Journal of Oil Palm Research
DOI: https://doi.org/10.21894/jopr.2023.0059

OIL PALM LEVEL OF RIPENESS CLASSIFICATION USING EFFICIENTDET-LITE CNN ARCHITECTURE

YOSUA ALVIN ADI SOETRISNO[1]*; EKO HANDOYO[1]; SUMARDI[1] and ENDA WISTA SINURAYA[1]

ABSTRACT

Determining the category of ripeness of oil palm fresh fruit bunches (FFB) based on maturity is essential in estimating suitable fruit for processing. Ripeness classification could prevent the oil palm from becoming over-ripe. When the oil palm becomes over-ripe, the quality of oil extracted is not optimal because of the increase in free-fatty acid level. The deep learning approach usually used in object detection could help in object classification based on the model trained with the oil palm dataset. Many kinds of research used shared convolutional neural network (CNN) architecture to detect the ripeness of oil palms fruits. This research contributes to finding the suitable CNN model specialised in oil palm FFB ripeness using an unequal scalable feature known as EfficientDet. We propose the most proper coefficient for object detection scaling with the following configuration. The compound coefficient D2's resolution is used as EfficientDet-Lite2 input size. EfficientDet-Lite2's backbone network is nearly identical to EfficientDet using D2. The bi-directional feature pyramid network (Bi-FPN) layer is five, and the box class per layer is three. The accuracy of the proposed EfficientDet-Lite2 using EfficientDet D2 input is 84% and has been tested in Indonesian plantations.

Keywords: CNN, EfficientDet, feature pyramid network, oil palm.

Received: 31 October 2022; **Accepted:** 27 August 2023; **Published online:** 3 November 2023.

INTRODUCTION

The development of Deep Learning (DL) makes functions such as image classification, object detection, image tracking, and image manipulation more accessible and accurate (Prasetyo *et al.*, 2020). The DL system has a layered architecture which could be organised as a convolutional and fully connected layer. The convolutional layer is for extracting automatic features based on the properties of the windowing filter applied in a particular area of the image. The fully connected layer collects the critical component of the object with an activation function to ensure that the feature belongs to the category. The activation function is a mathematical function that can activate the next layer based on previous criteria with various weights to be considered as a category based on the feature that belongs to that category.

DL has been applied in many applications, including agriculture. Palm oil is one strategic commodity in Indonesian agriculture known as crude palm oil (CPO) that is being exported to many countries because of the need for cooking oil and other industrial application. Oil palm ripeness as the CPO source must be detected on the right time sequence. A farmer harvests based on experience, and sometimes the decision is wrong because of unfocussed observation. The farmer could be assist by a computer vision system to select for ripe fresh fruit bunches (FFB) is ripe based on the grade classification (Suharjito *et al.*, 2021).

There are two categories of research: Detecting the ripeness of oil palm FFB from a close range and detecting the ripeness and classifying the trees from

[1] Department of Electrical Engineering, Faculty of Engineering, Diponegoro University, Semarang, Indonesia.

* *Corresponding author e-mail: yosua@live.undip.ac.id*

remote sensing images. Image classification can be achieved through the use of traditional machine learning methods, as well as through the application of deep learning techniques. Traditional machine learning combines image processing techniques to obtain unique features from statistical Red, Green, Blue (RGB) distribution in images of oil palm FFB, while deep learning extracts specific features through filter combinations.

In general, mature FFB is transformed from black to red but not completely red, so it needs to be considered the most significant region of interest (ROI) with red color (Alfatni *et al.*, 2020). The image processing technique combines machine learning to properly get the ROI's feature distribution. An artificial neural network (ANN) and support vector machine (SVM) algorithms classify the oil palm ripeness category. The SVM algorithm is used in the grain, not the whole oil palm fruit image.

Regarding DL, there is a variation in convolutional neural network (CNN) usage from simple to complicated. Simple CNN uses EfficientNet B0 with an accuracy of 89.3% (Suharjito *et al.*, 2021). There is a need for simple CNN because it will be implemented in the embedded device. Based on the architecture of EfficientNet, it would be developed with extra scaling fusion parametrisation in the term of EfficientDet in this research. Mounting the filter with unequal dimensions in EfficientDet could give another insight base on the oil palm's ripeness characteristic. There is a surrounding edge to each bunch and a color transition to determine which part is ripe.

Several kinds of research on the maturity level of oil palm FFB use different classification and DL models and additional techniques to help obtain more specific characteristics of the oil palm bunch. Machine learning needs several feature engineering. Feature engineering that needs to be tinkered with is wavelength input and RGB decomposition, preprocessing using the principal component analysis (PCA) method, and the artificial neural network (ANN) hyperparameter tuning.

The ANN approach is explained in research for near-bunch images and remote sensing. ANN uses a different approach to CNN architecture. Transfer learning is used from the typical architecture like AlexNet, which outperforms machine learning with SVM and extra preprocessing like Histogram of Oriented Gradient (HOG) (Ibrahim *et al.*, 2018). *Table 1* shows the comparison of methods and improvement suggestions of each research done in oil palm detection based on image processing and artificial neural network.

This research was not using regional segmentation because the complexity of the segmentation could affect the performance of the DL model applied to the Android device. There is also a need for more examples to represent the state of the oil palms (Prasetyo *et al.*, 2020). Although this research primarily focuses on the dataset derived from the plantations in Sumatra, it should be noted that the dataset from Kalimantan was relatively limited in size and scope. We incorporated data from both regions to understand plantation conditions across different areas better.

An extension for CNN is R-CNN, a combination of region proposals. There is extra insight from complex sight, overlapping objects, and diverse backgrounds, which could exclude the intersection of an object with another object, so the region of interest becomes clearer. Faster R-CNN consists of regional proposal network (RPN) and detector modules. RPN is a small network neuron in the last row of convolutional layers and is used to predict the existence of the required object and bounding box.

This research also aimed to build observation tools to validate the model directly on the plantation area, given the variation in tree height, ranging from approximately 2-20 m. Before utilising the observation tools in this research, farmers would employ elevated stationary platforms or tall observation towers to observe the upper sections of trees where the fruit bunches are located at heights of 10 m or higher. In this case, following the tools from this research, farmers and researchers could practically approach the task by utilising their smartphones' cameras along with a long stick. Farmers and researchers could extend their reach by attaching their phones to the stick and capturing images or videos of the FFB located at heights of 10 m or higher. This method allowed them to observe and monitor the FFB without needing elevated stationary platforms or tall observation towers. Subsequently, these captured images were classified using deep learning techniques, thereby enhancing the accuracy and efficiency of the classification process.

Several CNN is used to classify the oil palm FFB: LeNet, AlexNet, EfficientNet B0, and Faster R-CNN. The reason for choosing EfficientDet-Lite as the backbone for classification was that EfficientDet is based on the CNN model that achieved state-of-the-art performance (Guo *et al.*, 2022). A lighter and improved version of EfficientDet, called EfficientDet-Lite, does away with the squeeze-and-excite module because it is not mobile-optimised and replaces the ReLU6 activation function with a swish activation mechanism. This research began with a trial with MobileNetV2 and NasNetMobile before using the EfficientDet-Lite. The problem was that the MobileNetV2 could not do a stable classification with cases; sometimes, the object was undetected. Another research also showed that EfficientDet-D5 using a variation of scaling D5 gives better accuracy than YOLOv4 (Ammar *et al.*, 2021).

TABLE 1. FFB RIPENESS DETECTION COMPARISON

Researcher	Method	Algorithm	Sample size, number of classes, accuracy	Insight and improvement suggestions
Shiddiq *et al*. (2016)	Image Processing	RGB quantisation	Not mentioned, 4 classes not mentioned	The comparison should also be made using soft computing programs
Septiarini *et al*. (2019)	Image Processing and Machine Learning	Histogram statistical spread, otsu method, and morphology using SVM	160 images, 4 classes, about 90.00%	Need development of segmentation and classification methods to reduce error.
Sukemi and Sukrisno (2019)	Image Processing	Gray Level Co-occurrence Matrix (GLCM)	50 images, 2 classes, 90.00%	Need lighting for getting the 90% accuracy
Prasetyo *et al*. (2020)	ANN	Faster R-CNN	100 images, counting and giving a bunch's region area, 80.00%	Some things that caused the detection of bunches were the high level of occlusion, and there is the presence of several objects that covered the bunch
Suharjito *et al*. (2021)	ANN	EfficientNetB0	103 images, 4 classes, 89.3.0%	A tradeoff exists between accuracy, classification time, and the need for a more significant parameter.
Saleh and Liansitim (2020)	ANN	CNN	Accuracy is calculated in different research	Need to use datasets outside the training because the accuracy is high when using the training dataset.
Harsawardana *et al*. (2020)	ANN with Smart Crane Hardware	CNN using ResNet	400 images, 7 classes, 71.34%	The accuracy is only about 71%, and need for optimisation
Ibrahim *et al*. (2018)	ANN	Compare Fast Retina Keypoint, HOG using SVM, and CNN using AlexNet	120 images, 4 classes, 92.00%	A deep layer can lead to better results but at a slow processing time and need a deeper pre-trained CNN model.
Ammar *et al*. (2021)	ANN for Geospatial Image	YOLOv4 and EfficientDet-D5	258 images, not detecting oil palm ripeness, 99.00%	There is a tradeoff between speed and accuracy which are YOLOv4 and EfficientDetD5 are suitable for accuracy, while Faster R-CNN and YOLOv3 were significantly less accurate.
Khalid and Shahrol (2022)	ANN for Geospatial	CNN inside ArcGIS Pro compared with SVM	Not mentioned, not detecting oil palm ripeness, 91.00%	CNN gets better accuracy than SVM, but by using built-in CNN, the architecture could not be described clearly
Wibowo *et al*. (2022)	ANN for Remote Sensing	YOLOv3, YOLOv4, YOLOv5	Not mentioned, not detecting oil palm ripeness, 97.00%	There is a certain resolution that affects the result, and for the model, YOLOv3 and YOLOv4 are the best models
Mubin *et al*. (2019)	ANN	CNN using LeNet	260 images, 2 classes of remote sensing, 95.11%	There is a need for an optimiser like AdaGrad because the lowest loss value and high accuracy could be further improved by using high-resolution hyperspectral remote-sensing images.

Considering all the aspects above, this research contributes by optimising the hyperparameters and input size of the EfficientDet-Lite model, which leads to improved performance in the ripeness classification of the Dumpy variety. The findings suggest that with appropriate adjustments to the hyperparameters and input size, the model can achieve favorable results in accurately classifying the ripeness of Dumpy variety fruit bunches.

MATERIALS AND METHODS

The total dataset of oil palm FFB used as training data were 240 images. There were four classes: Unripe, underripe, ripe, and overripe, and each class consisted of 60 images. The consideration of choosing 240 images was because, as in *Table 1*, the usage of the sample images was about 100-200 images if they consisted of 2-4 classes. There is no specific minimum image standardisation that could be used in DL. The validation used 15 images for each category, totaling 60 images outside the training data. A variation of 170 images of unique oil palm FFB has been used. EfficientDet supports a variety of image preprocessing procedures. By default, the supplied image has a 50% probability of being horizontally flipped, serving as a simple image enhancement method.

There was a consideration that some oil palm FFB images were rotated and zoomed in to get an additional number of samples for about 2-5 images from the original 170 images. The variety of oil palms captured was all Dumpy from Pusat Penelitian Kelapa Sawit (PPKS) or Research Centre of Oil Palm. Dumpy is a variety of oil palm with the advantage of a slow growth rate, allowing it to grow for an economic lifespan of 30 years or longer compared to other types. The Dumpy variety of oil palm, also known as DyP Sungai Pancur I (SP-I), can be found in various locations, including Kebun Rayon B, PT Raya Padang Langkat, Ds. Bukit Mas II, Kec Besitang, Kab. Langkat, North Sumatra, Indonesia. The site of the plantation can be accessed through the following Google Maps link: https://maps.app.goo.gl/omzqUWJusKfNCnTs7. The Dumpy variety picture was also taken from several private plantations in Kalimantan. The distance between the camera and the oil palm FFB was about 3-5 m. The smartphone used to take the picture was the Samsung Galaxy J7 with specifications 13 MP, f/1.9, 28 mm (wide). A variation of the camera has been used: An X85 HD 720P Webcam with 5 MP high definition with a resolution of 1920 × 1080.

The researcher collected images of the oil palm FFB using a smartphone camera and varied the shooting angles and backgrounds to create different datasets. With the guidance of experts from Perhutani, predetermined classes were determined, and the data collection was supervised on the plantation. A strong foundation was laid for the oil palm FFB ripeness detection research by working directly with the experts and gathering a comprehensive data set. The criteria used to determine the class was based on the experience of the expert. The main reason for taking an image with supervision was to ensure that the photo's angle could describe the proper criteria for classification.

The second stage was image preprocessing based on the specification in *Table 2*. EfficientDet needs a specific resolution for the input, so the input images were considered using EfficientDet D2 because the backbone layer structure is the same as EfficientDet-Lite2. The image was resized to 768 × 768 from the original resolution of about 1080 × 1920.

The third stage was labeling and annotating each dataset image. Labeling is crucial in EfficientDet object detection by providing ground truth annotations as the reference standard for training the model. The assigned bounding boxes and class labels during the labeling process are used to instruct the model on identifying and locating objects of different classes within an image, considering the diverse backgrounds and conditions in the dataset shown in *Figure 1*. These annotations are instrumental in teaching the model the visual characteristics of various objects, enhancing its ability to generalise and make accurate predictions on unseen patterns. Four different environments and scenarios were used to collect oil palm images. There were 27 photos of palm bunches hanging from the tree, 183 photos on the ground, 11 photos capturing scenes with hanging FFB, and 19 photos of groups with grass in the background. The labeling and annotation process results are saved in an XML file containing information about the labeling box coordinate and width.

The fourth stage was conducting data training on images that had been labeled and annotated. The training process was performed using the Google Collaboratory application. EfficientDet-Lite2 was chosen among other models because the mean Average Precision (mAP) score is above average compared with EfficientDet-Lite0 until EfficientDet-Lite4 (Repák, 2021). This model was implemented as the detection and counting program on the Android device.

EfficientNet, as the base of EfficientDet, introduced a brand-new compound scaling technique that uniformly scales the network's depth, width, and resolution. A new baseline network was created with a neural architecture search. The EfficientNet scaling method evenly

Figure 1. Background and camera angle variation of the oil palm FFB dataset.

scales network breadth, depth, and resolution using a set of preset scaling coefficients, in contrast to standard practice, which scales these variables arbitrarily.

Compound scaling uses a compound coefficient Φ to uniformly scale a network according to guiding principles. The parameters denote the user-specified coefficient in Equation (1) and parameters α, β, and γ define how to apply this to the network depth, width, and resolution.

$$\begin{gathered} \text{Depth: } d = \alpha^{\phi} \text{ Width: } w = \beta^{\phi} \\ \text{Resolution: } r = \gamma^{\phi} \\ \mathrm{f} = \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \\ \alpha \geq 1, \beta \geq 1, \gamma \geq 1 \end{gathered} \tag{1}$$

Compound scaling aims to increase the network's length (Li), width (Ci), and resolution (Hi, Wi) while maintaining the baseline network's preset Fi. Although the design space is supposed to be reduced by fixing Fi, since Li, Ci, Hi, and Wi can still be explored for each layer, the design space is still quite wide. Every layer must be scaled consistently with a constant ratio to constrict the design space further. Compounding scaling, described as an optimisation problem, aims to maximise the model's accuracy given the resource constraints in Equation (2).

$$\begin{aligned} &\max_{d,w,r} && \text{Accuracy } [N(d,w,r)] \\ &\text{s.t.} && N(d,w,r) = \bigodot_{i=1\ldots s} \hat{F}_i^{d\cdot\hat{L}_i}(X_{\langle r\cdot\hat{H}_i, r\cdot\hat{W}_i, w\cdot\hat{C}_i\rangle}) \\ &&& \text{Memory } (N) \leq \text{Target\_memory} \\ &&& \text{FLOPS } (N) \leq \text{Target\_flops} \end{aligned} \tag{2}$$

Where N is Convnet, w, d, and r are coefficients for scaling the network width, depth, and resolution. A sign ⊙ is a list of a layer.

A bidirectional Feature Pyramid Network (BiFPN) improves cross-scale connections by deleting nodes with only one input edge. BiFPN adds an advantage from the original input to the output node based on some level conditions and treats each bidirectional path as a single feature network layer. Different input features have varied resolutions and typically contribute unequally to the output feature. Additional weight is applied to each input, allowing the network to learn the relative relevance of each input attribute.

BiFPN has five modifications over a normal FPN. Instead of only the top-down feature, it adds another bottom-up feature fusion branch, shown with the red arrow in *Figure 2*. It has to skip connections from the initial feature map to the fused feature map, so there is a repeated block with a purple arrow. Nodes with only one input are removed because they do not do much fusion as other nodes. The entire module is repeated multiple times. For different resolution feature maps to contribute to the fusion at various capacities, features are not added directly but instead using a weighted average. There is a normalization of backpropagation because unbounded weights cause issues. Although it worked, applying softmax to the weight values slowed down training. A simple average following Relu activation is utilised to normalise the weights.

The BiFPN, as an additional module in the network, which may also be scaled, necessitates the development of a new scaling technique. The width of BiFPN increases exponentially while the input resolution and depth of BiFPN increase linearly.

This section shows the top-down convolution in the 6th layer of BiFPN. The illustration of feature fusion at level 6 is shown in *Figure 2*. A top-down fusion P_6^{td} has to be computed as in Equation (3). A bottom-up feature fusion is calculated to produce the output from the current level shown in Equation (4). There is a resizing mechanism from P_7^{in} The layer shows a connection for the convolution from top-down and bottom-up as

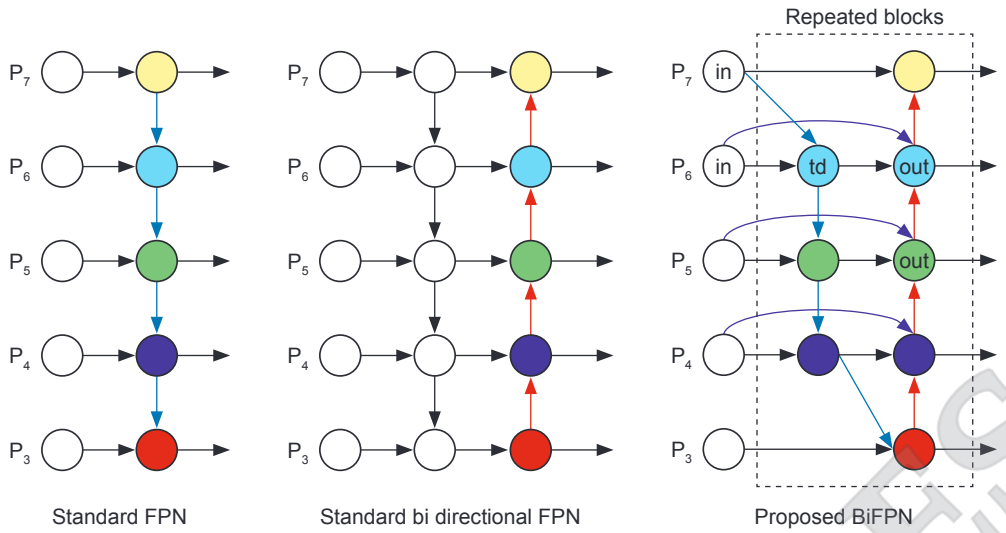

Figure 2. Weighted Bi-directional Feature Pyramid Network (FPN)

the backbone of FPN. Equation (4) shows the P_6^{out} that affected the convolutional in Equation (2) and connected with P_5^{out} so it is connected bi-directional.

$$P_6^{td} = \text{Conv}\left(\frac{w_1 \cdot P_6^{in} + w_2 \cdot Resize\,(P_7^{in})}{w_1 + w_2 + \epsilon}\right) \quad (3)$$

$$P_6^{out} = \text{Conv}\left(\frac{w_1' \cdot P_6^{in} + w_2' \cdot P_6^{td} + w_3' \cdot Resize\,(P_5^{out})}{w_1' + w_2' + w_3' + \epsilon}\right) \quad (4)$$

In general, Pi represents an output from the backbone's i-th layer and is an input to the i-th Bi-FPN level. w_1', w_2', w_3', are the weighted feature fusion. ϵ is a small constant (configurable as part of the constructor arguments).

Top-down and bottom-up fusion are explained in *Figure 3*. The left side in *Figure 3* is the bottom-up approach, which down-sampled the resolution by a factor of 0.5. The middle part is the top-down approach, starting with a coarser-resolution feature map and upsampling by 2. The upsampled map is merged with the corresponding bottom-up map, which walks on a 1 × 1 convolutional layer to reduce channel dimension. This process is iterated until the finest resolution map is generated.

Inspired by the merge scaling used in EfficientNets, a new combined scaling method is proposed for object detection. This method uses coefficients (Φ) to improve the fusion of all dimensions of the backbone network, BiFPN network, class network, and resolution. This research proposes an EfficientDet-Lite2 model using a D2 input size of 768 × 768 in input layer resolution. EfficientDet D2 and EfficientDet-Lite2 use five BiFPN layers to fuse the upsampling and downsampling features. Three ClassNet and BoxNet are used to predict categories and draw the location of the bounding box. The bounding box is the box with the coordinate of the object. An XML file describes the bounding box location while the model was fed to EfficientDet. As a result of the classification, a bounding box was used to determine the exact object location. A complete table of comparison is shown in *Table 2*.

TABLE 2. EFFICIENTDET INPUT, BIFPN, AND CLASSNET LAYER

Model	Input layer	BiFPN layers	ClassNet and BoxNet layers
D0 (Ø = 0)	512	3	3
D1 (Ø = 1)	640	4	3
D2 (Ø = 2)	768	5	3
D3 (Ø = 3)	896	6	4
D4 (Ø = 4)	1024	7	4
D5 (Ø = 5)	1280	7	4
D6 (Ø = 6)	1280	8	5
D7 (Ø = 7)	1536	8	5
Lite0	320	3	3
Lite1	384	4	3
Lite2	448	5	3
Lite3	512	6	4
Lite4	640	7	4

This research proposes EfficientDet-Lite2 based on the trial and error mechanism. The consideration for choosing EfficientDet-Lite2 is the size, latency, and mAP. Size describes the quantised integer model after optimisation. Integer quantisation is an optimisation technique that converts both model's weights and activation outputs from 32-bit floating-point numbers to the

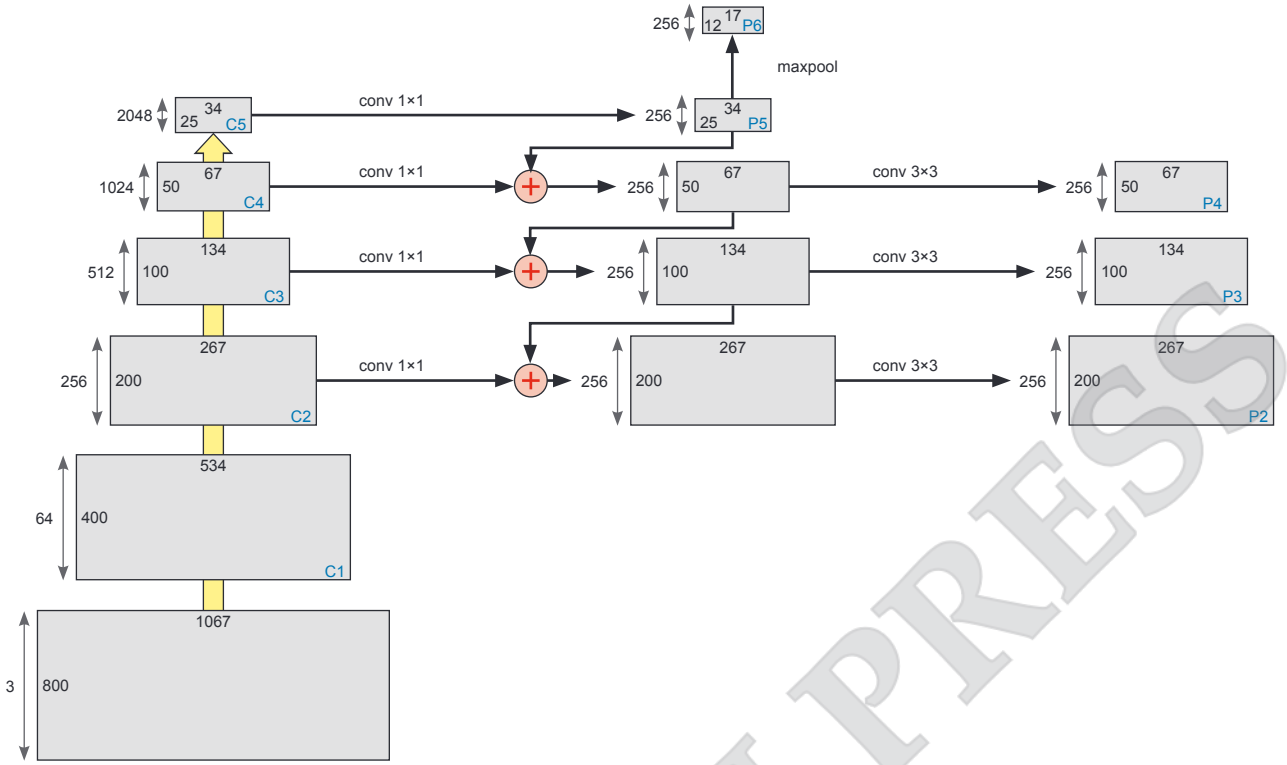

Figure 3. FPN bottom-up and top-down convolutional mechanism.

nearest 8-bit fixed-point numbers. Latency was measured on Google Pixel 4 Smartphone using four threads on the CPU. Average Precision is the mAP on the COCO 2017 validation dataset. EfficientDet-Lite3 mAP is higher but not very significant. The usage of EfficientDet-Lite2 is also a consideration that the model would be implemented on the Android device. The comparison of size, latency, and mAP levels is shown in *Table 3*.

TABLE 3. EFFICIENTDET-LITE SIZE, LATENCY, AND MAP COMPARISON

Model	Size (MB)	Latency (ms)	mAP
EfficientDet-Lite0	4.4	37	25.69%
EfficientDet-Lite1	5.8	49	30.55%
EfficientDet-Lite2	7.2	69	33.97%
EfficientDet-Lite3	11.4	116	37.70%
EfficientDet-Lite4	19.9	260	41.96%

Test on the system's performance was based on the data outside the dataset, batch and epoch variations, and independence accuracy testing. Data training with batch and epoch variations is divided into two: Original and preprocessing data training with batch variations: 8 and 16 and epochs: 20, 50, 100, 200, 300, 400, and 500. The results of data training with batch and epoch variations can be seen in *Table 4*.

RESULTS AND DISCUSSION

Batch size is the sample of images that propagate through the network. A larger batch size would make less backpropagation, so for the 240 samples, there was 15 backpropagation in batch 16 and 30 backpropagation in batch 8. The pattern was that in 300 epochs, each batch gained the top mAP percentage, and then the mAP decreased. There was no significant difference between batch 8 and batch 16 according to the duration of the training and the mAP score, but the mAP score was higher in batch 16.

Detection quality means recognising oil palm FFB objects and performing classification. The confidence level of the detection must be above 50%. The confidence score reflects how likely the box contains an object of interest and how confident the classifier is about it. If no object exists in that box, the confidence score should ideally be zero. Generally, the confidence score tends to be higher for tighter bounding boxes in the case of object detection. The TensorFlow model validation testing only shows one category with the highest confidence. A confidence level below 50% often shows incorrect results, so the threshold is set to 50%.

The test results were entered into a table called the confusion matrix or error matrix, which was used to deliver information on comparing the classification results. The confusion matrix is

shown in *Table 5*. The performance parameter of classification is shown in *Table 6*.

Testing is done on the 60 images beside the dataset captured directly in the plantation in variation with images provided on the internet. While color features in pictures on the internet can still offer some reliability, they may not be as accurate as images taken directly from the fruit on the plantation due to possible variations in lighting and other factors. Still, the aim is to test the possibility of the interpretation and how general the model could classify. The classification performance was not much higher than the accuracy in the previous literature finding, which is above 90% or more.

The testing scenario used images outside the training process and a direct capture at the plantation, as in *Figure 2*. For example, in *Figure 2*,

TABLE 4. EFFICIENTDET-LITE2 BATCH TRAINING 8 AND 16

No	EfficientDet-Lite Version (Batch training 8)	Epoch	Duration (hh:mm:ss)	mAP (%)
1	2	20	00:06:20	58,00366
2	2	50	00:12:42	55,70173
3	2	100	00:23:59	55,09633
4	2	200	00:47:49	56,46802
5	2	300	01:10:00	57,66782
6	2	400	01:22:21	53,56957
7	2	500	02:12:50	52,73226

No	EfficientDet-Lite Version (Batch traning 16)	Epoch	Duration (hh:mm:ss)	mAP (%)
1	2	20	00:07:08	55,47828
2	2	50	00:13:32	58,89936
3	2	100	00:26:18	57,62151
4	2	200	00:53:33	58,88936
5	2	300	01:20:00	59,80644
6	2	400	01:30:00	57,99332
7	2	500	02:16:34	55,68957

TABLE 5. CONFUSION MATRIX OF EFFICIENTDET-LITE2

Category		Prediction				Total
		Overripe	Ripe	Underripe	Unripe	
Actual	Overripe	21	4	0	0	25
	Ripe	1	22	2	0	25
	Underripe	0	3	17	4	25
	Unripe	0	0	1	24	25
	Total	22	29	20	28	100

TABLE 6. CLASSIFICATION PERFORMANCE

No	CNN Model	Device	Accuracy (%)	Precision (%)	Recall (%)	F1-Score
1	EfficientDet-Lite2 (Proposed)	Android	84.00	82.00	85.00	84.00
2	LeNet (Mubin *et al.*, 2019)	PC	95.11			
3	EfficientDet-D5 and YOLOv4 (Ammar *et al.*, 2021)	PC	99.00			
4	AlexNet (Ibrahim *et al.*, 2018)	PC	92.00			
5	ResNet (Harsawardana *et al.*, 2020)	PC	71.34			
6	Fast R-CNN (Prasetyo *et al.*, 2020)	PC	80.00			

the image was directly downloaded from Google search. There were about 50 images that were downloaded from Google that were used for testing. Before the testing process, the consideration for each class was clarified with the expert. The result of the detection was underripe because the FFB was ripe but with juxtaposition with unripe. The detection threshold was set at about 0.5 to ensure that the confidence rate of the intersection of the detection box is relatively high. This mechanism is built on the EfficientDet-Lite2 and could prove that the classification could be done. *Figure 4* shows examples of the classification result for each class.

The flowchart of the Android detection application and usage process is shown in *Figure 3*. There are two modes of the camera: Directly using the mobile phone camera or using the webcam USB-based camera connected to the mobile phone. After choosing the camera mode, the classifier must scan the aiming process with the TensorFlow Lite model, as shown in *Figure 5*. If there is a detection, the process continues to capture the data. If no detection results have occurred, then the process of taking the picture is repeated.

TensorFlow Lite would determine the class and bounding box based on the classification of the EfficientDet-Lite2 embedded in the application.

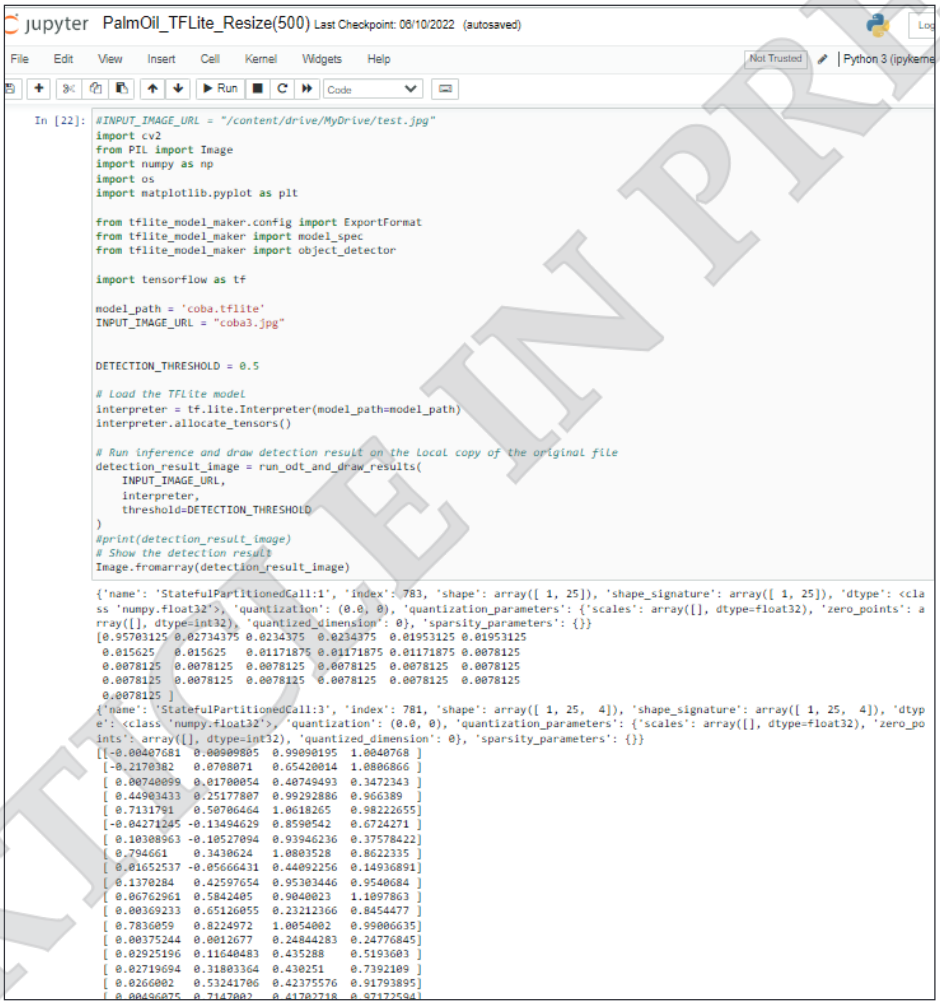

Figure 4. Classification mechanism in FFB ripeness detection.

After the detection was done and shown in the bounding box, the approved counting result was saved to the phone database. Approved counting result was done by selecting the save button icon, as shown in *Figure 6*. After being approved, the result was saved in the Android SQLite database.

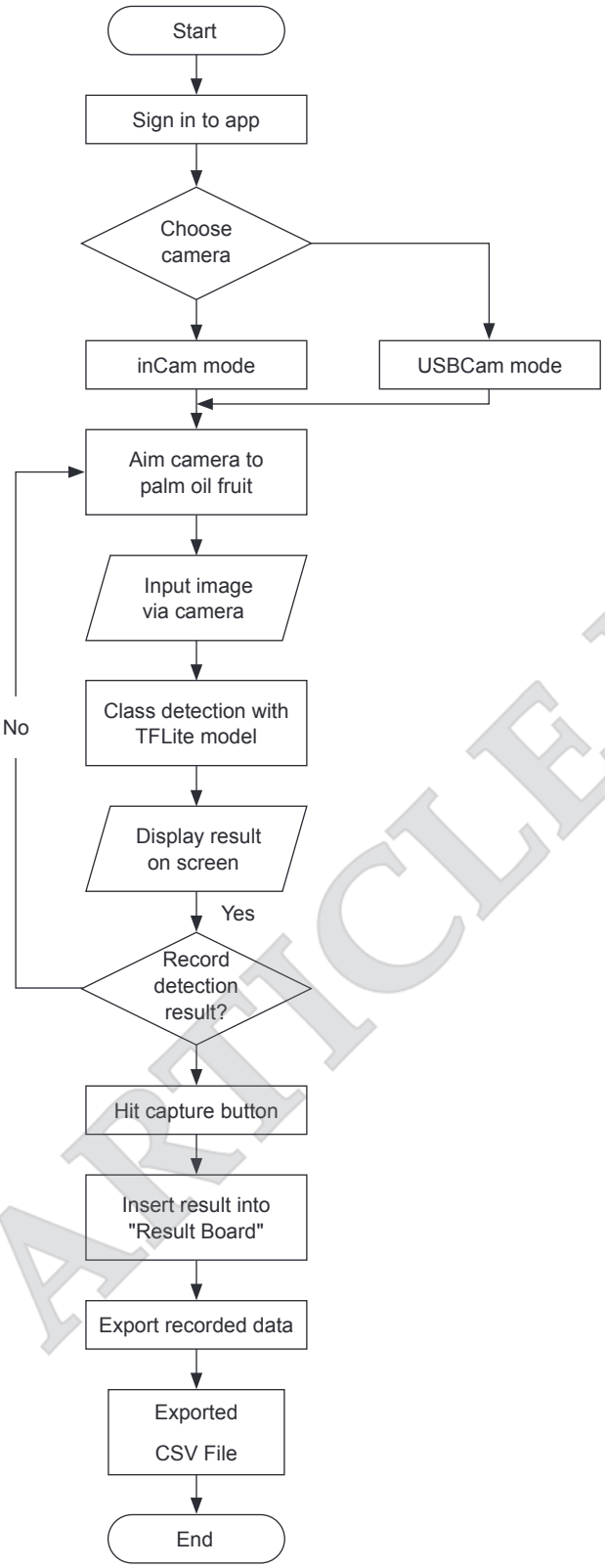

Figure 5. Classification Flowchart on the mobile phone.

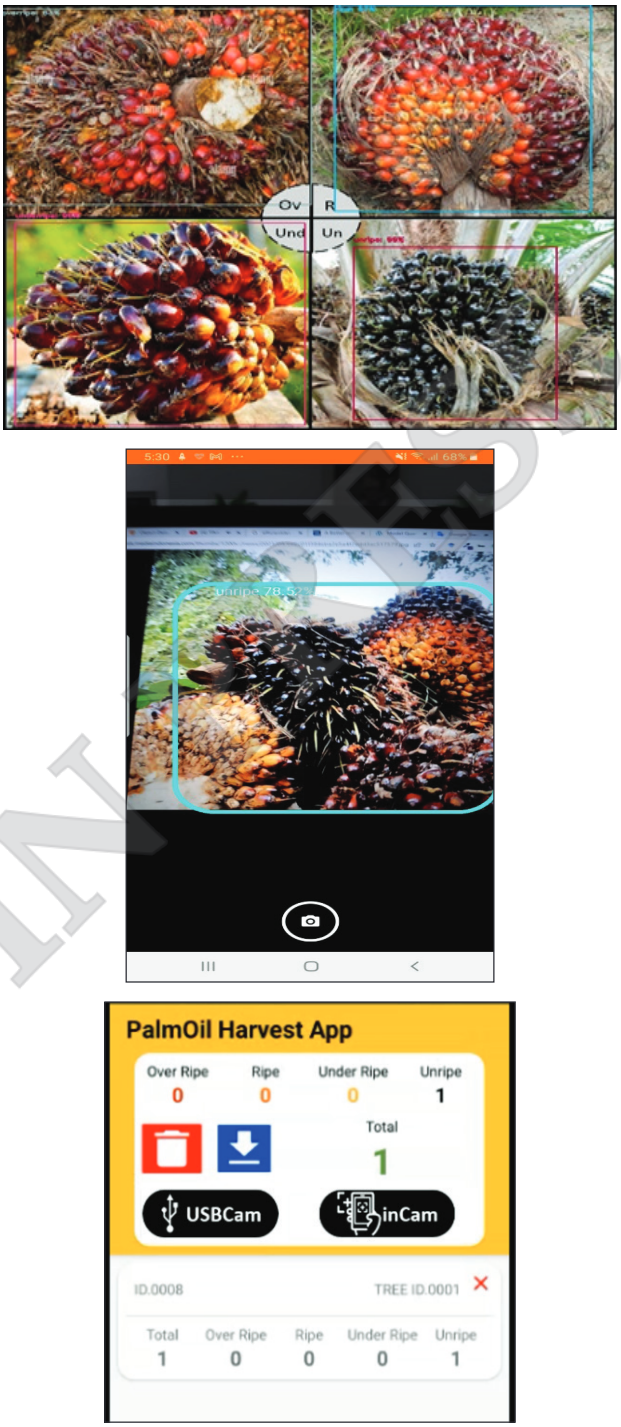

Figure 6. Classification result after taking a picture directly of the oil palm FFB.

The accuracy of EfficientDet-Lite2 is about 84%, which shows that the model could fit the situation using an image outside the dataset. Despite having slightly lower accuracy than previous models, with around 90% accuracy, the developed model demonstrated its feasibility for application in a real-world plantation setting, including FFB on the trees. The challenge is that the state-owned enterprise plantation has a high standard to determine ripeness, whereas, in private plantations,

the ripeness standard is lower, so the half-ripe or underripe can be considered ripe. In the testing scenario of the Android application, sometimes the class and bounding box results could differ in a flash condition when there is a fusion area between black-colored and orange-red parts. The overall performance is acceptable to be used in the big plantation situation.

CONCLUSION

This research applied the EfficientDet-Lite2 with proposed hyperparameter tuning in the Android application. This application can classify oil palm ripeness into four categories: Unripe, underripe, ripe, and overripe. Using a back-and-forth convolutional process, Bi-FPN could do the scaling variation to capture the image's feature from the coarse once related until a more detailed feature. EfficientDet, as the embodiment of the EfficientNet backbone with the D model, showed that resolution variation could effectively help the detection of the oil palm with striking differences in certain conditions. The proposed model uses EfficientDet-Lite2 as the backbone network and uses the input size of EfficientDet D2 to get more detailed features. EfficientDet-Lite2 had a good mAP score and could be used on mobile phones with an accuracy of 84%.

ACKNOWLEDGEMENT

This work is supported and fully funded by the Strategic Research Annual Budget Grant from the Electrical Engineering Department, part of the Faculty of Engineering at Diponegoro University.

REFERENCES

Alfatni, M S M; Mohamed Shariff, A R ; Ben Saaed, O M; Albhbah, A M and Mustapha, A (2020). Colour feature extraction techniques for real time system of oil palm fresh fruit bunch maturity grading. *IOP Conf. Series: Earth Environ. Sci., 540(1)*: 012092. DOI: 10.1088/1755-1315/540/1/012092.

Ammar, A; Koubaa, A and Benjdira, B (2021). Deep-learning-based automated palm tree counting and geolocation in large farms from aerial geotagged images. *Agronomy, 11(8)*: 1458. DOI: 10.3390/agronomy11081458.

Guo, J M; Yang, J S; Seshathiri, S and Wu, H W (2022). A light-weight CNN for object detection with sparse model and knowledge distillation. *Electronics, 11*: 575. DOI: 10.3390/electronics11040575.

Harsawardana; Rahutomo, R; Mahesworo, B; Cenggoro, T W; Budiarto, A; Suparyanto, T; Surya Atmaja, D B; Samoedro, B and Pardamean, B (2020). AI-based ripeness grading for oil palm fresh fruit bunch in smart crane grabber. *IOP Conf. Series: Earth Environ. Sci., 426(1)*: 012147. DOI: 10.1088/1755-1315/426/1/012147.

Ibrahim, Z; Sabri, N and Isa, D (2018). Palm oil fresh fruit bunch ripeness grading recognition using convolutional neural network. *J. Telecommun. Electron. Comput. Engin., 10(3-2)*: 109-113.

Khalid, N and Shahrol, N A (2022). Evaluation the accuracy of oil palm tree detection using deep learning and support vector machine classifiers. *IOP Conf. Ser.: Earth Environ. Sci., 1051(1)*: 012028. DOI: 10.1088/1755-1315/1051/1/012028.

Khan, N; Kamaruddin, M A; Sheikh, U U; Yusup, Y and Bakht, M P (2021). Oil palm and machine learning: Reviewing one decade of ideas, innovations, applications, and gaps. *Agriculture 2021, 11*: 832. DOI: 10.3390/agriculture11090832.

Mubin, N A; Nadarajoo, E; Shafri, H Z M and Hamedianfar, A (2019). Young and mature oil palm tree detection and counting using convolutional neural network deep learning method. *Int. J. Remote Sens., 40*: 7500-7515. DOI: 10.1080/01431161.2019.1569282.

Prasetyo, N A; Pranowo, P and Santoso, A J (2020). Automatic detection and calculation of palm oil fresh fruit bunches using faster R-CNN. *Int. J. Appl. Sci. Eng., 17*: 121-134. DOI: 10.6703/IJASE.202005_17(2).121.

Repák, T (2021). Fast object detection on mobile platforms using neural networks. Bachelor Thesis, Masaryk University.

Saleh, A Y and Liansitim, E (2020). Palm oil classification using deep learning. *Sci. Infor. Tech. Letters, 1(1)*: 1-8. DOI: 10.31763/sitech.v1i1.1

Septiarini, A; Hamdani, H; Hatta, H R and Kasim, A A (2019). Image-based processing for ripeness classification of oil palm fruit. *2019 5th International Conference on Science in Information Technology (ICSITech)*. IEEE, Indonesia. p. 23-26.

Septiarini, A (2021). Machine vision for the maturity classification of oil palm fresh fruit bunches based on color and texture features. *Sci. Hortic., 286(2021): 110245*. DOI: 10.1016/j.scienta.2021.110245.

Shiddiq, M; Salambue, R; Poja, R and Solistio, A T (2016). Analysis of physical properties of oil palm

fresh fruit bunches using ImageJ. *Appl. Sci. Technol., 1(1)*: 388-394.

Suharjito; Elwirehardja, G N and Prayoga, J S (2021). Oil palm fresh fruit bunch ripeness classification on mobile devices using deep learning approaches. *Comput. Electron. Agric., 188*: 106359. DOI: 10.1016/j.compag.2021.106359.

Sukemi and Sukrisno, E (2019). Identification using the K-means clustering and gray level co-occurance matrix (GLCM) at maturity fruit oil head. *2019 Fourth International Conference on Informatics and Computing (ICIC)*. IEEE, Indonesia. p. 1-4.

Tan, M; Pang, R and Le, Q V (2020). EfficientDet: Scalable and efficient object detection. *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (2020)*. p. 1-10

Wibowo, H; Sitanggang, I S; Mushthofa, M and Adrianto, H A (2022). Large-scale oil palm trees detection from high-resolution remote sensing images using deep learning. B*ig Data Cogn. Comput. 2022, 6(3):* 89.