Detection the Ripeness of Oil Palm Fresh Fruit Bunch Using Pretrained and Improved Models of YOLOv8

Saysunee Jumrat [1,2], Pattanapong Saeleung [1], Yutthapong Pianroj [1,2], Piyanart Chotikawanid [1], Teerasak Punvichai [2,3], Doungrat Chitcharoen [4], and Jirapond Muangprathub [1,2,*]

[1] Faculty of Science and Industrial Technology, Prince of Songkla University, Surat Thani, Thailand
[2] Integrated High-Value of Oleochemical (IHVO) Research Center, Surat Thani Campus, Prince of Songkla University, Surat Thani, Thailand
[3] Faculty of Innovative Agriculture, Fisheries and Food, Prince of Songkla University, Surat Thani Campus, Surat Thani, Thailand
[4] Faculty of Science, Chandrakasem Rajabhat University, Ratchadaphisek Road, Bangkok, Thailand
Email: saysunee.j@psu.ac.th (S.J.); 6540320304@email.psu.ac.th (P.S.); yutthapong.p@psu.ac.th (Y.P.); piyanart.ko@psu.ac.th (P.C.); teerasak.p@psu.ac.th (T.P.); doungrat.c@chandra.ac.th (D.C.); jirapond.m@psu.ac.th (J.M.)
*Corresponding author

***Abstract*—The ripeness of oil palm fresh fruit bunches significantly impacts the quality and economic value of palm oil production. This study proposes a novel approach using deep learning, particularly You Only Look Once (YOLO)v8, to classify and detect the ripeness of oil palm fruit bunches. By leveraging pretrained models and fine-tuning techniques, this study aims to improve detection accuracy while reducing inference time. The methodology involves data preparation, model training, and evaluation, utilizing a dataset comprising 427 images of oil palm fruit bunches classified into raw, half-ripe, and ripe categories. Additionally, the images were augmented to increase the dataset size. The YOLOv8 architecture, known for its scalability and efficiency, is applied to improve the classification process. Results demonstrate that YOLOv8 provides a balance between accuracy and processing speed, making it a suitable tool for real-time applications in the palm oil industry. This study contributes to reducing the reliance on manual techniques, lowering operational costs, and increasing the overall efficiency of the oil palm industry.**

***Keywords*—deep learning, You Only Look Once (YOLO)v8, object detection, oil palm fresh fruit bunch, computer vision**

I. INTRODUCTION

One of the most influential crops in Southeast Asia is oil palm, particularly in countries such as Indonesia, Malaysia, and Thailand [1–3]. Also, the main economic crop in Thailand is oil palm. Palm oil production in Thailand is estimated at 11–13 million tons per year. Therefore, the oil palm industry holds significant importance in Thailand. In the southern region of Thailand, the majority of farmers focus on palm oil growing. Typically, the percentage of oil palm per bunch is 16–18%, indicating good quality. However, some smallholder harvest oil palm early, resulting in low quality and reduced percentage of oil palm around 14–15%. This variation in harvesting practices leads to varying palm quality sold to small and large industries, ultimately affecting the overall oil palm quality [4].

In the context of classification, oil palm was measured by ripeness, which normally consists of raw, half and ripe [5]. As a result, small and large industries expend significant costs and resources to classify the ripeness of oil palm, such as hiring professionals and investing time. Some industries lack the technology to immediately assess palm quality, increasing the risk of processing lower quality palm. Traditional harvesters employ manual techniques like observing color and counting the number of fallen fruits from palm trees to classify the quality of the palm [6–8]. While professionals in industries use their eyes to observe the palm quality and classify the ripeness level of oil palm by looking through the pouring yard. Additionally, each time customers sell palm bunches, professionals find that most customers ensure stable palm quality. Therefore, applying information technology to assist in observation tasks within industries reduces the time and resources needed to classify oil palm, predicts ripeness using artificial intelligence, particularly deep learning methods. Deep learning method encompasses various tasks including object detection, image segmentation, image recognition, feature detection, and more [9]. Furthermore, deep learning can aid the palm industry by decreasing the time required for object evaluation, lowering professional hiring costs, and minimizing human bias in tasks. Therefore, applying deep learning methods to classify ripeness and detect oil palm can enhance accuracy, speed up processing time, and reduce human hiring costs. According to previous research

Manuscript received November 13, 2025; revised December 31, 2025; accepted February 11, 2026; published June 10, 2026.

doi: 10.12720/jait.17.6.1028-1040

various deep learning techniques have been applied to detect ripeness of oil palm [10] such as EfficientNetB0, NASNet Mobile, MobileNetV1, MobileNetV2 for classifying ripeness of oil palm [11]. ResNet50 and You Only Look Once (YOLO)v3 deep learning techniques were utilized to detect the maturity of oil palm, and both models were compared [12]. Similarly, Junos *et al.* [13] applied YOLOv2 and YOLOv3 to find the most suitable models.

Previous research has demonstrated that applying deep learning methods, such as YOLO, achieves high accuracy in classifying the ripeness of oil palm [14] and is suitable for real-time object detection [15]. In contrast, while Convolutional Neural Network (CNN) is not designed specifically for real time object detection, they perform exceptionally in high-accuracy tasks like image classification and feature extraction. Presently, deep learning methods have a lot of techniques to enhance accuracy for detecting objects, such as the transfer learning technique, where a pretrained model is used as the starting point for a new task [16]. CNN modules typically operate as two-stage detection models. In the first stage, they propose potential regions in the input image likely to contain objects of interest. In the second stage, they focus on each proposed region individually to classify the objects within and refine their localization. In contrast, YOLO functions as a single-stage detection model, predicting bounding boxes and class probabilities directly from the entire image in a single pass without explicit region proposal. In this research, a deep learning method and transfer learning technique were applied, leveraging the latest YOLOv8 architecture, to enhance capabilities in classifying and detecting the ripeness of oil palm.

The aim of this research will be focusing on training the YOLOv8 architecture model with pretrained models that's provided by Ultralystic and fine-tuning the model for enhance accuracy and reduce detection time. The model was trained using pretrained models, compared, and the best performing one was selected to enhance accuracy and reduce detection time through fine-tuning by freezing blocks of network layers. This research is structured as follows: Section II presents the related work. Section III outlines the research methodology, focusing on three main processes: data preparation, data training and fine tune model, and evaluation models. Section IV presents the results and evaluation of the models. Finally, Section V presents the conclusion.

To address this problem, this study proposes a systematic framework for oil palm fresh fruit bunch ripeness detection using pre-trained and fine-tuned YOLOv8 models. First, a dataset of oil palm fruit bunch images was collected, annotated, and augmented to improve data diversity. Next, multiple YOLOv8 pretrained variants (*n*, *s*, *m*, *l*, and *x*) were evaluated to identify the most suitable backbone for this task. The best-performing model was then further optimized through a layer-freezing fine-tuning strategy to balance detection accuracy and inference speed. Finally, the models were evaluated using standard object detection metrics to assess their effectiveness and practical applicability.

Unlike previous studies that primarily focus on applying or modifying YOLO architectures, this work systematically investigates the effect of layer-freezing strategies in YOLOv8 for agricultural object detection. The main contributions of this study are threefold: (1) a focused three-class ripeness detection problem aligned with industrial grading practices, (2) a detailed evaluation of freezing different backbone depths to balance accuracy and efficiency, and (3) an experimental analysis demonstrating that selective freezing can significantly reduce inference time without sacrificing detection performance.

II. Related Work

Over the years, object detection has involved the identification and localization of objects within an image. Various methodologies have been developed to improve the efficiency and accuracy of object detection algorithms. These methods can be broadly categorized into two-stage and single-stage object detection approaches. Two-stage object detection is a traditional technique for identifying objects in images. Presently, single-stage object detection has become increasingly popular in recent years due to its faster inference times and competitive accuracy. Redmon *et al.* [17] proposed YOLO, a novel object detection technique that repurposes classifiers for detection. YOLO utilizes features from the entire image to predict each bounding box and can also predict classes for the detected objects. The result is a fast method, capable of running at 45 frames per second, making it suitable for real-time applications with improved accuracy. YOLOv2 proposed Darknet-19 as the backbone network, achieving more accuracy and handling more than 9,000 object categories [18]. YOLOv3 introduced residual blocks and multi-scale predictions, utilizing Darknet-53 as the backbone network, which improved accuracy while maintaining speed [19]. Bochkovskiy *et al.* [20] proposed YOLOv4 incorporated CSPDarknet53, Mish activation, PANet, and SPP blocks, along with advanced augmentation techniques, resulting in improvements in both speed and accuracy. YOLOv5 used CSPDarknet53 for the backbone and combined FPN and PAN for upsampling in the neck network, continuing the balance of speed and accuracy [21]. YOLOv6 introduced EfficientRep (RepVGG blocks) in the backbone and an efficient decoupled head for improved efficiency [22]. Wang *et al.* [23] proposed YOLOv7 presented Extended Efficient Layer Aggregation Networks (E-ELAN) for model scaling methods in real-time detection to improve feature extraction. While Dillon *et al*. [24] mentioned that YOLOv8 was trained on a blend of the COCO dataset and several other datasets, YOLOv5 was primarily trained on the COCO dataset. YOLOv8, similar to YOLOv5, uses the methods of FPN and PAN for its neck. For the head, YOLOv8 employs a decoupled head similar to YOLOv6. Additionally, YOLOv8 utilizes Soft-Non-Maximum Suppression (NMS), a variant of the NMS technique used in YOLOv5, which applies a soft threshold to overlapping bounding boxes instead of discarding them. The comparison of different YOLO versions is shown in Table I.

TABLE I. COMPARISON YOLO IN DIFFERENT VERSIONS

YOLO Version	Advantage
YOLOv1	Single stage object detection to predict bounding boxes and class probabilities.
YOLOv2	Introduced multi-scale training, enhance speed and accuracy.
YOLOv3	Complex backbone includes residual blocks for better feature extraction.
YOLOv4	Adding various data augmentation techniques.
YOLOv5	CBS is utilized for feature extraction, also FPN and PAN is utilized to upsampling the output feature map.
YOLOv6	EfficientRep architecture with RepVGG blocks, decoupled head, etc.
YOLOv7	Introduced E-ELAN for improved feature extraction and model scaling method.
YOLOv8	Scalability in various model sizes (*n*, *s*, *m*, *l*, *x*) catering to different performance needs.

YOLO has diverse applications in agriculture. Li *et al.* [25] proposed YOLOv7-CS, a lightweight model optimized for bayberry target detection, which involves three key modifications to the YOLOv7 architecture: replacing ELAN with CNxP modules and incorporating SPPFCSPC in the backbone network to enhance efficiency and accuracy, using RepGFPN for multiscale feature fusion and GAM for enriched output details, and implementing the SPD module for better feature extraction along with Wise-IoU to improve the detection of overlapping objects, resulting in improved accuracy while reducing model parameters and computational requirements.

Lai *et al.* [26] proposed a YOLOv4-based real-time detection system for ripe oil palm fresh fruit bunches. This system uses an RGB camera (Intel Realsense D435) to capture images of oil palm trees, which are processed by a single board computer (Nvidia Jetson NX) running a YOLOv4 inference model. The trained model identifies the target object, records its coordinates, and sends positional information to the robotic harvesting mechanism within the Robot Operating System (ROS). The model achieved a mean Average Precision (mAP) of 87.9% and a speed of 21 frames per second. However, the training dataset requires a large number of iterations, and the system is limited to detecting only ripe oil palm fresh fruit bunches, not other maturity stages, such as raw or half-ripe oil palms. Thereby, Suharjito *et al.* [27] used the YOLOv4 model to detect the ripeness of oil palm fruit bunches, classifying them into categories such as unripe, underripe, ripe, overripe, empty bunch, and abnormal fresh fruit bunch. They modified hyperparameters for training data, which resulted in effective detection of oil palm fruit bunches. However, the paper does not mention augmenting the dataset to increase the number of images in the oil palm fruit dataset.

Lai *et al.* [28] proposed an improved YOLOv7 model for field environments, adapting the original YOLOv7 architecture with several modifications. First, SimAM is integrated into the backbone network to enhance feature extraction capabilities and minimize complex background interference without increasing the number of network parameters. SimAM can be embedded at any position within the model. Additionally, the MPConv structure is improved to reduce feature loss during the downsampling process. The soft NMS algorithm is also enhanced to improve detection accuracy when pineapples are occluded or overlapping. These improvements significantly boost the detection accuracy and speed of the YOLOv7 model in complex field environments for different maturity stages of pineapples. Similarly, Gu *et al.* [29] improved the YOLOv7-Tiny version for citrus detection in complex environments. They replaced ELAN with ELAN-DW in the backbone network to reduce the number of model parameters and used CA to replace CBL in the neck network, enhancing the model's feature extraction ability. Additionally, a Dynamic Head was integrated into the head network to improve the model's capability to detect citrus fruits at different scales by fusing multi-layer features. This improved model, named YOLO-DCA, outperforms the original YOLOv7-Tiny in terms of lightweight design and can be deployed on embedded devices with limited computational resources, accurately detecting citrus fruits in complex environments and on various devices.

Using YOLOv8 demonstrated improved accuracy for classifying and detecting oil palm fresh fruit bunches while also reducing inference times. Gunawan *et al.* [30] proposed a YOLOv8 model trained with a pretrained model using a dataset of oil palm fresh fruit bunches at six validated maturity levels: raw, underripe, ripe, overripe, empty bunch, and abnormal. The dataset comprised 7,171 images and 14,559 objects. The authors utilized high-performance hardware to ensure optimal analysis and execution, with a batch size fixed at 16 and a learning rate fixed at 0.01 during training. The results of this research showed that YOLOv8m achieved a good balance between accuracy and training time, taking 0.899 h for training and achieving a mAP50–95 of 0.927. Meanwhile, YOLOv8x attained a slightly higher mAP50–95 of 0.921 but required a significantly longer training time of 9.019 h. Additionally, Naftali and Hugo [3] presented various YOLO models to evaluate performance using a dataset of oil palm fresh fruit bunches captured in plantations, including five maturity levels: abnormal, ripe, underripe, unripe, and flower. The challenges in this research included partially visible objects, low-contrast scenes, occluded and small objects, and blurry images. This study proposed YOLOv8s Depth-wise and compared it with YOLOv6s, YOLOv61, YOLOv7 Tiny, YOLOv7l, YOLOv8s, and YOLOv8l. Data augmentation was used to enhance model performance. The results demonstrated that YOLOv8s Depthwise achieved a balanced performance with fast inference (0.027 s), a mAP of 0.75, and a model size of 10.6 MB, with a training time of 2 h, 18 min, and 30 s. The data augmentation positively impacted model performance, making YOLOv8s Depth-wise ideal for real-time palm oil harvesting applications.

Numerous research in the field of agriculture modified the YOLO architecture to improved accuracy and increase the speed of inference time. This research utilizes the benefits of YOLOv8 for improved accuracy and fine-tuning technique such as frozen blocks on the network layers for reduce training time and reduce model size to classify the maturity of oil palm fresh fruit bunches. While recent studies have applied YOLOv8 to oil palm detection, they mainly emphasize architectural variants or extensive datasets. In contrast, this study addresses a practical deployment challenge by optimizing model efficiency through systematic fine-tuning and layer freezing, making the approach suitable for real-world industrial environments with limited computational resources.

Recent studies have explored architectural enhancements to YOLOv8 for agricultural applications, such as improved feature fusion and lightweight designs (e.g., LEF-YOLO). While these approaches demonstrate performance gains through architectural modification, they often increase model complexity [31–33]. In contrast, this study focuses on training-level optimization without altering the base architecture, offering an alternative perspective on improving efficiency in practical deployment scenarios.

III. RESEARCH METHODOLOGY

The research methodology, illustrated in Fig. 1, is divided into three main steps: data preparation, data training and model fine-tuning, and evaluation of all model results. Each stage is designed to systematically refine the detection model and ensure fair comparison across different YOLOv8 variants. These steps are detailed below.

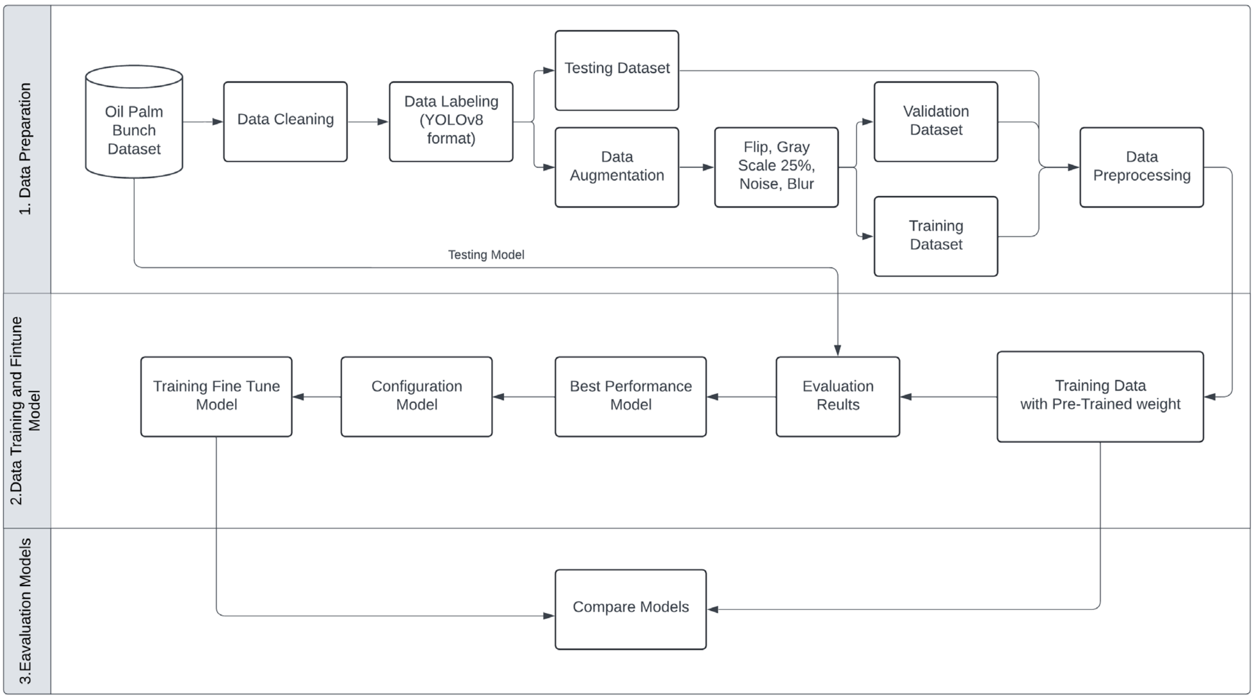


Fig. 1. Research framework.

A. Data Preparation

In this subsection, the first step is preparing the data before training with the YOLOv8 algorithm. The dataset, collected by Thaksin Palm Company, contains images of oil palm fruit bunches at various stages of ripeness. The dataset was cleaned before labeling, resulting in a total of 427 cleaned images. The dataset consists of three ripeness classes: raw, half-ripe, and ripe. As summarized in Table II, the class distribution includes 146 images of raw fruit bunches, 127 images of half-ripe fruit bunches, and 154 images of ripe fruit bunches. Although the dataset is not perfectly balanced, the class proportions are comparable, allowing the models to learn representative features for each ripeness stage.

Once the data cleaning is completed, the next step in data preparation is labeling the images. This research utilizes Roboflow software to label the images. Each image in the dataset is annotated with a class of oil palm fruit bunch, raw, half, or ripe, using bounding boxes to define the class of the image. This process is time consuming to ensure that the dataset does not contain any garbage data for the training process. After the annotation process is finished, the dataset is exported into YOLOv8 format files, including image and label (.txt) folders. Files in the label folder consist of [class id, x, y, w, h], where x and y are the coordinates of the box, w is the width, and h is the height. The filenames in the image and label folders are the same, ensuring that each image corresponds to its respective class along with its bounding box.

TABLE II. DATASET OF OIL PALM FRUIT BUNCH

Class	Number of Images
Raw	146
Half	127
Ripe	154
Total	427

The annotation process was performed using the Roboflow platform. Each image was manually annotated using bounding boxes to label oil palm fruit bunches as raw, half-ripe, or ripe. The labeling was conducted by trained personnel familiar with oil palm grading practices and subsequently reviewed to ensure consistency and accuracy. This manual verification step helped minimize labeling errors and improve dataset quality.

To ensure a balanced evaluation of model performance while maintaining sufficient training data, the dataset was divided into training, validation, and testing sets using a 7:2:1 ratio. This split was selected to provide enough samples for effective model learning while preserving an independent test set for unbiased performance assessment. Data augmentation techniques, including horizontal flipping, grayscale transformation, Gaussian noise, and blur, were applied exclusively to the training set to increase data diversity and improve generalization. These augmentation strategies were chosen to simulate real-world variations commonly encountered in industrial environments, such as lighting changes, motion blur, and sensor noise, thereby enhancing the robustness of the trained models under practical operating conditions. The total number of dataset images after applying the augmentation techniques consists of 891 images in the training set, 84 images in the validation set, and 46 images in the testing set, as shown in Table III. With these steps completed, the data preprocessing is finished, and the dataset is ready for training.

TABLE III. DATASET AFTER AUGMENTATION OF OIL PALM FRUIT BUNCH

Type	Number of Images
Training Set	891
Validation Set	84
Testing Set	46
Total	1,021

Data augmentation was applied exclusively to the training set to improve model generalization. Horizontal flipping was applied with a probability of 0.5. A grayscale transformation was applied to 25% of the training images to simulate illumination variations commonly observed in industrial environments. Gaussian noise was added with a mean of 0 and a variance of 0.01 to model sensor noise. Gaussian blur was applied using a kernel size of 5×5 with a probability of 0.3 to simulate motion blur and defocus effects. These parameters were empirically selected to reflect realistic variations while preserving discriminative visual features.

All images were collected at Thaksin Palm Company under real industrial conditions. The images were captured using a standard RGB camera at varying distances and angles to reflect practical inspection scenarios. Data acquisition was conducted under natural lighting conditions, including variations in illumination and background clutter, in order to represent realistic operating environments in palm oil processing facilities. These dataset characteristics ensure that the proposed model is evaluated under realistic conditions, supporting its applicability to real-world oil palm ripeness assessment tasks.

B. *Data Training and Fine-Tuning Model*

The next step is training the data and fine-tuning the model. In this subsection, the dataset is trained using the YOLOv8 algorithm with pretrained weights. YOLOv8 was design for object detection, classification, segmentation, and tracking tasks, Additional YOLOv8 offers various pretrained models, such as YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x, which have been trained on the COCO dataset containing 80 classes. The size of the parameters for these models is shown in Table IV.

TABLE IV. YOLOV8 PRE-TRAINED MODELS INCLUDE 80 PRE-TRAINED CLASSES TRAINED ON COCO DATASET

Model	Size (pixels)	mAP50–95	Params (m)	FLOPs (b)
YOLOv8n (nano)	640	37.3	3.2	8.7
YOLOv8s (small)	640	44.9	11.2	28.6
YOLOv8m (medium)	640	50.2	25.9	78.9
YOLOv8l (large)	640	52.9	43.7	165.2
YOLOv8x (xlarge)	640	53.9	68.2	257.8

Rather than modifying the original YOLOv8 architecture, this study adopts a fine-tuning strategy based on selective layer freezing. This approach was chosen to preserve general visual features learned from large-scale datasets while adapting higher-level representations to the specific task of oil palm ripeness classification. Layer freezing also reduces the number of trainable parameters, leading to faster convergence and lower computational cost, which is advantageous when working with limited training data.

All available YOLOv8 pretrained variants (*n*, *s*, *m*, *l*, and *x*) were employed to train the oil palm fresh fruit bunch ripeness dataset described in the previous subsection. The use of pretrained models enables the reuse of robust visual features learned from large-scale datasets, thereby improving detection performance and enhancing generalization to unseen data. Each model was trained for 100 epochs using the Adam optimizer with an initial learning rate of 0.001 and a cosine learning rate decay schedule. A uniform batch size of 16 was applied across all experiments to ensure a fair comparison among different YOLOv8 variants. Early stopping was not employed, as all models exhibited stable convergence within the predefined training duration.

After training with various pretrained models, the evaluation results were used to select the best performing model for fine-tuning. The evaluation metrics in this research included precision, recall, F1-Score, mAP50, and mAP50–95. In the next step, the model configuration involves using the technique of freezing blocks of network layers for fine-tuning. Freezing network layer blocks is beneficial for improving model performance. By freezing the initial layers of a pre-trained model, the fundamental features learned are retained, while the unfrozen layers are trained to enhance performance. This technique not only improves performance but also reduces computational load and speeds up the training process. In this research, the network layers are frozen in 10, 13, 16, and 19 blocks, as shown in Fig. 2.

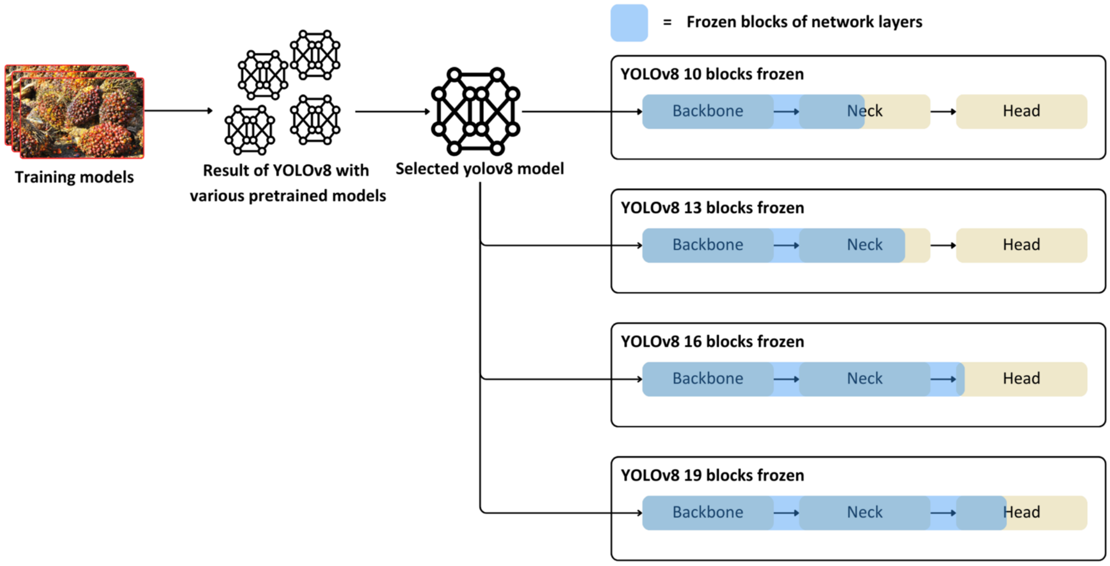


Fig. 2. YOLOv8 structure for blocks of frozen network layers.

The improvement strategy is based on freezing different numbers of backbone blocks in the pretrained YOLOv8x model (10, 13, 16, and 19 blocks), while allowing the remaining layers in the neck and detection head to be retrained. The frozen backbone layers preserve low-level and mid-level visual features learned from large-scale datasets, whereas the unfrozen layers adapt high-level semantic representations to the specific characteristics of oil palm fruit bunch ripeness.

Fine-tuning was performed using a layer-freezing strategy to balance model adaptability and computational efficiency. In deep convolutional networks, early layers typically capture low-level visual features such as edges and textures, while deeper layers encode high-level semantic representations. By freezing a selected number of backbone blocks, the pretrained feature representations learned from the large-scale COCO dataset are preserved, while the unfrozen layers are allowed to adapt to the specific characteristics of oil palm fruit bunch ripeness. In this study, four freezing configurations (10, 13, 16, and 19 blocks) were investigated to analyze the trade-off between detection accuracy, training stability, and inference speed. This systematic exploration enables a more rigorous evaluation of how different fine-tuning depths influence performance.

During the fine-tuning phase, the learning rate was reduced to 0.0001 to prevent large gradient updates from disrupting pretrained representations. The same optimizer and batch size were retained to isolate the effect of layer freezing. Freezing was applied only to the backbone network, while the neck and detection head remained fully trainable.

Finally, the results for each configuration corresponding to the freezing of 10, 13, 16, and 19 blocks will be obtained and compared with all the trained models.

C. Evaluation Models

In this step, the models trained with pretrained weights and fine-tuned models were compared using various metric values such as precision, recall, F1-Score, mAP50, and mAP50–95 to evaluate their performance. The effectiveness of the models in detecting the ripeness of oil palm bunches was measured by Recall, while the accuracy of the models was measured by Precision. The F1-Score, which is the harmonic mean of Precision and Recall, was also used for evaluation. Additionally, mAP50 and mAP50–95 were used to evaluate the mean average precision at an IoU threshold of 50% and at IoU thresholds from 50% to 95%. Furthermore, the detection time for each model was considered to compare the speed of the models. The equations for precision, recall, and F1-Score are shown in Eqs. (1)–(3):

$$Precision = \frac{TP}{FP + TP} \tag{1}$$

$$Recall = \frac{TP}{FN + TP} \tag{2}$$

$$F1 - Score = \frac{Precision \times Recall}{Precision + Recall} \tag{3}$$

where True Positive (TP) denotes correctly detected objects, False Positive (FP) represents incorrectly detected objects, and False Negative (FN) indicates the objects that were not detected by the model.

IV. Results and Evaluation Models

A. Training and Testing Results

It should be noted that this study does not introduce structural modifications to the original YOLOv8 architecture. Instead, the proposed improvement focuses on an enhanced training strategy using selective layer freezing applied to pretrained YOLOv8 models. The backbone, neck, and detection head architectures remain unchanged, ensuring compatibility with the original YOLOv8 design.

In this study, which focused on training and improving models for a specific task, YOLOv8 was utilized to detect the ripeness of palm fruit bunches. Each model was trained for 100 epochs. A pretrained YOLOv8 model was employed to enhance the data training process and enable faster adaptation to detect the ripeness of oil palm fruit bunch. Upon identifying the model with the best performance, transfer learning was applied using the freeze network layer method. The training outcomes were evaluated using the following metrics: precision, recall, F1-Score, mAP@0.50, mAP@0.50–0.95, and detection time (ms). The training was conducted on a computer with the following specifications: 12th Gen Intel(R) Core (TM) i5-12400 2.50 GHz CPU, 32 GB RAM, 1 TB DISK, and an Nvidia GeForce GTX 3070 TI GPU, as shown on Table V. The results of the training are presented in Table VI.

TABLE V. SPECIFICATION OF COMPUTER DEVICE USED

Device Specification	Details
CPU	12th Gen Intel(R) Core (TM) i5-12400 2.50 GHz
RAM	32 GB
DISK	1 TB
GPU	Nvidia GeForce GTX 3070TI

TABLE VI. COMPARISON OF ALL METRIC VALUES MODELS

Model	Precision	Recall	F1-Score	mAP@0.50	mAP@0.50–0.95	Detection Time (ms)	Size (MB)
YOLOv8n	0.827	0.93	0.876	0.956	0.709	2.9	5.96
YOLOv8s	0.924	0.927	0.926	0.975	0.756	10.5	22.5
YOLOv8m	0.917	0.918	0.918	0.959	0.75	7.2	49.64
YOLOv8l	0.922	0.952	0.937	0.974	0.763	20.6	83.6
YOLOv8x	0.918	0.964	0.94	0.975	0.761	29.3	136.7
YOLOv8x10	0.956	0.964	0.96	0.989	0.763	16.3	130.39
YOLOv8x13	0.92	0.931	0.926	0.975	0.755	16.7	130.39
YOLOv8x16	0.94	0.909	0.924	0.969	0.757	15.3	130.39
YOLOv8x19	0.966	0.968	0.967	0.988	0.773	16.2	130.39

To facilitate analysis, a graphical presentation is used. The graph illustrates the training results of both the pretrained model and the fine-tuned model. Fig. 3 shows the metric value analysis for the pretrained model, starting with the precision value. The lowest precision value is 0.83 for the YOLOv8n model, while the highest is 0.92 for every pretrained model. Recall values increased from 0.92 for YOLOv8m to 0.96 for YOLOv8x. The highest mAP@0.50 and mAP@0.50:0.95 values are found in the YOLOv8x, YOLOv8l, and YOLOv8s models, with values of 0.99 and 0.76, respectively. Comparing the F1-Score values, the most efficient pretrained model is YOLOv8x.

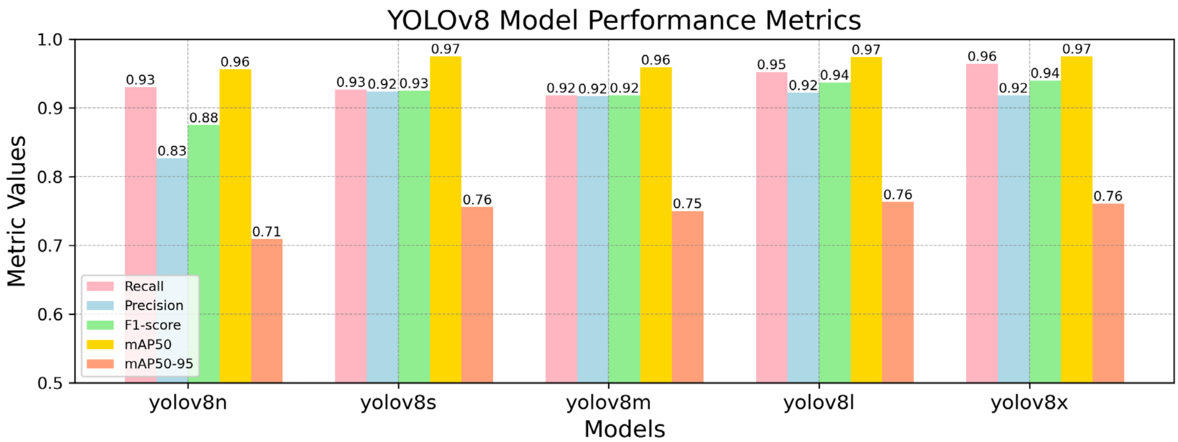


Fig. 3. Metric value comparison with pretrained models.

Based on the comparative results of the pretrained models, YOLOv8x was selected for further fine-tuning due to its superior balance between detection accuracy and robustness across all evaluation metrics. Although lighter models achieved faster inference times, YOLOv8x demonstrated consistently higher mAP and recall values, making it a suitable candidate for optimization through layer freezing to improve efficiency without sacrificing accuracy.

Although YOLOv8x achieved the highest recall and F1-Score among all pretrained models, its inference time (29.3 ms) is considerably higher than lighter variants such as YOLOv8n (2.9 ms). This highlights a practical trade-off between detection accuracy and computational efficiency. While YOLOv8n is suitable for ultra-low-latency applications, its lower accuracy may limit reliability in quality-sensitive industrial contexts. Therefore, YOLOv8x was selected as the base model for fine-tuning, with the objective of reducing inference time while preserving high detection performance.

In addition to inference time measured in milliseconds, Frames Per Second (FPS) were calculated to better reflect real-time performance. The proposed YOLOv8x19 model achieved an average inference time of 16.2 ms per image, corresponding to approximately 61.7 FPS. This performance satisfies real-time processing requirements for industrial inspection systems.

The experimental results indicate that the YOLOv8x19 model consistently outperforms both the fully fine-tuned YOLOv8x model and other pretrained variants across most evaluation metrics. This improvement can be attributed to the preservation of high-level semantic features through deeper layer freezing, which reduces overfitting while maintaining strong discriminative capability for ripeness classification. Compared to the original YOLOv8x model, YOLOv8x19 achieves comparable or higher mAP values while significantly reducing detection time, demonstrating a more efficient balance between accuracy and computational cost. These findings suggest that selective layer freezing is particularly effective for domain-specific agricultural tasks with limited datasets.

Model complexity was also considered to evaluate deployment feasibility. The YOLOv8x19 model has a model size of 136.7 MB, which is smaller than the fully trainable YOLOv8x model while maintaining superior detection performance. This reduction in model complexity is a direct result of the selective layer freezing strategy.

Fig. 4 presents the metric value analysis for the fine-tuned models, indicating that all the highest metric values are in the YOLOv8x19 model. Fig. 5 illustrates the detection times (ms) of various YOLOv8 models. YOLOv8n achieved the lowest detection time at 2.9 ms, while YOLOv8x achieved the highest detection time at 29.3 ms. The best performance model for detection and classify ripeness of oil palm bunch is YOLOv8x19 and this model demonstrated a reduction in detection time was reduce 13.1 ms when compared with the YOLOv8x model, additionally increasing accuracy from 0.97 to 0.99 on the mAP@0.50 metric. In practical terms, YOLOv8x19 combines high accuracy with a moderate detection speed per image or video on a typical system, making it suitable for practical applications.

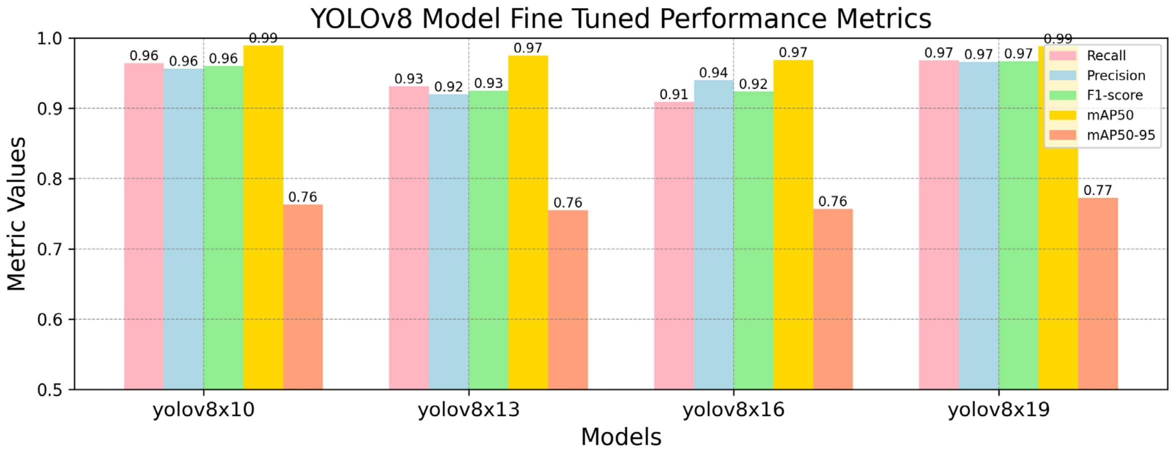


Fig. 4. Metric value comparison with fine-tuned models.

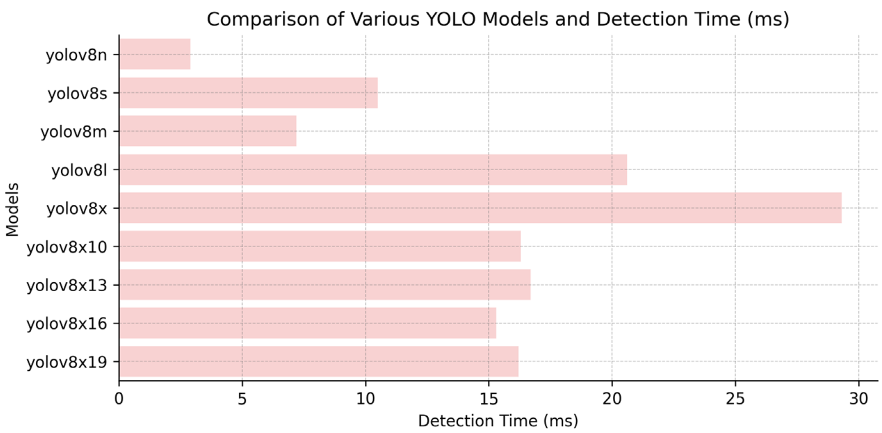


Fig. 5. Models' comparison with detection time (ms).

Freezing 19 backbone blocks yielded the best overall performance because these layers primarily capture high-level semantic features learned from large-scale datasets such as COCO. By preserving these representations and allowing only the deeper layers in the neck and detection head to adapt, the model avoids overfitting while maintaining strong discriminative capability. In contrast, freezing fewer blocks allows excessive parameter updates, which may lead to instability when training with a relatively small dataset.

B. Evaluation Model

As a representation in Fig. 6 presents an extensive visualization of the training and validation metrics for the

YOLOv8x19 model across 100 epochs. It illustrates both the loss and performance metrics, which are critical for evaluating the efficiency and accuracy of the model during its training process.

The training and validation curves in Fig. 6 show a stable learning process for YOLOv8x19. Both training and validation losses decrease steadily and converge without significant divergence, indicating no severe overfitting. Furthermore, the validation mAP metrics plateau after approximately 70 epochs, suggesting that the model reaches convergence and that further training yields marginal improvements.

The observed performance improvement of the YOLOv8x19 model can be attributed to the balance achieved between feature preservation and task-specific adaptation. By freezing a larger portion of the backbone, the model retains robust high-level representations while reducing unnecessary parameter updates during training. This leads to improved generalization, reduced overfitting, and lower inference time compared to the fully trainable YOLOv8x model.

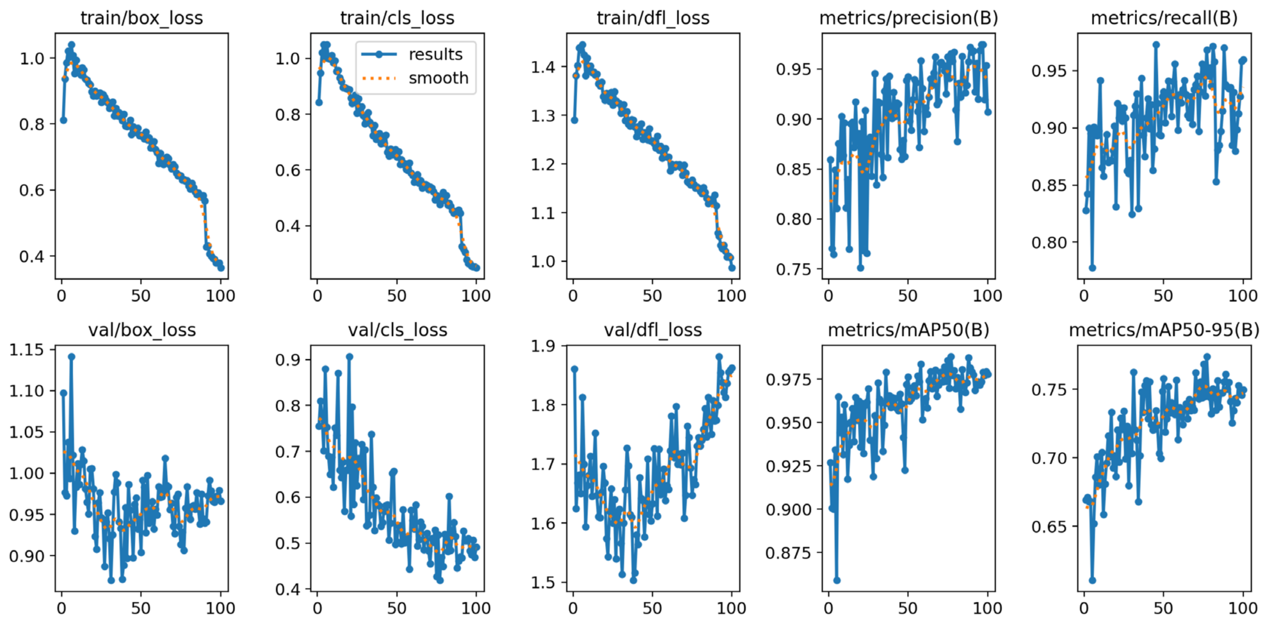


Fig. 6. Training and validation of YOLOv8x19.

The training loss metrics consist of three metrics: First, the box_loss represents the box regression loss during the training phase, which starts high and progressively decreases. Second, the cls_loss represents classification loss, which also decreases over the epochs, suggesting that the model is learning to classify objects within the bounding boxes more accurately. Lastly, the dfl_loss represents the distribution focal loss during training. A steady decrease in this loss is observed, indicating the model's enhanced capability to predict the correct focal distribution over the epochs. The training metrics consist of four key metrics: First, precision measures the accuracy of positive predictions. Second, recall measures the model's ability to capture all positive instances, with an increasing graph indicating improved object identification. Next, mAP50, or mean average precision at a 50% IoU threshold, provides a comprehensive measure of precision and recall across all classes, with an increasing trend signifying overall performance improvement. Lastly, mAP50–95 measures the same as mAP50 but across IoU thresholds from 50% to 95%. The validation loss includes three metrics similar to the training loss but measured on the validation set.

The training/validation curves for YOLOv8x19 (Fig. 6) show a stable learning process, with losses decreasing and metrics increasing consistently across epochs. The overall loss metrics (box, cls, dfl) display a general decreasing trend, indicating that the YOLOv8x19 model is learning effectively with each epoch. The initial high loss values decrease and adjust with every epoch. Focusing on the precision, recall, mAP50, and mAP50–95 trend lines, we observe an increase in all metrics, signifying the model's improved accuracy and ability to detect relevant objects. Overall, the YOLOv8x19 model has been enhanced through fine-tuning techniques to detect and classify the ripeness of oil palm fruit bunches, achieving a precision and recall of 0.97, an mAP50 of 0.99, and an mAP50–95 of 0.77.

To better observe the detection accuracy and performance of the model, Fig. 7 presents a confusion matrix for YOLOv8x19, with 19 blocks of the YOLOv8x model frozen. The x-axis represents the true classes, while the y-axis shows the classes predicted by the model. Focusing on the diagonal elements (top-left to bottom-right), we see high values for the predicted and true classes, indicating that this model accurately detects the true class ripeness of oil palm fruit bunches. The model shows high accuracy for "raw", "half", and "ripe" classes, with both "raw" and "ripe" achieving 100% correct classification. The "half" class also demonstrates high accuracy, achieving 89% correct classification, with only slight misclassifications.

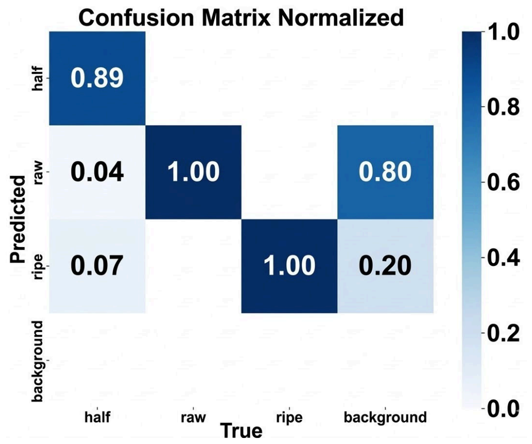
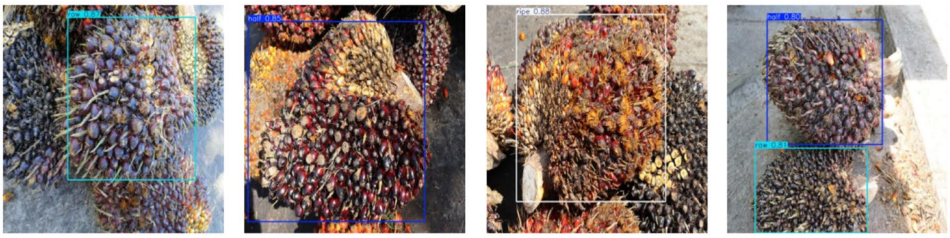


Fig. 7. Confusion matrix of YOLOv8x19.

From a practical perspective, the improved performance of YOLOv8x19 highlights its suitability for real-time deployment in palm oil processing facilities and plantation environments. The reduced inference time, combined with high precision and recall across all ripeness classes, enables rapid and reliable quality assessment without reliance on manual inspection. This capability can support operational decision-making, reduce labor costs, and improve consistency in quality control processes, thereby contributing to increased efficiency within the palm oil industry.

In Fig. 8, the detection ripeness of oil palm using the YOLOv8x19 model is illustrated. Image (a) displays a prediction for the class 'raw' with an accuracy of 87%, image (b) shows a prediction for the class "half" with an accuracy of 89%, image (c) presents a prediction for the class "ripe" with an accuracy of 88%, and image (d) presents multiple classes in class 'half' and 'raw' with an accuracy of 80% and 81% respectively. The predictions were made on a testing set of 45 images. The model achieved an inference time of 32.8 ms, with preprocessing taking 1.8 ms and postprocessing taking 1.3 ms, resulting in a total detection time of 35.9 ms, averaging 32 ms per image. Therefore, the term "improved YOLOv8" in this study refers to an optimized fine-tuning strategy rather than a modification of the original network architecture. These results indicate that the YOLOv8x19 model is highly effective at detecting the ripeness of oil palm fruits, demonstrating strong performance across different ripeness classes with high accuracy and efficient processing times. This makes the model suitable for real-time applications in the field of agriculture.



(a) (b) (c) (d)

Fig. 8. Oil palm detection on test set using YOLOv8x19 model. (a) Raw class (b) Half class (c) Ripe class (d) Multiple classes.

Despite the strong overall performance, several failure cases were observed. Most misclassifications occurred between the "half-ripe" and "ripe" classes, as indicated by the confusion matrix. This confusion can be attributed to visual similarities in color and texture between these intermediate ripeness stages, particularly under varying illumination conditions. In addition, partial occlusion of fruit bunches and background clutter occasionally led to incorrect predictions, especially when multiple bunches appeared in a single image.

From a technical perspective, these failure cases suggest that RGB-based visual features alone may be insufficient to fully discriminate subtle ripeness transitions. Furthermore, the limited dataset size restricts the model's exposure to rare or ambiguous visual patterns, which may reduce robustness when encountering challenging scenarios such as overlapping fruit bunches or non-uniform lighting.

Recent studies published in 2024–2025 have reported improved performance using YOLOv8 variants and lightweight adaptations for agricultural object detection. For example, Naftali and Hugo [3] emphasized deployment efficiency on embedded devices using depthwise convolutions, while Gunawan *et al.* [30] achieved high accuracy by training YOLOv8 models on large-scale datasets with multiple ripeness classes. Compared with these approaches, the proposed YOLOv8x19 model focuses on optimizing performance through training strategy rather than architectural modification.

The discrepancies in reported performance across recent studies can be attributed to several factors. First, dataset scale and class definition differ substantially, with many recent models trained on thousands of images and more ripeness categories, whereas this study focuses on a three-class industrial grading problem using a smaller but carefully curated dataset. Second, architectural modifications and lightweight designs often prioritize inference speed at the expense of accuracy, while the proposed approach maintains the original YOLOv8 architecture and improves efficiency through selective layer freezing. Third, differences in evaluation protocols and deployment targets further contribute to variations in reported metrics.

Table VII presents a comparative analysis between the proposed YOLOv8x19 model and representative YOLO-based approaches reported in recent literature for oil palm and agricultural object detection tasks. Previous studies predominantly focus on architectural modifications (e.g., YOLOv4- and YOLOv7-based variants) or rely on large-scale datasets to achieve high detection accuracy. For example, Gunawan *et al.* [30] employed YOLOv8 models trained on over 7,000 images and achieved competitive mAP values, while Naftali and Hugo [3] emphasized lightweight YOLOv8 variants for embedded deployment.

TABLE VII. PERFORMANCE COMPARISON WITH EXISTING YOLO-BASED APPROACHES FOR OIL PALM AND AGRICULTURAL OBJECT DETECTION

Study	Model	Target Application	No. of Classes	Dataset Size	mAP@0.50	Inference Time	Key Characteristics
Lai *et al.* [26]	YOLOv4	Oil palm FFB (ripe only)	1	–	0.879	21 FPS	Robotic harvesting system
Suharjito *et al.* [27]	YOLOv4	Oil palm FFB piles	6	–	–	–	Hyperparameter tuning
Gunawan *et al.* [30]	YOLOv8m	Oil palm FFB	6	7,171 images	0.927	–	Large dataset, no layer freezing
Naftali and Hugo [3]	YOLOv8s-DW	Oil palm plantation	5	–	0.75	27 ms	Lightweight, embedded focus
Li *et al.* [25]	YOLOv7-CS	Fruit detection	1	–	–	–	Architectural modification
This study	YOLOv8x19	Oil palm FFB ripeness	3	427 images (+ augmentation)	0.99	16.2 ms	Fine-tuning with selective layer freezing

In contrast, the proposed approach demonstrates that competitive and even superior performance can be achieved through a systematic fine-tuning strategy without altering the original YOLOv8 architecture. By selectively freezing backbone layers, the YOLOv8x19 model effectively balances feature preservation and task-specific adaptation, resulting in a high mAP@0.50 of 0.99 while reducing inference time to 16.2 ms. This inference speed is significantly lower than that of the fully trainable YOLOv8x model and comparable to lightweight variants, despite using a deeper backbone.

Furthermore, unlike prior studies that primarily focus on binary or single-class detection (e.g., ripe fruit only), this work addresses a practical three-class ripeness classification problem aligned with industrial grading requirements. The results indicate that the proposed fine-tuning strategy enables robust performance even with a relatively small dataset, highlighting its suitability for real-world deployment where data collection and computational resources may be limited.

Overall, this comparison confirms that the proposed YOLOv8x19 model achieves a favorable trade-off between accuracy and efficiency, positioning it as a competitive alternative to both heavyweight and lightweight YOLO-based approaches for agricultural object detection.

Despite the promising results, this study has several limitations. First, the dataset size is relatively small compared to large-scale benchmarks, which may limit generalization to highly diverse field conditions. Second, the classification task is restricted to three ripeness classes, and additional categories such as overripe or abnormal fruit bunches were not considered. Third, although the proposed fine-tuning strategy improves efficiency, the YOLOv8x-based model still requires relatively high computational resources compared to lightweight architectures, which may constrain deployment on low-end edge devices. Moreover, qualitative inspection of test results indicates that the model is sensitive to challenging conditions such as non-uniform illumination, partial occlusion, and complex backgrounds. Misclassifications were more frequent under these conditions, highlighting the need for more diverse training data to improve robustness in real-world deployment.

V. CONCLUSION

This study systematically investigated the use of pre-trained and fine-tuned YOLOv8 models for oil palm fresh fruit bunch ripeness detection, progressing from dataset preparation to model selection, fine-tuning, and performance evaluation. This research has demonstrated the capability of YOLOv8 in detecting and classifying the ripeness of oil palm bunches by using a fine-tuning technique to increase accuracy and reduce detection time. Including this study involved fine-tuning the YOLOv8 models specifically YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x by freezing different numbers of blocks to enhance performance for a specific task. For result, YOLOv8x demonstrated the best overall performance with precision, recall, F1-Score, mAP@0.50, and mAP@0.50–0.95 scores of 0.92, 0.96, 0.94, 0.97, and 0.76, respectively, prompting its selection for fine-tuning. The fine-tuned variant, YOLOv8x19, with 19 frozen blocks, achieved even higher Precision, Recall, and F1-Score at 0.97, and mAP@0.50, mAP@0.50–0.95 at 0.99, 0.77 respectively. Additionally, YOLOv8x19 boasted a detection time speed of 16.2 ms and a model size of 136.7 MB.

Beyond demonstrating high detection accuracy, this study contributes methodologically by providing a systematic analysis of fine-tuning strategies for YOLOv8 in an agricultural context. Rather than merely applying a pretrained detection model, the proposed layer-freezing approach offers practical guidance on adapting deep learning models to domain-specific tasks with limited data. The findings highlight how selective freezing can improve both accuracy and inference efficiency, reinforcing the applicability of deep learning-based object detection for real-world agricultural automation. Furthermore, comparative analysis with recent YOLO-based studies confirms that the proposed fine-tuning strategy enables YOLOv8 to achieve competitive performance without introducing architectural complexity, reinforcing its applicability for real-world agricultural deployment.

From a deployment perspective, the achieved inference speed and model complexity indicate that the proposed YOLOv8x19 model is suitable for real-time or near-real-time applications in palm oil processing facilities. The model can be deployed on GPU-equipped industrial PCs or edge servers for automated ripeness inspection. However, deployment on low-end edge devices may require further model compression or the use of lighter YOLOv8 variants.

Future work will focus on expanding the dataset to include more diverse environmental conditions and additional ripeness categories. Furthermore, lightweight YOLOv8 variants and model compression techniques will be explored to facilitate deployment on edge devices with limited computational resources. In addition, the integration with a web application was planned by using Flask to create a YOLO algorithm service for uploading, detecting, and returning the ripeness results of oil palm bunches. Following this, a backend server will be developed to receive data from the YOLO algorithm service, store it in a database, and send it to the web application's frontend. The frontend will feature the history of ripeness results, graphs showing the quality and quantity of ripe oil palm bunches, a dashboard for business analysis, and an image upload feature for detection. The advantage of this web application is that the model can be applied in business, supporting better decision-making, resource management, and strategic planning.

Future research will address these failure cases by incorporating additional data modalities, such as multispectral or hyperspectral imaging, to better capture biochemical differences between ripeness stages. Expanding the dataset to include more diverse environmental conditions and ambiguous samples will further improve robustness. Additionally, advanced techniques such as attention mechanisms or temporal information from video sequences may help mitigate misclassification caused by occlusion and visual similarity. Lightweight model variants and model compression techniques will also be explored to enhance deployment feasibility on edge devices.

Conflict of Interest

The authors declare no conflicts of interest.

Author Contributions

S.J. collected the data, made contributions to data analysis, and made significant contributions to the conceptualization, methodology, data analysis, supervision, validation, and writing, review, and editing of the study. Y.P. collected the data, made contributions to data analysis, and made significant contributions to the conceptualization, methodology, data analysis, supervision, validation, and writing, review, and editing of the study. J.M. collected the data, made contributions to data analysis, and results visualization, and wrote the manuscript. P.S. and P.C. performed data analysis and results visualization and wrote the manuscript. T.P. and D.C. made significant contributions to the conceptualization, methodology, data analysis, supervision, validation, and writing, review, and editing of the study. All authors read and approved the final manuscript.

Acknowledgment

The authors are deeply grateful to the Faculty of Science and Industrial Technology, Prince of Songkla University, Surat Thani campus, Thailand.

Funding

This research was financially supported by Prince of Songkla University, Surat Thani Campus and Hat Yai Campus for research grants.

References

[1] X. Liu, K. H. Ghazali, F. Han, and I. I. Mohamed, "Automatic detection of oil palm tree from UAV images based on the deep learning method," *Applied Artificial Intelligence*, vol. 35, no. 1, pp. 13–24, 2021.

[2] M. Asrol, D. N. Utama, and F. A. Junior, "Real-time oil palm fruit grading system using smartphone and modified YOLOv4," *IEEE Access*, vol. 11, pp. 59758–59773, 2023.

[3] M. G. Naftali and G. Hugo, "Palm oil counter: State-of-the-art deep learning models for detection and counting in plantations," *IEEE Access*, vol. 12, pp. 90395–90417, 2024.

[4] K. Yarak, A. Witayangkurn, K. Kritiyutanont *et al*., "Oil palm tree detection and health classification on high-resolution imagery using deep learning," *Agriculture*, vol. 11, no. 2, 183, 2021.

[5] A. Septiarini, A. Sunyoto, H. Hamdani *et al*., "Machine vision for the maturity classification of oil palm fresh fruit bunches based on color and texture features," *Scientia Horticulturae*, vol. 286, 110245, 2021.

[6] N. Sabri, Z. Ibrahim, S. Syahlan *et al*., "Palm oil fresh fruit bunch ripeness grading identification using color features," *Journal of Fundamental and Applied Sciences*, vol. 9, pp. 563–579, 2017.

[7] M. Makky and P. Soni, "In situ quality assessment of intact oil palm fresh fruit bunches using rapid portable non-contact and non-destructive approach," *Journal of Food Engineering*, vol. 120, pp. 248–259, 2014.

[8] T. Raj, F. H. Hashim, A. B. Huddin *et al*., "Classification of oil palm fresh fruit maturity based on carotene content from Raman spectra," *Scientific Reports*, vol. 11, no. 1, 18315, 2021.

[9] K. Choudhary, B. DeCost, C. Chen *et al*., "Recent advances and applications of deep learning methods in materials science," *NPJ Computational Materials*, vol. 8, no. 1, 59, 2022.

[10] Z. Ibrahim, N. Sabri, and D. Isa, "Palm oil fresh fruit bunch ripeness grading recognition using convolutional neural network," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 10, no. 3–2, pp. 109–113, 2018.

[11] G. N. Elwirehardja and J. S. Prayoga, "Oil palm fresh fruit bunch ripeness classification on mobile devices using deep learning approaches," *Computers and Electronics in Agriculture*, vol. 188, 106359, 2021.

[12] N. Khamis, H. Selamat, S. Ghazalli *et al*., "Comparison of palm oil Fresh Fruit Bunches (FFB) ripeness classification technique using deep learning method," in *Proc. 2022 13th Asian Control Conference (ASCC)*, 2022, pp. 64–68.

[13] M. H. Junos, A. S. Mohd-Khairuddin, S. Thannirmalai, and M. Dahari, "Automatic detection of oil palm fruits from UAV images using an improved YOLO model," *The Visual Computer*, vol. 38, no. 7, pp. 2341–2355, 2022.

[14] E. Salim, "Hyperparameter optimization of YOLOv4 tiny for palm oil fresh fruit bunches maturity detection using genetics algorithms," *Smart Agricultural Technology*, vol. 6, 100364, 2023.

[15] F. A. Junior, "Video based oil palm ripeness detection model using deep learning," *Heliyon*, vol. 9, no. 1, 2023.

[16] E. Olivas, J. Guerrero, M. Martinez-Sober, J. Magdalena-Benedito, and A. Serrano López eds. *Handbook of Research on Machine*

Learning Applications and Trends: Algorithms, Methods, and Techniques, 2009, pp. 242–264. https://doi.org/10.4018/978-1-60566-766-9.ch011
[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.
[18] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7263–7271, 2017.
[19] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *Computer Vision and Pattern Recognition*, vol. 1804, pp. 1–6, 2018.
[20] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," arXiv preprint, arXiv:2004.10934, 2020.
[21] Ultralytics. (2022). YOLOv5: A scalable object detection model. [Online]. Available: https://github.com/ultralytics/yolov5
[22] C. Li, L. Li, H. Jiang *et al*., "YOLOv6: A single-stage object detection framework for industrial applications," arXiv preprint, arXiv:2209.02976, 2022.
[23] C. Y. Wang, A. Bochkovskiy, and H. Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7464–7475.
[24] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-time flying object detection with YOLOv8," arXiv preprint, arXiv:2305.09972, 2023.
[25] S. Li, T. Tao, Y. Zhang *et al*., "YOLO v7-CS: A YOLO v7-based model for lightweight bayberry target detection count," *Agronomy*, vol. 13, no. 12, 2952, 2023.
[26] J. W. Lai, H. R. Ramli, L. I. Ismail, and W. Z. W. Hasan, "Real-time detection of ripe oil palm fresh fruit bunch based on YOLOv4," *IEEE Access*, vol. 10, pp. 95763–95770, 2022.
[27] Suharjito, F. A. Junior, Y. P. Koeswandy *et al*., "Annotated datasets of oil palm fruit bunch piles for ripeness grading using deep learning," *Scientific Data*, vol. 10, no. 1, 72, 2023.
[28] Y. Lai, R. Ma, Y. Chen *et al*., "A pineapple target detection method in a field environment based on improved yolov7," *Applied Sciences*, vol. 13, no. 4, 2691, 2023.
[29] B. Gu, C. Wen, X. Liu *et al*., "Improved YOLOv7-tiny complex environment citrus detection based on lightweighting," *Agronomy*, vol. 13, no. 11, 2667, 2023.
[30] T. S. Gunawan, M. Kartiwi, H.Mansor, and N. M. Yusoff, "Palm fruit ripeness detection and classification using various YOLOv8 Models," in *Proc. 2023 IEEE 9th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA)*, 2023, pp. 193–198.
[31] X. Wu, Y. Tian, and Z. Zeng, "LEF-YOLO: A lightweight cherry tomato detection YOLOv8 network with enhanced feature fusion," in *Proc. International Conference on Intelligent Computing*, 2025, pp. 474–488.
[32] M. S. Alfatni, S. Khairunniza-Bejo, M. H. Marhaban *et al*., "Towards a real-time oil palm fruit maturity system using supervised classifiers based on feature analysis," *Agriculture*, vol. 12, no. 9, 1461, 2022.
[33] J. W. Lai, H. R. Ramli, L. I. Ismail, and W. Z. Wan-Hasan, "Oil palm fresh fruit bunch ripeness detection methods: A systematic review," *Agriculture*, vol. 13, no. 1, 156, 2023.