Heliyon 9 (2023) e13036

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

Video based oil palm ripeness detection model using deep learning

Franz Adeta Junior [a], Suharjito [b,*]

[a] Computer Science Department, BINUS Graduate Program – Master of Computer Science, Bina Nusantara University, Jakarta, 10480, Indonesia
[b] Industrial Engineering Department, BINUS Graduate Program – Master of Industrial Engineering, Bina Nusantara University, Jakarta, 11480, Indonesia

ARTICLE INFO

Keywords:
Oil palm
Object detection
Real-time
Deep learning

ABSTRACT

Research on oil palm detection has been carried out for years, but there are only a few research that have conducted research using video datasets and only focus on development using non-sequential image. The use of the video dataset aims to adjust to the detection conditions carried out in real time so that it can automatically harvest directly from oil palm trees to increase efficiency in harvesting. To solve this problem, in this research, we develop an object detection model using a video dataset in training and testing. We used the 3 series YOLOv4 architecture to develop the model using video. Model development is done by means of hyperparameter tuning and frozen layer with data augmentation consisting of photometric and geometric augmentation experiment. To validate the outcomes of the YOLOv4 model development, a comparison of SSD-MobileNetV2 FPN and EfficientDet-D0 was performed. The results obtained show that YOLOv4-Tiny 3L is the most suitable architecture for use in real time object detection conditions with an mAP of 90.56% for single class category detection and 70.21% for multi class category detection with a detection speed of almost 4× faster than YOLOv4-CSPDarknet53, 5× faster than SSD-MobileNetV2 FPN, and 9× faster than EfficientDet-D0.

1. Introduction

Palm oil is one of the largest export commodities in Indonesia and one of the largest contributors to the world's demand for processed palm oil products. Because processed palm oil products can be utilized for a variety of essential necessities such as food, fuel, and oleo-chemicals, the need for palm oil grows up to 18.43% in 2020. The maturity level of oil palm is critical since it demonstrates the quality of the oil palm itself and influences the amount of oil extracted (OER) [1]. Procedure of determining the maturity level of oil palm occurs during the harvesting session and recorded in a log book. However, the method used in general is to determine the maturity level manually by human resources. Because of the high subjectivity of the manual approach, the determined level of palm oil maturity is inconsistent, resulting in frequent confrontations between the determinants of the maturity level and the customer [2]. Furthermore, the manual technique incurs high production costs [3] due to the vast number of human resource requirements and takes a long time to sort oil palm, causing the operations to become inefficient [4]. Based on the problems described, it is critical to establish an accurate detection of oil palm maturity level utilizing real-time computer vision so that the palm oil industry's production operation becomes more efficient in terms of production costs and time used to sort oil palm.

Majority of current research on the maturity level of oil palm is conducted utilizing a classification system without object localization. The method used is traditional machine learning [5–7] and convolutional neural network (CNN) [2–4,8,9]. According to

* Corresponding author.
E-mail address: suharjito@binus.edu (Suharjito).

https://doi.org/10.1016/j.heliyon.2023.e13036
Received 8 August 2022; Received in revised form 11 January 2023; Accepted 12 January 2023
Available online 18 January 2023

existing studies, CNN outperforms traditional machine learning in image processing in the form of still images and video [10]. There is an object detection study using You Only Look Once version 4 (YOLOv4) on the level of oil palm maturity, but the number of classes used is limited to 3 classes. The dataset used is a single image [1]. In general, the limitation of previous studies is that there is no method to detect the maturity level of oil palm FFB using a video dataset. Input information in the form of video is critical for real-time model implementation. There are some features in video data that are similar to how the human eye behaves, such as dynamic illumination, things being obstructed by other objects, and motion blur while transitioning between frames [11].

Despite the limitations of past research, we purpose a method based on a real-time approach with object detection and video datasets for detect the maturity level of palm oil. Object detection is a method in the field of computer vision to detect the location of objects in an image by providing a bounding box for the object to be detected. Object detection can be applied in various fields such as performing automatic calculations [12], monitoring anomalies [13], autonomous driving on vehicles [14], face mask detection [15] and can be combined with embedded systems to perform other specific tasks [16]. The object detection algorithm can identify the desired object by giving a bounding box label as an input to train the algorithm to be used or more commonly known as labeling. The development of object detection can be said to be quite impressive because it can be applied to various fields, for example in conducting object detection in the agricultural industry. In the agricultural industry, there are challenges in cutting operational costs such as laborers who are tasked with harvesting fruit and determining maturity levels accurately. Therefore, object detection in real time is one solution to be able to cut operational costs. However, there are still not many studies that evaluate the object detection of the maturity level of oil palm FFB using data in the form of videos which of course will be very useful for detection purposes in real time. Since video dataset are rich in temporal information and have a wide range of camera angles, they can provide the model with a knowledge of real-time situations [17].

YOLOv4 is state of the art in object detection which is the result of the development of YOLOv3 research [18] and has been tested in agriculture with satisfactory results such as automating pear counting [12], apple detection [19] and citrus fruit ripeness detection [20]. YOLOv4 employs a DarkNet framework based on the low-level programming language to do fast computing, which is ideal for implementing conditions in real time. The backbone, neck, and head comprise the basic structure of YOLOv4. The backbone extracts feature from the input, whilst the neck performs feature aggregation to collect feature information from the shallow layer, which really is useful for object localization, and semantic information from the deeper layers. All features that have been collected are used for object detection in the head section with different feature scales to detect small, medium, and large objects [21]. The YOLOv4 method is the focus of this research's model development, and it will be compared to state-of-the-art baseline methods along with Single Shot Multibox Detector (SSD) [22] and EfficientDet-D0 [23]. SSD consists of two major components, feature extraction and additional convolution layers for object recognition. SSD employed VGG16 to extract features in the early stages of development [22]. The object detection process then employs numerous feature maps of varying scales to detect things of various sizes. Recently, there has been a model development on SSD using MobilenetV2 for feature extraction and feature pyramid layer (FPN) which is applied to harvest crops in the agricultural sector [24]. Meanwhile, EfficientDet uses a weighted bi-directional feature pyramid network (BiFPN) and compound scaling to increase the efficiency of the object detection architecture. There have been many studies using EfficientDet and this method is often compared to YOLO such as in the study of lemon fruit detection [25] and automatic apple harvesting using robot [26]. EfficientDet is a one stage detector that has a good detection speed so that it can be applied for real-time detection. In this study, a comparison between YOLOv4 with SSD-MobileNetV2 FPN and EfficientDet-D0 will be carried out.

There are various types of architecture from YOLOv4, in this research (1) YOLOv4-CSPDarknet, (2) YOLOv4-Tiny, and (3) YOLOv4-Tiny 3L which will be compared with the baseline model of SSD-MobileNetv2 FPN and EfficientDet-D0. In this research, we conducted an experiment to detect 6 levels of maturity of oil palm FFB using a video dataset as a novel method for oil palm fruit detection. This study uses 3 types of YOLOv4 architecture to compare in terms of model performance. There are 4 stages in developing the model: (1) Augmentation data experiment, (2) Hyperparameter fine-tuning, (3) Frozen layer experiment, and (4) tuning combination. The following are the main contributions made in this research:

- Develop a model in detecting the maturity level of oil palm fresh fruit bunches using a real-time detection approach with a dataset in the form of videos on YOLOv4
- The effect of photometric and geometric augmentation on the detection of oil palm fruit maturity level using video dataset
- Find the optimal hyperparameters tune for detecting and classifying oil palm fruit maturity levels by learning rate and batch size combination scenario
- Perform training using video dataset and perform test on videos that have only 1 category and videos that have combination of class category.
- Comparing the performance of YOLOv4 with 2 different state-of-the-art methods to determine the best model in terms of detection speed and accuracy in video test set
- Tested the impact of the number of frozen layers used during training on the detection of oil palm maturity levels with the YOLOv4-CSPDarknet53 architecture

2. Materials and methods

2.1. Dataset

We use the dataset in the form of videos. In the video shooting process, oil palm fruit bunches with various types of maturity are collected at the oil palm grading site in Central Kalimantan. The video is taken using a smartphone in an outdoor condition with

various positions and lighting conditions. When recording the videos, 360° rotation is carried out so that the entire part of the oil palm fruit bunch can be seen, Fig. 1(A)–(F) is an example of frame chunks in captured video. The dataset has 6 categories of oil palm maturity: (1) ripe, (2) over_ripe, (3) under_ripe, (4) unripe, (5) empty, and (6) abnormal. The format of videos is saved in MP4 (Motion Picture Experts Group-4) format with 1280 × 720 pixels resolution. The captured video has 30 frames every 1 s. All images from each category can be seen in Fig. 2(A)–(F). Total of data that we use in this research are 47 videos for training and 10 videos outside of training for testing needs. The ratio of train and validation used is 7:3. Video dataset is used to adapt to the scenario of performing real-time detection. The model in this work was developed using a dataset with picture quality collected using a smartphone camera so that it may be more applied at the time of detection using affordable equipment. In performing real-time palm oil detection, it does not require a super high resolution data because the object detection target is an object with a relatively large size. Object detection against sequential images of oil palm fresh fruit bunches can be developed back into the next research such as object tracking. However, first we want to know the performance of the object detection model trained using the video dataset. Current research focuses on non-sequential image data so we propose a new method to detect the maturity level of oil palm fruit bunches using a video dataset.

2.2. The purpose method

The method utilized in this research consist of pre-processing step that extracts images from the collected video for use as input into the object detection. During this pre-processing session, data augmentation is also performed on images extracted from videos. After that, there are four stages of model development: (1) Data augmentation experiment, (2) Hyperparameter tuning, (3) Frozen layer experiment, and (4) Tuning combination. After all the experiments are completed, we will combine the best configuration from each experiment and do the final model evaluation with video test set. We use YOLOv4 with CSPDarknet53 backbone, YOLOv4 Tiny with 2 detection layer, and YOLOV4 Tiny with 3 detection layer architectures for object detection. Illustration of the pre-processing method can be seen in Fig. 3, and the model development stages can be seen in Fig. 4. Various kinds of model development are carried out aiming to determine the type of configuration that is suitable for the type of dataset used so that the model becomes more robust. We use 3 different YOLOv4 architectures, SSD-MobileNetv2 FPN, and EfficientDet-D0 to compare performance in terms of accuracy and speed in detecting the maturity level of oil palm fresh fruit bunches.

2.2.1. Pre-processing

The preprocessing stage starts from extracting frames on the video dataset for use in the training process. Frame extraction is done

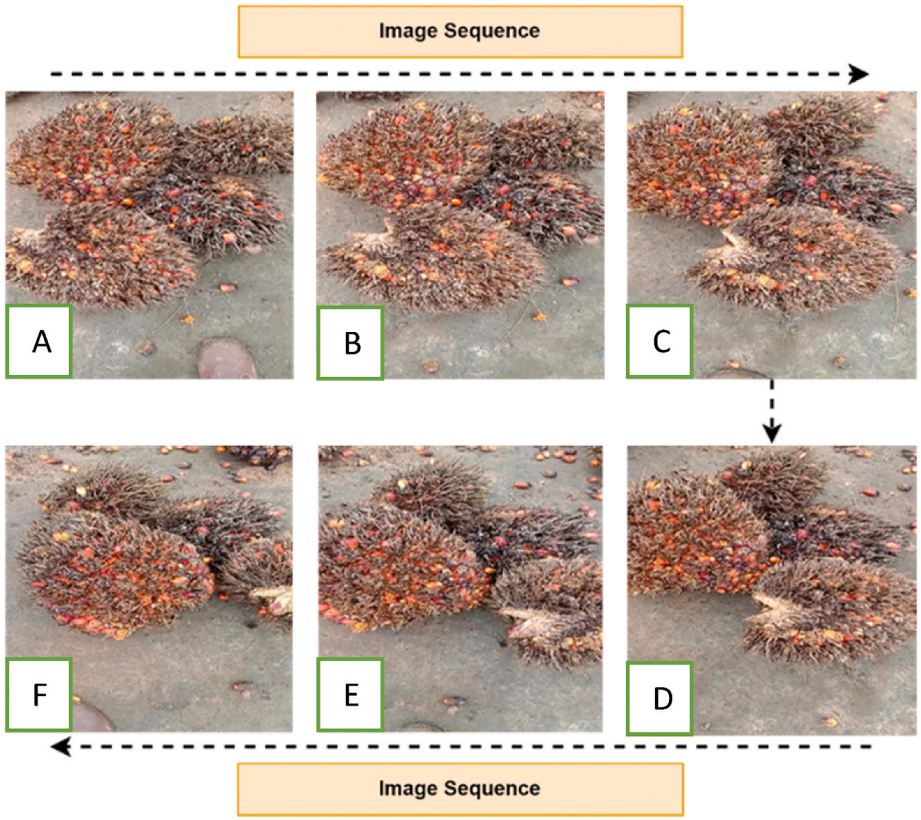


Fig. 1. Sample of sequence frame in captured video show different position of palm oil fresh fruit bunch. Sequence of frames starting from (A) to (F).

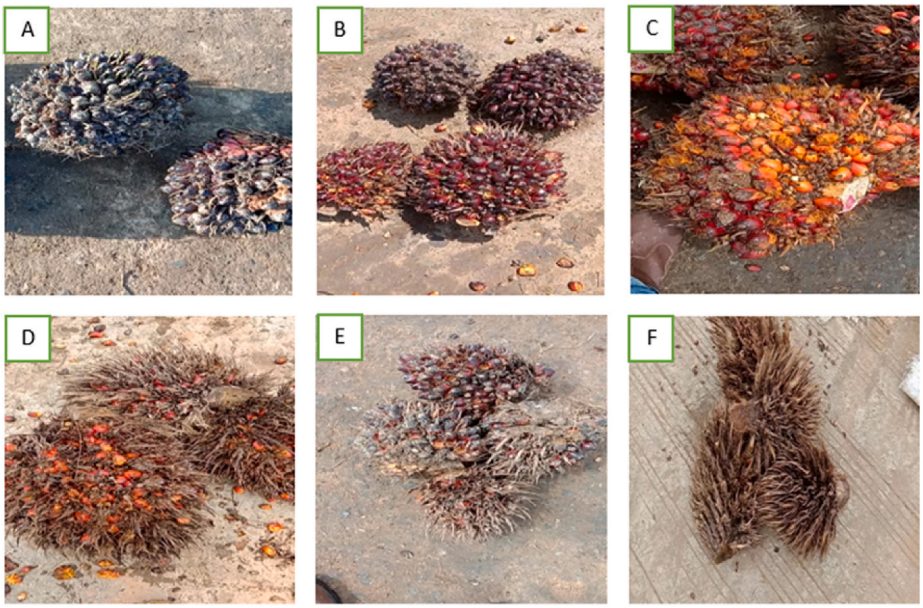


Fig. 2. The example of 6 oil palm FFB categories: (A) unripe, (B) under-ripe, (C) ripe, (D) over-ripe, (E) abnormal, (F) empty fruit bunch.

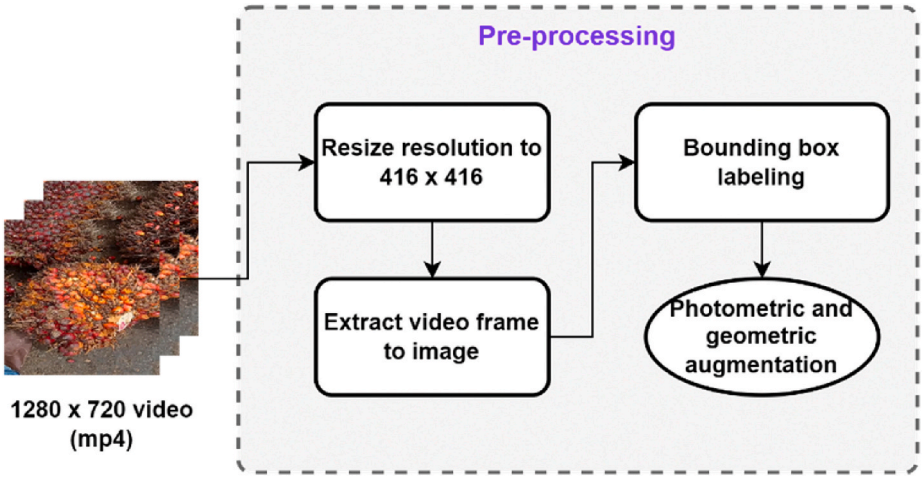


Fig. 3. Illustration of pre-processing stage in this research.

using the VLC media player application using a ratio of 30 which means it takes 1 image frame every 1 s to avoid redundancy [27]. The images obtained for training are 929 frames extracted from 49 videos, after that proceed to the stage of providing bounding boxes for each category. Data that has been given a bounding box will be augmented with 4 types of augmentation: (1) Gaussian blur, (2) random brightness, (3) 45° and 90° rotation and (4) image translation. The images generated from the augmentation process are 4645 images. Rotation by 45° as many as 420 images and rotation by 90° by 509 images. Examples of images that have been augmented can be seen in Fig. 5(A)–(F). The ratio of train and validation used is 7:3. Photometric and geometric data augmentation is carried out so that the model becomes robust in detecting the maturity level of oil palm fresh fruit bunches against different luminance and fruit positions. The test data used in this study consisted of 2 types of video data: (1) single class category, and (2) multi class category. Single class category data consists of 6 videos, each class has 1 video to test and multi category data has 4 videos consisting of a combination of maturity levels: (1) abnormal-underripe, (2) empty-ripe, (3) overripe-empty, and (4) overripe-unripe. The difference between the two types of test video data can be seen in Fig. 6(A) and (B).

2.2.2. Object detection model

2.2.2.1. YOLOv4. YOLOv4 is a model that has a variety of architectural options. In this study, there are 3 types of architecture used:

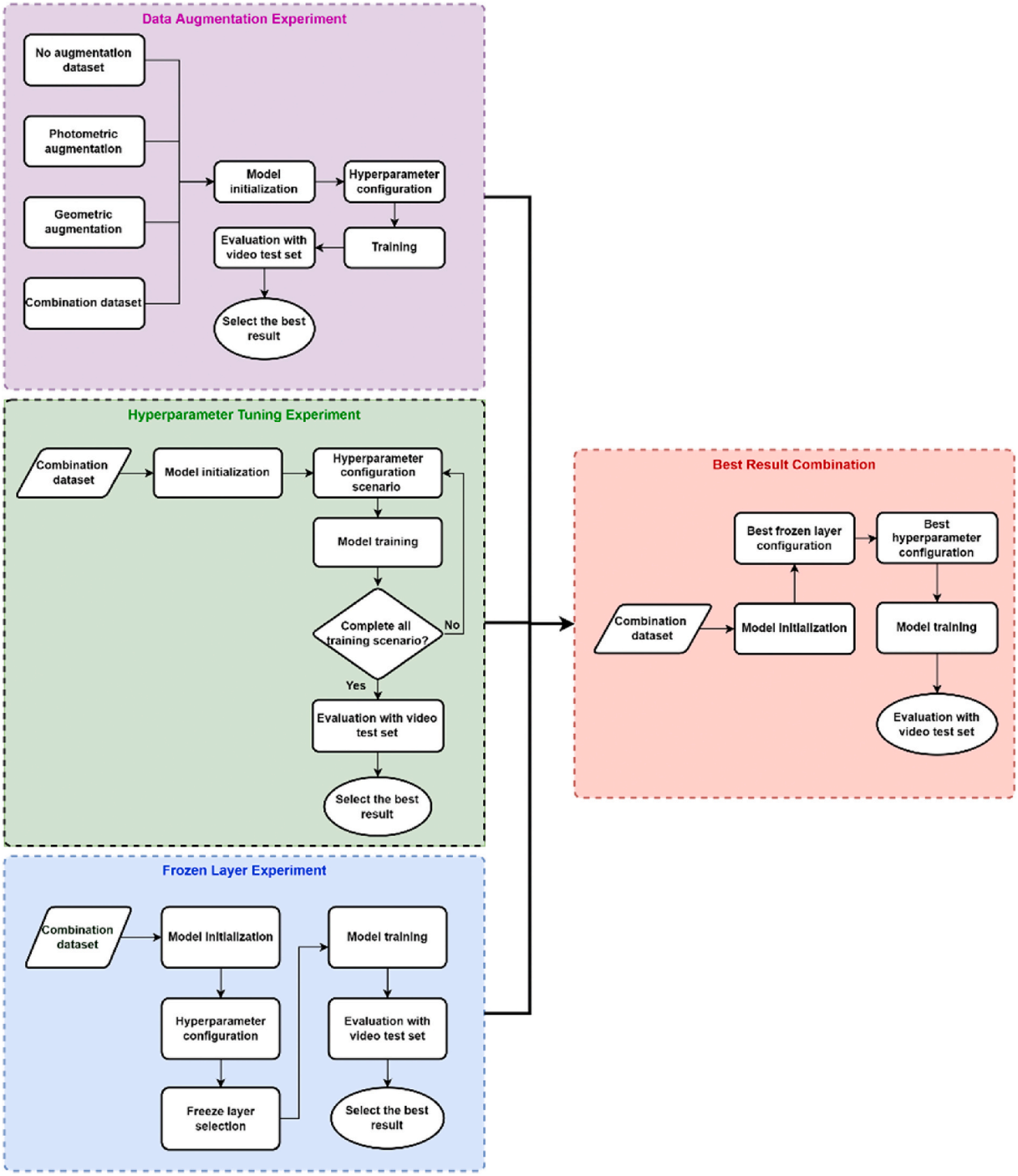


Fig. 4. Illustration of model development stages in this research.

(1) YOLOv4-CSPDarknet53 [28], (2) YOLOv4-Tiny [29], and (3) YOLOv4-Tiny 3L [30]. In every architecture generally has 3 important parts: (1) Backbone, (2) Neck, and (3) Head. The backbone is used to perform feature extraction. Neck functions to aggregate the features contained in the backbone and head to detect the objects. CSPDarknet53 is the result of a combination of Darknet53 [18] and CSPNet [31] architectures. Darknet53 itself has a depth of 53 layers containing convolutional layers, batch normalization, and activation functions. The merging of CSP blocks makes Darknet53 have a better ability to overcome vanishing gradients by combining the original features in the base layer with features that have gone through the convolution process. CSP allows the architecture to achieve a rich feature combination by dividing the base layer into 2 parts: (1) the base layer as input to the dense layer which contains

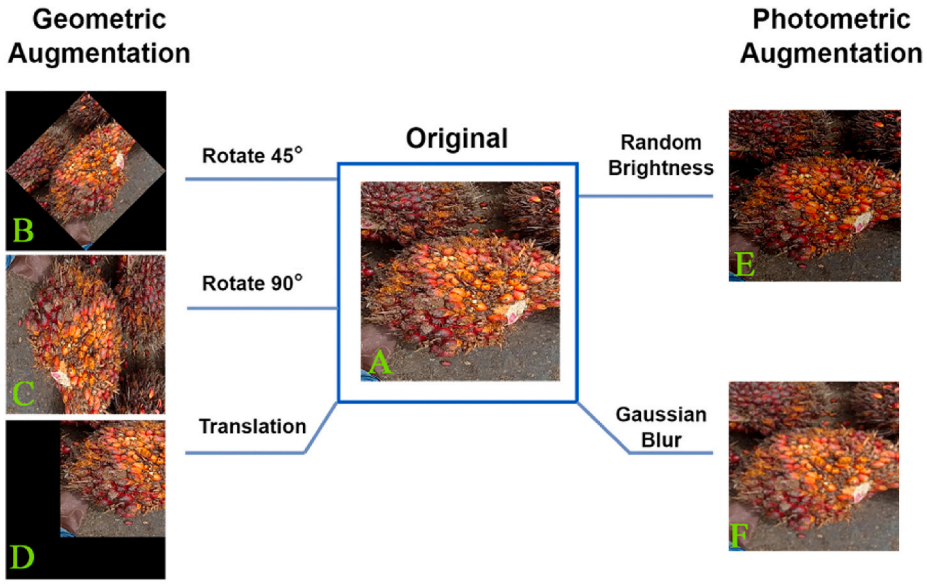


Fig. 5. The example of augmentation types result: (A) Original image, (B) 45° rotate, (C) 90° rotate, (D) Image translation, (E) Random brightness, (F) Gaussian Blur.

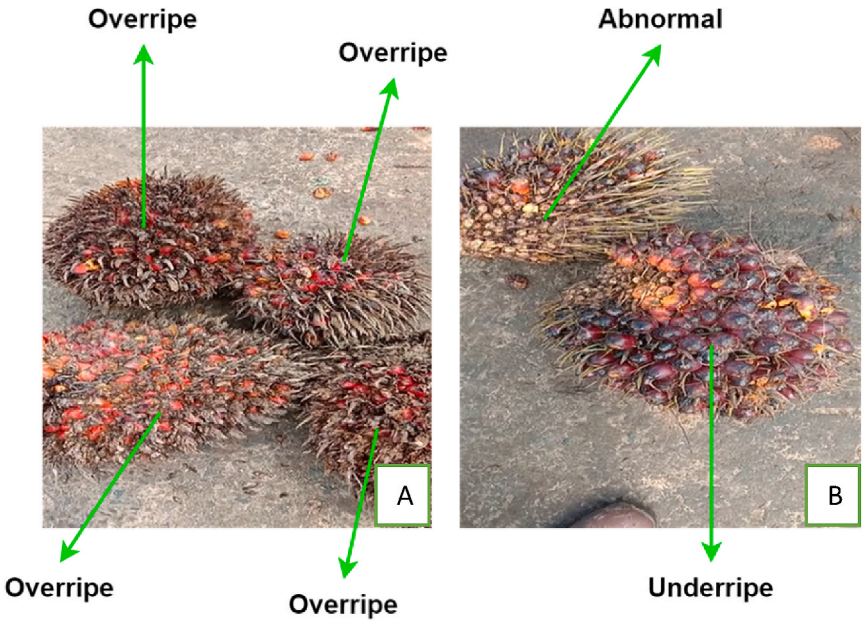


Fig. 6. Different types of test data: (A) single category, (B) multi-category.

convolutional, batch normalization, and activation functions, and (2) the ordinary base layer. Then the output of the 2 parts is concatenated. The other difference between CSPDarknet53 and CSPDarknet53-Tiny lies in the fewer number of convolution layers in the CSP blocks [32]. SPP is also used in the YOLOv4-CSPDarknet53 architecture before entering the fully connected layer. SPP performs pooling using spatial bins on 3 different image scales, the size of the spatial bins is proportional to the image. This method avoids performing repetitive convolution calculations and can increase the accuracy of the detector [33]. For the neck section PANet is used which has 3 objectives: (1) Abbreviate the information path and improve the feature pyramid with precise localization signals existing in low levels, (2) Recover broken data path between every proposal and all feature levels, and (3) Capture different perspectives from each proposal [34]. YOLOv3 [18] is used as a detector on the head to detect images at different scales. YOLOv4-CSPDarknet53 and YOLOv4-Tiny 3L use 3 different scales: (1) $52 \times 52 \times 33$, (2) $26 \times 26 \times 33$, and (3) $13 \times 13 \times 33$. While on YOLOv4-Tiny it uses only 2 scales: (1) $13 \times 13 \times 33$, and (2) $26 \times 26 \times 33$. In addition, the number of layers used in YOLOv4-CSPDarknet53, YOLOv4-Tiny, and

YOLOv4-Tiny 3L respectively are 161, 37, and 44. We made a few changes to the activation function YOLOv4-CSPDarknet53 [28] by changing all activation functions to Mish and eliminating down sampling in the second and third detection layers. Illustration of the YOLOv4 architecture used can be seen in Figs. 7–9. Three types of YOLOv4 architecture are used to compare model performance in terms of mAP, IoU and detection speed.

2.2.2.2. SSD and EfficientDet. The models used for comparison are SSD-MobileNetV2 FPN and EfficientDet-D0 in the repository provided by the researchers as an API for training, inference and evaluation of several object detection models [35]. The available models are pre-trained models that were trained using the MS COCO Dataset. The input data used on the SSD-MobileNetV2 FPN is 320 × 320, while the EfficientDet-D0 uses 512 × 512 input. In Table 1, it can be seen in more detail the baseline hyperparameter configuration used.

2.2.2.3. Model training. The training of YOLOv4 models is done by using DarkNet framework in Google Colaboratory so that the costs used to develop the model are relatively low. DarkNet is a versatile framework that is compatible with this research since it has created

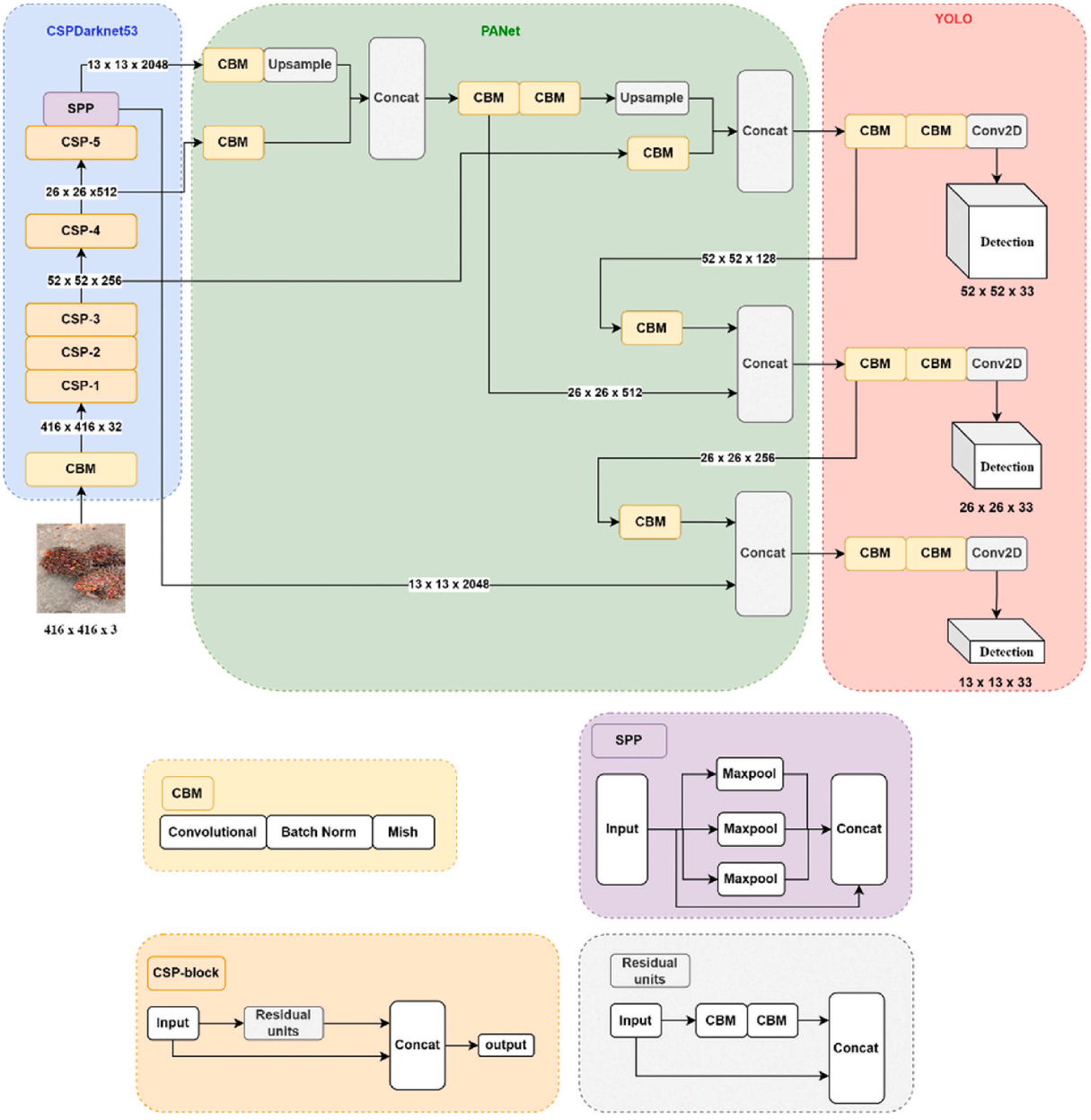


Fig. 7. YOLOv4-CSPDarknet53 architecture used in this research.

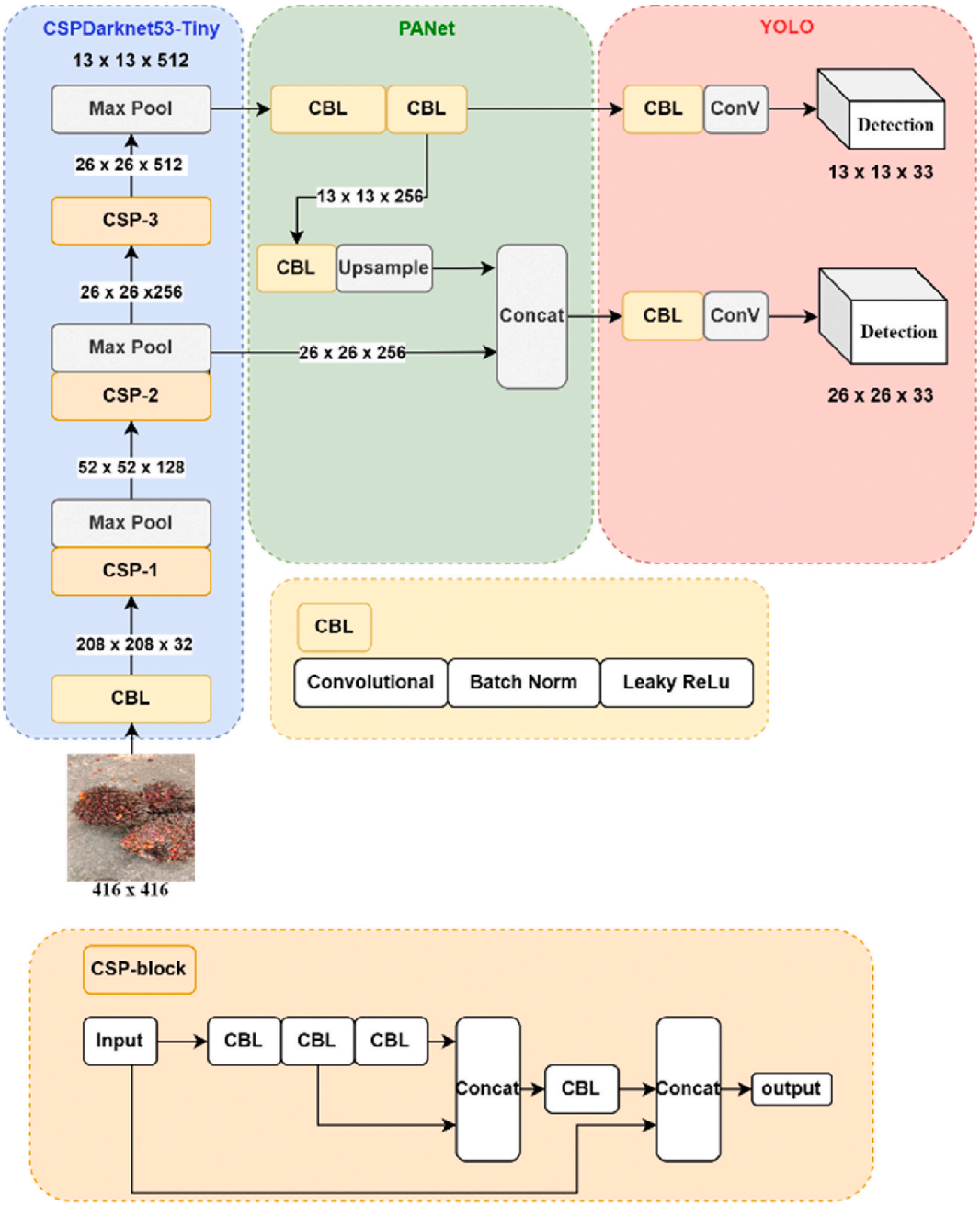


Fig. 8. YOLOv4-Tiny architecture used in this research.

cutting-edge real-time object identification algorithms like YOLOv2 [36], YOLOv3 [18], and YOLOv4 [21]. The hardware specifications used are Nvidia Tesla P100 with 54.8 GB RAM (Random Access Memory) with CUDA version 11.2. The configuration used is 12, 000 max batches based on the calculations suggested in the YOLOv4 research in Eq. (1) as follows:

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

In optimizing, a learning rate schedule is used based on steps with a scale of 0.1, 0.1 in steps 9600 and 10,800, which means the initial learning rate will be multiplied by 0.1 at the 9600 iterations and 0.1 again at the 10,800 iterations. Momentum value used in YOLOv4-CSPDarknet53 is 0.949 and in YOLOv4-Tiny series it is 0.9. The decay value for all architectures used is 0.0005. Meanwhile, the value of the learning rate and batch size is determined based on the hyperparameter tuning experiment in Tables 3 and 4. The input size used is 416 × 416. We use transfer learning method with initial weights that have been trained using MS COCO (Miscrosoft Common Objects in Context) dataset because transfer learning techniques have been proven to increase the accuracy of the model used

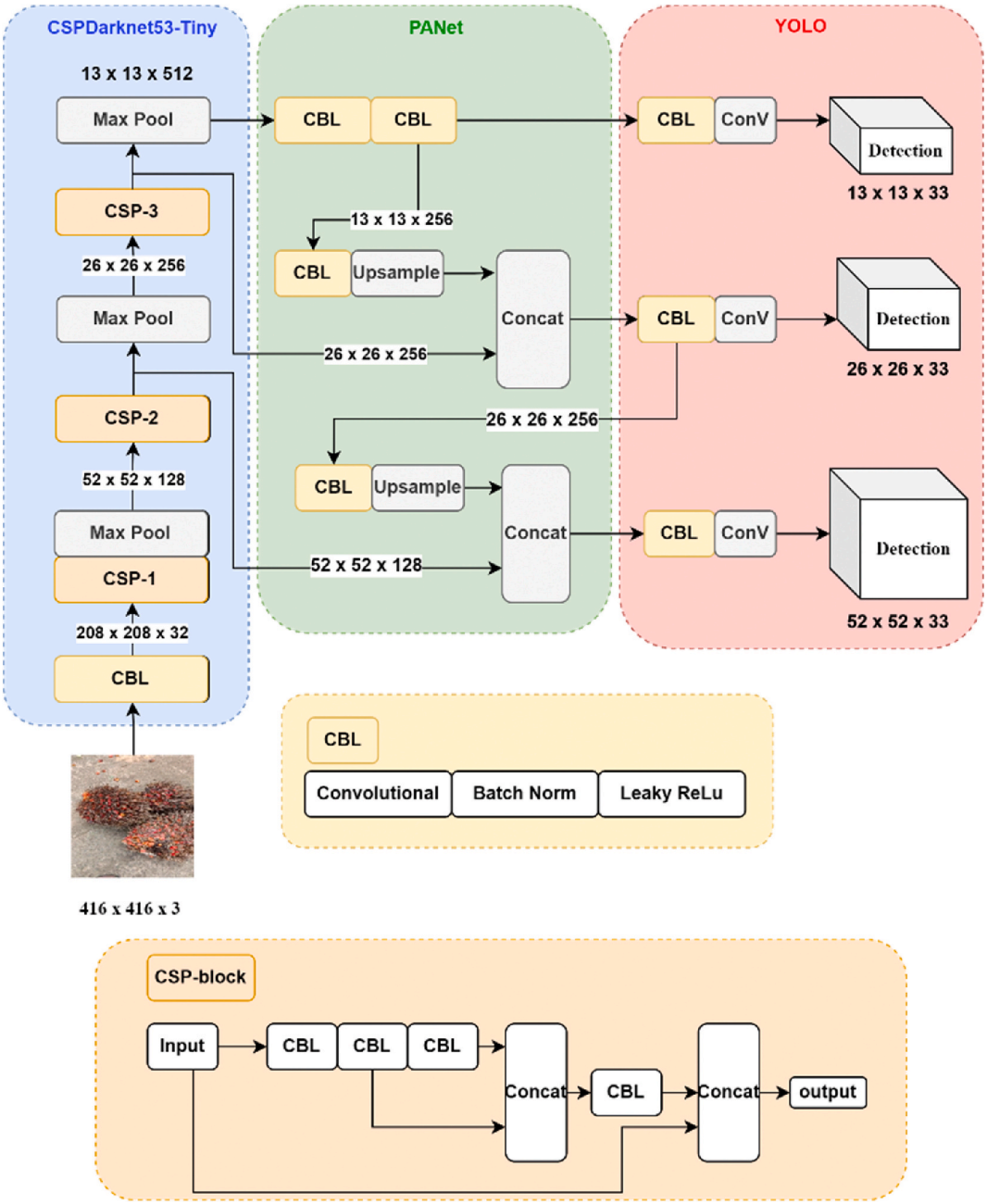


Fig. 9. YOLOv4-Tiny 3L architecture used in this research.

Table 1
TensorFlow Object Detection API model configuration.

Model	Input size	Learning rate	Batch size	Steps
SSD-MobileNetV2 FPN	320 × 320	0.08	16	50,000
EfficientDet-D0	512 × 512	0.08	16	300,000

when compared to doing training from the scratch [37].

2.2.2.4. Model optimization. To improve the performance of the model, two approaches were taken to the YOLOv4 model: (1) Bag of Freebies (BoF), and (2) Bag of Specials. BoF is a method to improve overall model performance without extra cost by changing the

Table 2
Total used images in each dataset experiment.

Augmentation Type	Original Images	Random Brightness	Gaussian Blur	45° Rotate	90° Rotate	Translation	Total images
No Augmentation (Original)	929	–	–	–	–	–	929
Photometric	929	929	929	–	–	–	2787
Geometric	929	–	–	420	509	929	2787
Combination	929	929	929	420	509	929	4645

Table 3
YOLOv4-CSPDarknet53 and YOLOv4-Tiny hyperparameter tuning scenario.

Model	Batch Size	Learning Rate		
		0.001	0.0001	0.01
YOLOv4-CSPDarknet53	64	Model_1 [12]	Model _3	Model _5
	32	Model _2	Model _4	Model _6
YOLOv4-Tiny	64	Model _7 [29]	Model _9	Model _11
	32	Model _8	Model _10	Model _12

Table 4
YOLOv4-Tiny 3L hyperparameter tuning scenario.

Model	Batch Size	Learning Rate		
		0.0012	0.0024	0.001
YOLOv4-Tiny 3L	64	Model_13 [30]	Model_15	Model _17
	32	Model_14	Model_16	Model_18

training strategy [38], meanwhile BoS is a technique to increase object detection accuracy significantly with only a small increase in inference cost [21]. The BoF used in this study is used to replace the traditional bounding box loss function shown in Eqs. (2) and (3):

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

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

where $R(B, B^{gt})$ is the penalty from B which is the predicted bounding box and B^{gt} is the actual bounding box. Meanwhile, the bounding box regression equation used is Complete IoU (CIoU) which has better consistency to the aspect ratio [39] and only needs fewer iterations to reach convergence, the equation of the consistency of the aspect ratio shown in Eqs. (4),(5),(6):

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

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

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

where ρ is the euclidean distance of the bounding box, C is the diagonal length of the smallest enclosing box covering the two boxes, α is the positive trade-off parameter, and v is the consistency of the aspect ratio. From the equations above, it can be obtained an equation to determine the loss function of the bounding box for CIoU, which can be defined in Eq. (7):

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

Then on the BoS used, applied to 2 parts of YOLOv4: (1) Backbone, and (2) Neck. In the backbone section, the CSP method is used as described in section 2.2.2.1. While on the neck using SPP, and PANet which has been described in section 2.2.2.1. For the activation function in YOLOv4 CSPDarknet53 uses Mish activation function, while in YOLOv4 Tiny series only uses Leaky ReLu. Mish activation is one type of activation function that can solve vanishing and exploding gradient problems which can trigger slow to reach the convergence point and very large changes in weight values. Mish activation help the deep learning model to obtain better accuracy and generalization [40], the calculations for the mish activation function are shown in Eqs. (8) and (9):

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

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

On the other hand, function of Leaky ReLu makes it possible to prevent a malfunctioning neuron because it has a value of "0" by defining a negative value as a very small value, Eq. (10) is the equation for the Leaky ReLu activation function:

$$f(x) = \max\ (0.01\ .x, x) \tag{10}$$

However, the Mish activation function can handle vanishing gradients and exploding gradients better than Leaky ReLu.

2.2.2.5. Evaluation. For the model evaluation we use mAP, F1-score, and IoU. The value of the F1-score is the harmonic value of precision and recall so that it can conclude how good the precision and recall are. Mean Average Precision (mAP) compares the ground-truth bounding box with the predicted box. The IoU used in the evaluation refers to the CIoU loss value described in section 2.2.2.3. Eqs. (11),(12),(13),(14) show the detail calculations for mAP and F1-Score:

$$\text{mAP} = \frac{1}{N}\ \Sigma_{i=1}^{N} AP_i \tag{11}$$

$$F1\ Score = 2\ .\frac{precision\ .\ recall}{precision + recall} \tag{12}$$

$$Precision = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{13}$$

$$Recall = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{14}$$

Precision is a parameter to measure the ratio between correctly identified data and all positive data results. Recall is a comparison between a lot of data that has a positive prediction and is actually positive with a lot of data that is actually positive. TP is a true positive which means that the correct prediction results are in accordance with the truth of the actual data. FP (false positive) is a prediction that does not match the real truth and FN (false negative) is a negative value that is wrong or in other words the value should be positive, but the prediction results show a negative value.

2.2.3. Model development

2.2.3.1. Augmentation data experiment. In the augmentation experiment, 4 experimental stages were carried out. First, carry out the training process on data that is not augmented. Second, carry out the training process with photometric transformation. Third, do the training process with geometric transformation and finally combination of photometric and geometric transformation. The number of images used in each experiment can be seen in more detail in Table 2. Photometric and geometric transformation are 2 augmentation methods that are quite commonly used in agriculture [41], the example of augmented images can be seen in Fig. 5(A)–(F). Deep learning architecture used for the image augmentation experiment is yolov4 with CSPDarknet53 as backbone. The hyperparameter configuration used is a learning rate of 0.001, a batch size of 64 and a subdivision of 16, which is known to be the best configuration for the YOLOv4 architecture in detecting objects in the agricultural sector [12]. We use Albumentation framework to enhance the generalization and robustness of deep learning models [42]. The following is the configuration used in performing augmentation with Albumentation framework:

- Random brightness range from −40% to +60%
- Random blur, noise variance from 7% to 9%
- Image rotation 45° and 90°
- Image translation with ratio 0.5

2.2.3.2. Hyperparameter fine-tuning. Hyperparameters that will be tuned are learning rate and batch size. The training scenario for hyperparameter tuning carried out will use a combination of different learning rate and batch size values, more detailed about training scenario can be seen in Tables 3 and 4. The value of learning rate and batch size uses the reference in previous research on Model_1 [12], Model_7 [29] and Model_13 [30]. From the results of the hyperparameter tuning, 1 best model based on validation result of each architecture will be selected for testing using video data. The test data used is divided into 2 types, firstly the video of 1 category and secondly, videos that have combination of categories.

2.2.3.3. Frozen layer experiment. Freeze layer is a transfer learning technique where the layers in the architecture are frozen so that the resulting weight values from the previous training process will not change while the layers that are not frozen will experience weight changes during the training process. Several studies have proven that training on certain frozen layers will improve the accuracy performance of the model in making predictions [43]. The use of a large or small dataset affects the number of layers that are frozen, if the dataset is relatively small, it would be better if you increase the number of layers that are frozen, while for a relatively large dataset, it would be better to just freeze the initial layer [44]. In this experimental session, there will be 4 stages of freeze layers: (1) not

performing freeze layer, (2) freeze the first 10 layers, (3) freeze the first 23 layers, and (4) freeze the first 54 layers. The observation is done to see the impact of frozen layer, which used 4645 images consisting of original images with a combination of photometric and geometric data augmentation for training purpose. Hyperparameter used in the training process is a learning rate of 0.001, a batch size of 64, and subdivision of 16.

3. Results and discussions

3.1. Data augmentation result

The results of the data augmentation in 4 stage experiment: (1) Without data augmentation, (2) Photometric transformation augmentation, (3) Geometric transformation augmentation, and (4) Combination of photometric and geometric augmentation are shown in Table 5 All the augmentation dataset can improve the mAP_{50} and IoU Score of models. From the results obtained show that the type of geometric augmentation has a better level of accuracy compared to other types of augmentation. The validation IoU value of geometric augmentation in Table 5 has a value of 1.5% better than photometric augmentation and 1.15% better than the combination of photometric and geometric augmentation, this indicates that geometric augmentation is able to detect the fresh fruit bunches more precisely than photometric augmentation. The test results strengthen the evidence that geometric data augmentation can perform object localization better than photometric augmentation, the IoU test on geometric augmentation was 4.56% better than the IoU test on photometric augmentation. Even more clearly can be seen in the experimental results in performing video detection in Fig. 10 (B) which shows the dataset with geometric augmentation is able to detect objects better than photometric augmentation in Fig. 10 (A), but by combining the two types of augmentation can improve both object localization and classification of fresh fruit bunches as shown in Fig. 10 (C), the effect of photometric data provides new variations thereby increasing the model's ability to detect better while geometric data allows the model to detect better from various angles of objects.

3.2. Hyperparameter tuning and testing result

The following are the validation results of the hyperparameter tuning from the training scenario Tables 3 and 4 in section 2.2.3.2. The model training was done by using a train validation ratio of 7:3. The amount of data used is 4645 images consisting of original, photometric and geometric augmentation data. Overall, the experiments carried out at this stage started from performing hyperparameter tuning scenarios and then selecting the best 1 model from each different type of YOLOv4 architecture. The model that has been selected will be used to test videos that have only 1 category and videos that have combination of categories. The results of training and validation results from Table 6, indicates the model is not overfitting. Experiments on the YOLOv4-CSPDarknet53 model showed only slight changes in each training scenario, which means that the model has reached the optimal point in classifying and detecting. The highest IoU validation value was obtained in the Model_5 scenario with a value of 2.03% higher than Model_1. We chose Model_5, Model_12, and Model_16 for test 2 different types of video data. A high IoU value is an important consideration parameter in choosing the model used in conducting the test. Table 7 shows the results of detecting the maturity level of oil palm fresh fruit bunches from video that contain only 1 class category, the total video data used is 6 videos where each class uses 1 video data for testing with 3238 ground truth annotations. YOLOv4-CSPDarknet53 has a better test result than YOLOv4-Tiny series, this is normal because YOLOv4-CSPDarknet53 has more layer depth than YOLOv4-Tiny. However, the YOLOv4-Tiny series also has good performance because the mAP_{50} value is above 80%. YOLOv4-Tiny 3L can detect and classify the object better than YOLOv4-Tiny because it has three detection layers, which means it can detect objects at three different object size scales: (1) small, (2) medium, and (3) large. The increase was not significant because the size of the oil palm fresh fruit bunches was relatively consistent. Although the difference in mAP values between YOLOv4-CSP and YOLOv4-Tiny series is slightly different, YOLOv4-CSPDarknet53 is still better in terms of object localization and class prediction of the maturity level of oil palm fresh fruit bunches as shown in Fig. 11 (A) and (D); Fig. 12 (A) and (D); Fig. 13 (A) and (D). For YOLOv4-Tiny testing on single category test data can be seen in Fig. 11 (B) and (E); Fig. 12 (B) and (E); Fig. 13 (B) and (E). Meanwhile, the YOLOv4-Tiny 3L test can be seen in Fig. 11 (C) and (F); Fig. 12 (C) and (F); Fig. 13 (C) and (F).

We test data on videos that have combination class category (multi-category) to see if the model can distinguish many objects in 1 frame with total 1288 annotations. The results from Table 8 show that the entire YOLOv4 model has not been able to detect more than 1 class well in a video frame due to the training data used only has 1 class in each image. The best results are obtained from YOLOv4-Tiny 3L with the lowest FP and FN value. Detection test results can be seen in Fig. 14(A)–(F), and Fig. 15(A)–(F), YOLOv4-Tiny 3L is able to detect better than YOLOv4-CSPDarknet53.

Overall, from the results of tests carried out at the hyper-parameter tuning stage, YOLOv4-Tiny 3L is the best model in detecting the maturity level of oil palm fresh fruit bunches. The detection speed is only 2 s different and the mAP rate is much better than YOLOv4-

Table 5
Data augmentation validation and test result.

Augmentation Type	Validation mAP_{50}	Validation F1 Score	Validation IoU	Test mAP_{50}	Test F1 Score	Test IoU
No Augmentation	83.4%	0.74	62.28%	91.82%	0.83	64.72%
Photometric	98%	0.95	82.88%	92.87%	0.82	63.17%
Geometric	**99.1%**	**0.97**	**84.14%**	93.93%	**0.87**	**67.73%**
Combination	98.7%	0.96	83.18%	**94.98%**	0.87	67.38%

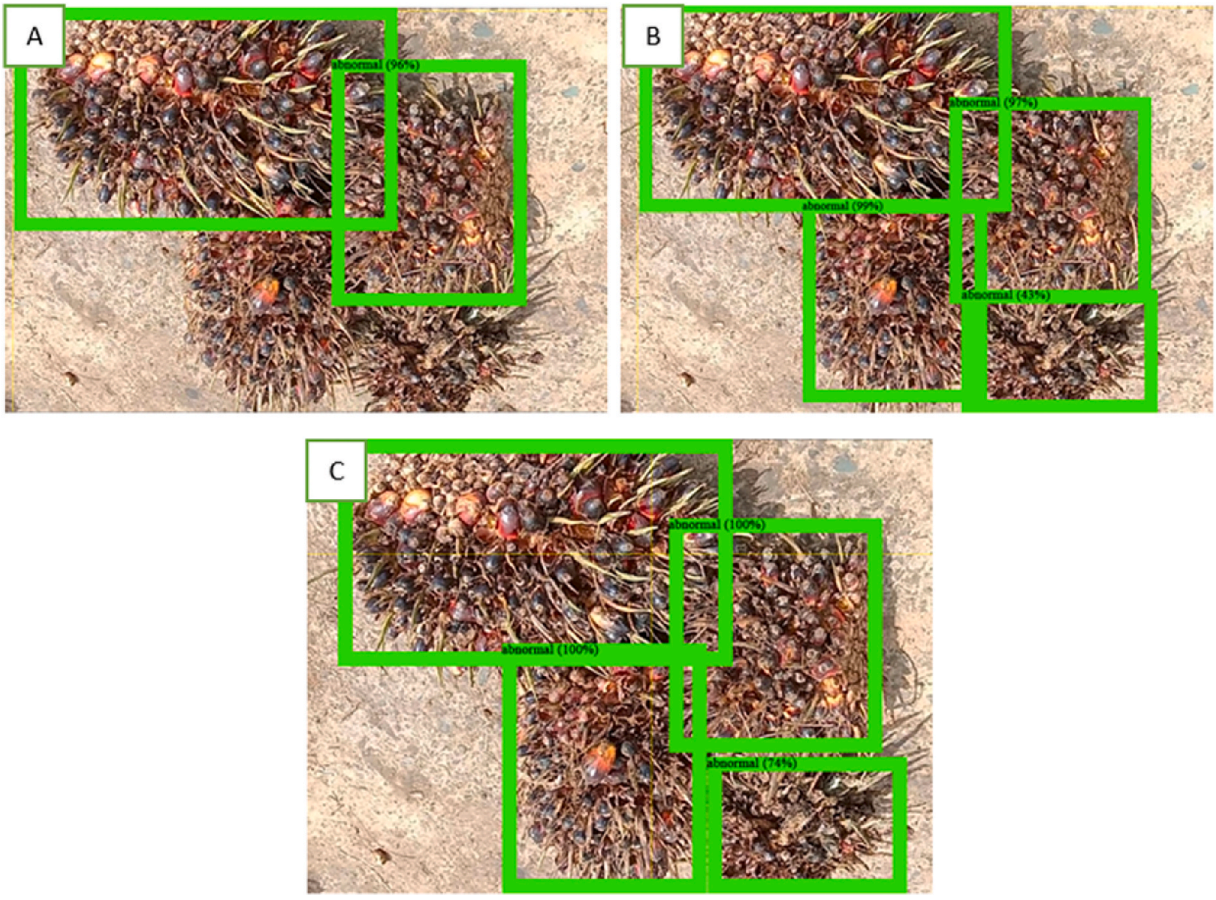


Fig. 10. The augmentation experiment on abnormal class video test set. (A) photometric augmentation result not able to detect all the fresh fruit bunches: (1) abnormal 96%, (B) geometric augmentation result able to detect all the fresh fruit bunches: (1) abnormal 97%, (2) abnormal 99%, (3) abnormal 43%, (C) The combination of photometric and geometric augmentation able to detect all fresh fruit bunches with a better class prediction score: (1) abnormal 100%, (2) abnormal 100%, (3) abnormal.

Table 6
Train and validation results from hyperparameter tuning scenario.

Model	Scenario	Train mAP_{50}	Train IoU	Validation mAP_{50}	Validation IoU
YOLOv4-CSPDarknet53	Model_1 [12]	99.83%	86.8%	**98.83%**	82.41%
	Model_2	99.56%	84.65%	98.47%	80.96%
	Model_3	99.52%	82.22%	98.28%	78.78%
	Model_4	99.05%	79.95%	97.30%	75.67%
	Model_5	**99.94%**	**89.26%**	98.52%	**84.12%**
	Model_6	99.37%	84.62%	98.57%	81.33%
YOLOv4-Tiny	Model_7 [29]	96.72%	72.53%	94.17%	66.31%
	Model_8	90.09%	62.98%	85.43%	57.32%
	Model_9	54.4%	50.07%	51.30%	48.01%
	Model_10	41.47%	47.12%	41.07%	44.10%
	Model_11	99.08%	80.62%	97.61%	74.99%
	Model_12	**99.44%**	**81.86%**	**97.88%**	**76.69%**
YOLOv4-Tiny 3L	Model_13 [30]	99.67%	**82.94%**	98.26%	77.11%
	Model_14	99.15%	81.78%	97.74%	76.47%
	Model_15	**99.74%**	82.46%	98.33%	77.13%
	Model_16	99.55%	82.70%	98.08%	**77.78%**
	Model_17	99.58%	81.04%	**98.39%**	75.19%
	Model_18	88.19%	62.79%	84.22%	58.59%

Table 7
Test result from video data that contain only 1 class category for each videos.

Model	mAP_{50}	IoU	F1 Score	Detection time	AVG FPS
YOLOv4-CSPDarknet53 (Model_5)	**97.64%**	**73.36%**	**0.93**	19 s	41.5
YOLOv4-Tiny (Model_12)	83.57%	56.90%	0.78	**4 s**	**105.03**
YOLOv4-Tiny 3L (Model_16)	90.56%	58.35%	0.79	5 s	104.95

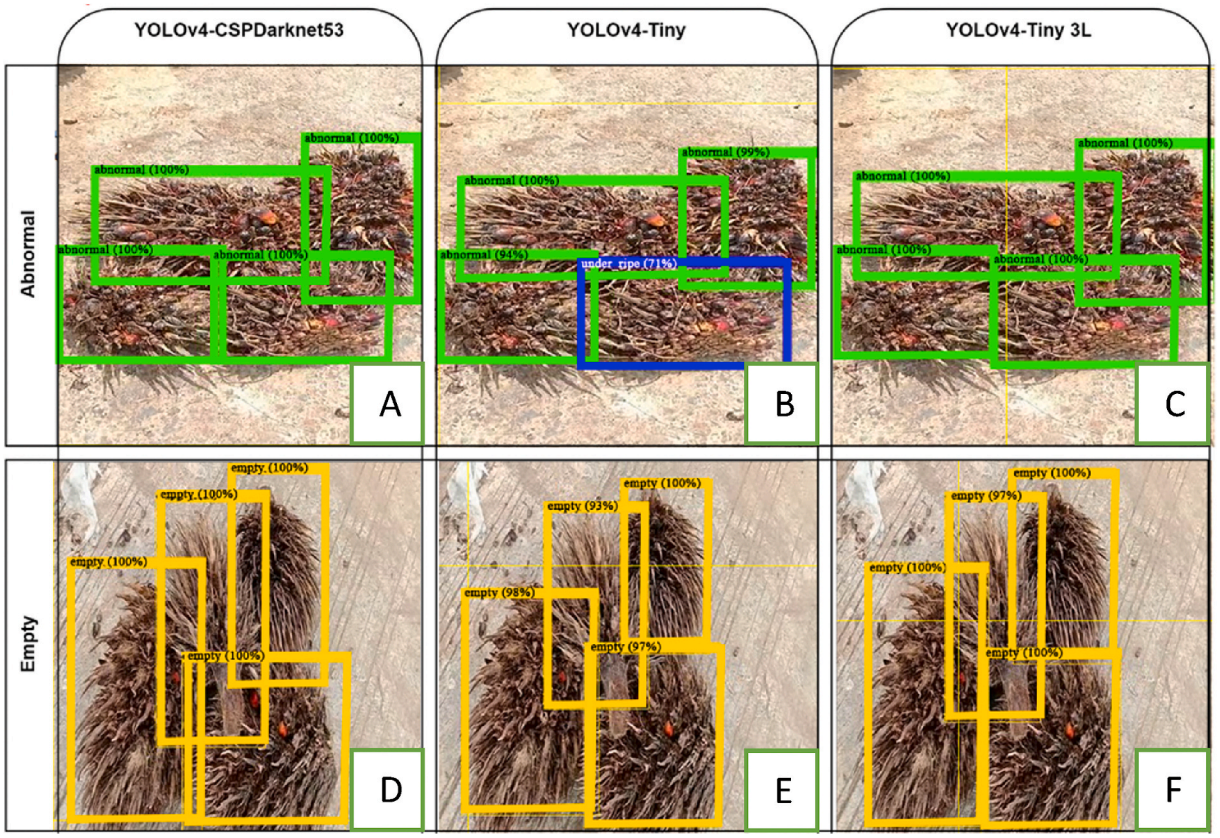


Fig. 11. Test result for video that contain 1 class category. Row is the class type of abnormal and empty, the column is the model used for detection. (A), (B) and (C) are 'abnormal' class data that were tested using YOLOv4-CSPDarknet53, YOLOv4-Tiny and YOLOv4-Tiny 3L respectively. (D), (E) and (F) are 'empty' class data that were tested using YOLOv4-CSPDarknet53, YOLOv4-Tiny and YOLOv4-Tiny 3L respectively.

Tiny and has a much lighter computational load than YOLOv4-CSPDarknet53. Meanwhile, YOLOv4-CSPDarknet53 has not been able to detect the maturity level of palm oil when using videos that contain combination category as well as YOLOv4-Tiny 3L. The results of experiments conducted showed that training using a dataset with only 1 category resulted in a model that was not good at testing data that had combination category.

To prove it even more clearly, we conducted a detection test on all models with multi-category videos. The results of the mAP_{50} and IoU tests can be seen in Figs. 16 and 17. It turns out that the YOLOv4-Tiny Series is proven to be able to make better generalizations, especially on the YOLOv4-Tiny 3L. In the YOLOv4-Tiny series, the use of a smaller batch size causes the convergence level to take longer to be achieved so that when the epoch has ended, the loss level has not had time to reach the global minimum point so that the accuracy obtained is not good. YOLOv4-Tiny 3L on Model_14 got the highest mAP_{50} test value, this is because the number of batch sizes with the appropriate learning rate values, but still stuck at the local minimum and can be increased again if the specified step value is increased larger. Layer depth and the number of parameters are not determinants of model performance. YOLOv4-CSPDarknet53 has a layer that is too deep so it is not suitable for use on datasets with only 4645 images. In general, the test results show that the YOLOv4-Tiny Series tends to increase in performance if the learning rate value is inversely proportional to the batch size value, so if the learning rate value is large, it is better to use a small batch size. Meanwhile, in YOLOv4-CSPDarknet53, if you use a large learning rate and a large batch size, the performance will increase.

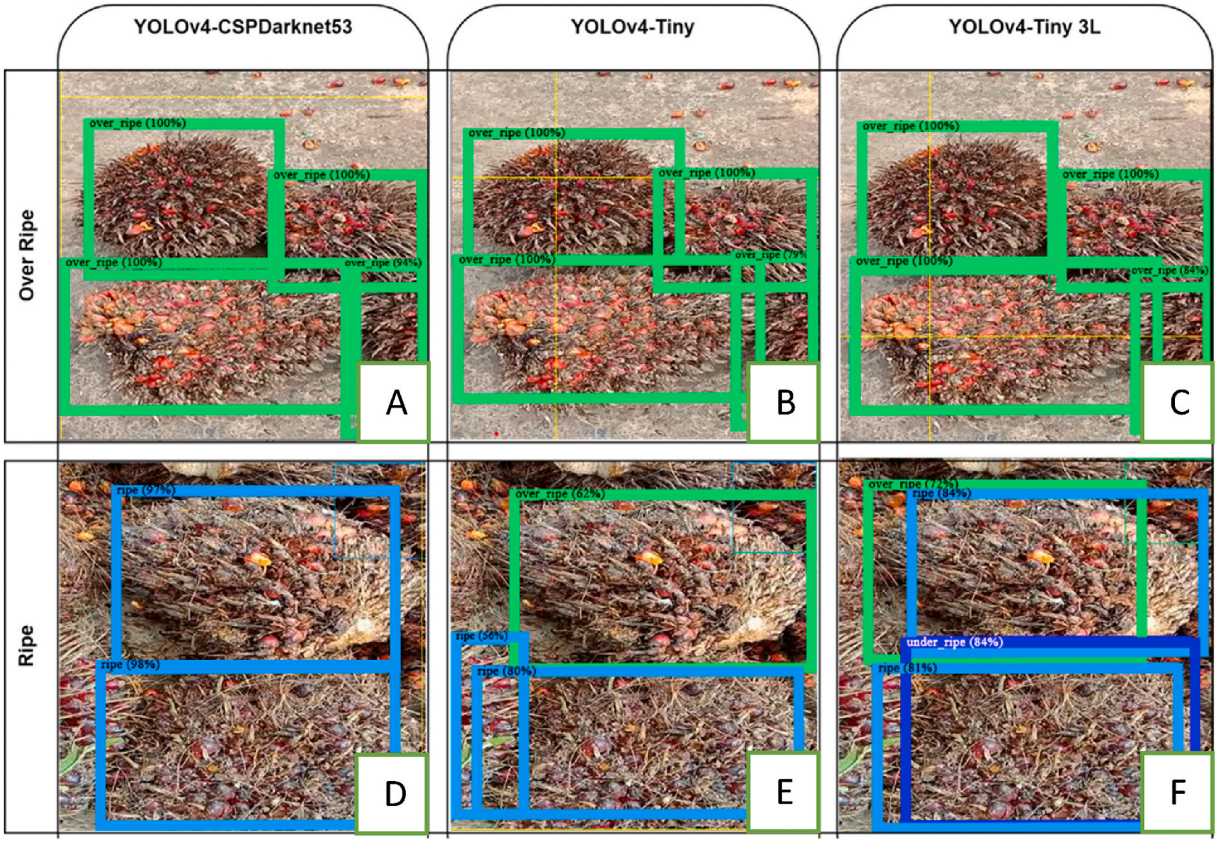


Fig. 12. Test result for video that contain 1 class category. Row is the class type of over ripe and ripe, the column is the model used for detection. (A), (B) and (C) are 'over-ripe' class data that were tested using YOLOv4-CSPDarknet53, YOLOv4-Tiny and YOLOv4-Tiny 3L respectively. (D), (E) and (F) are 'ripe' class data that were tested using YOLOv4-CSPDarknet53, YOLOv4-Tiny and YOLOv4-Tiny 3L respectively.

3.3. Freeze layer effect

The performance of the training model on the frozen layer has evaluation results that are not much different in terms of mAP and IoU validation set. Freeze layer on the model only reduces 0.22%–0.32% for mAP_{50} validation and can reduce the required training time of approximately 2–3 h. The initial weight in the pre-trained model is optimal enough so that it has a validation result that is not much different from the model that does not freeze layer. The results of experiment can be seen in Table 9. Testing session which is carried out using a video that has 1 class category using freeze layer with the right composition is proven to improve performance. This increase is due to the optimal weight value from the results of previous training using a very large dataset which is then combined with the adjusted weight using the oil palm fresh fruit bunch dataset. The IoU value increased by 6.93% when compared to not freezing the layer.

3.4. Discussions

Detecting oil palm maturity in real-time is challenging due to obstacles such as piled objects, dynamic lighting, objects with different angles of view, and shadows that can cause discoloration of the oil palm fruit to get darker. Previous oil palm detection studies employed augmentations such as random brightness and Gaussian blur [1]. The emphasis on geometric augmentation is also carried out in research on the classification of oil palm maturity with additional Gaussian blur augmentation [3], but applying photometric or geometric augmentation alone is insufficient because objects can be seen in real time conditions from various camera points of view and dynamic lighting conditions. The data augmentation experiment conducted in Table 5 demonstrates that data augmentation successfully overcomes the challenge of detecting oil palm maturity by combining photometric and geometric augmentation, with an indication of improvement from mAP_{50} where objects can be properly identified and IoU where the model can do object localization better. The dataset offers a wide variety of item tilt angles and dynamic illumination that is better suited to outdoor scenarios as a result of photometric and geometric augmentation. Furthermore, the speed of detection is a critical factor in improving time efficiency in assessing the maturity stage of oil palm.

The model development process aims to improve model performance by adjusting the hyperparameter configuration with the dataset used. The training outcomes on all YOLOv4 architectures are compared in Fig. 18(A)–(D). Each training result can reflect the

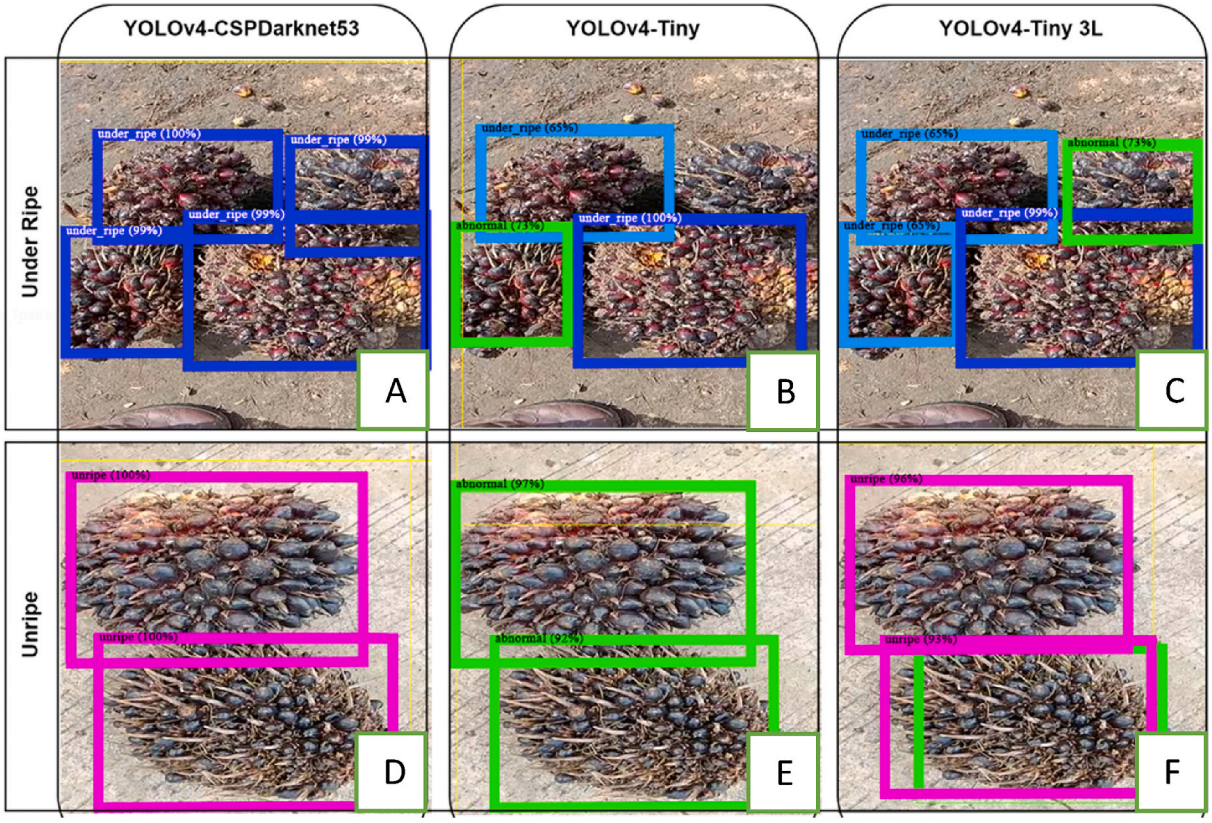


Fig. 13. Test result for video that contain 1 class category. Row is the class type of under ripe and unripe, the column is the model used for detection. (A), (B) and (C) are 'under-ripe' class data that were tested using YOLOv4-CSPDarknet53, YOLOv4-Tiny and YOLOv4-Tiny 3L respectively. (D), (E) and (F) are 'unripe' class data that were tested using YOLOv4-CSPDarknet53, YOLOv4-Tiny and YOLOv4-Tiny 3L respectively.

Table 8
Test result from video data that contain combination category.

Model	mAP_{50}	IoU	F1 Score	Detection time	AVG FPS
YOLOv4-CSPDarknet53 (Model_5)	48.54%	33.13%	0.43	13 s	44.8
YOLOv4-Tiny (Model_12)	59.17%	42.83%	0.59	**2 s**	**118.6**
YOLOv4-Tiny 3L (Model_16)	**70.21%**	**45.50%**	0.63	4 s	108.2

properties of the final model. YOLOv4-CSPDarknet53, YOLOv4-Tiny and YOLOv4 (combination tuning) from Fig. 18 (D) show that the model has not yet reached the global minimum average loss point, which is due to an insufficient number of training steps. Meanwhile, YOLOv4-Tiny 3L has the minimum loss value, indicating that the model is already performing well, but a higher step value is still needed for maximum performance.

A comparison was made between the YOLOv4 architecture and other state-of-the-art object detection models such as SSD and EfficientDet to verify the performance of the developed model in detecting the maturity level of oil palm. Tables 10 and 11, show the detection results from single category and multi category data. In general, all models can detect single category data adequately, while multi-category data recognition remains poor. This could be because the dataset used for training is a single category with only one type of class in the video data for each category. The YOLOv4-Tiny 3L is highly recommended for real-time deployments because to its balanced mAP value and excellent detection speed. The IoU value on EfficientDet-D0 and SSD-MobileNetV2 FPN is much higher than YOLOv4 because the methods for generating anchor boxes differ. YOLOv4 uses a k-means cluster to determine 9 anchor box points based on the position of the bounding box in the dataset, whereas the model with the TensorFlow framework uses a multiscale anchor box. To demonstrate the model's capabilities, 10 samples of the same video frame were utilized. For single category test data, it uses the last 10 frames from a total of 186 frames in the underripe class video test, while for multi-category test data it uses the last 10 frames out of a total of 146 frames in the test video with a mix of overripe and empty classes. A video test on single category data is shown in Fig. 19. Although the YOLOv4 model's total IoU value in Fig. 19 is lower than EfficientDet-D0 and SSD-MobileNetV2 FPN, YOLOv4 detects all objects more precisely than SSD-MobileNetV2, and the resulting prediction value exceeds EfficientDet-D0.

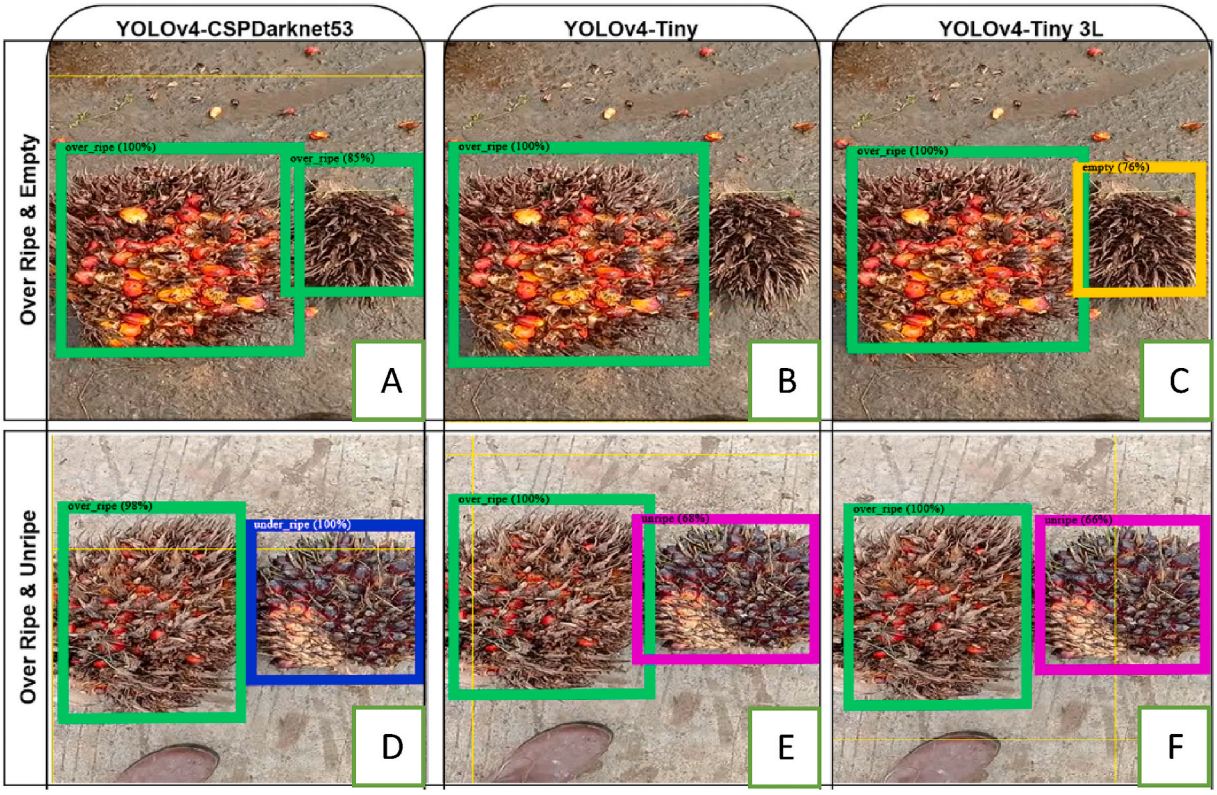


Fig. 14. Test result for video that contain combination class category. In the row is a chunk of frames with overripe and empty classes, while in the second row is an overripe and unripe classes. The column is the model used for detection. (A), (B) and (C) are 'over-ripe' and 'empty' class data that were tested using YOLOv4-CSPDarknet53, YOLOv4-Tiny and YOLOv4-Tiny 3L respectively. (D), (E) and (F) are 'over-ripe' and 'unripe' class data that were tested using YOLOv4-CSPDarknet53, YOLOv4-Tiny and YOLOv4-Tiny 3L respectively.

Meanwhile, the results of multi-category video detection in Fig. 20 show a better detection ability on YOLOv4-Tiny 3L. The YOLOv4-Tiny 3L detects better than the SSD-MobileNetV2 FPN. The object detection model with the Tensorflow framework has a significant advantage in terms of model size. SSD-MobileNetV2 and EfficientDet-D0 have more reliable and better IoU values than YOLOv4-based models. Meanwhile, mAP does not differ considerably, but the YOLOv4 model has a much faster detection speed, making it more suitable for application in real-time scenarios.

The process of grading the maturity level of oil palm is often carried out after harvesting and after the fresh fruit bunches of oil palm are received by the palm oil processing mill. This research could be applied to oil palm plantations, although it is more appropriate for palm oil processing plants. Previous research on oil palm detection is more appropriate for use while performing automatic harvesting [1,4], but this study is more focused on assessing the maturity level of fresh fruit bunches of oil palm, which is more suited for use in sorting maturity levels in palm oil processing factories. The key advancement produced in this research is in the form of a video dataset with six classes of oil palm maturity levels, which is more suitable to real-world settings than non-sequential image datasets. This research investigated multi-category video data because the current research has not tested multi-category data, therefore the findings of the existing research are unsatisfactory, and most of the previous research employs a classification model [2,3,5,6] that cannot recognize several objects in one picture frame. Even so, using video datasets necessitates a large amount of data in order for model detection to perform well. According to Table 11, it performs poorly in the multi-category situation because the training dataset contains only single category data. Collecting video data takes longer than collecting non-sequential images due to several problems such as unpredictable weather and limited time for ideal lighting, such as between 11:00 a.m. and 01:00 p.m.

4. Conclusion

The development of the model at the experimental stage of data augmentation shows that geometric augmentation is able to better localize objects due to the influence of various angles of objects making the model understand more about the characteristics of the object. Meanwhile, photometric data augmentation makes the model more robust against illumination on objects, thereby increasing object classification capabilities. At the hyperparameter experimental stage, it can be concluded that the number of layers that are too large is not suitable for the dataset used because it is relatively small, this is evident in the YOLOv4-CSPDarknet53 model which has not been able to generalize well on the results of the multiple category videos test. Meanwhile, the model that can generalize objects well

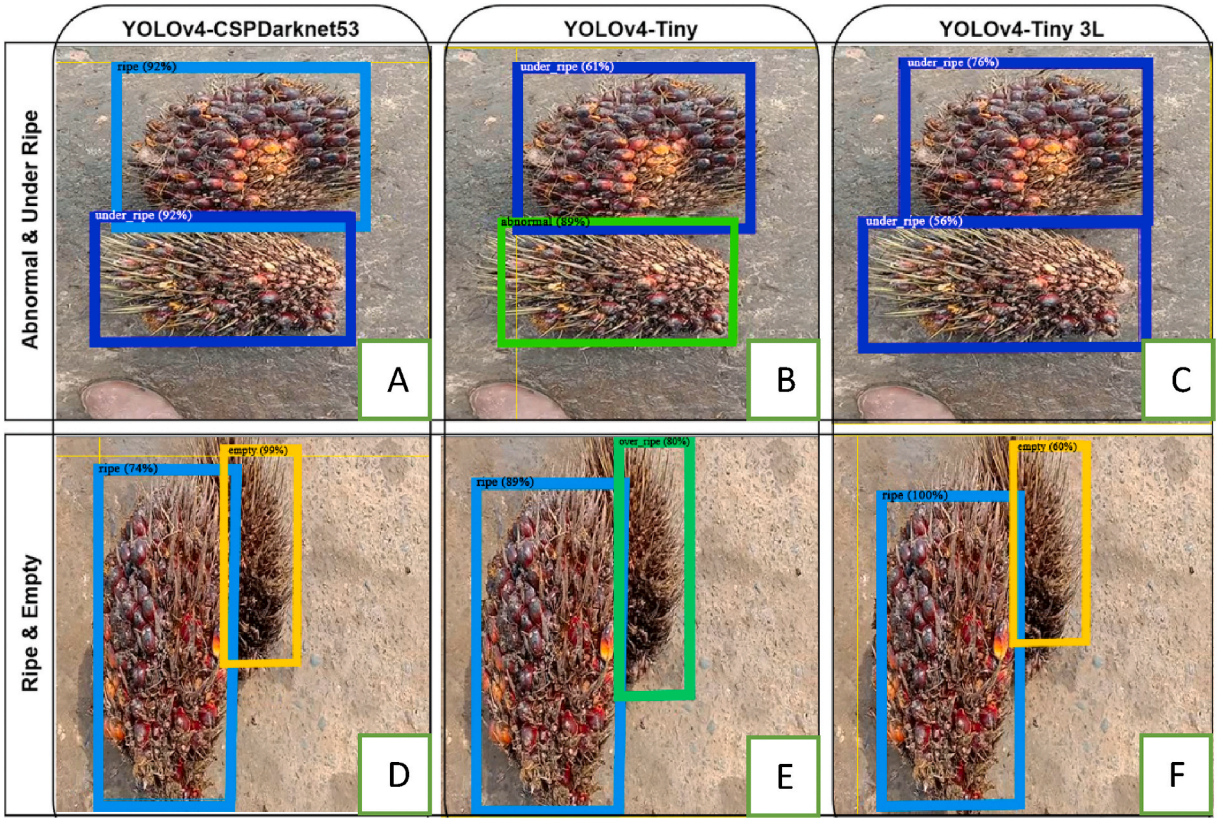


Fig. 15. Test result for video that contain combination class category. In the row is a chunk of frames with abnormal and ripe classes, while in the second row is a ripe and empty classes. The column is the model used for detection. (A), (B) and (C) are 'abnormla' and 'under-ripe' class data that were tested using YOLOv4-CSPDarknet53, YOLOv4-Tiny and YOLOv4-Tiny 3L respectively. (D), (E) and (F) are 'ripe' and 'empty' class data that were tested using YOLOv4-CSPDarknet53, YOLOv4-Tiny and YOLOv4-Tiny 3L respectively.

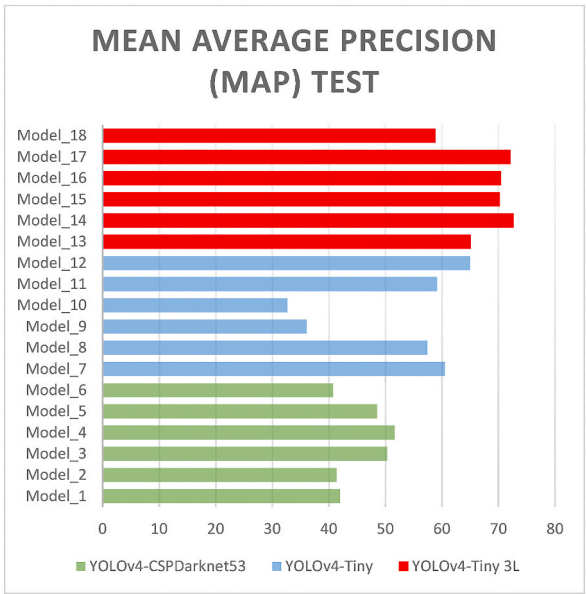


Fig. 16. All models mAP_{50} test result for multi-category videos.

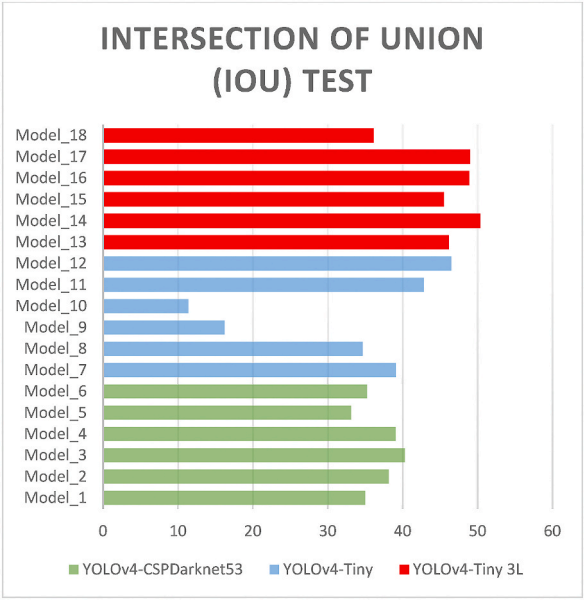


Fig. 17. All models IoU test result for multi-category videos.

Table 9
Train validation and test result using frozen layers.

Frozen layers	Unfrozen layers	Train mAP_{50}	Train IoU	Validation mAP_{50}	Validation IoU	Test mAP_{50}	Test IoU
0	161	99.83%	86.80%	**98.83%**	82.41%	94.98%	67.38%
10	151	99.80%	**88.41%**	98.54%	83.64%	94.68%	69.69%
23	138	**99.84%**	88.30%	98.51%	**84.05%**	**98.53%**	**74.31%**
54	107	99.70%	87.86%	98.61%	83.88%	96.83%	73.42%

among other models is YOLOv4-Tiny 3L using a learning rate of 0.001 and a batch size of 64 with an mAP50 test value of 72.15% on multiple category videos. However, these results cannot be said to be maximal because the model is stuck at the local minimum, this is because the number of steps in the training is still not enough. However, all YOLOv4 architectures perform very well to detect single category class videos. Using frozen layers doesn't significantly improve model performance, it only improves object localization capabilities. This is proven when combining tuning models using the best hyperparameters and the best frozen layer performance, which increases the IoU by 3.76%. We succeeded in detecting the type of maturity level of oil palm fresh fruit bunches using a dataset in the form of video or sequential image which was applied to the object detection model as a novelty, where previous studies only used non-sequential images. When compared to models such as SSD-MobileNetV2 FPN and EfficientDet-D0, we believe that the future one stage object detector will see a promising improvement in terms of real-time detection accuracy. Possible development that can be done from this research is to automate in calculating each class that has been detected by the camera that can be used in real time.

We suggest adding geometric augmentation with a variety of more varied tilt angles to increase accuracy in object localization. Meanwhile, to improve detection accuracy in videos that have combination of class category, it is recommended to add multi-category data for training. YOLOv4-Tiny 3L architecture is the most balanced among other architectures because it has fast training time, fast detection rate and light computing level but has relatively high accuracy but with a concern YOLOv4-Tiny 3L is more suitable for relatively small datasets. We believe that the use of large datasets will be more suitable against YOLOv4-CSPDarknet53. So it can be considered as a good choice for real-time detection.

Author contribution statement

Franz Adeta Junior: Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Wrote the paper.

Suharjito: Conceived and designed the experiments; Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper.

Funding statement

Dr. Suharjito Suharjito was supported by The Directorate General of Higher Education Ministry of Education, Culture, Research,

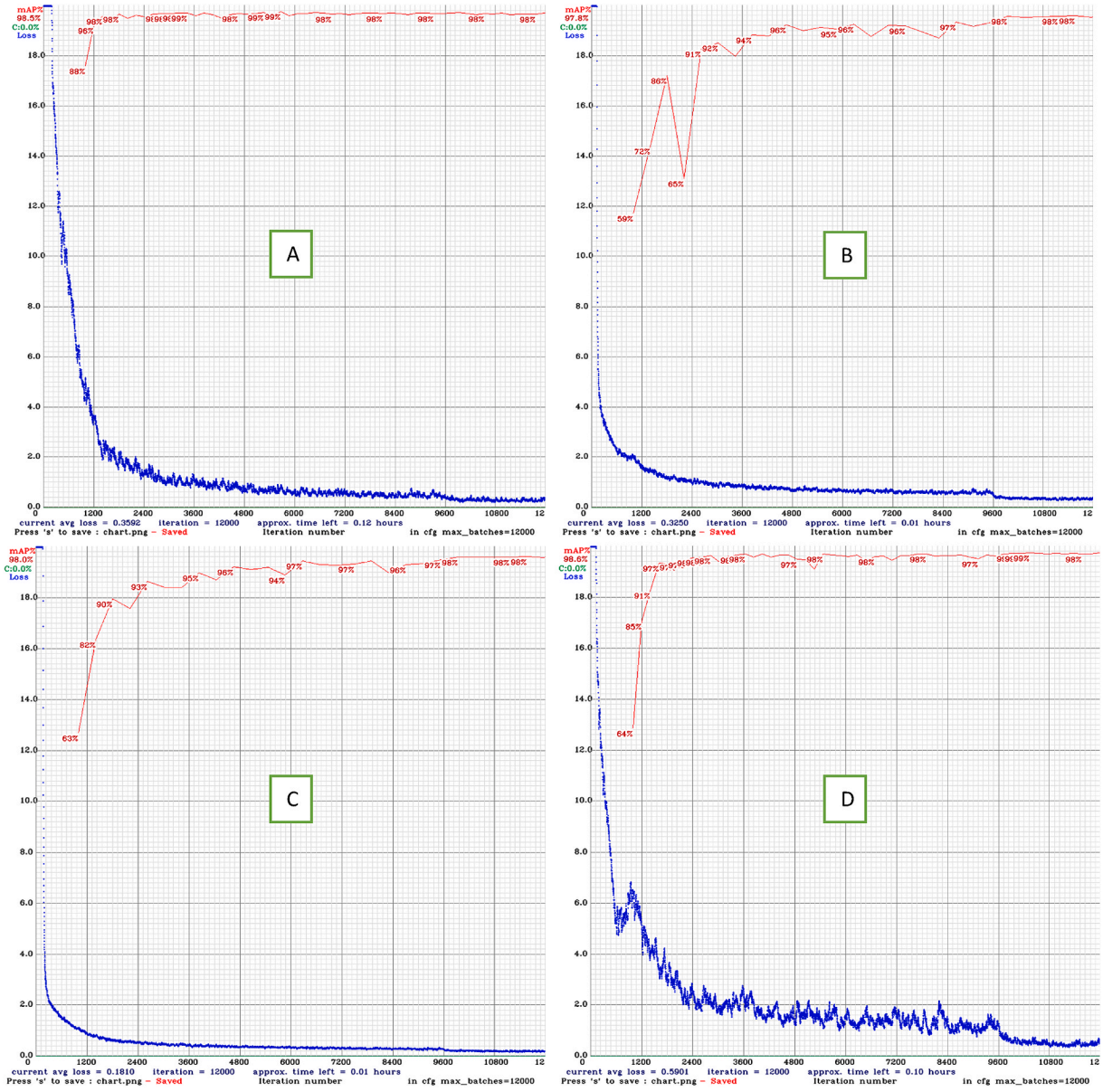


Fig. 18. Loss value from training result: (A) YOLOv4-CSPDarknet53, (B) YOLOv4-Tiny, (C) YOLOv4-Tiny 3L, (D) YOLOv4-CSPDarknet53 with combination tuning.

Table 10
Object detection model comparison using single category dataset.

Model	Framework	Parameters (Million)	Detection speed (second)	mAP_{50}	IoU	Model Size (MB)
Model_1 [12]	DarkNet	53	18	94.98%	67.38%	244.3
Model_5	DarkNet	53	19	97.64%	73.36%	244.3
Model_7 [29]	DarkNet	<10	5	86.02%	52.23%	22.5
Model_12	DarkNet	<10	5	83.57%	56.90%	22.5
Model_13 [30]	DarkNet	<10	4	90.18%	59.47%	23.4
Model_16	DarkNet	<10	4	90.56%	58.35%	23.4
YOLOv4-CSPDarknet53 (23 layer frozen)	DarkNet	53	17	98.53%	74.31%	244.3
YOLOv4-CSPDarknet53 (Combine tuning)	DarkNet	53	20	81.37%	59.5%	244.3
EfficientDet-D0 [35]	Tensorflow	3.9	52	99.30%	87.10%	40.1
SSD-MobileNetV2 FPN [35]	Tensorflow	15	30	98.51%	87.12%	17.6

Table 11
Object detection model comparison using multi-category dataset.

Model	Framework	Parameters (Million)	Detection speed (second)	mAP_{50}	IoU	Model Size (MB)
Model_1 [12]	DarkNet	53	12	41.97%	35.04%	244.3
Model_5	DarkNet	53	13	48.54%	33.13%	244.3
Model_7 [29]	DarkNet	<10	3	60.53%	39.1%	22.5
Model_12	DarkNet	<10	3	59.17%	42.84%	22.5
Model_13 [30]	DarkNet	<10	3	65.05%	46.16%	23.4
Model_16	DarkNet	<10	3	70.21%	45.50%	23.4
YOLOv4-CSPDarknet53 (23 layer frozen)	DarkNet	53	11	43.56%	39.03%	244.3
YOLOv4-CSPDarknet53 (Combine tuning)	DarkNet	53	13	44.91%	36.86%	244.3
EfficientDet-D0 [35]	Tensorflow	3.9	36	54.51%	85.11%	40.1
SSD-MobileNetV2 FPN [35]	Tensorflow	15	20	70.04%	84.47%	17.6

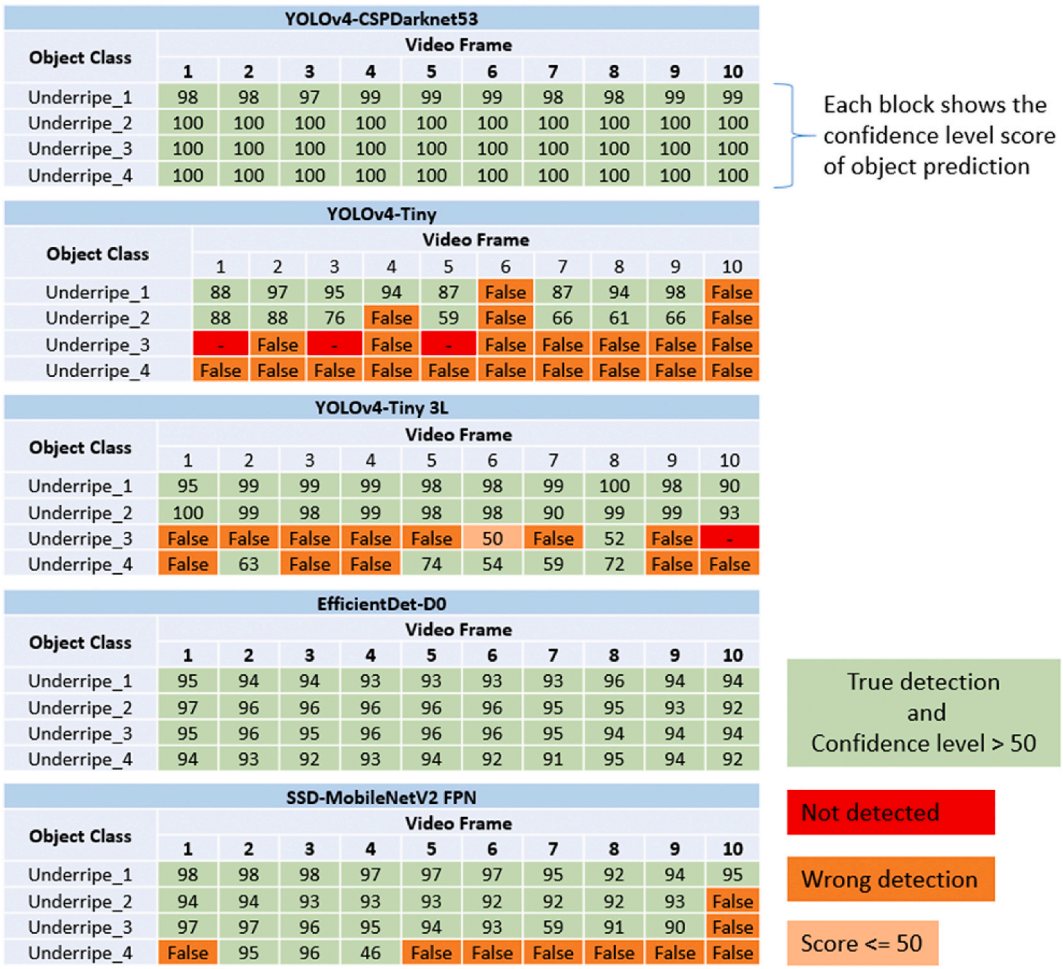


Fig. 19. The results of the analysis of the sample with 10 video frames in the underripe class (single category data).

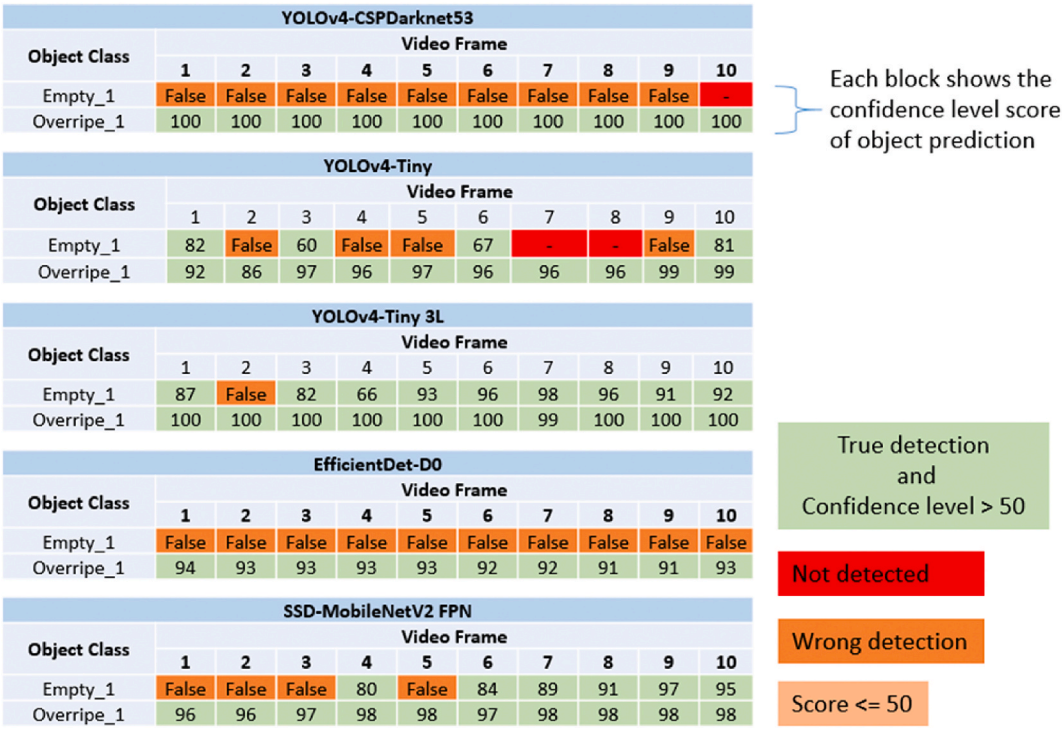
and Technology together with the Education Fund Management Institute [085/E4.1/AK.04.RA/2021].

Data availability statement

Data will be made available on request.

Declaration of interest's statement

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to



YOLOv4-CSPDarknet53

Object Class	Video Frame 1	2	3	4	5	6	7	8	9	10
Empty_1	False	False	False	False	False	False	False	False	False	-
Overripe_1	100	100	100	100	100	100	100	100	100	100

YOLOv4-Tiny

Object Class	Video Frame 1	2	3	4	5	6	7	8	9	10
Empty_1	82	False	60	False	False	67	-	-	False	81
Overripe_1	92	86	97	96	97	96	96	96	99	99

YOLOv4-Tiny 3L

Object Class	Video Frame 1	2	3	4	5	6	7	8	9	10
Empty_1	87	False	82	66	93	96	98	96	91	92
Overripe_1	100	100	100	100	100	100	99	100	100	100

EfficientDet-D0

Object Class	Video Frame 1	2	3	4	5	6	7	8	9	10
Empty_1	False	False	False	False	False	False	False	False	False	False
Overripe_1	94	93	93	93	93	92	92	91	91	93

SSD-MobileNetV2 FPN

Object Class	Video Frame 1	2	3	4	5	6	7	8	9	10
Empty_1	False	False	False	80	False	84	89	91	97	95
Overripe_1	96	96	97	98	98	97	98	98	98	98

Fig. 20. The results of the analysis of the sample with 10 video frames in the multi-category data (empty-overripe).

influence the work reported in this paper.

Acknowledgement

The authors would like to express their gratitude to BINUS University for all of their support and the oil palm mill for supporting the preparation of dataset.

References

[1] M.H. Junos, A.S. Mohd Khairuddin, S. Thannirmalai, M. Dahari, An optimized YOLO-based object detection model for crop harvesting system, IET Image Process. 15 (2021), https://doi.org/10.1049/ipr2.12181.
[2] A.Y. Saleh, E. Liansitim, Palm oil classification using deep learning, Sci. Inf. Technol. Lett. 1 (2020) 1–8, https://doi.org/10.31763/sitech.v1i1.1.
[3] Suharjito, G.N. Elwirehardja, J.S. Prayoga, Oil palm fresh fruit bunch ripeness classification on mobile devices using deep learning approaches, Comput. Electron. Agric. 188 (2021), 106359, https://doi.org/10.1016/j.compag.2021.106359.
[4] N.A. Prasetyo, Pranowo, A.J. Santoso, Automatic detection and calculation of palm oil fresh fruit bunches using faster R-CNN, Int. J. Appl. Sci. Eng. 17 (2020) 121–134, https://doi.org/10.6703/IJASE.202005_17(2).121.
[5] A. Septiarini, H. Hamdani, H.R. Hatta, A.A. Kasim, Image-based Processing for Ripeness Classification of Oil Palm Fruit, in: Proceeding - 2019 5th Int. Conf. Sci. Inf. Technol. Embrac. Ind. 4.0 Towar. Innov. Cyber Phys. Syst. ICSITech 2019, 2019, pp. 23–26, https://doi.org/10.1109/ICSITech46713.2019.8987575.
[6] I.F. Astuti, F.D. Nuryanto, P.P. Widagdo, D. Cahyadi, Oil palm fruit ripeness detection using K-Nearest neighbour, J. Phys. Conf. Ser. 1277 (2019), https://doi.org/10.1088/1742-6596/1277/1/012028.
[7] M.S.M. Alfatni, A.R.M. Shariff, S.K. Bejo, O.B.M. Saaed, A. Mustapha, Real-time oil palm FFB ripeness grading system based on ANN, KNN and SVM classifiers, IOP Conf. Ser. Earth Environ. Sci. 169 (2018), https://doi.org/10.1088/1755-1315/169/1/012067.
[8] Harsawardana, R. Rahutomo, B. Mahesworo, T.W. Cenggoro, A. Budiarto, T. Suparyanto, D.B. Surya Atmaja, B. Samoedro, B. Pardamean, AI-based Ripeness Grading for Oil Palm Fresh Fruit Bunch in Smart Crane Grabber, in: IOP Conference Series: Earth and Environmental Science, 2020, https://doi.org/10.1088/1755-1315/426/1/012147.
[9] Z. Ibrahim, N. Sabri, D. Isa, Palm oil fresh fruit bunch ripeness grading recognition using convolutional neural network, J. Telecommun. Electron. Comput. Eng. 10 (2018).
[10] S. Albawi, T.A. Mohammed, S. Al-Zawi, Understanding of a Convolutional Neural Network, in: Proc. 2017 Int. Conf. Eng. Technol. ICET 2017, 2018, pp. 1–6, https://doi.org/10.1109/ICEngTechnol.2017.8308186.
[11] J. Wang, T. Zhang, Y. Cheng, N. Al-Nabhan, New generation deep learning for video object detection: a survey, Comput. Syst. Sci. Eng. 38 (2021) 165–182, https://doi.org/10.32604/CSSE.2021.017016.
[12] A.I.B. Parico, T. Ahamed, Real time pear fruit detection and counting using YOLOv4 models and deep SORT, Sensors (Switzerland) 21 (2021) 1–32, https://doi.org/10.3390/s21144803.
[13] S. Chaudhary, M.A. Khan, C. Bhatnagar, Multiple Anomalous Activity Detection in Videos, in: Procedia Computer Science, 2018, https://doi.org/10.1016/j.procs.2017.12.045.
[14] R. Nabati, H. Qi, RRPN: Radar Region Proposal Network for Object Detection in Autonomous Vehicles, in: Proceedings - International Conference on Image Processing, ICIP, 2019, pp. 3093–3097, https://doi.org/10.1109/ICIP.2019.8803392.
[15] S. Singh, U. Ahuja, M. Kumar, K. Kumar, M. Sachdeva, Face mask detection using YOLOv3 and faster R-CNN models: COVID-19 environment, Multimed. Tool. Appl. 80 (2021) 19753–19768, https://doi.org/10.1007/s11042-021-10711-8.

[16] S. Singha, B. Aydin, Automated drone detection using YOLOv4, Drones 5 (2021), https://doi.org/10.3390/drones5030095.
[17] F. Xiao, Y.J. Lee, Video object detection with an aligned spatial-temporal memory. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), 11212 LNCS (2018) 494–510, https://doi.org/10.1007/978-3-030-01237-3_30.
[18] J. Redmon, A. Farhadi, YOLOv3: an Incremental Improvement, 2018.
[19] W. Chen, J. Zhang, B. Guo, Q. Wei, Z. Zhu, An apple detection method based on des-YOLO v4 algorithm for harvesting robots in complex environment, Math. Probl Eng. 2021 (2021) 1–12, https://doi.org/10.1155/2021/7351470.
[20] S. Zheng, Z. Lin, J. Xie, M. Liao, S. Gao, X. Zhang, T. Qiu, Maturity recognition of citrus fruits by Yolov4 neural network. 2021 IEEE 2nd Int. Conf. Big Data, Artif. Intell. Internet Things Eng. ICBAIE 2021 (2021) 564–569, https://doi.org/10.1109/ICBAIE52039.2021.9389879.
[21] A. Bochkovskiy, C.-Y. Wang, H.-Y.M. Liao, YOLOv4: Optimal Speed and Accuracy of Object Detection, 2020.
[22] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, A.C. Berg, SSD: single shot multibox detector. Lect. Notes comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), 9905 LNCS (2016) 21–37, https://doi.org/10.1007/978-3-319-46448-0_2.
[23] M. Tan, R. Pang, Q.V. Le, EfficientDet: Scalable and Efficient Object Detection, in: Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 2020, pp. 10778–10787, https://doi.org/10.1109/CVPR42600.2020.01079.
[24] G.J. Horng, M.X. Liu, C.C. Chen, The Smart image recognition mechanism for crop harvesting system in intelligent agriculture, IEEE Sensor. J. 20 (2020) 2766–2781, https://doi.org/10.1109/JSEN.2019.2954287.
[25] G. Li, X. Huang, J. Ai, Z. Yi, W. Xie, Lemon-YOLO: an efficient object detection method for lemons in the natural environment, IET Image Process. 15 (2021), https://doi.org/10.1049/ipr2.12171.
[26] B. Yan, P. Fan, X. Lei, Z. Liu, F. Yang, A real-time apple targets detection method for picking robot based on improved YOLOv5, Rem. Sens. 13 (2021), https://doi.org/10.3390/rs13091619.
[27] H. Zhu, H. Wei, B. Li, X. Yuan, N. Kehtarnavaz, A review of video object detection: datasets, metrics and methods, Appl. Sci. 10 (2020), https://doi.org/10.3390/app10217834.
[28] Y. Li, H. Wang, L.M. Dang, T.N. Nguyen, D. Han, A. Lee, I. Jang, H. Moon, A deep learning-based hybrid framework for object detection and recognition in autonomous driving, IEEE Access 8 (2020) 194228–194239, https://doi.org/10.1109/ACCESS.2020.3033289.
[29] M.A. Genaev, E.G. Komyshev, O.D. Shishkina, N.V. Adonyeva, E.K. Karpova, N.E. Gruntenko, L.P. Zakharenko, V.S. Koval, D.A. Afonnikov, Classification of fruit flies by gender in images using smartphones and the YOLOv4-Tiny neural network, Mathematics 10 (2022), https://doi.org/10.3390/math10030295.
[30] F. Li, Z. Liu, W. Shen, Y. Wang, Y. Wang, C. Ge, F. Sun, P. Lan, A remote Sensing and Airborne edge-computing based detection system for pine wilt disease, IEEE Access 9 (2021) 66346–66360, https://doi.org/10.1109/ACCESS.2021.3073929.
[31] C.Y. Wang, H.Y. Mark Liao, Y.H. Wu, P.Y. Chen, J.W. Hsieh, I.H. Yeh, CSPNet: A New Backbone that Can Enhance Learning Capability of CNN, in: IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work., 2020, pp. 1571–1580, https://doi.org/10.1109/CVPRW50498.2020.00203.
[32] P. Xu, Q. Li, B. Zhang, F. Wu, K. Zhao, X. Du, C. Yang, R. Zhong, On-board real-time ship detection in hisea-1 sar images based on cfar and lightweight deep learning, Rem. Sens. 13 (2021), https://doi.org/10.3390/rs13101995.
[33] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), 8691 LNCS (2014) 346–361, https://doi.org/10.1007/978-3-319-10578-9_23.
[34] S. Liu, L. Qi, H. Qin, J. Shi, J. Jia, Path Aggregation Network for Instance Segmentation, in: Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 2018, pp. 8759–8768, https://doi.org/10.1109/CVPR.2018.00913.
[35] H. Yu, C. Chen, X. Du, Y. Li, A. Rashwan, L. Hou, P. Jin, F. Yang, F. Liu, J. Kim, et al., TensorFlow Model Garden, 2020. https://github.com/tensorflow/models.
[36] J. Redmon, A. Farhadi, YOLO9000: Better, Faster, Stronger, in: Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017, 2017, pp. 6517–6525, https://doi.org/10.1109/CVPR.2017.690.
[37] A. Kaya, A.S. Keceli, C. Catal, H.Y. Yalic, H. Temucin, B. Tekinerdogan, Analysis of transfer learning for deep neural network based plant classification models, Comput. Electron. Agric. 158 (2019), https://doi.org/10.1016/j.compag.2019.01.041.
[38] Z. Zhang, T. He, H. Zhang, Z. Zhang, J. Xie, M. Li, Bag of Freebies for Training Object Detection Neural Networks, 2019.
[39] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, D. Ren, Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression, in: AAAI 2020 - 34th AAAI Conf. Artif. Intell., 2020, pp. 12993–13000, https://doi.org/10.1609/aaai.v34i07.6999.
[40] K. Wu, C. Bai, D. Wang, Z. Liu, T. Huang, H. Zheng, Improved object detection algorithm of YOLOv3 remote Sensing image, IEEE Access 9 (2021), https://doi.org/10.1109/ACCESS.2021.3103522.
[41] P.M. Blok, F.K. van Evert, A.P.M. Tielen, E.J. van Henten, 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. 38 (2021), https://doi.org/10.1002/rob.21975.
[42] A. Buslaev, V.I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, A.A. Kalinin, Albumentations: fast and flexible image augmentations, OR Inf. 11 (2020), https://doi.org/10.3390/info11020125.
[43] A. Rehman, S. Naz, M.I. Razzak, F. Akram, M. Imran, A deep learning-based framework for automatic Brain Tumors classification using transfer learning, Circ. Syst. Signal Process. 39 (2020), https://doi.org/10.1007/s00034-019-01246-3.
[44] M. Kruithof, H. Bouma, N. Fischer, K. Schutte, Object Recognition Using Deep Convolutional Neural Networks with Complete Transfer and Partial Frozen Layers, in: Optics and Photonics for Counterterrorism, Crime Fighting, and Defence XII, vol. 9995, SPEI, 2016, pp. 159–165, https://doi.org/10.1117/12.2241177.