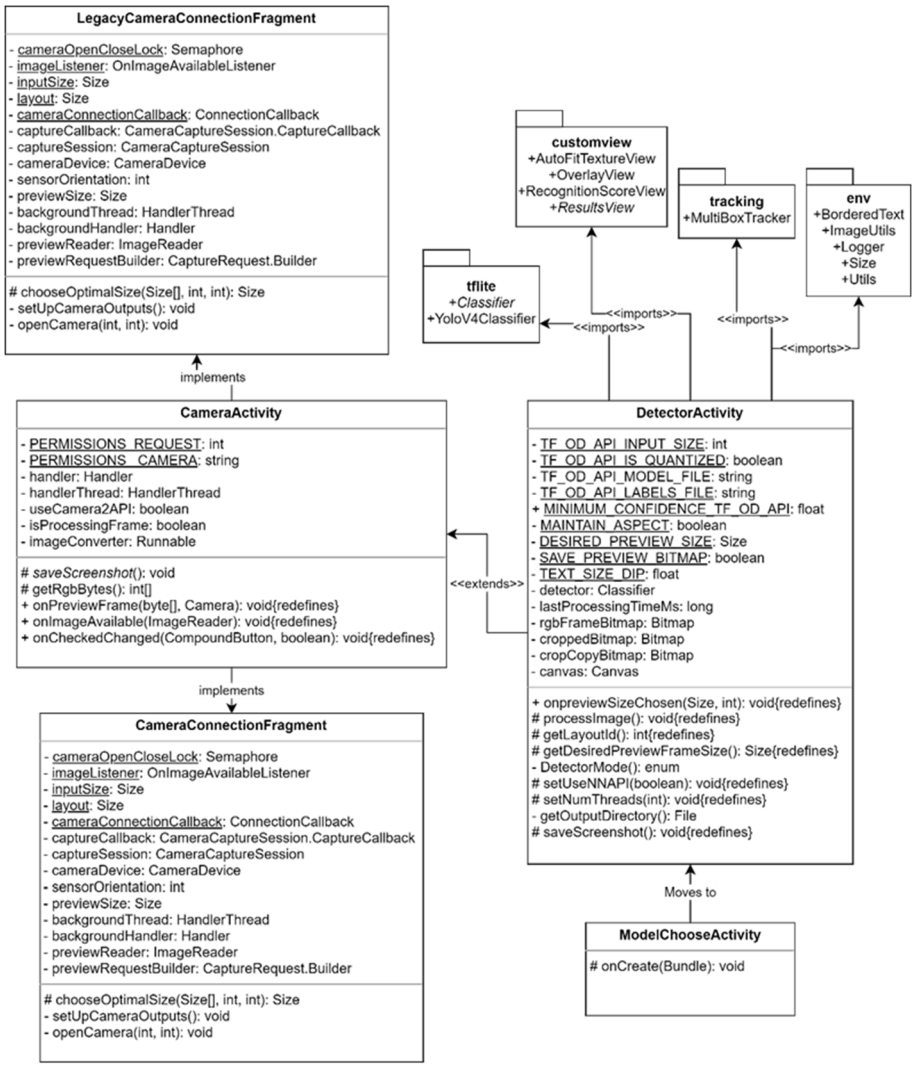

FIGURE 7. Class diagram design of mobile application.

and abnormal fruit bunches—will be used to classify the ripeness of oil palm fruit.

The Camera Activity class is the Activity class inherited by DetectorActivity, where the function of CameraActivity is to manage camera usage, so prepare camera access permissions, then camera layouts/views, then also the screenshot button. There is a CameraConnectionFragment and a LegacyCameraConnectionFragment in this activity; LegacyCameraConnectionFragment is LegacyCode or old camera code that is used as a fallback/backup if the CameraConnectionFragment cannot be used for various reasons, as seen in the setFragment function in CameraActivity.

I. FIELD TESTING AND EVALUATION OF MOBILE APPS

Evaluating images begins with selectively selecting input images from the camera's input stream for inference. Selection was made using the variable flag. The images that go through the selection stage will be converted to RGB values. After applying de-noising and resizing to match the interpreter's input size, the image will be a byte buffer that can be classified by the interpreter. The resulting image classification will be shown to the user by displaying bounding boxes, class name text, and the confidence value for each object in the image. Mobile application testing and evaluation are carried out directly at the palm oil mill in the grading section to determine the performance of the model or application prototype with video input using a smartphone. To evaluate the application's effectiveness in detecting and categorizing the oil palm FFB's state of ripeness, several piles of fruit bunches were used. Data were collected 50 times for each model tested, and the level of accuracy was then calculated.

The model's ability to perform object detection will be measured through the Intersection over Union (IoU) metric, which quantifies the closeness between two types of bounding boxes: truth boxes and prediction boxes. The truth box is the location of the actual object box, while the prediction box is the location of the box predicted by the model. IoU is the ratio between the junction area and the union area for the

TABLE 2. Characteristic of dataset.

Category	#Image	#Object
Unripe	1141	2941
Under Ripe	1306	2609
Ripe	1904	3012
Over Ripe	1169	2679
Empty Bunch	480	879
Abnormal	1256	2637
Total	7256	14757

two boxes [44]. The model's ability to perform classification will be measured through the mean Average Precision (mAP), F1-score, and confidence score metrics. In calculating these metrics, we use several other metrics that are defined as True positive (TP) means that the model predicts there is a bounding box at a location, and that is true; False positive (FP) means that the model predicts there is a bounding box at a location, but it is wrong; False negative (FN) means that the model does not predict that there is a bounding box at a location, but actually there is; True negative (TN) means that the model does not predict a bounding box at a location, and that is true.

Precision is the ratio between the number of true positives and the sum of all the data that the model considers positive. This is a measure of model precision, meaning that of all the positive predictions issued by the model, what proportion of them are correct.

$$IoU = \frac{Area\ of\ prediction\ box \cap Area\ of\ truth\ box}{Area\ of\ prediciton\ box \cup Area\ of\ truth\ box} \quad (14)$$

III. RESULTS AND DISCUSSION

A. DATASET CHARACTERISTICS

The dataset utilized in this study has 4,214 files altogether and is made up of image files of a stack of oil palm FFB that were converted from videos that were recorded using a smartphone at the grading area of the palm oil mill. Each file may contain several images of oil palm FFB, classed as unripe, underripe, ripe, overripe, abnormal, and empty bunches. There may be multiple photos of oil palm fruit in each image file, and Table 2 provides information on the number of images and objects for each ripeness category from the oil palm FFB dataset utilized in this study.

According to Table 2, there were 7,256 total images utilized in this study and 14,757 total objects. This means that there was an average of 2 objects per image because each image included a pile of oil palm FFB as seen in Figures 1 and 2. The number of images for each category is not equal to the number of image objects, with the largest number of images being the ripe fruit bunches, followed by the underripe fruit bunches, and finally, the empty bunches. The proportion in each category and the ripeness level have relatively the same value, so it can be used to develop a detection and classification model for the ripeness level of the oil palm fruit. The dataset we use presently evolved from the dataset used in previous research [45], which only used a single category of oil palm maturity level in each image.

TABLE 3. The comparison of evaluation results of YOLOv4-320 model with and without data augmentation.

Metric	YOLOv4-320 (with data Augmentation)	YOLOV4-320 (without data Augmentation)
mAP@0.50 validation	99.90%	99.77%
mAP@0.50 test	99.89%	99.81%
Precision	0.99	0.97
Recall	1.00	0.99
F1-score	0.99	0.98
Avg IoU	88.01%	86.78%
TP	1406	1401
FP	20	36
FN	4	9

B. DATA AUGMENTATION

In order to ensure that the developed model already uses a sufficient number of datasets, the model is tested by comparing the YOLOv4 model with augmentation data and without augmentation data. The comparison of the results of testing the YOLOv4-320 model with and without augmentation data to detect and classify oil palm fruit can be seen in Table 3.

For each test parameter, namely mAP@0.50, precision, recall, F1 score, and average IoU, Table 3 demonstrates that the performance of the YOLOv4-320 model using augmentation data is better than using data without augmentation. Therefore, for the development of the other YOLOv4 models, data augmentation will be used to carry out training and validation to get the best YOLOv4 model.

C. COMPARISON PERFORMANCE EVALUATION RESULTS OF THE MODEL

To compare the performance of the YOLOv4 model with different experimental parameters, the next model development process used augmentation data. This was done because the results of earlier data experiments showed that using augmentation data was superior to using data without augmentation. Tables 4 and 5 display the findings of experiments conducted on various YOLOv4 models.

In terms of Avg-IoU performance, the YOLOv4-608 model performs the best for determining the ripeness level of oil palm FFB, followed by the YOLOv4-512 and YOLOv4-426 models, while the YOLOv4-CSP-640 model has the lowest Avg-IoU value. Then, from the performance of the F1-score, the YOLOv4 model has a relatively equal value, with the largest F1-score value of 0.99 for the YOLOv4-320, YOLOv4-512, and YOLOv4-608 models, while the YOLOv4-CSP-640 model has the smallest F1-score value. Additionally, model performance measured by recall and precision values reveals that YOLOv4 outperforms YOLOv4-CSP in determining the degree of ripeness of oil palm FFB.

Table 5 shows that the best mAP@50 value in the validation process was obtained in the YOLOv4-320 model, followed by the YOLOv4-416 and YOLOv4-512 models.

Then, from the results of model testing, the highest mAP@50 value was 99.98% obtained in YOLOv4-320, followed by the YOLOv4-512 and YOLOv4-416 models with values of 99.97% and 99.93%, indicating that the YOLOv4 model has a better average precision level in determining the ripeness level of oil palm FFB than YOLOv4-CSP. Then, based on the FPS value, it was found that the YOLOv4-320 model had the largest FPS value (59.7 seconds), and YOLOv4-608 had the smallest FPS value (33.1 seconds), meaning that the inference process in detecting the ripeness level of oil palm FFB can be carried out the fastest by the YOLOv4-320 model because the input file size is smaller than the others. Furthermore, from the measurement of BFlops values, the YLOv4-320 model has the smallest value, followed by the YOLOv4-416 model. Therefore, from the results of this performance comparison, the YOLOv4-320 model and the YOLOv4-416 model will be selected, which are used to be implemented on Android-based smartphone applications. The YOLOv4-416 model will be used in the application development process for mobile devices because its average IoU value is higher than that of the YOLOv4-320 model. Thus, TensorflowLite will be used to transform this model for implementation on mobile platforms. However, due to the computational constraints of this mobile device model, quantization must be performed using both the 8-bits and 16-bits techniques. Figure 8 provides numerous illustrations of how the results of model testing in detecting and classifying the oil palm FFB's degree of ripeness were visualized. This was done in order to evaluate the YOLOv4 model using the data stated in Tables 4 and 5. Figure 8a demonstrates the model's ability to identify unripe and abnormal FFB, while Figures 8b and 8c demonstrate the model's accuracy in identifying ripe and unripe FFB as well as stacks of unripe, abnormal, ripe, and overripe oil palm FFB. The ability of the model to recognize and distinguish between ripe and underripe of oil palm FFB is seen in Figure 8d.

A similar research study focused on detecting the maturity level of oil palm using YOLOv4 [45], albeit with a slightly different dataset. This research used a video dataset, but each frame contained only one maturity-level class with multiple palm oil FFB objects. For single-class detection, the highest mAP achieved was 97.64% using YOLOv4-CSPDarknet53. On the other hand, for multi-class detection within a single video frame, the highest mAP obtained was 70.21% using YOLOv4-Tiny. Table 5 demonstrates that training the model with multi-class data on maturity levels within a single frame enhances its ability to detect different levels of oil palm maturity compared to previous research. Detecting multiple maturity levels in a video frame represents a more realistic and dynamic scenario compared to detecting only one type of maturity level in a video frame.

D. ANDROID-BASED PALM OIL FFB RIPENESS DETECTION APPLICATION USER INTERFACE

The user interface of the Android-based oil palm FFB ripeness detection application can be explained in Figure 9. Figure 9a shows the appearance of the application when it is first opened, where a select model menu is used to select a detection model before the user presses the start button to start the detection process. After the user selects the model to be used for detection by clicking on the drop-down menu (Figure 9b), the user can click the start button so that a tracking overlay appears from the application for FFB detection according to its ripeness level (Figure 9c). Besides that, the user can also click the white circle button to save the detection results to a file. After the detection process is carried out, the user can find out the length of time the inference process takes from the model used by pulling the bottom sheet overlay down (Figure 9d). From this view, it can be revealed that the use of the oil palm FFB detection process application is relatively easy for users to use with the stages of selecting a detection model, then directing the camera to the pile of oil palm FFB so that the application can display a bounding box according to the level of ripeness of the FFB that appears on the mobile application screen.

E. EXPERIMENTAL RESULTS

The YOLOv4-416 model was selected from the results of initial model testing using TensorFlow for the development of a prototype application for detecting the oil palm FFB ripeness degree on smartphones. This model was then converted to TensorFlow Lite so that it could be utilized in smartphone applications. For the conversion on TensorFlow Lite, the quantization approach is applied on YOLOv4, and its performance will be compared with the YOLO model, specifically for smartphones, namely YOLO-tiny. Therefore, in this prototype test, four models were used to be compared, namely YOLOv4-quantized-fp16, YOLOv4-416-quantized-int8, YOLOv4-tiny, and YOLOv4-tiny-quantized-fp16. For evaluation, the TensorFlow Lite model was applied to 10 videos from each ripeness class category, namely empty fruit bunches, not ripe, underripe, ripe, overripe, and abnormal, along with 10 videos from the combined FFB collection. In this test, the results of the model inference were collected for 5 frames in each video. Thus, 350 frames of FFB images were applied for each type of model. The model runs on the Samsung Galaxy S20+ smartphone with an Octa-Core 3 Processor CPU, ARM Mali-G77 MP11 GPU (800 MHz), and 8 GB LPDDR5 RAM. The camera resolution can reach up to 4032 $\times$ 3024 pixels.

The model's capacity to locate oil palm FFB at the precise spot marked by setting a bounding box at the right coordinates is demonstrated by the model test results based on the IoU value. In Table 6, the IoU values for each model in each class category are compared. According to the findings of this comparison, the underripe category has the highest IoU value, which is 92.4%, while the abnormal FFB category has the lowest IoU, which is 74.65%. This value states that in the under-ripe FFB category, the model can detect the location of FFB better than the abnormal category class. Then, based on the model, the one with the highest IoU value is the YOLOv4-q-fp16 model, and the lowest is YOLOv4-

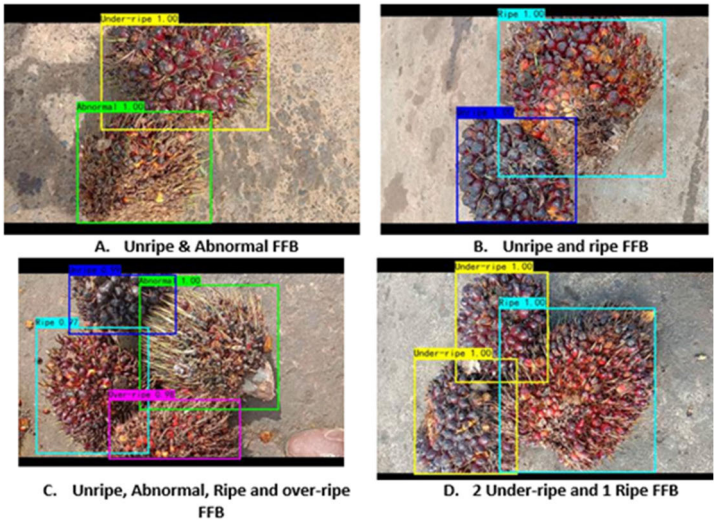

FIGURE 8. Testing results of YOLOv4.

TABLE 4. Comparison of model testing results based on F1-score and Avg-IoU.

Models	Precision	Recall	F1-Score	Avg IoU	TP	FP	FN
YOLOv4-320	0.99	1.00	0.99	88.01%	1406	20	4
YOLOv4-416	0.97	1.00	0.98	88.27%	1404	37	6
YOLOv4-512	0.99	1.00	0.99	88.94%	1408	20	2
YOLOv4-608	0.99	1.00	0.99	91.87%	1407	12	3
YOLOv4-CSP-512	0.94	0.99	0.96	85.68%	1395	95	15
YOLOv4-CSP-640	0.85	0.99	0.91	78.66%	1399	252	11

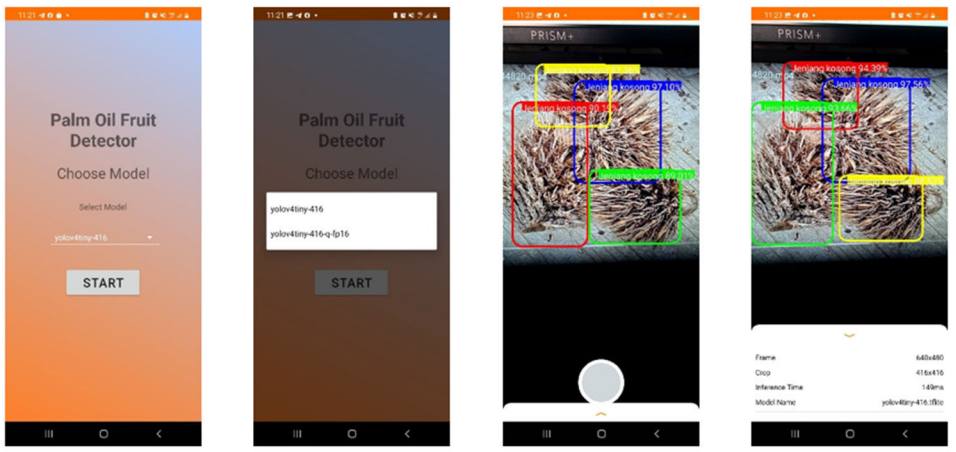

Figure 9a. *Interface chooser* **menu**

Figure 9b. YOLO model selection display

Figure 9c. *Tracking overlay in detection view*

Figure 9d. *Tracking overlay with bottom sheet view*

FIGURE 9. User interface of oil palm FFB ripeness detection application.

tiny-q-fp16, which means that the YOLOv4 model with a quantification of 16 has the best ability to detect oil palm FFB compared to other models. In addition, the IoU value for the combined oil palm FFB class has a value below the average because, in this class, the objects have various FFB categories in terms of size and ripeness level, so the model has poor performance compared to FFB piles with uniform class categories.

The same thing happened to the abnormal FFB category, which had different FFB sizes, so the IoU value was smaller than the average value. To test the model's ability to classify the ripeness level of oil palm FFB according to its class

TABLE 5. Comparison of model evaluation results based on mAP, FPS, and BFLOPS.

Models	mAP@0.50 validation	mAP@0.50 testing	FPS	BFlops
YOLOv4-320	99.90%	99.98%	59.7	35.266
YOLOv4-416	99.89%	99.93%	50.9	59.599
YOLOv4-512	99.89%	99.97%	43.7	90.281
YOLOv4-608	99.75%	99.88%	33.1	127.310
YOLOv4-CSP-512	99.78%	99.84%	48.6	76.188
YOLOv4-CSP-640	99.84%	99.93%	36.0	119.044

TABLE 6. Results comparison of IoU for each category.

Model	Empty bunch	Unripe	Under-ripe	Ripe	Over-ripe	Abnormal	Combination	Average
YOLOv4-q-fp16	88.20%	90.15%	92.40%	83.40%	85.75%	75.89%	79.85%	85.09%
YOLOv4-q-int8	87.90%	86.70%	91.45%	86.45%	82.34%	74.65%	74.69%	83.45%
YOLOv4-tiny	86.50%	87.95%	90.35%	79.80%	84.86%	74.78%	82.30%	83.79%
YOLOv4-tiny-q-fp16	84.25%	86.70%	88.92%	82.34%	83.47%	76.89%	75.68%	82.61%

TABLE 7. Results comparison of F1-score & mAP for each category.

Model	Evaluation metrics	Empty bunch	Unripe	Under-ripe	Ripe	Over-ripe	Abnormal	Cobination	Average
YOLOv4-q-fp16	mAP	98.20%	99.90%	40.20%	67.89%	56.70%	60.10%	35.70%	65.53%
	F1-score	0.99	0.99	0.53	0.56	0.53	0.60	0.30	0.64
YOLOv4-q-int8	mAP	98.00%	99.90%	40.15%	64.35%	53.20%	60.15%	35.26%	64.43%
	F1-score	0.99	0.99	0.47	0.60	0.52	0.60	0.24	0.63
YOLOv4-tiny	mAP	97.10%	99.75%	15.60%	53.60%	43.25%	57.80%	25.70%	56.11%
	F1-score	0.99	0.99	0.25	0.45	0.39	0.47	0.20	0.53
YOLOv4tiny-q-fp16	**mAP**	97.02%	99.73%	14.20%	53.20%	38.90%	46.75%	23.40%	53.31%
	F1-score	0.99	0.99	0.23	0.42	0.38	0.45	0.15	0.52

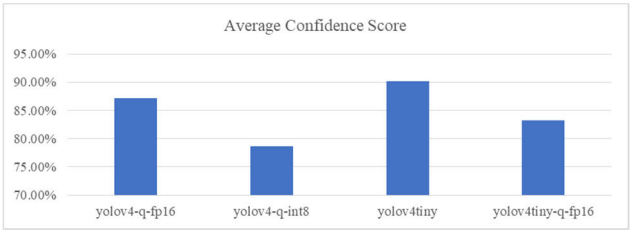

FIGURE 10. Comparison of the average confidence score models.

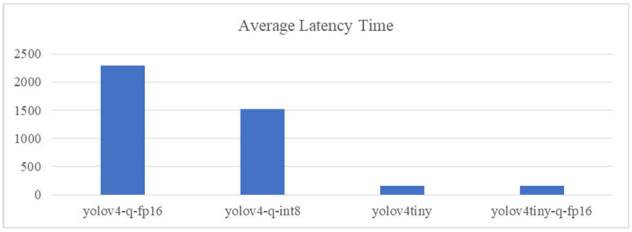

FIGURE 11. Comparison of the average latency time of the models.

category, model performance measurements based on the F1-score and mAP metrics are used. The comparison results of the performance testing of the YOLOv4 model with these two metrics can be explained in Table 7. The results of the comparison of the F1-score values found that the highest average F1-score was obtained in the YOLOv4-q-fp16 model with a value of 0.64, while the lowest value was obtained on the YOLOv4tiny-q-fp16 model with a value of 0.52. This value indicates that the model that can classify the best ripeness level when implemented on smartphones is YOLOv4tiny-q-fp16. The same thing happened to the average mAP value, with the highest value being 65.53% in the YOLOv4tiny-q-fp16 model and the lowest value in the YOLOv4tiny-q-fp16 model with a value of 53.31%.

The model's ability to classify the ripeness level of FFB also depends on the data from each category. It can be seen that the empty and immature bunches category has the highest F1 score and mAP value, which is above 90%, while the under-ripe category has the lowest F1 score. This indicates that the model can classify objects with very different characteristics well, while it still has difficulty classifying objects with almost similar characteristics. The characteristics of objects in the unripe FFB category class are relatively similar, namely having a color that tends to be black, so models can categorize these objects more easily, as well as in the empty fruit bunch category class, which tends to have a similar color, namely brown, so it can be easily distinguished from other objects. However, the model has a less consistent performance for oil palm FFB with more complex characteristics in terms of color gradations, such as underripe and ripe FFB.

Table 7 shows that the model can classify the ripeness level of FFB very well for the empty and unripe categories, achieving an mAP of more than 97% and an F-score of 0.99. However, the model began to lose accuracy in classifying

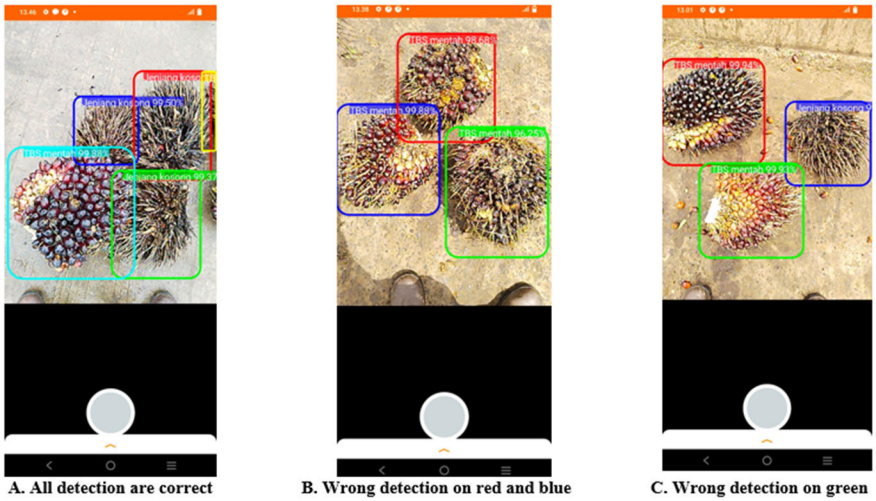

FIGURE 12. Example of testing results of mobile-based detection for the YOLOv4-tiny model.

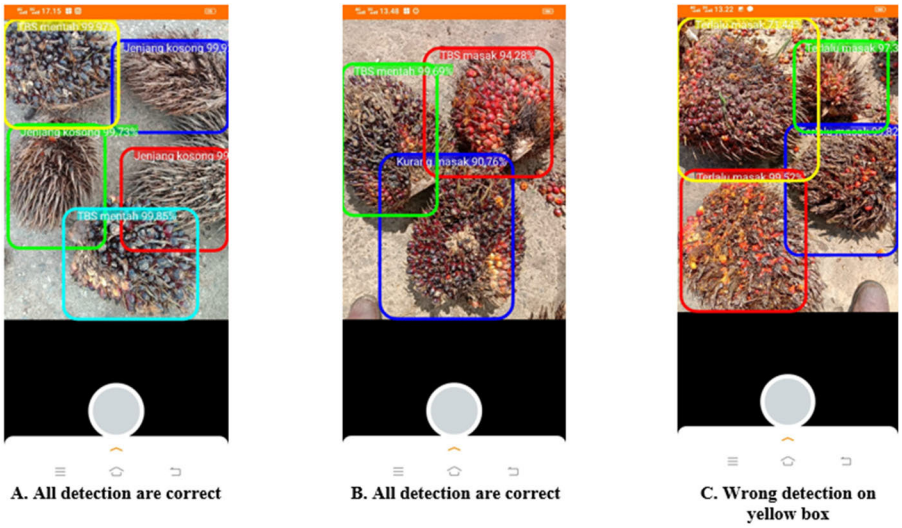

FIGURE 13. Example of the testing result of mobile-based detection for YOLOv4-quantized model.

ripe, abnormal, and overripe FFB, with a mAP of 40–60% and an F-score of 0.30–0.50. Models are often confused in classifying ripe and overripe FFB due to the similarity in color and shape. In the same way, the model has a poor level of accuracy in the underripe category because it is often classified as unripe. The majority of underripe FFB is classified as unripe; hence, the mAP falls below 20% for the YOLOv4-tiny model. The model also has poor accuracy for videos in the combined category due to difficulty differentiating between the many types of FFB, where the model can only achieve a mAP level of 20–40% and an F-score of 0.15–0.30.

Models were evaluated using a confidence score in order to gauge their object detection capability. The overall average confidence score for the tested palm FFB categories is shown in Figure 10 for all classes. The test's findings show that all models have a strong confidence score that is greater than 75%. This value indicates that the model can detect oil palm FFB image objects with certainty, although this does not yet indicate that the classification results are correct. However, a model that has a high average confidence score will have a more stable output. It is clear from the comparison of the confidence score values that the YOLOv4-tiny model received the highest score, followed by the YOLOv4-q-fp16 model with scores of 90% and 87%, respectively. Meanwhile, the lowest value was achieved by the YOLOv4-q-fp8 model. These results indicate that the YOLOv4-tiny model performs the best object detection when implemented on smartphones because this model was indeed developed to be implemented on mobile devices. However, the YOLOv4-q-fp16 model still has quite good capabilities when implemented on mobile devices, which are almost like the YOLOv4-tiny model.

To measure the speed of the model in classifying and detecting objects, the speed of inference is measured with the latency time value of each tested model, as shown in Figure 11. Figure 11 shows that the YOLOv4-quantized model has a much longer latency than YOLOv4-tiny. This indicates that the inference time for the YOLOv4-quantized model is slower than the YOLOv4-tiny model. This inference process is greatly influenced by the size of the model because the YOLOv4-quantized model has a larger size than the YOLOv4-tiny model, which causes the inference process to take longer. However, the speed of this inference does not necessarily provide information related to the accuracy of the model. Therefore, it is necessary to test the accuracy of each model in order to determine the accuracy of the inference.

The application prototype was tested under actual testing conditions at the palm oil mill grading location in order to determine the performance of the proposed model. Figures 12 and 13 show the outcomes of model testing for detecting and classifying the pile of oil palm FFB applied on a smartphone. Figure 12 compares the detection results of the YOLOv4-tiny method used on mobile devices to determine the degree of ripeness of oil palm FFB in various categories. From the test results, it was found that this model has a good level of detection accuracy on FFB with empty bunches and unripe fruit categories, as shown in Figure 12A. However, as shown in Figures 12B and 12C, the YOLOv4-tiny approach still frequently results in detection mistakes for both ripe and unripe fruit. This shows that although the YOLOv4-tiny model can carry out the detection process quickly, it has a poor level of accuracy according to the F1 score and mAP values shown in Table 7.

Figure 13 compares the detection results of the YOLOv4-quantized model implemented on a smartphone for various ripeness levels of oil palm FFB. From the results of this comparison, it can be seen that the model can detect oil palm FFB ripeness level categories with a good level of generalization, as shown in Figure 13A, which can detect FFB with empty bunches and unripe fruit bunch categories, as well as in Figure 13C, which can detect unripe, underripe, and properly ripe fruit categories. However, an error still occurs when detecting ripe fruit combined with overripe fruit, as shown in Figure 13C. This shows that, in contrast to the YOLOv4-tiny model, the YOLOv4-quantized model can distinguish more intricate class categories. In comparison to the YOLOv4-tiny model, which includes complex object and image characteristics, the YOLOv4-quantized model has a greater generalization power for recognizing the ripeness degree of oil palm FFB in the piles.

In broad sense, the YOLOv4-q-fp16 model that we propose has a higher level of detection precision than YOLOv4tiny. Although the parameters used by YOLOv4-q-fp16 are still larger than those used by YOLOv4tiny, which is one of the supporting factors for better detection, YOLOv4-q-fp16 is still feasible to implement on mobile devices despite having a higher latency. As an alternative, YOLOv4-q-int8 can be used to make the model lighter while still maintaining detection precision because it only has a 1.1% decrease in mAP compared to YOLOv4-q-fp16.

IV. CONCLUSION

The separation of oil palm FFB that does not meet quality standards is an important thing that needs to be done in the palm oil production process. However, currently, it is done manually, so the level of consistency is less controllable. Therefore, it is necessary to develop a system that can automatically separate oil palm FFB so that the results are more consistent and have better accuracy. This research proposes a model for detecting and classifying oil palm FFB using smartphones and deep learning. The model used in this study is YOLOv4, with various modifications both in terms of the dataset used and the model hyperparameters. The test results of the YOLOv4 model show that the use of augmentation data produces better performance than data without augmentation. Then, to get the best YOLOv4 model that can detect and classify the ripeness level of oil palm FFB, it has been compared with various YOLOv4 models, with the best results being the YOLOv4-416 model with mAP, IoU, FPS, and BFlops criteria. Furthermore, the YOLOv4 model is converted into a model that can be implemented on smartphones with limited memory and processing speed using quantification. The results of the quantification of the selected Its performance with the YOLOv4-tiny model, which can be utilized to identify and categorize oil palm FFB categories, was compared to that of the YOLOv4 model. The test findings demonstrate that although the quantized YOLOv4 model's inference process is slower than that of the YOLOv4-tiny model, it is more accurate when used on mobile devices. These findings show that the suggested model can detect and categorize the ripeness level of oil palm FFB, which has complicated color and size features, on a smartphone with video input so that it can detect and categorize in real-time. From the results of this experiment, there is still an opportunity to improve the performance of models on smartphones because the performance of detection and classification with the combined video class category still has lower performance than the video class with a single category. One of the proposed improvements is to increase the training data on multi-category datasets.

The model utilized in this research demonstrates its effectiveness in detecting post-harvest palm oil FFB, making it suitable for sorting the maturity level of oil palm. However, due to the diverse datasets required for detecting palm oil FFB before harvesting, the application of this detection model in oil palm plantation areas is limited. The implementation is currently restricted to mobile devices as the model parameters need to be reduced to match the hardware capabilities. Nevertheless, there is a possibility that future implementations will utilize more suitable embedded devices, enabling improvements in model performance. Ultimately, there is still significant potential for further research in computer vision within the palm oil industry, both in terms of expanding datasets and enhancing existing object detection models.

REFERENCES

[1] F. B. Ahmad, Z. Zhang, W. O. S. Doherty, and I. M. O'Hara, "The outlook of the production of advanced fuels and chemicals from integrated oil palm biomass biorefinery," *Renew. Sustain. Energy Rev.*, vol. 109, pp. 386–411, Jul. 2019.

[2] IndexMundi. (2022). *Palm Oil Production by Country in 1000 MT*. [Online]. Available: https://www.indexmundi.com/agriculture/?commodity=palm-oil

[3] C. L. Chew, B. A. Tan, J. Y. S. Low, N. I. N. M. Hakimi, S. F. Kua, and C. M. Lim, "Exogenous ethylene application on postharvest oil palm fruit bunches improves crude palm oil quality," *Food Sci. Nutrition*, vol. 9, no. 10, pp. 5335–5343, Oct. 2021.

[4] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *Proc. Int. Conf. Eng. Technol. (ICET)*, Aug. 2017, pp. 1–6.

[5] M. A. Hedjazi, I. Kourbane, and Y. Genc, "On identifying leaves: A comparison of CNN with classical ML methods," in *Proc. 25th Signal Process. Commun. Appl. Conf. (SIU)*, May 2017, pp. 1–4.

[6] S. H. Lee, C. S. Chan, P. Wilkin, and P. Remagnino, "Deep-plant: Plant identification with convolutional neural networks," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 452–456.

[7] Y.-D. Zhang, Z. Dong, X. Chen, W. Jia, S. Du, K. Muhammad, and S.-H. Wang, "Image-based fruit category classification by 13-layer deep convolutional neural network and data augmentation," *J. Multimedia Tools Appl.*, vol. 78, no. 3, pp. 1–20, 2017.

[8] W. I. W. Ishak and R. M. Hudzari, "Image based modeling for oil palm fruit maturity prediction," *J. Food, Agricult. Environ.*, vol. 8, no. 2, pp. 469–476, 2010.

[9] N. Fadilah, J. Mohamad-Saleh, Z. A. Halim, H. Ibrahim, and S. S. Ali, "Intelligent color vision system for ripeness classification of oil palm fresh fruit bunch," *Sensors*, vol. 12, no. 10, pp. 14179–14195, Oct. 2012.

[10] Y. Yesiansyah and M. Murinto, "Aplikasi deteksi kematangan buah sawit menggunakan metode perbandingan histogram citra," *JSTIE Jurnal Sarjana Teknik Informatika (E-J.)*, vol. 4, no. 3, pp. 86–95, 2016.

[11] Z. Ibrahim, N. Sabri, and D. Isa, "Palm oil fresh fruit bunch ripeness grading recognition using convolutional neural network," *J. Telecommun., Electron. Comput. Eng. (JTEC)*, vol. 10, no. 3, pp. 109–113, 2018.

[12] H. Herman, A. Susanto, T. W. Cenggoro, S. Suharjito, and B. Pardamean, "Oil palm fruit image ripeness classification with computer vision using deep learning and visual attention," *J. Telecommun., Electron. Comput. Eng. (JTEC)*, vol. 12, no. 2, pp. 21–27, 2020.

[13] Z. Y. Wong, W. J. Chew, and S. K. Phang, "Computer vision algorithm development for classification of palm fruit ripeness," *AIP Conf. Proc.*, vol. 2233, no. 1, 2020, Art. no. 030012.

[14] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[15] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[16] M. Tan and Q. Le, "EfficientNet-rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.

[17] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "MnasNet: Platform-aware neural architecture search for mobile," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2815–2823.

[18] Suharjito, G. N. Elwirehardja, and J. S. Prayoga, "Oil palm fresh fruit bunch ripeness classification on mobile devices using deep learning approaches," *Comput. Electron. Agricult.*, vol. 188, Sep. 2021, Art. no. 106359.

[19] Q. Fang, H. Li, X. Luo, L. Ding, H. Luo, T. M. Rose, and W. An, "Detecting non-hardhat-use by a deep learning method from far-field surveillance videos," *Autom. Construct.*, vol. 85, pp. 1–9, Jan. 2018.

[20] X. Long, W. Cui, and Z. Zheng, "Safety helmet wearing detection based on deep learning," in *Proc. IEEE 3rd Inf. Technol., Netw., Electron. Autom. Control Conf. (ITNEC)*, Mar. 2019, pp. 2495–2499.

[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 779–788.

[22] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[23] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[24] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 379–387.

[25] F. Li, Z. Liu, W. Shen, Y. Wang, Y. Wang, C. Ge, F. Sun, and P. Lan, "A remote sensing and airborne edge-computing based detection system for pine wilt disease," *IEEE Access*, vol. 9, pp. 66346–66360, 2021.

[26] M. G. Moghaddam, "Detection and localization of ripe tomatoes using machine vision," *Turkish J. Comput. Math. Edus.*, vol. 12, no. 13, pp. 7584–7592, 2021.

[27] A. I. B. Parico and T. Ahamed, "Real time pear fruit detection and counting using YOLOv4 models and deep SORT," *Sensors*, vol. 21, no. 14, p. 4803, 2021, doi: 10.3390/s21144803.2021.

[28] M. A. Genaev, E. G. Komyshev, O. D. Shishkina, N. V. Adonyeva, E. K. Karpova, N. E. Gruntenko, L. P. Zakharenko, V. S. Koval, and D. A. Afonnikov, "Classification of fruit flies by gender in images using smartphones and the YOLOv4-tiny neural network," *Mathematics*, vol. 10, no. 3, p. 295, Jan. 2022, doi: 10.3390/math10030295.

[29] R. Gai, N. Chen, and H. Yuan, "A detection algorithm for cherry fruits based on the improved YOLO-v4 model," *Neural Comput. Appl.*, vol. 35, pp. 1–12, May 2021, doi: 10.1007/s00521-021-06029-z.2021.

[30] W. Chen, J. Zhang, B. Guo, Q. Wei, and Z. Zhu, "An apple detection method based on Des-YOLO v4 algorithm for harvesting robots in complex environment," *Math. Problems Eng.*, vol. 2021, pp. 1–12, Oct. 2021, doi: 10.1155/2021/7351470.

[31] M. H. Junos, A. S. M. Khairuddin, S. Thannirmalai, and M. Dahari, "Automatic detection of oil palm fruits from UAV images using an improved YOLO model," *Vis. Comput.*, vol. 38, no. 7, pp. 2341–2355, Jul. 2022.

[32] J.-W. Chen, W.-J. Lin, H.-J. Cheng, C.-L. Hung, C.-Y. Lin, and S.-P. Chen, "A smartphone-based application for scale pest detection using multiple-object detection methods," *Electronics*, vol. 10, no. 4, p. 372, Feb. 2021.

[33] F. A. Junior, Y. P. Koeswandy, Debi, P. W. Nurhayati, M. Asrol, and Marimin, "Annotated datasets of oil palm fruit bunch piles for ripeness grading using deep learning," *Sci. Data*, vol. 10, no. 1, p. 72, Feb. 2023.

[34] P. M. Blok, F. K. Evert, A. P. M. Tielen, E. J. Henten, and G. Kootstra, "The effect of data augmentation and network simplification on the image-based detection of broccoli heads with mask R-CNN," *J. Field Robot.*, vol. 38, no. 1, pp. 85–104, Jan. 2021.

[35] P. Xu, Q. Li, B. Zhang, F. Wu, K. Zhao, X. Du, C. Yang, and R. Zhong, "On-board real-time ship detection in HISEA-1 SAR images based on CFAR and lightweight deep learning," *Remote Sens.*, vol. 13, no. 10, p. 1995, May 2021, doi: 10.3390/rs13101995.

[36] C. Wang, H. M. Liao, Y. Wu, P. Chen, J. Hsieh, and I. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 1571–1580.

[37] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[39] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8759–8768, doi: 10.1109/CVPR.2018.00913.

[40] A. Kaya, A. S. Keceli, C. Catal, H. Y. Yalic, H. Temucin, and B. Tekinerdogan, "Analysis of transfer learning for deep neural network based plant classification models," *Comput. Electron. Agricult.*, vol. 158, pp. 20–29, Mar. 2019.

[41] Z. Zhang, T. He, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of freebies for training object detection neural networks," 2019, *arXiv:1902.04103*.

[42] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 7, Apr. 2020, pp. 12993–13000.

[43] K. Wu, C. Bai, D. Wang, Z. Liu, T. Huang, and H. Zheng, "Improved object detection algorithm of YOLOv3 remote sensing image," *IEEE Access*, vol. 9, pp. 113889–113900, 2021.
[44] A. Anwar. (May 13, 2022). *What is Average Precision in Object Detection & Localization Algorithms and How to Calculate It*. Accessed: Nov. 23, 2022. [Online]. Available: https://towardsdatascience.com/what-is-average-precision-in-object-detection-localization-algorithms-and-how-to-calculate-it-3f330efe697b
[45] F. A. Junior, "Video based oil palm ripeness detection model using deep learning," *Heliyon*, vol. 9, no. 1, Jan. 2023, Art. no. e13036.

SUHARJITO (Member, IEEE) received the master's degree in information engineering from the Sepuluh Nopember Institute of Technology (ITS), in 2000, and the Ph.D. degree in agro-industrial engineering from Bogor Agricultural University, Indonesia, in 2011. His areas of expertise are computer science, engineering, decision sciences, soft computing, and information engineering. He is currently a Senior Lecturer with the Department of Master of Industrial Engineering, Binus Graduate Program, Bina Nusantara University. His current research interest includes computer vision, especially the use of computer vision in agriculture with the topic of detecting the maturity level of oil palm. The research, he is currently carrying out is supported by the Indonesian Directorate General of Higher Education, Research and Technology.

MUHAMMAD ASROL received the Doctor of Engineering degree from Bogor Agricultural Institute, Indonesia, in 2019. He is interested in supply chain management, engineering optimization, machine learning, and intelligence system as his major publications and research. He was an Awardee of PMDSU Scholarship from the Ministry of Higher Education, Indonesia, from 2015 to 2019, to achieve his master's and Ph.D. degrees in four years. He currently leads the Department of Industrial Engineering, Bina Nusantara University.

DITDIT NUGERAHA UTAMA received the bachelor's degree from the Informatics Program, Bina Nusantara University, Indonesia, in 1998, the master's degree in information system from Bina Nusantara University, in 2000, the Master of Commerce degree in IS from Curtin University, Australia, in 2001, the Ph.D. degree in agriculture industrial engineering from Bogor Agriculture University, Indonesia, in 2012, and the Ph.D. degree from the Mathematics and Informatics Program, Göttingen Universität, Germany, in 2015. His Ph.D. research applied in the domains of IS applied and environmental informatics. He is currently a senior researcher in computer science, specifically in the research domain of decision support model.

FRANZ ADETA JUNIOR received the Bachelor of Science (B.Sc.) degree in computer engineering from Bina Nusantara University, Indonesia, in 2021. He was awarded funding to study computer science at the master's level from Bina Nusantara University. In 2022, he successfully finished his thesis. His research interests include computer vision and image processing.

MARIMIN (Member, IEEE) received the B.S. degree (Hons.) in agro-industrial technology from IPB University (Bogor Agricultural University), Bogor, Indonesia, in 1984, the M.Sc. degree in computer science from the University of Western Ontario, Canada, in 1990, and the Ph.D. degree from the Faculty of Engineering Science, Osaka University, Japan, in 1997. Since 2003, he has been a Professor in systems engineering with IPB University. His research interests include intelligent and fuzzy expert systems, multiple criteria decision making, intelligent decision support systems, and sustainable supply chain management. He is a member of the Indonesian Engineer Association and Indonesia Logistic and Supply Chain Management.