sustainability

MDPI

Article

Enhancing Image Annotation Technique of Fruit Classification Using a Deep Learning Approach

Normaisharah Mamat [1], Mohd Fauzi Othman [1,*], Rawad Abdulghafor [2,*], Ali A. Alwan [3] and Yonis Gulzar [4]

[1] Department of Electronic System Engineering, Malaysia-Japan International Institute of Technology, Universiti Teknologi Malaysia, Jalan Sultan Yahya Petra, Kuala Lumpur 54100, Malaysia
[2] Computational Intelligence Group Research, Faculty of Information and Communication Technology, International Islamic University Malaysia, Kuala Lumpur 53100, Malaysia
[3] Schools of Theoretical and Applied Science, Ramapo College of New Jersey, Mahwah, NJ 07430, USA
[4] Department of Management Information Systems, College of Business Administration, King Faisal University, Al-Ahsa 31982, Saudi Arabia
* Correspondence: mdfauzi@utm.my (M.F.O.); rawad@iium.edu.my (R.A.)

Abstract: An accurate image retrieval technique is required due to the rapidly increasing number of images. It is important to implement image annotation techniques that are fast, simple, and, most importantly, automatically annotate. Image annotation has recently received much attention due to the massive rise in image data volume. Focusing on the agriculture field, this study implements automatic image annotation, namely, a repetitive annotation task technique, to classify the ripeness of oil palm fruit and recognize a variety of fruits. This approach assists farmers to enhance the classification of fruit methods and increase their production. This study proposes simple and effective models using a deep learning approach with You Only Look Once (YOLO) versions. The models were developed through transfer learning where the dataset was trained with 100 images of oil fruit palm and 400 images of a variety of fruit in RGB images. Model performance and accuracy of automatically annotating the images with 3500 fruits were examined. The results show that the annotation technique successfully annotated a large number of images accurately. The *mAP* result achieved for oil palm fruit was 98.7% and the variety of fruit was 99.5%.

Keywords: image annotation; repetitive annotation task; large dataset; oil palm FFB; classification

Citation: Mamat, N.; Othman, M.F.; Abdulghafor, R.; Alwan, A.A.; Gulzar, Y. Enhancing Image Annotation Technique of Fruit Classification Using a Deep Learning Approach. *Sustainability* **2023**, *15*, 901. https://doi.org/10.3390/su15020901

Academic Editor: Hong Tang

Received: 1 December 2022
Revised: 23 December 2022
Accepted: 30 December 2022
Published: 4 January 2023

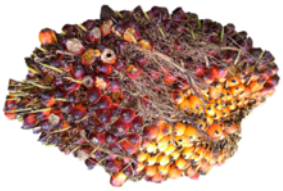
1. Introduction

Annotation is a work that involves a lot of repetition when performed completely manually. Numerous artificial intelligence-related tasks require massive datasets. Although it could be possible for a person to annotate everything, doing so might not be desired. The automatic image annotation (AIA) technique is a technology that can annotate images automatically with its semantic tags. Significant features of AIA include image retrieval [1], classification [2], recognition [3], and medical diagnostics [4,5]. AIA is able to adapt to complex patterns as more training data become available, and deploys a common strategy to annotate a new image. In order to annotate, firstly, similar images from the training set are retrieved, and then labels are ranked based on their frequency in the retrieval set. The most frequent labels in the neighborhood are thus transferred to the test image to achieve automatic image annotation [6].

Computer vision in agriculture automation is challenging due to the considerable variation within a class of fruit species as well as their similarities in color, size, and shape. As a result, manually annotating fruit takes time and effort. The detection and accurate classification of fruits is a fascinating issue associated with enhancing the quality and economic potential of fruits, especially in an industrial field. The challenge become more significant when automating tasks such as matching fruit quality with other information

such as nutritional facts and pricing [7]. The classification of oil palm fresh fruit bunch (FFB) ripeness is significant in ensuring the quality of the oil. The ripeness of oil palm fruits dictates the quality of palm oil produced and overall marketability. The color of the oil palm FFB may be used to estimate its ripeness. Color is one of the most important characteristics for determining fruit ripeness [8]. The color of an item is determined by the light reflected from it. Therefore, these changes serve as a foundation for image processing and analysis. The primary components of the color coding are red, green, and blue (RGB). The Malaysian Palm Oil Board [9] has classified unripe, underripe, ripe, and overripe based on color, as shown in Table 1.

Table 1. Oil palm fruit ripeness and color classification.

Category	Figure	Color
Unripe		Black
Underripe		Reddish black
Ripe		Red
Overripe		Reddish orange

Manually grading oil palm fruit is a typical technique for identifying its quality, but this technique is time-consuming and may result in human error [10,11]. It is crucial to identify the ripeness of oil palm fruit. Incorporating artificial intelligence, computer technology has provided a variety of solutions to alleviate this dependence. Many researchers have recently used artificial intelligence techniques for object detection and classification challenges, with beneficial outcomes [12]. A reliable, fast, and accurate approach for detecting oil palm FFB ripeness is required. Therefore, AIA using a deep learning approach gives both academic and commercial applications that benefit greatly from this technique. The automatic annotating of oil palm fruit classification can assist farmers in increasing production and make work easier. Oil palm is often used to make margarine, candles, soaps, home cooking oil, and snacks, and it is Malaysia's main agricultural commodity export [13].

Despite the prevalent deep learning-based strategies for improving AIA framework implementation, AIA remains vulnerable to several critical issues. Among these difficulties is the need for a large number of data to make an accurate prediction. The control of

inconsistent keyword distribution, as well as the selection of relevant characteristics, are the other two primary AIA problems [14]. With the development of artificial intelligence, deep learning is widely used in image annotation. Deep learning, which encompasses artificial neural networks and computational models, is a subset of the machine learning process. Its method is designed to replicate the topology of biological neural networks and mimic the function of the brain [14]. When the brain acquires new information, it seeks to make sense of it by comparing it to previously acquired knowledge. Deep learning decodes information using the same approach that the brain employs to categorize and identify items. Deep learning accelerates and simplifies this process, which is particularly beneficial to data scientists who are tasked with obtaining, analyzing, and interpreting huge volumes of data [15,16]. YOLO is a regression issue that combines target classification and localization. A YOLO network uses regression to recognize targets in an image without the need for RPN. The network has the ability of the human visual system to recognize objects instantly [17]. Moreover, YOLO is extremely efficient and works impressively well for real-time object detection [18]. Nowadays, there are several YOLO variants with various architectures. The original YOLO has 24 convolutional layers preceded by two fully connected layers.

Annotation strategies that are fast and simple to use are recommended for effectively overcoming such obstacles. AIA approaches aim to develop a model from the training data and then use the trained model to automatically give semantic labels to the new image. With the recent attention and development of AIA in contributing to significant tasks, this study is about enhancing automatically annotated image techniques, namely, repetitive annotation tasks. This AIA method-enhancing technique contributes to solving the problem of massive image data and thus contribute to the time consumption and human energy needed to manually annotate an image. A repetitive training task to annotate images and implement deep learning techniques will increase the accuracy and efficiency of the AIA technique. The proposed repetitive annotation technique can be applied in various deep learning methods to automatically annotate the object. However, to evaluate the effectiveness of the proposed technique, this study chooses YOLOv5 as the algorithm platform to generate accurate predictions, as YOLOv5 generates high accuracy and fast performance. The annotation of oil palm FFB using a repetitive annotation task assists farmer with identifying the ripeness of oil palm FFB from the process of harvesting until the milling process.

2. Related Works

In recent decades, computer vision researchers have successfully endeavored to invent computer systems capable of imitating this human skill. AIA is a step ahead in this approach, detecting each item in an image and assigning appropriate tags to explain its content. AIA has made breakthroughs in the agricultural industry through numerous advanced equipment systems and procedures, making this field more productive and profitable. Various works presented in the literature address the technique of AIA in agriculture. Nemade and Sonavane [19] examined the annotation of fruit by deploying co-occurrence patterns. Identifying fruit quality categories and combination attributes that contribute to co-occurrence patterns can be accomplished with the aid of this. The findings indicate that, for the fruit categories, the co-occurrence pattern using SVM yields an overall accuracy of 97.3%. Instead of the traditional two-step procedure of acquisition followed by human annotation, Samiei et al. [20] evaluated the value of several egocentric vision approaches for performing joint acquisition and AIA. This approach is used in automatic apple segmentation and obtained high performance in annotating images by implementing a machine learning application. The review of image annotation techniques in the agriculture field has been proposed by Mamat et al. [21]. The study summarized the implementation of deep learning techniques, the image annotation approach, and the various applications of deep learning techniques in the agriculture industry.

A lack of accessibility to efficient categorization systems might be a problem for farmers. The texture, shape, and color of a fruit are used to grade its ripeness, which may

lead to variations and inefficiency in grading. Many methods have been introduced to address the obstacle and implement deep learning techniques to categorize the ripeness of oil palm fruit. Jamil et al. [22] established the first artificial intelligence system for oil palm fruit ripeness classification in 2009. Their AI system uses a Neuro-Fuzzy model that had been trained on color data collected from 90 images. The algorithm correctly classified 45 test photos with 73.3% accuracy [23]. Using the deep learning method in the agriculture field, Khamis et al. [24] proposed YOLOv3, Elwirehardja and Prayoga [10] deployed MobileNetv1, Liu et al. [25] deployed YOLOv4-tiny, Janowski et al. [26] implemented YOLOv5 in detecting apples, and Herman [27] used DenseNet to classify the ripeness of oil palm fruit in their study. The application of AIA techniques is useful in increasing the harvesting fruit process. The implementation of a harvesting robot [28] using computer vision was used to pluck fruit from the tree used on farmers' requirements. Furthermore, these AI-enabled computers are developed using training datasets generated by image annotation. Tang et al. [29] reviewed all the applications of fruit-picking robots using machine vision and related developing technology that have enormous promise in sophisticated agriculture applications.

YOLO update version 4, commonly known as YOLOv4, was released in early 2020 by Alexey Bochkovskiy [30], a Russian developer who produced the first three versions of YOLO utilizing Joseph Redmon's [31] Darknet architecture. Glenn Jocher [32] and his ultralytics LLC research division, who developed YOLO algorithms using the PyTorch framework, released YOLOv5 a month after YOLOv4. YOLOv5 is simple and efficient. It requires far fewer CPU resources than other designs while producing equivalent results and performing significantly faster than previous YOLO versions [33]. The significance of YOLOv5 makes it widely used in agricultural areas [34,35]. Wang et al. [36] detected real-time apple stems by deploying YOLOv5. The study was first conducted by figuring out the hyper-parameter and using transfer learning as a training approach to achieve stronger detection performance. Next, networks with different depths and widths were trained to find the baseline detection. Subsequently, the YOLOv5 was optimized for this task by using the detection of head searching, layer, and channel pruning. The results from the study showed that YOLOv5 was easier to use under the same setting and could be chosen as the baseline network based on how well it detected things. Other applications of YOLOv5 in agriculture has been proposed in crop detection by Yan et al. [37], classification by Wang et al. [38], disease recognition by Chen et al. [39], and counting by Lyu et al. [40].

Inspired by the previous research, this study chooses YOLOv5 as the method to investigate the proposed repetitive annotation task technique, since this method is founded on excellence in the detection of an object. YOLOv5 is compared to other variations of YOLO, which are YOLOv3 and YOLOv4, to evaluate its performance.

3. Materials and Method

3.1. Dataset

The images of oil palm fruit were collected in the orchards located in Felda Tenang, Jerteh, Terengganu, Malaysia. A total of 400 images of oil palm FFB were collected for four different categories, which are unripe, underripe, ripe, and overripe. These images were then expanded to 600 images using the data-augmentation method. The collected images were captured by using a smartphone and unmanned aerial vehicle (UAV) with DJI Phantom 4 with 3472 × 4640 pixel and 3840 × 2160 pixel resolution. The image was taken with red, green, and blue (RGB) colors to identify the ripeness of the fruit. Figure 1 shows a drone used to capture the image of a tall oil palm tree. All the image sizes were resized to 416 × 416 pixel resolution to fit the size of the common required deep learning algorithm. This work was implemented in Python in the Google Collaboratory platform running on Windows 10. Utilizing Google's environment offers free access to the graphics processing unit and requires some configurations. The system configuration has a 16 GB RAM Intel(R) Core(TM) i5 processor. At first, only 152 images of oil palm fruit were annotated manually by using the LabelImg tool. The categories were drawn manually and classified using

bounding boxes. A variety of fruit, consisting of rambutan, dragon fruit, pineapple, and mangosteen, were downloaded from Google Image and Kaggle datasets. The variety of these fruits was to evaluate the performance of the capability of automatically annotating the image in large datasets. A total of 3400 fruit images were used and only 400 images were firstly manually annotated. The dataset used in this study is elaborated in Tables 2 and 3.

Figure 1. Drone used to capture images and samples images from the drone dataset.

Table 2. Dataset of oil palm fruit.

	Number of Training Images	Testing Image
Unripe	38	150
Underripe	38	150
Ripe	38	150
Overripe	38	150
Total	152	600

Table 3. Dataset of fruits.

	Number of Training Images	Testing Image
Rambutan	100	875
Dragon fruit	100	875
Pineapple	100	875
Mangosteen	100	875
Total	400	3500

3.2. Deep Learning Method

3.2.1. YOLO

The YOLO neural network architecture predicts a set of bounding boxes and class probabilities. Figure 2 is an illustration of the YOLO framework. The fundamental idea is to split the input image into $S \times S$ grid cells and perform detections in each grid cell. Each cell predicts B bounding boxes as well as their confidence. The confidence may indicate whether or not an item exists in the grid cell, as well as the intersection over union (IoU) of the ground truth and predictions. Equation (1) is utilized to express confidence [41]. Pr signifies the probability that the cell contains an object within the predicted bounding box and IoU is the intersection of the predicted bounding box and the ground truth.

$$Confidence = Pr(object) \times IoU(Ground\ truth,\ prediction) \quad (1)$$

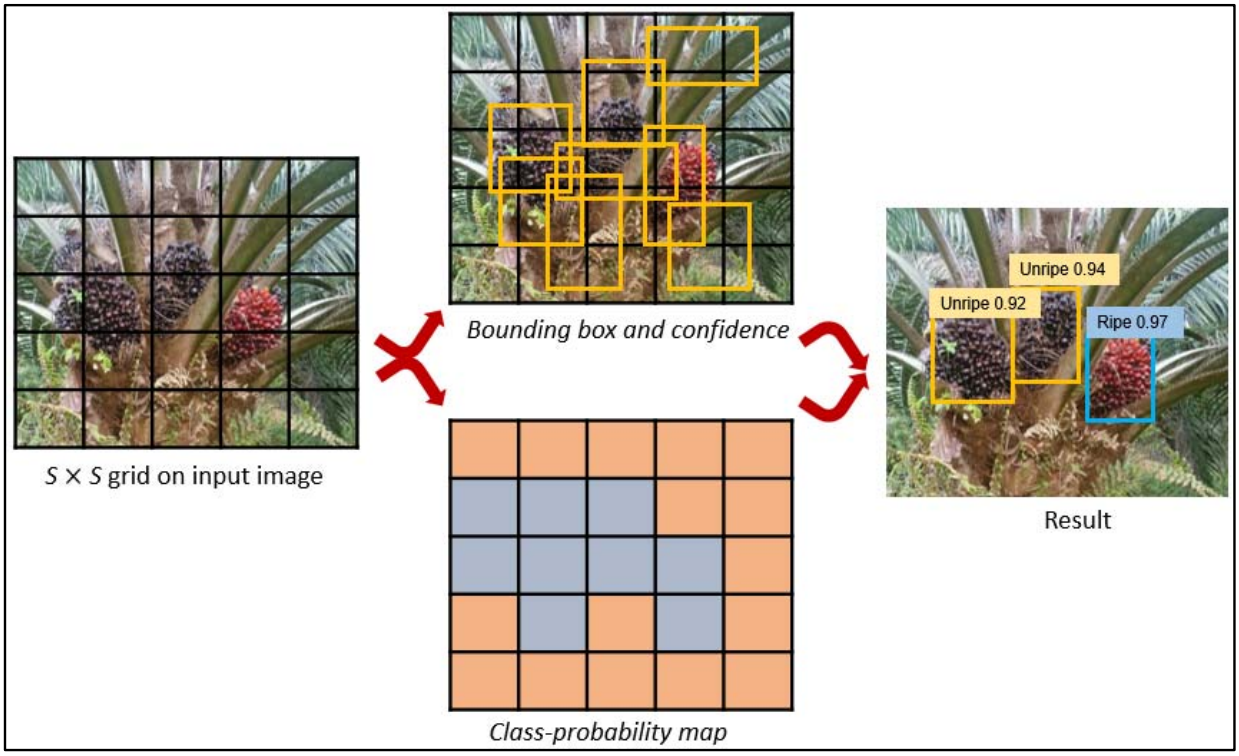


Figure 2. YOLO model.

YOLO is a one-stage object detector that detects objects quickly from beginning to end. Images are downsized to a reduced resolution in YOLO algorithms and then a single CNN runs on the images, returning detection results based on the model's confidence threshold. YOLO's first version was developed to reduce the sum of squared errors (loss function). This optimization improves identification speed but decreases accuracy in comparison to state-of-the-art object detection models. YOLO comes in a variety of forms. The feature extraction backbone of Darknet19, which struggled with detecting small objects in YOLOv3, was changed to Darknet53 in YOLOv3. Residual blocks, skip connections, and up-sampling were introduced in that work, significantly improving the algorithm's accuracy. The feature extractor's backbone was changed to CSPDarknet53 in YOLOv4, which significantly improved the algorithm's speed and accuracy. YOLOv5 is the lightest version of previous YOLO algorithms and it employs the PyTorch framework rather than the Darknet framework. YOLOv5 is mainly utilized in this study for object identification and categorization.

3.2.2. Network Architecture

The entire architecture of YOLOv5, which includes the backbone, detection neck, and detection head, is depicted in Figure 3.

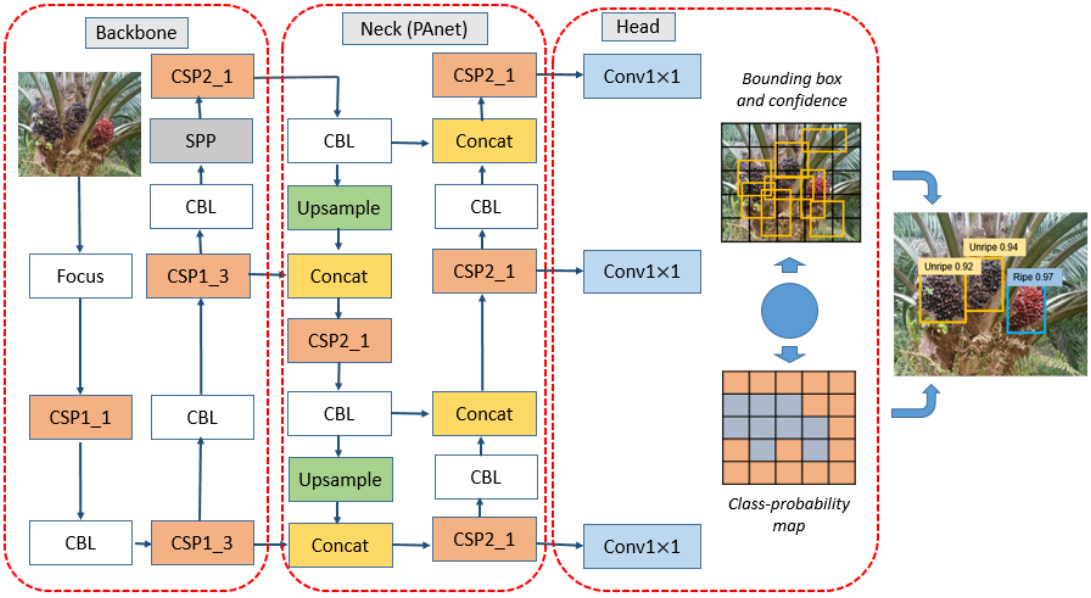


Figure 3. YOLOv5 architecture representation.

(i) Backbone model

The model backbone is mostly utilized to extract important information from an input image. The focusing layer is the first layer of the backbone network and is used to simplify the model calculation and speed-up training. Second, concatenation is employed to integrate the four segments in depth. The output feature map is then created using a convolutional layer comprised of 32 convolution kernels. Finally, the results are fed into the next layer through the batch normalization layer and the activation functions. The bottleneckCSP module is the third layer of the backbone network, and it is intended to efficiently extract the image's detailed information. BottleneckCSP is essentially composed of a bottleneck module, which is a residual network architecture that connects a convolutional layer. The bottleneck module's complete output is the sum of this part's output of the beginning input through the residual structure. The spatial pyramid pooling (SPP) module is the backbone network's ninth layer, and it is intended to boost the network's receptive field by adapting any size of the feature map into a fixed-size feature vector. After being subsampled via three concurrent max-pooling layers, this feature map and the output feature map are coupled in-depth [37].

(ii) Neck model

The model neck is primarily utilized in the generation of feature pyramids. It is formed of a feature-pyramid network (FPN) and a path-aggregation network (PAN). When it comes to object scaling, feature pyramids help models generalize effectively. As a consequence, it makes it easier to identify the same object in different sizes and scales [35].

(iii) Head model

The front segment of the network accomplishes the entire fusion of low-level features and high-level features via the feature pyramid structure and PAN to generate rich feature maps. The final detection stage is essentially the responsibility of the model head, which employs anchor boxes to create final output vectors containing class probabilities, accuracies, and bounding boxes.

YOLOv5's loss function is the sum of the regression function for the bounding box, the confidence loss, and the classification loss. It is determined as Equation (2), where l_{bx} is the regression function for the bounding box, l_j is the confidence loss function, and l_s is the classification loss function [42].

$$Loss\ fucntion = l_{bx} + l_s + l_j \quad (2)$$

The variables of l_{bx}, l_s, and l_j are calculated as shown in Equations (3)–(5). h' and w' are the height and width of the target, y_i and x_i are the correct coordinates of the target, λ_{cd} is the indicator function of whether cell i contains an object, λ_s is the classification loss function, λ_{noj} is the category loss coefficient, c is the confidence score, and c_l is the class.

$$l_{bx} = \lambda_{cd} \sum_{i=0}^{s^2} \sum_{m=0}^{b} I_{i,m}^{i} bj(2 - W_i \times h_i)[\left(x_i - x'^{m}_{i}\right)^2 + \left(y_i - y'^{m}_{i}\right)^2 + \left(w_i - w'^{m}_{i}\right)^2 + \left(h_i - h'^{m}_{i}\right)^2] \quad (3)$$

$$l_s = \lambda_s \sum_{i=0}^{s^2} \sum_{m=0}^{b} I_{i,m}^{i} \sum_{C \in cl} V_i(c) \log(VV_i(c)) \quad (4)$$

$$l_j = \lambda_{noj} \sum_{i=0}^{s^2} \sum_{m=0}^{b} I_{i,m}^{noj} (c_i - c_l)^2 + \lambda_j \sum_{i=0}^{s^2} \sum_{m=0}^{b} I_{i,m}^{j} (c_i - cc_l)^2 \quad (5)$$

3.3. Transfer Learning

Deep learning has a complicated structure. Overfitting and performance issues arise as training data decrease. Its performance improves as the number of training data increase. As a result, in various deep learning applications, a transfer learning approach that trains information from a particular field with a pre-trained system in advance by abundant data in a related field is extensively utilized [43]. The initial layers in the convolutional process extract the general characteristics, and, as the process to the final layers, the transition to features that are more specialized to the dataset is trained on. Transfer learning has evolved as a result of these layer feature transfers. As a consequence, the model's characteristics learned on the main task are used in transfer learning for an unrelated next task [44]. During deep learning training, the model is fed a large number of data and accumulates model weight and bias. These weights are then used to test different network models. The new model can begin with weights that have already been trained [45]. Figure 4 shows the process of transfer learning. Transfer learning is a handy technique for fast retraining of a model on fresh data without retraining the whole network. Instead, a portion of the initial weights is held constant, while the remainder of the weights are utilized to calculate loss and are updated by the algorithm.

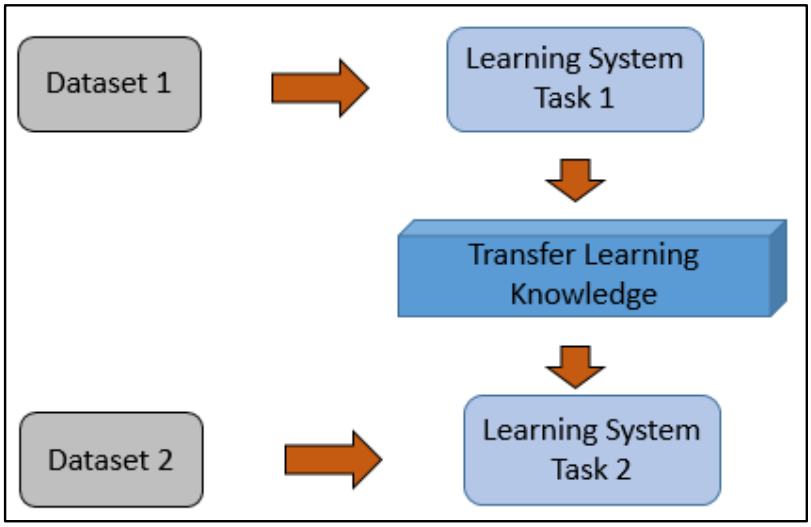


Figure 4. Transfer learning process.

3.4. Automatic Image Annotation

AIA has been a prominent study area in recent years since it has the ability to annotate enormous datasets. To address the so-called semantic gap issue, AIA approaches are developed. In contrast to content-based image retrieval, automated annotations may benefit

from image search by using high concepts automatically. The AIA approach is considered to be a method that is quick for text-based image retrieval. However, this approach is not sophisticated enough to extract complete semantic meanings. Many researchers have analyzed image annotation techniques in response to the increasing need for image annotation. Therefore, this study proposed a repetitive annotation method to automatically annotate large datasets, as shown in Figure 5. The first dataset is processed to generate transfer learning to annotate new dataset images. Next, the test image is automatically annotated and repeated in the system and combined as a new dataset. This process increases the accuracy performance in the annotation of an object. The process is repeated until optimum accuracy and high efficiency are obtained.

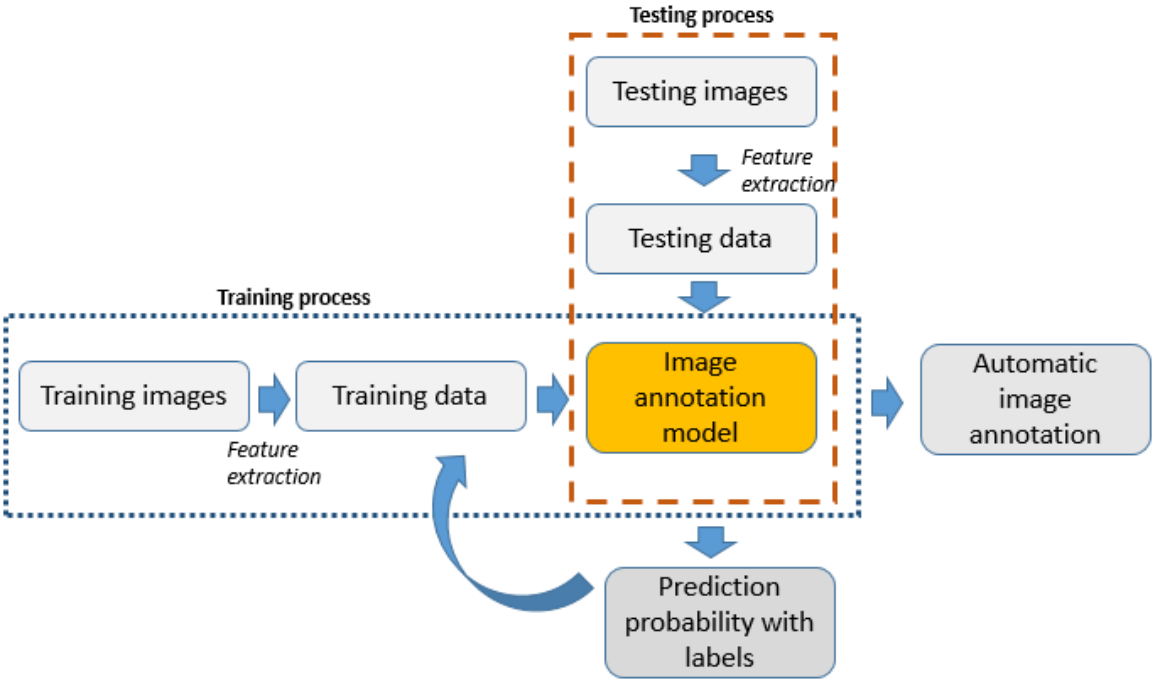


Figure 5. Automatic image annotation model based on a repetitive annotation task method.

3.5. Performance Metrics

Several parameters may be utilized to evaluate the effectiveness of the YOLO algorithm. The average precision (*AP*), recall, and mean average precision (*mAP*) are the performance metrics assessed in this study.

The expression of these evaluations is described as follows:

$$P = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{6}$$

$$R = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{7}$$

$$AP = \int_0^1 P(R)dR \tag{8}$$

$$mAP = \frac{1}{|Q_R|} \sum_{q=Q_R} AP(q) \tag{9}$$

The average accuracy when the *P* index is integral to the *R* index or the area under the *P*–*R* curve is denoted by *AP*, and *mAP* is the average accuracy of the mean calculated by diving the total of *AP* values for all categories. The *mAP* calculates a score based on how accurately the detected bounding box matches the ground-truth box. In this study,

evaluating *mAP* is denoted by the notation of *mAP*@0.5, meaning that *mAP* is calculated at an *IoU* threshold 0.5.

4. Results and Discussion

Three versions of YOLO were developed to evaluate the training performance for the oil palm fruit dataset. Table 4 shows the results obtained for all version models after the first training dataset including precision, recall, *mAP*, and training time. The accuracy generated by YOLOv5 is higher compared to the other versions. In fact, the training time produced is faster, with 0.609 h compared to YOLOv3 and YOLOv4 with 0.896 and 0.876, respectively. Figure 6a–c show the detection results of oil palm FFB for these three versions of YOLO for the classification of the ripeness of oil palm FFB. All the YOLO version's learning rates were set at 0.01 and the model training batch size was set to 32. The value of the *IoU* was set to 0.2. At the optimized rapid performance, the training epoch value was set to 100. The model was continuously trained and performed effectively. The last weight result for the model was stored after training and the test set of 1000 images was used to assess the model's performance. Next, the test images were deployed as a new dataset and combined with the first trained dataset. This method was utilized to increase the annotation accuracy for further test images.

Table 4. Training model for three versions of YOLO.

Models	Precision	Recall	*mAP*@50	Training Time (h)
YOLOv3	67%	70%	78%	0.896
YOLOv4	74%	83%	85%	0.876
YOLOv5	98%	97%	99%	0.609

The YOLOv5 model, which was trained on the custom dataset, was fine-tuned. The first test dataset, consisting of 150 images, was used to classify their ripeness using previously trained oil palm fruit detection algorithms. Precision, recall, and *mAP*@50 were used in the comparison. Furthermore, annotation speed was measured in frames per second (FPS) for each model to investigate the feasibility of using previously trained models in real-time applications. As the test images were unfamiliar to the training models, the metrics produced on this test dataset varied from the previously calculated metrics.

The repetitive annotation method at the second annotation generated 98.7% for oil palm FFB and was then tested on another 1000 new images, and the results are shown in Figure 7a–h. The ripeness classification of oil palm fruit was successfully automatically annotated with a bounding box and accuracy value. The algorithm also trained with 20, 40, 60, 80, and 100 epochs to examine the accuracy performance and the efficiency of the model. The results obtained for each epoch and each performance are shown in Figure 8a–f. The TensorBoard tool was used to visualize all of the network's statistical data. According to the figures, the accuracy value for *mAP*@50 at 100 epochs achieves a training accuracy of nearly 100%. Moreover, the network from which we observed the validation loss graph decreases concurrently with the training loss. Given higher accuracy and lower losses at 100 epochs, this study fixed the training epochs at 100 epochs to generate high efficiency in object detection and annotation tasks.

Table 5 shows the outcomes of the annotation precision, recall, *mAP*, and time comparison for the training, second annotation, and third annotation process using repetitive annotation tasks. Each image's annotation time was calculated using all of the annotation methods. There were statistically significant differences between the second annotation process, training process, and second annotation task. The average detection speed for the ripeness classification for the training process, first train, and second train were 0.55 ms, 0.43 ms, and 0.3 ms, respectively. The training time for the annotation process increased to generate a better result, however, the test speed FPS outcome was faster. A faster the test dataset is significant in the application of real-time capturing images and harvesting robots.

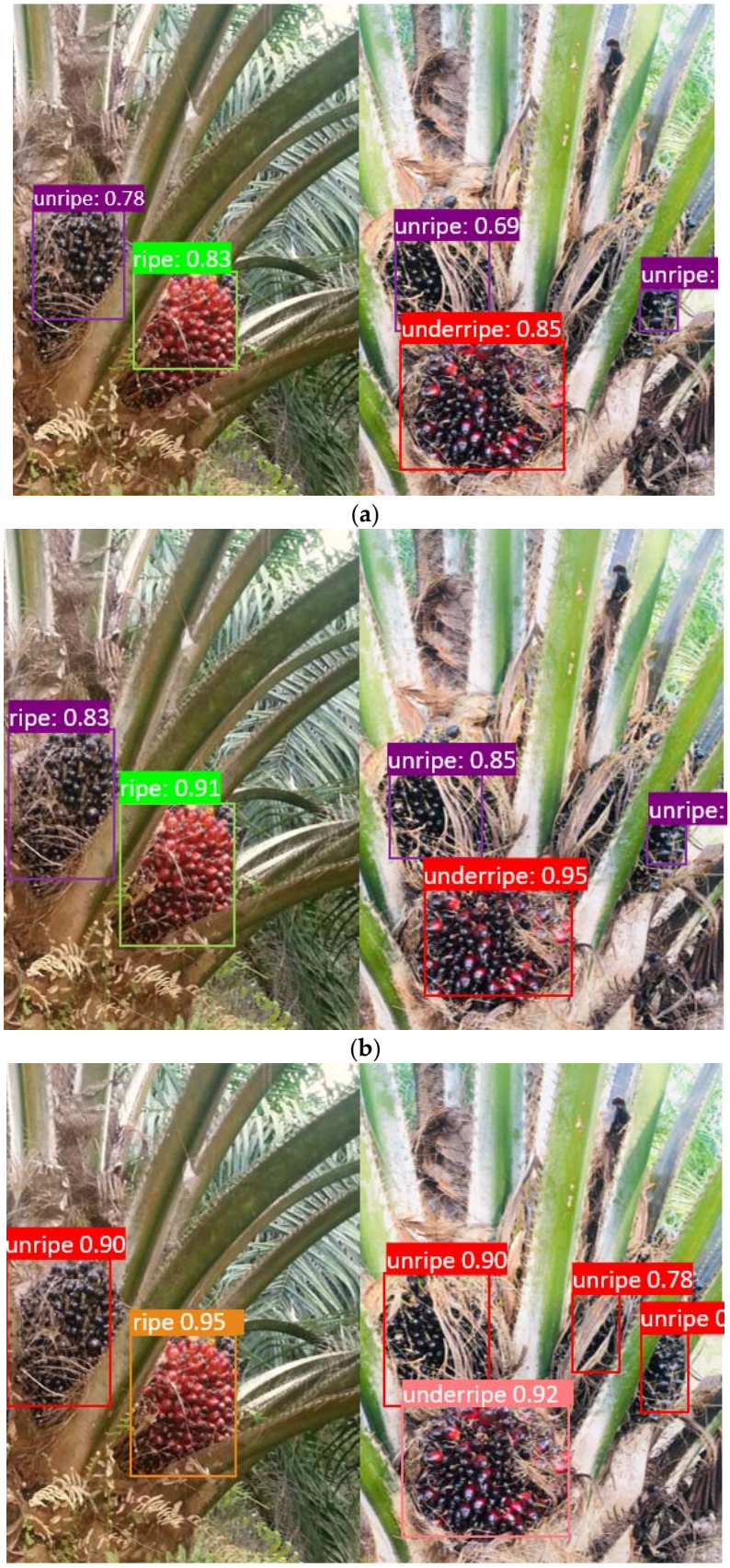


Figure 6. Training results for (**a**) YOLOv3, (**b**) YOLOv4, and (**c**) YOLOv5.

Figure 7. Automatically annotate image of oil palm fruit: (**a**) unripe, (**b**) underripe, (**c**) ripe, (**d**) overripe, and (**e**–**h**) all classifications.

The technique of repetitive annotation was then evaluated with the larger dataset, tested on a variety of fruits consisting of rambutan, dragon fruit, pineapple, and mangosteen. The epoch was set to 30 for the training task. The annotation results with the bounding box obtained after the second annotation process are shown in Figure 9.The performance curves for *mAP*, precision, recall, bounding box regression loss and classification loss depicted by red lines are shown in Figure 10a–f. The outcomes of the annotation precision, recall, *mAP*, and time comparison for the various fruit dataset are shown in Table 6. The accuracy recorded for the second training for a variety of fruit was 99.5%. The accuracy obtained was better compared to the accuracy of oil palm fruit due to the large volume of the dataset used with the variety of fruit, thus producing better predictive performance. Moreover, a larger dataset enhances the probability that the data may include relevant information. There are unstable values for precision and recall. However, in the detection case, most of the cases are evaluated based on the *mAP* due to its value produced by calculating the average precision for each class and then averaging across several classes. Moreover, *mAP* takes into consideration both false positives (FP) and false negatives (FN), and reflects the trade-off between accuracy and recall. Based on this feature, *mAP* is a good measure for most detection applications. There is no accuracy improvement for the

first and second annotations, which may occur because the model eventually reaches a point where increasing a dataset will not improve the accuracy. At this point, the model can be playing around with the learning rate or epoch values. Even though there is no enhancement accuracy, the time required to generate an annotation for a new test image is decreased. This benefit may lower the time required to classify further huge numbers of images. Since the accuracy value achieved is almost 100%, this result obtains satisfactory performance shown in employing the repetitive annotation task method. The average detection speed for the fruit classification for the training process, first annotation, and second annotation recorded are 0.44 ms, 0.32 ms, and 0.25 ms.

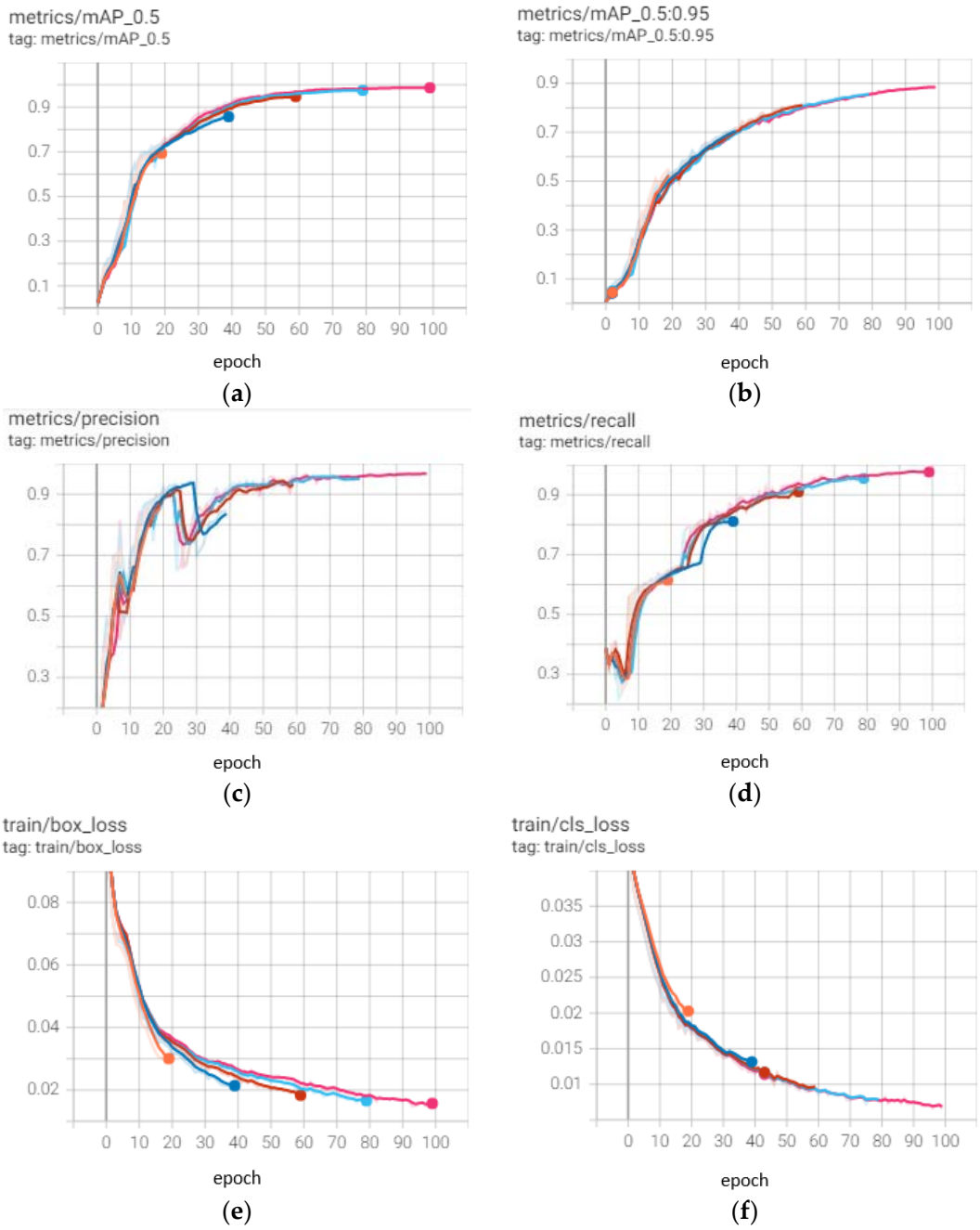


Figure 8. Performance of the last training on the fruit dataset for epochs 20, 40, 60, 80, and 100 are depicted as red, blue, maroon, sea blue, and dark pink, respectively: (**a**) *mAP*@0.5 (**b**) *mAP*@0.95 (**c**) precision (**d**) recall (**e**) bounding box regression loss and (**f**) classification loss.

Table 5. Repetitive annotation task results for oil palm fruit.

Models		Training Process	First Annotation	Second Annotation
Precision	Unripe	97.1%	98.5%	98.7%
	Underripe	97.3%	97.6%	98.0%
	Ripe	98.6%	98.7%	97.5%
	Overripe	99.7%	91.4%	90.0%
	Total	98.2%	96.6%	96.7%
Recall	Unripe	98.6%	97.3%	97.9%
	Underripe	95.7%	97.5%	97.8%
	Ripe	95.9%	98.7%	98.4%
	Overripe	84.4%	87.6%	88.9%
	Total	93.6%	97%	97.5%
mAP@50	Unripe	99.3%	99.4%	99.4%
	Underripe	98.5%	99.2%	99.2%
	Ripe	99.2%	99.4%	99.4%
	Overripe	95.1%	98.9%	96.9%
	Total	98%	98.2%	98.7%
Training time (h)		0.187	0.306	0.412
Test dataset speed (FPS)		0.54 ms	0.43 ms	0.3 ms

Figure 9. Automatically annotated images of variety of fruit.

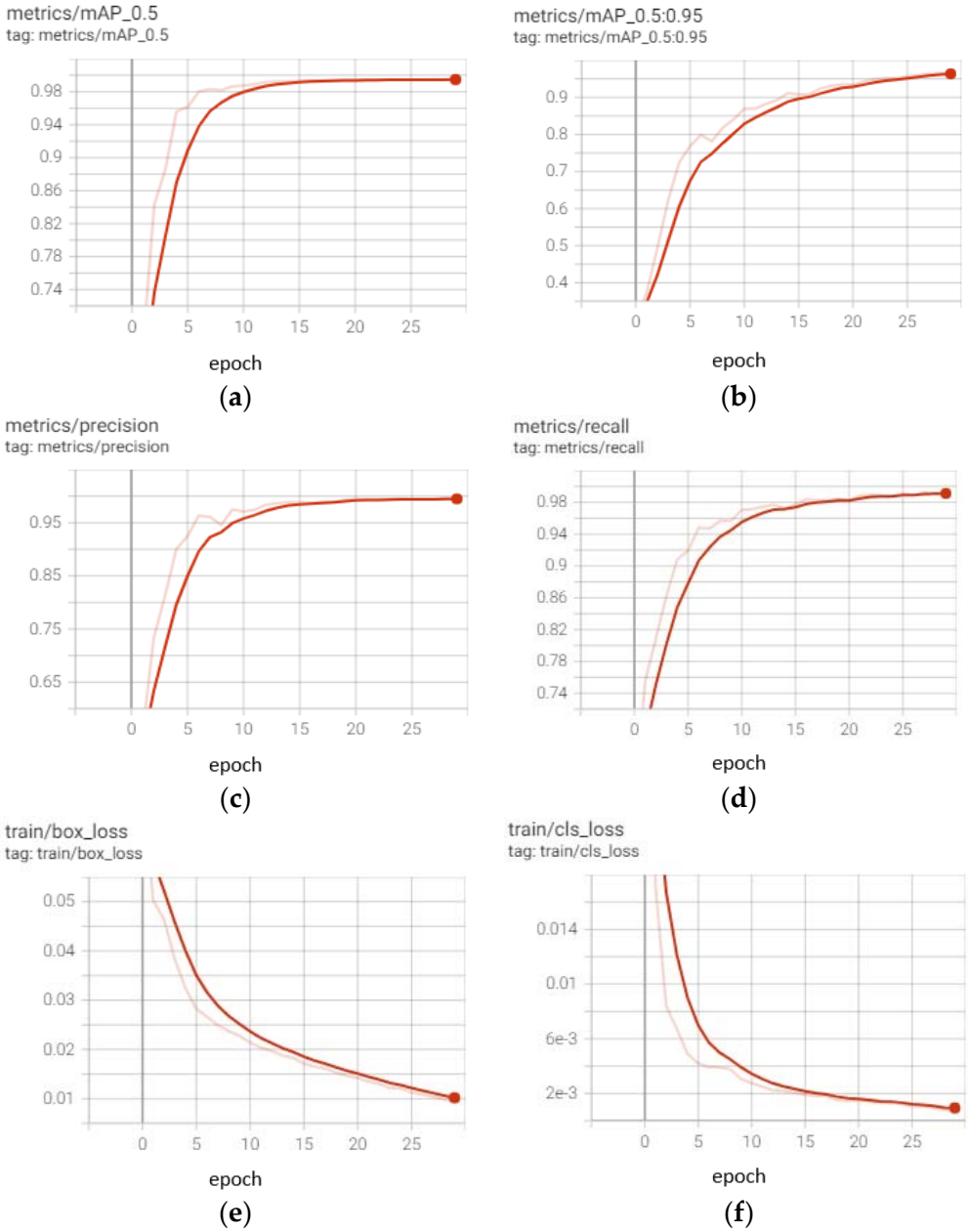


Figure 10. Performances of the last training for the fruit dataset: (**a**) *mAP*@0.5 (**b**) *mAP*@0.95 (**c**) precision (**d**) recall (**e**) bounding box regression loss and (**f**) classification loss.

Based on the findings, it can be demonstrated that the approach technique of repetitive annotation tasks in automatic image annotation has effectively annotated new images with high accuracy. With accurately annotated data, computer vision systems can identify and classify a variety of objects in a huge number of images. In contrast, the proposed method based on the YOLOv5 architecture performed well with the provided dataset. The classification of oil palm fruit maturity or ripeness determines the quality of palm oil produced and its overall marketability. Using this proposed method, the classification of FFB could be employed to address an obstacle in fruit processing for oil production.

Table 6. Repetitive annotation task results for variety of fruit.

Models		Training Process	First Annotation	Second Annotation
Precision	Rambutan	96.6%	98.6%	99.0%
	Dragon fruit	96.3%	96.2%	99.9%
	Pineapple	97.6%	97.2%	99.9%
	Mangosteen	98.9%	99.2%	99.0%
	Total	97.3%	98.8%	99.5%
Recall	Rambutan	94.8%	98.3%	98.7%
	Dragon fruit	99.6%	99.7%	99.8%
	Pineapple	98.9%	98.6%	99.6%
	Mangosteen	99.3%	98.5%	98.6%
	Total	99.1%	99.3%	99.2%
mAP@50	Rambutan	99.5%	99.5%	99.5%
	Dragon fruit	99.2%	99.5%	99.5%
	Pineapple	99.1%	99.5%	99.5%
	Mangosteen	98.1%	99.5%	99.5%
	Total	99.0%	99.5%	99.5%
Training time (h)		0.116	0.452	0.690
Test dataset speed (FPS)		0.44 ms	0.32 ms	0.25 ms

5. Conclusions

In the agricultural sector, robotics, drones, and AI-enabled machines are employed to accomplish a variety of jobs. All of this equipment is based on computer vision technology. When image annotation is performed for the agriculture industry, numerous crops and plants are annotated according to model requirements, such as their ripeness and disease. Therefore, this study proposed an automatic image annotation advancement approach that employs repetitive annotation tasks to automatically annotate an object. This study's dataset includes oil palm FFB and a variety of fruits, with a vast number of data. The YOLOv5 model, a deep learning approach, is chosen for automatically annotating images using the repetitive annotation task technique. The developed method was tested on a large dataset to determine its performance and accuracy in the annotation. The findings reveal that the trained network can correctly classify an object in an image. Furthermore, to demonstrate the superiority of the suggested technique, two alternative YOLO versions, YOLOv3 and YOLOv4, were trained and evaluated on the same dataset, and their results were compared to those obtained by the proposed approach. The comparative results demonstrated the proposed method's efficacy and superiority for the task of fruit categorization. In addition, the repetitive annotation task method is able to increase efficiency in automatically annotating an object in an image. The accuracy for the last training dataset achieves 98.7% for oil palm fruit and 99.5% for a variety of fruit. Therefore, the design of this method is proven fast in annotating a new image and successfully achieves high accuracy. Additionally, this automated method can greatly reduce the amount of time required to classify fruit, while also addressing the difficulty caused by a massive number of unlabeled images. Other than YOLO, the proposed repetitive annotation task technique is recommended to be deployed in any deep learning technique as the advancement of deep learning evolves.

Author Contributions: Writing—original draft preparation, N.M.; supervision and methodology, M.F.O.; writing—review and editing, R.A., A.A.A. and Y.G.; funding acquisition, R.A. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to acknowledge Universiti Teknologi Malaysia (research grant Professional Development Research University: Q.K130000.21A2.05E38) for financially supporting this research. Furthermore, the authors would like to thank the Research Management Center, Malaysia International Islamic University for funding this work by Grant RMCG20-023-0023. The publication of this work was supported by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia, under Project GRANT2,263.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data are not publicly available due to privacy restrictions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Radenović, F.; Tolias, G.; Chum, O. Fine-tuning CNN image retrieval with no human annotation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 1655–1668. [CrossRef] [PubMed]
2. Pliner, H.A.; Shendure, J.; Trapnell, C. Supervised classification enables rapid annotation of cell atlases. *Nat. Methods* **2019**, *16*, 983–986. [CrossRef] [PubMed]
3. Alay, N.; Al-Baity, H.H. Deep Learning Approach for Multimodal Biometric Recognition System Based on Fusion of Iris, Face, and Finger Vein Traits. *Sensors* **2020**, *20*, 5523. [CrossRef] [PubMed]
4. Yin, S.; Bi, J. Medical image annotation based on deep transfer learning. In Proceedings of the 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Halifax, NS, Canada, 30 July–3 August 2018; pp. 47–49.
5. Yang, H.; Gao, H. Toward sustainable virtualized healthcare: Extracting medical entities from Chinese online health consultations using deep neural networks. *Sustainability* **2018**, *10*, 3292. [CrossRef]
6. Uricchio, T.; Ballan, L.; Seidenari, L.; Del Bimbo, A. Automatic image annotation via label transfer in the semantic space. *Pattern Recognit.* **2017**, *71*, 144–157. [CrossRef]
7. Albarrak, K.; Gulzar, Y.; Hamid, Y.; Mehmood, A.; Soomro, A.B. A Deep Learning-Based Model for Date Fruit Classification. *Sustainability* **2022**, *14*, 6339. [CrossRef]
8. Alfatni, M.S.M.; Shariff, A.R.M.; Shafri, H.Z.M.; Saaed, O.M.B.; Eshanta, O.M. Oil palm fruit bunch grading system using red, green and blue digital number. *J. Appl. Sci.* **2008**, *8*, 1444–1452. [CrossRef]
9. Board, M.P.O. *Oil Plam Fruit Grading Manual*; Lembaga Minyak Sawit Malaysia (MPOB): Kajan, Malaysia, 2003.
10. Elwirehardja, G.N.; Prayoga, J.S. Oil palm fresh fruit bunch ripeness classification on mobile devices using deep learning approaches. *Comput. Electron. Agric.* **2021**, *188*, 106359.
11. Gulzar, Y.; Hamid, Y.; Soomro, A.B.; Alwan, A.A.; Journaux, L. A convolution neural network-based seed classification system. *Symmetry* **2020**, *12*, 2018. [CrossRef]
12. Hamid, Y.; Wani, S.; Soomro, A.B.; Alwan, A.A.; Gulzar, Y. Smart Seed Classification System Based on MobileNetV2 Architecture. In Proceedings of the 2022 2nd International Conference on Computing and Information Technology (ICCIT), Tabuk, Saudi Arabia, 25–27 January 2022; pp. 217–222.
13. Shabdin, M.K.; Shariff, A.R.M.; Johari, M.N.A.; Saat, N.K.; Abbas, Z. A study on the oil palm fresh fruit bunch (FFB) ripeness detection by using Hue, Saturation and Intensity (HSI) approach. In Proceedings of the IOP Conference Series: Earth and Environmental Science, Kuala Lumpur, Malaysia, 13–14 April 2016; Volume 37, p. 12039.
14. Adnan, M.M.; Rahim, M.S.M.; Rehman, A.; Mehmood, Z.; Saba, T.; Naqvi, R.A. Automatic image annotation based on deep learning models: A systematic review and future challenges. *IEEE Access* **2021**, *9*, 50253–50264. [CrossRef]
15. Jakhar, D.; Kaur, I. Artificial intelligence, machine learning and deep learning: Definitions and differences. *Clin. Exp. Dermatol.* **2020**, *45*, 131–132. [CrossRef] [PubMed]
16. Piam, E.H.; Mahmud, A.; Abdulghafor, R.; Wani, S.; Ibrahim, A.A.; Olowolayemo, A. Face Authentication-Based Online Voting System. *Int. J. Perceptive Cogn. Comput.* **2022**, *8*, 19–23.
17. Jung, H.; Rhee, J. Application of YOLO and ResNet in Heat Staking Process Inspection. *Sustainability* **2022**, *14*, 15892. [CrossRef]
18. Mujahid, A.; Awan, M.J.; Yasin, A.; Mohammed, M.A.; Damaševičius, R.; Maskeliūnas, R.; Abdulkareem, K.H. Real-Time Hand Gesture Recognition Based on Deep Learning YOLOv3 Model. *Appl. Sci.* **2021**, *11*, 4164. [CrossRef]
19. Nemade, S.B.; Sonavane, S.P. Co-occurrence patterns based fruit quality detection for hierarchical fruit image annotation. *J. King Saud Univ. Inf. Sci.* **2020**, *34*, 4592–4606. [CrossRef]
20. Samiei, S.; Rasti, P.; Richard, P.; Galopin, G.; Rousseau, D. Toward joint acquisition-annotation of images with egocentric devices for a lower-cost machine learning application to apple detection. *Sensors* **2020**, *20*, 4173. [CrossRef]

21. Mamat, N.; Othman, M.F.; Abdoulghafor, R.; Belhaouari, S.B.; Mamat, N.; Mohd Hussein, S.F. Advanced Technology in Agriculture Industry by Implementing Image Annotation Technique and Deep Learning Approach: A Review. *Agriculture* **2022**, *12*, 1033. [CrossRef]
22. Jamil, N.; Mohamed, A.; Abdullah, S. Automated grading of palm oil fresh fruit bunches (FFB) using neuro-fuzzy technique. In Proceedings of the 2009 International Conference of Soft Computing and Pattern Recognition, Malacca, Malysia, 4–7 December 2009; pp. 245–249.
23. Rahutomo, R.; Mahesworo, B.; Cenggoro, T.W.; Budiarto, A.; Suparyanto, T.; Atmaja, D.B.S.; Samoedro, B.; Pardamean, B. AI-based ripeness grading for oil palm fresh fruit bunch in smart crane grabber. In Proceedings of the IOP Conference Series: Earth and Environmental Science, Solo, Indonesia, 13–14 November 2020; Volume 426, p. 12147.
24. Khamis, N.; Selamat, H.; Ghazalli, S.; Saleh, N.I.M.; Yusoff, N. Comparison of Palm Oil Fresh Fruit Bunches (FFB) Ripeness Classification Technique using Deep Learning Method. In Proceedings of the 2022 13th Asian Control Conference (ASCC), Jeju Island, Korea, 4–7 May 2022; pp. 64–68.
25. Liu, S.; Jin, Y.; Ruan, Z.; Ma, Z.; Gao, R.; Su, Z. Real-Time Detection of Seedling Maize Weeds in Sustainable Agriculture. *Sustainability* **2022**, *14*, 15088. [CrossRef]
26. Janowski, A.; Kaźmierczak, R.; Kowalczyk, C.; Szulwic, J. Detecting Apples in the Wild: Potential for Harvest Quantity Estimation. *Sustainability* **2021**, *13*, 8054. [CrossRef]
27. Herman, H.; Cenggoro, T.W.; Susanto, A.; Pardamean, B. Deep Learning for Oil Palm Fruit Ripeness Classification with DenseNet. In Proceedings of the 2021 International Conference on Information Management and Technology (ICIMTech), Jakarta, Indonesia, 19–20 August 2021; Volume 1, pp. 116–119.
28. Kang, H.; Chen, C. Fast implementation of real-time fruit detection in apple orchards using deep learning. *Comput. Electron. Agric.* **2020**, *168*, 105108. [CrossRef]
29. Tang, Y.; Chen, M.; Wang, C.; Luo, L.; Li, J.; Lian, G.; Zou, X. Recognition and localization methods for vision-based fruit picking robots: A review. *Front. Plant Sci.* **2020**, *11*, 510. [CrossRef] [PubMed]
30. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
31. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
32. Jocher, G.; Nishimura, K.; Mineeva, T.; Vilariño, R. Yolov5. Code Repos. 2020. Available online: https//github.com/ultralytics/yolov5 (accessed on 1 October 2022).
33. Choiński, M.; Rogowski, M.; Tynecki, P.; Kuijper, D.P.J.; Churski, M.; Bubnicki, J.W. A first step towards automated species recognition from camera trap images of mammals using AI in a European temperate forest. In Proceedings of the International Conference on Computer Information Systems and Industrial Management, Elk, Poland, 17 September 2021; Springer: Cham, Switzerland, 2021; pp. 299–310.
34. Sozzi, M.; Cantalamessa, S.; Cogato, A.; Kayad, A.; Marinello, F. Automatic Bunch Detection in White Grape Varieties Using YOLOv3, YOLOv4, and YOLOv5 Deep Learning Algorithms. *Agronomy* **2022**, *12*, 319. [CrossRef]
35. Yao, J.; Qi, J.; Zhang, J.; Shao, H.; Yang, J.; Li, X. A real-time detection algorithm for Kiwifruit defects based on YOLOv5. *Electronics* **2021**, *10*, 1711. [CrossRef]
36. Wang, Z.; Jin, L.; Wang, S.; Xu, H. Apple stem/calyx real-time recognition using YOLO-v5 algorithm for fruit automatic loading system. *Postharvest Biol. Technol.* **2022**, *185*, 111808. [CrossRef]
37. Yan, B.; Fan, P.; Lei, X.; Liu, Z.; Yang, F. A real-time apple targets detection method for picking robot based on improved YOLOv5. *Remote Sens.* **2021**, *13*, 1619. [CrossRef]
38. Wang, H.; Shang, S.; Wang, D.; He, X.; Feng, K.; Zhu, H. Plant disease detection and classification method based on the optimized lightweight YOLOv5 model. *Agriculture* **2022**, *12*, 931. [CrossRef]
39. Chen, Z.; Wu, R.; Lin, Y.; Li, C.; Chen, S.; Yuan, Z.; Chen, S.; Zou, X. Plant disease recognition model based on improved YOLOv5. *Agronomy* **2022**, *12*, 365. [CrossRef]
40. Lyu, S.; Li, R.; Zhao, Y.; Li, Z.; Fan, R.; Liu, S. Green Citrus Detection and Counting in Orchards Based on YOLOv5-CS and AI Edge System. *Sensors* **2022**, *22*, 576. [CrossRef]
41. Kim, J.; Cho, J. A Set of Single YOLO Modalities to Detect Occluded Entities via Viewpoint Conversion. *Appl. Sci.* **2021**, *11*, 6016. [CrossRef]
42. Al-Qubaydhi, N.; Alenezi, A.; Alanazi, T.; Senyor, A.; Alanezi, N.; Alotaibi, B.; Alotaibi, M.; Razaque, A.; Abdelhamid, A.A.; Alotaibi, A. Detection of Unauthorized Unmanned Aerial Vehicles Using YOLOv5 and Transfer Learning. *Electronics* **2022**, *11*, 2669. [CrossRef]
43. Kwak, N.; Kim, D. A study on Detecting the Safety helmet wearing using YOLOv5-S model and transfer learning. *Int. J. Adv. Cult. Technol.* **2022**, *10*, 302–309.

44. Naushad, R.; Kaur, T.; Ghaderpour, E. Deep Transfer Learning for Land Use and Land Cover Classification: A Comparative Study. *Sensors* **2021**, *21*, 8083. [CrossRef] [PubMed]
45. Krishna, S.T.; Kalluri, H.K. Deep learning and transfer learning approaches for image classification. *Int. J. Recent Technol. Eng.* **2019**, *7*, 427–432.