J. Umm Al-Qura Univ. Eng.Archit.
https://doi.org/10.1007/s43995-025-00164-7

ORIGINAL ARTICLE

Efficient real-time palm oil tree detection and counting using YOLOv8 deployed on edge devices

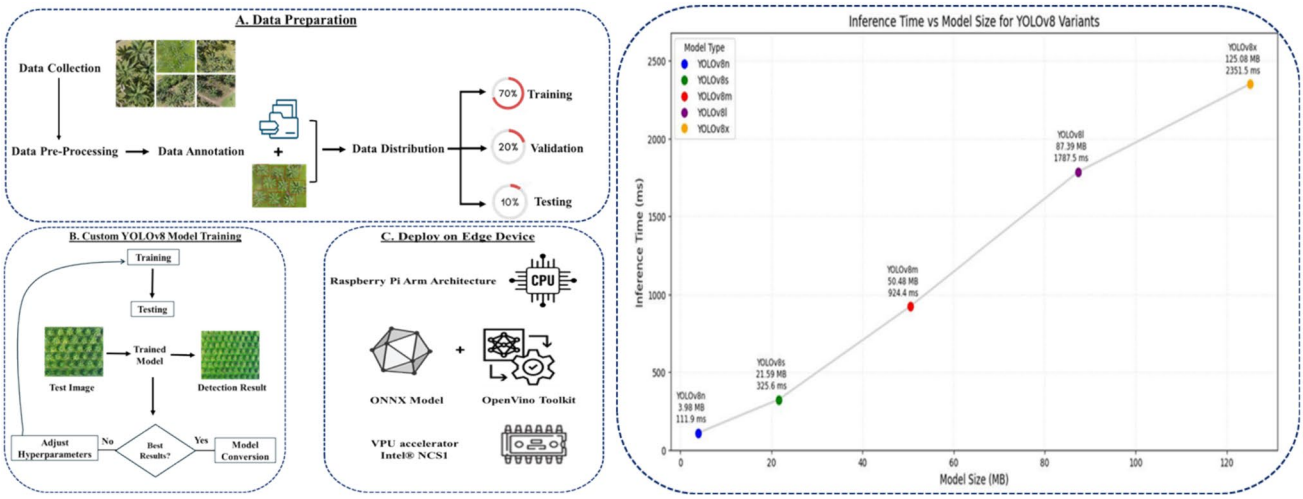
Mohammad Farhan[1] · **Mohammad Nishat Akhtar**[1] · **Elmi Abu Bakar**[1]

Received: 12 March 2025 / Accepted: 28 May 2025

Abstract

This study investigates the application of YOLOv8 object detection models for identifying and counting oil palm trees in plantation management, with a focus on deployment on edge devices. Various YOLOv8 architectures were trained and evaluated using drone-captured images of palm oil plantations. The YOLOv8 nano model delivered superior performance, achieving 95.6% precision, 93.3% recall, and 98% mAP @0.5. The optimized model was deployed on a Raspberry Pi 4B (RPi) equipped with an Intel Neural Compute Stick 1 (NCS1) accelerator, enabling real-time detection under field conditions. Post-deployment performance analysis indicated an average inference time of 0.4 s per image (2.4 FPS) with minimal accuracy reduction (97% mAP @0.5). The system demonstrated low power consumption (1.6W peak) and efficient memory usage (450 MB RAM), emphasizing its suitability for edge computing in precision agriculture. This research demonstrates the feasibility of employing compact, efficient deep-learning models on resource-constrained devices like the RPi for palm oil tree detection and counting. The proposed system provides a portable and energy-efficient solution for real-time monitoring of palm oil plantations, offering significant potential to enhance plantation management by enabling data analysis and decision-making in remote or challenging environments. Despite promising results, the system's performance was evaluated under specific environmental conditions, potentially limiting its generalizability across diverse plantation landscapes. Future research should aim to expand detection capabilities to include disease identification and yield estimation.

Graphical abstract



Keywords Deep learning · Neural computing stick · Object detection · Palm oil tree counting · Raspberry Pi · YOLOv8

Abbreviations

RPi	Raspberry Pi
FPS	Frames per second

Extended author information available on the last page of the article

Published online: 11 July 2025

CSP	Cross Stage Partial
YOLO	You only look once
IoT	Internet of things
NCS	Neural computing stick
UAV	Unmanned aerial vehicle
mAP	Mean average precision
SPP	Spatial pyramid pooling
SPPF	Spatial pyramid pooling fast
ONNX	Open neural network exchange
COCO	Common Objects in Context dataset

1 Introduction and background

Agriculture is an important sector in Malaysia's economy. This is evident from the Gross Domestic Product (GDP) contribution, which is approximately 7.4% in 2023. The plantation industry has a high potential to contribute to the agriculture sector. In 2023, plantations contributed 3.4% of the overall GDP and 46.3% of the agricultural sector [1, 2]

The palm tree is one of the vegetable oil plants with the highest production rates globally, according to Ebongue et al. [3], the increase in palm oil output over the preceding 30 years supports this claim. According to the Food and Agriculture Organization (FAO), in 1995, global palm oil production was around 15 million tonnes. This production rate increased rapidly, with the total production reaching 78 million tonnes in 2023, a more than five-fold increase [4]. Malaysia is currently a major global producer of palm oil, generating 19.25 million tonnes in 2023, according to FAO. This production volume positions Malaysia as the world's second-largest palm oil producer, behind Indonesia, which topped 47 million tonnes in the same year. The statistics reported by the Department of Statistics Malaysia (DOSM) and approved by the FAO indicate that Malaysia's domestic palm oil production reached 19.2 million tonnes in 2023, with the state of Sabah being the largest contributor, accounting for approximately 20.4% of the national output. Thailand follows as the third-largest producer, generating 3.28 million tonnes of palm oil in 2023 [4, 5].

The productivity of oil palm cultivation is a complex subject influenced by numerous variables. The key factors that drive oil palm growth and yield remain a challenge to determine due to the multifaceted nature of the system [6]. However, technological innovations, such as the implementation of precision agriculture techniques, offer a promising solution to this challenge [7]. Precision agriculture can help optimize resource use and minimize negative environmental impacts, creating an opportunity to enhance the sustainability of the significant palm oil industry in Malaysia. As a major global producer, Malaysia is well-positioned to adopt further and leverage precision agriculture to improve the local economy and maintain environmentally responsible practices simultaneously [8].

Hence, conducting a survey of oil palm trees is crucial for precise production predictions and efficient plantation management. This survey entails the counting, positioning, and mapping of the trees' patterns and distribution [9]. However, the manual survey process encounters challenges such as labor-intensive requirements, high costs, and difficulties in accessing remote locations. Technological solutions are necessary to overcome these obstacles. Aerial monitoring of large-scale palm oil plantations can be conducted using satellite imagery, manned aircraft, or unmanned aerial vehicles (UAVs) [10]. Satellite imaging faces challenges due to cloud cover, which can impair image quality and hinder the detection of palm trees [11]. Manned aircraft are unsuitable for tree monitoring because of their high operational expenses and risk of environmental pollution [12].

In contrast, drones (a type of UAV) offer a more affordable and environmentally friendly alternative. Drones are equipped with high-resolution cameras capable of capturing medium to high-quality aerial imagery with a wide field of view, depending on the flight altitude. Drones have enabled various applications, including surveillance, search and rescue, and precision agriculture [13, 14]. However, the use of drones is often subject to legal and regulatory restrictions, such as minimum flight altitudes and permitted locations, which can pose challenges for capturing crucial video footage for agricultural purposes [15, 16]. Additionally, the small size of palm trees and potential occlusion by other objects may complicate their detection and recognition using computer vision techniques.

Remote sensing technologies have found widespread application in mining, agriculture, and plantation management, enabling remote monitoring of plantation areas. This is advantageous given the challenges associated with manual on-site monitoring, particularly in large-scale Malaysian oil palm farms with extensive cultivation areas and sometimes rugged terrain. Several techniques have been developed to optimize the detection of individual trees within plantation imagery, broadly categorized into three main methods: digital image processing [17–19], machine learning [20], and deep learning, the latter further divided into CNN classification [21], and object identification [22, 23] approaches. The digital image processing method involves four stages: pre-processing, treetop detection, tree crown delineation, and post-processing [24]. And the machine learning method extracts features and trains a classification model for tree detection using a sliding window technique. Deep learning-based approaches, including R-CNN, Fast R-CNN, Faster R-CNN, and YOLO (You Only Look Once), have gained popularity due to their high detection accuracy through automatic object-based techniques [25, 26]. Traditional

machine learning techniques, such as decision trees and support vector machines, rely heavily on manual feature extraction. This process is labour-intensive and often insufficient for capturing the complex visual patterns of oil palm trees in varied environments. These methods typically struggle with challenges like occlusion, shadows, and overlapping canopies common in plantation settings, leading to reduced detection accuracy and generalization capabilities across different lighting conditions and seasonal variations. In contrast, deep learning approaches, particularly convolutional neural networks (CNNs), automatically learn hierarchical feature representations from data, enhancing robustness and effectiveness in complex visual tasks. Models like YOLO have demonstrated superior accuracy and computational efficiency for real-time object detection tasks, making them particularly suitable for deployment on resource-constrained edge devices in remote plantation areas [27, 28].

YOLO is a popular object detection method which utilizes Convolutional Neural Networks (CNNs) as the backbone [29]. The YOLO model was developed to reduce the detection time of popular object detection methods such as R-CNN [30], Fast R-CNN [31], and Faster R-CNN [32]. It has demonstrated reliable performance as well as accuracy, making it ideal for real-time object detection applications. Since its initial release [29], YOLO has evolved into more effective variants, including YOLOv1, YOLOv2 (YOLO9000), YOLOv3 [33], YOLOv4 [34], YOLOv5 [35], YOLOv6 [36], YOLOv7 [37], and YOLOv8 [38], which have been used extensively in numerous fields, including medical [39], remote sensing [40], transport [41], and farming [42]. While YOLOv8 has shown promising results in identifying small objects [38], it is essential to note that pre-trained YOLOv8 models may not be suitable for detecting palm oil trees due to the difference in camera viewpoint between human installed CCTV cameras and drone mounted cameras. As a result, a crucial step in the proposed strategy should be retraining the chosen model for object detection from an aerial perspective.

This study developed a model that counts palm oil trees using YOLOv8, a method that has already shown potential in identifying palm oil tree and tree damages. [23]. The experimental results of this study compared the palm oil tree identification accuracy results obtained using five unique variants of YOLOv8: YOLOv8n (nano), YOLOv8s (small), YOLOv8m (medium), YOLOv8l (large), and YOLOv8x (extra-large) [38]. The methodology to detect the oil palm tree through drone imagery technique is described, and experimental data analysis is discussed in detail. Prospects for further research in this field of study are also recommended.

2 Methodology

The proposed oil palm tree detection system comprises two primary phases: training (encompassing both training and validation) and testing. Data sets are collected for the training phase, requiring thorough preparation before initiating the process. The study followed a staged approach, with each phase, including data preparation, training, validation, testing, and evaluation, stated in the following subsections and illustrated in Fig. 1.

2.1 Data capture and preparation

This research used a DJI Mavic Air drone equipped with a 640×512 resolution thermal camera and a 9.1 mm lens to acquire imagery. With permission from the university authorities, the drone conducted flights over specific palm oil farms at a certain height around the Universiti Sains Malaysia Engineering Campus, Nibong Tebal, South Seberang Perai, Penang, Malaysia shown in Fig. 2. The plantation features diverse topography, including both flat and hilly areas. In the flat areas, the plant canopies are sparse and close together. In contrast, the canopies overlap and intersect with other vegetation in the hilly areas, posing challenges for object detection algorithms.

The path plan was planned by inbuilt path planner software program of the drone, with 70% horizontal and 20% vertical overlaps. The DJI Mavic Air has a great speed suitable for efficient data collection and captures thermal and RGB imagery. To ensure comprehensive coverage and data quality, the images were captured with specified overlaps (70% front lap and 20% side lap).

The images were captured in January and February 2024, on clear sky days, between 9:00 a.m. and 1:00 p.m. (GMT + 8) to leverage consistent natural lighting conditions and minimize the effects of shadows and low-angle sunlight, which can introduce distortions and reduce image clarity. Capturing images during this time frame ensures optimal illumination, enhancing feature visibility and contributing to more accurate object detection. This approach aligns with best practices in aerial imaging, where consistent lighting conditions are crucial for reliable data acquisition and analysis. Figure 3 presents two examples of raw images acquired by the drone camera while looking vertically downward from the drone. The images depict oil palm trees of various sizes that are irregularly and unevenly distributed. Some oil palm trees appear to be dead or have sparse foliage. It is important to note that these specific trees are not considered within the scope of the research presented in the present research.

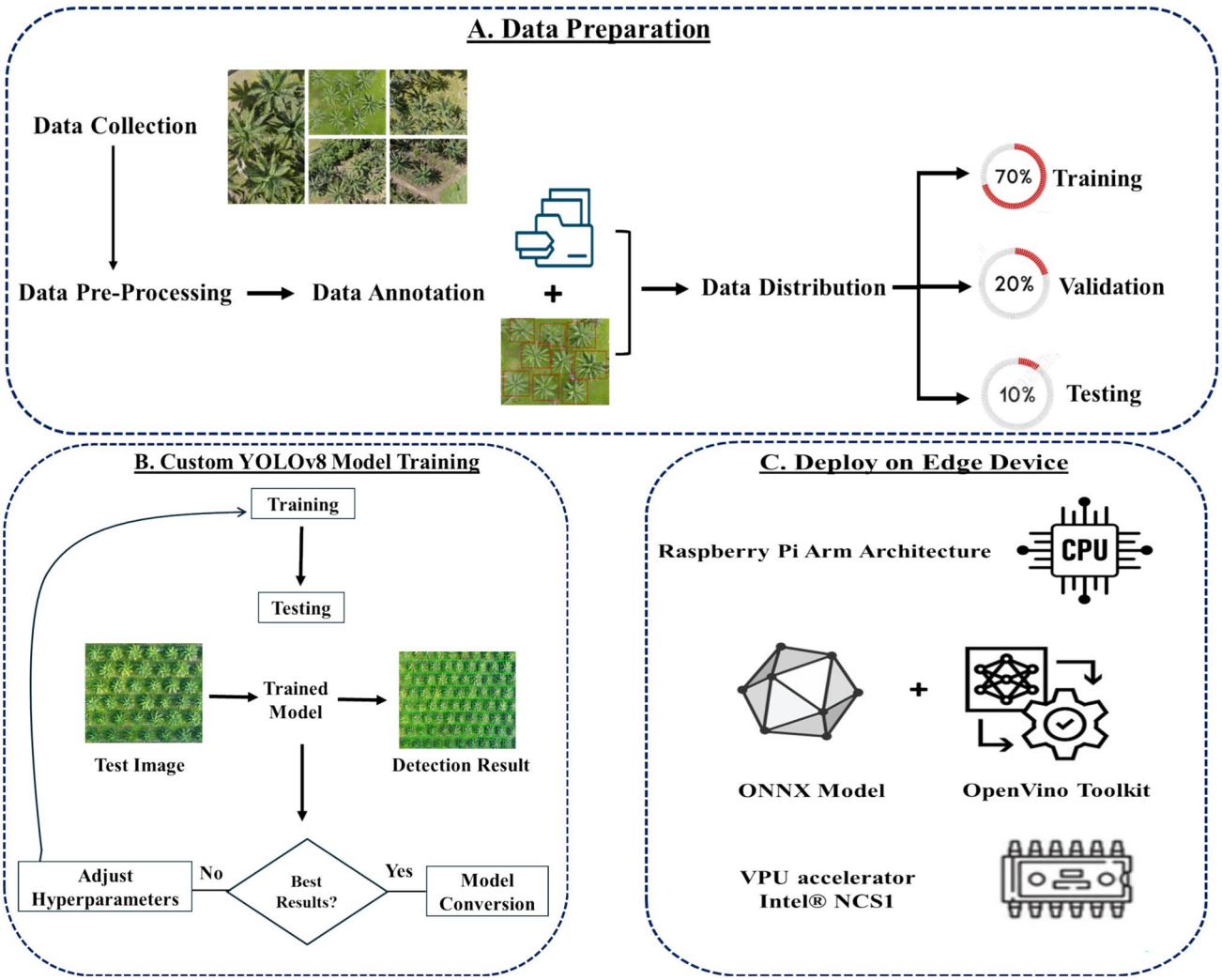


Fig. 1 Workflow of the palm oil tree detection system, including aerial image acquisition, annotation and dataset preparation, YOLOv8 model training with hyperparameter tuning, and deployment on a Raspberry Pi with Intel NCS1

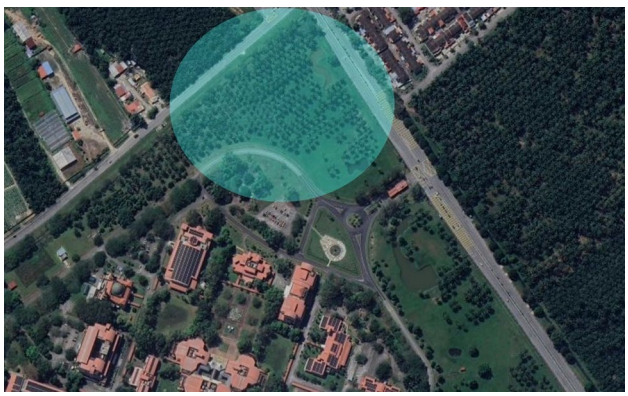


Fig. 2 Study area: Drone-based imaging of palm oil farm for dataset creation and model training

The collected data underwent further processing during the data preparation stage to support the development of the YOLOv8 model. This process involved data regeneration and reformatting. A total of 482 images, each with dimensions of 1280 × 959 pixels, were used. The Roboflow web application facilitated data regeneration, and the dataset was divided into three subsets: training (70%), validation (20%), and testing (10%), comprising 338, 96, and 48 images, respectively. During the training phase, the Yolo Label tool was utilized to annotate all oil palm trees in the training and validation datasets with bounding boxes labelled as "palm."

The training dataset is used to train the YOLOv8 network. The trained network detected only one object which is a palm oil tree. Similarly, the validation dataset is used during training to evaluate the performance of the model. Finally, the performance of the palm oil tree detection models is

Fig. 3 Digital images of captured palm oil trees

Table 1 Total Number of Labelled palm trees

Datasets	Number of labelled palm oil trees
Training	10,364
Validation	1068
Testing	574

tested using the test dataset. Table 1 shows the total number of labelled palm trees across the training, validation, and test datasets, representing the foundation for our detection model. Oil palm trees are enclosed in bounding boxes when annotating data for training and validation. These bounding boxes may contain oil palm trees of various sizes, which may overlap or be occluded by other oil palm trees, as well as different backgrounds (e.g., other plants). It is essential to have bounding boxes of picture pixels that capture these variations for testing and training to effectively train the YOLOv8 network and then apply the trained model for palm oil tree detection.

2.2 Model training

2.2.1 YOLOv8 architecture

In the present research, five variants of the YOLOv8 network, which includes YOLOv8 (n, s, m, l, x) mentioned above in Sect. 1 were trained using the dataset. The backbone, neck, and head are the three main components that make up the network architecture of all YOLOv8 versions, but they differ in terms of model depth and complexity [38]. The different YOLOv8 variants, n, s, m, l, and x refer to the increasing depth and complexity of the network architecture used. Figure 4 illustrates the network architecture of YOLOv8n, which acts as the foundation for all variants of YOLOv8. The backbone of YOLOv8 model extracts crucial information from an input image. It is based on the Cross Stage Partial Network (CSPNet) [43], which extracts high-level features while retaining accuracy and reducing model processing time. In YOLOv8, the model neck collects feature maps from various stages of the model backbone to build feature pyramids. YOLOv8 continues to produce feature pyramids utilizing an enhanced version of the Path Aggregation Network (PANet) [44] within the model neck. Feature pyramids help the model in identifying and matching the same kind of object at various sizes and scales. This is especially important in our study for detecting oil palm trees of various sizes over flat and hilly terrain.

In YOLOv8, the model head finally completes the last stage of object detection. The final output vectors are generated by the model head and consist of coordinates for the bounding boxes surrounding the objects that have been found, objectness scores, and class probabilities (in this example, the probability that an object is a palm oil tree). Unlike its predecessors, YOLOv8 has had a significant update in its detection head. It uses a decoupled head to divide the classification responsibilities and bounding box regression. This decoupled approach facilitates more focused learning of each task, possibly improving classification accuracy and bounding box predictions.

As shown in Fig. 4, YOLOv8 has several vital components within each part of its network, such as C2f (Cross-Stage Partial Network with 2 Convolutions and Feature Fusion), SPPF (Spatial Pyramid Pooling—Fast), and the decoupled detection head. YOLOv8 introduces significant architectural changes compared to its older versions, aiming for improved performance and efficiency. The backbone of YOLOv8 is based on an enhanced CSPDarknet, which uses C2f modules. These C2f modules evolve the CSP (Cross-Stage Partial) connections used in previous versions. They are designed to improve the learning ability of the model while maintaining computational efficiency. The C2f module uses a more efficient design than the earlier CSP modules, allowing for deeper feature extraction without significantly increasing computational cost.

The SPPF module in YOLOv8 replaces the SPP (Spatial Pyramid Pooling) module used in earlier versions. SPPF is a more efficient implementation that achieves similar results to SPP but with reduced computational overhead. It uses

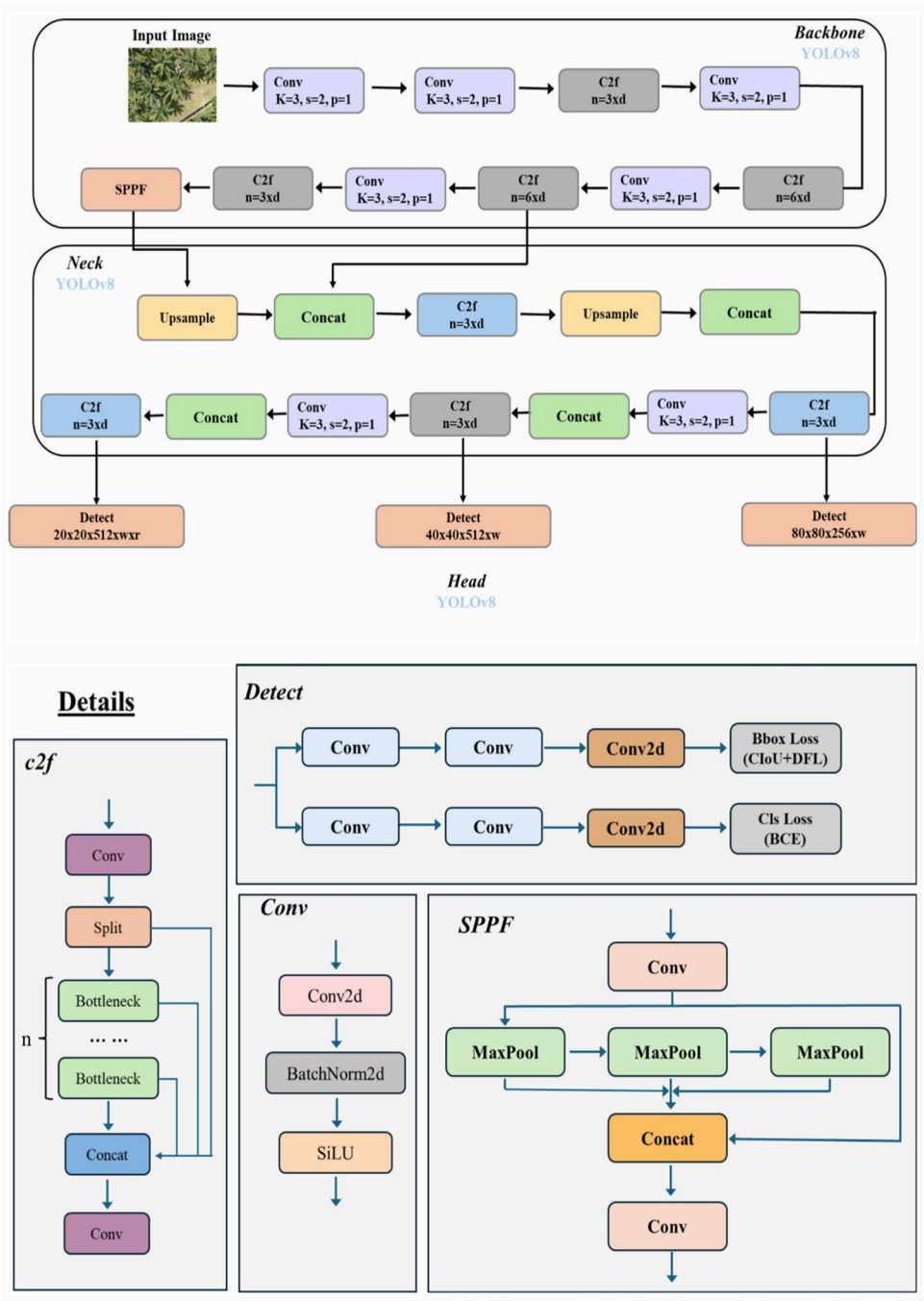


Fig. 4 The network architecture of YOLOv8n

a series of max pooling operations to capture multi-scale features, which is particularly useful for detecting oil palm trees of various sizes in our study. The neck of YOLOv8 still utilizes a PANet structure but with optimizations for better feature fusion across different scales. This is crucial for our application, as it allows the model to effectively detect oil palm trees in both densely packed and sparsely distributed areas of the plantation.

A significant change in YOLOv8 is its detection head. Unlike previous versions that used a single head for classification and bounding box regression, YOLOv8 employs a decoupled head. This means it has separate branches for classification and bounding box prediction. This decoupled design allows each task to be optimized independently, potentially leading to improved object classification and localization accuracy. Moreover, YOLOv8 adopts an anchor-free approach, eliminating the need for predefined anchor boxes. This improvement improves the identification process and may improve performance, especially for objects with varied aspect ratios, such as oil palm

trees observed from various viewpoints or growth stages. These architectural modifications in YOLOv8 are expected to improve the ability to accurately detect and count oil palm trees in complex plantation environments, effectively addressing challenges such as overlapping canopies in hilly areas and varying tree densities across the plantation.

The network architecture of YOLOv8 is scalable, with different versions offering a balance between speed and accuracy. This scalability allows us to choose the most appropriate model size for our specific palm oil tree detection requirements, which balances the need for accuracy with the computational constraints of edge devices like the RPi.

2.2.2 Training the network

To enhance the diversity of the training image dataset, data augmentation was applied during the training phase using various transformations, including auto orientation, resizing to 640 × 640 pixels, cropping, rotation, and adjustments to hue, saturation, and brightness. The specific 640 × 640 pixels resolution was selected as it represents the standard input dimension for YOLOv8 architectures, providing an optimal balance between detection accuracy and computational efficiency which is critical consideration for subsequent deployment on resource-constrained edge devices. This square format also eliminates aspect ratio distortions during batch processing and facilitates effective mosaic augmentation. Moreover, standardizing image dimensions facilitates consistent input during training and inference, which is crucial for model stability and accuracy. These augmentation techniques increased the total number of training images to 1,014, tripling the original dataset. Consequently, the number of annotated oil palm trees used for training has significantly increased, improving the model's ability to recognize trees under varying conditions.

A mosaic data augmentation technique was also used, a default feature in YOLOv8. Mosaic augmentation, initially introduced with YOLOv4, combines randomly cropped parts of four training images into one. This process helps the model recognize objects outside their normal context. Mosaic augmentation further increases the diversity and adequate size of the data used in training without requiring additional storage or preprocessing time. The combined use of these data augmentation methods significantly enhances the robustness and accuracy of the YOLOv8 model. The model training process employed transfer learning, utilizing pre-trained weights from a model originally trained on the large Common Objects in Context (COCO) dataset, for 100 epochs. Although the COCO dataset includes 80 object categories, palm oil trees are not among them. Each YOLOv8 network variant underwent fine-tuning for up to 300 epochs, though some variants achieved optimal performance before reaching the maximum.

Hyperparameters were carefully selected to optimize training efficiency and model convergence. The process involved a batch size of 16, which balances computational efficiency and model generalization; a momentum of 0.8 to accelerate gradient descent and dampen oscillations; a weight decay of 0.005 to prevent overfitting by penalizing large weights; and an initial learning rate of 0.01 to ensure stable and efficient convergence, and other parameters detailed in Table 2. These parameters were chosen based on a combination of default settings recommended by the YOLOv8 framework to optimize the training efficiency and model convergence, ensuring that the YOLOv8 network could effectively learn to detect date palm trees from the provided dataset.

2.2.3 Evaluating the trained model

After completing the training process, the trained models from each variant of the YOLOv8 network were used to detect palm oil trees in the test dataset. The model underwent evaluation using both validation and test data. To test the model, validation data was used at each epoch of the training process, and testing data evaluated the final YOLOv8 model. The network automatically resized 1,280 × 959 pixels test images to 640 × 640 pixels while maintaining the original aspect ratio using image padding techniques. The trained networks then processed these square pictures, extracting features and generating multi-scale feature maps for prediction. YOLOv8, an anchor-free model, predicts directly on feature maps rather than using predefined anchor boxes. The model makes predictions at various scales, allowing it to identify oil palm plants of different sizes. Fine-grained feature maps detect smaller trees, while coarser feature maps are used for more giant trees.

Each prediction contains the centre coordinates (x, y), width, height, confidence score, and class probability for the detected oil palm tree. The (x, y) coordinates indicate the centre of the bounding box. The confidence score indicates the model's certainty that the bounding box contains an oil palm tree. The confidence score of bounding boxes of 0 or very low implies that the box is unlikely to contain

Table 2 Hyperparameters used in the model

Hyperparameters	Values
Epochs	300
Image size	640 × 640
Batch size	16
Learning rate	0.01
Momentum	0.8
Weight decay	0.0005

any object of interest. Bounding boxes with low confidence scores can be removed by specifying a threshold to reduce false positives. This technique ensures that only the bounding boxes with the highest probability of containing oil palm trees are retained, increasing detection accuracy overall.

2.3 Model optimization and deployment for edge device

This research utilizes edge computational architectures, specifically the Raspberry Pi (RPi), for the implementation of YOLOv8. Deploying the YOLOv8 network on embedded devices necessitates careful consideration of hardware limitations, computational resources, and optimization methods to achieve effective and real-time object detection performance. After evaluating all YOLOv8 variants, the YOLOv8 nano variant was selected for deployment on the RPi 4B due to its optimal balance between accuracy and computational efficiency. The RPi 4B, equipped with an ARM Broadcom BCM2711 Cortex-A72 4-core CPU operating at 1.5 GHz, 4 GB of RAM, and running the Debian 10 Buster 64-bit operating system, provides a benchmark for deploying real-time object detection and image analysis under resource-constrained environments [45].

Several techniques were applied to optimize the YOLOv8 nano model for deployment on the RPi 4B. Post-training quantization was used to reduce the model size and inference time by converting the model weights from 32-bit floating point to 8-bit integers, while still maintaining acceptable accuracy. Additionally, less important connections in the network were pruned to decrease the model's size and reduce computational demand. To ensure compatibility with the Intel® NCS1, the trained YOLOv8 nano model was first converted from its original PyTorch format into the Open Neural Network Exchange (ONNX) format. This conversion served as a standardization step, offering a flexible intermediate model representation that is compatible with the OpenVINO toolkit. As a result, the model became more lightweight and efficient, enabling faster inference, which is essential for real-time oil palm tree detection on the RPi 4B.

The Intel® NCS1 was used as an AI accelerator to improve the processing performance of the RPi 4B while keeping the system compact and energy efficient. The deployment technique involves optimizing the YOLOv8 nano model using the Intel OpenVINO toolkit. Through OpenVINO Model Optimizer, the ONNX model was converted into the Intermediate Representation (IR) format, producing an XML file that describes the model architecture and a BIN file containing the weights. During this conversion, the model underwent several hardware-level optimizations, such as layer fusion, constant folding, and the removal of unnecessary operations. To match the processing design of the NCS1, the IR model was quantized to FP16 precision, reducing memory usage and computational load without affecting detection accuracy. The RPi 4B was configured with the required drivers and the OpenVINO runtime to establish communication with the NCS1. The optimized model was deployed using the OpenVINO Inference Engine, which efficiently handled the workload between the RPi and the NCS1. The system was fine-tuned to balance the load effectively between the RPi and the NCS1, achieving optimal performance for real-time oil palm tree detection.

This integrated system takes advantage of the RPi 4B versatility and the NCS1 dedicated neural network processing capabilities, achieving a portable, energy-efficient solution capable of performing real-time oil palm tree detection and counting in the field, even in areas with limited connectivity or power resources. Significant speedups were achieved using ONNX conversion features, compared to native execution on the same platform. The installation of Intel® NCS1 resulted in a significant increase in frame-per-second throughput, showing the advantages of outsourcing computations to dedicated AI hardware. To improve model execution on the NCS1, the OpenVINO toolkit, an open-source deep learning package with a high-level Python inference engine API, used to simplify the deployment of pre-trained models on Intel hardware. Figure 5 shows the experimental environment for implementation on the RPi 4B platform.

3 Results and analysis

Following the training phase, the YOLOv8 model was deployed to detect oil palm trees within the test dataset. To assess the model's effectiveness and performance, several key metrics were utilized, including Precision, Recall, F1-score, and mean Average Precision (mAP). These metrics provide a comprehensive evaluation of the model's ability to accurately identify and classify oil palm trees in varied conditions.

Precision evaluates the accuracy of the model in predicting objects, calculated as the ratio of correctly predicted positive observations to the total predicted positive observations. It reflects the proportion of detected objects that are relevant. In this context, a True Positive (TP) refers to correctly identified oil palm trees, while a False Positive (FP) denotes instances where an object is mistakenly classified as a tree. The precision value is calculated using Eq. (1).

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{1}$$

Recall assesses the model's effectiveness in identifying relevant objects. It is the ratio of correctly predicted positive observations to the total number of actual positive observations, reflecting the model's ability to detect relevant objects among all those that are truly present.

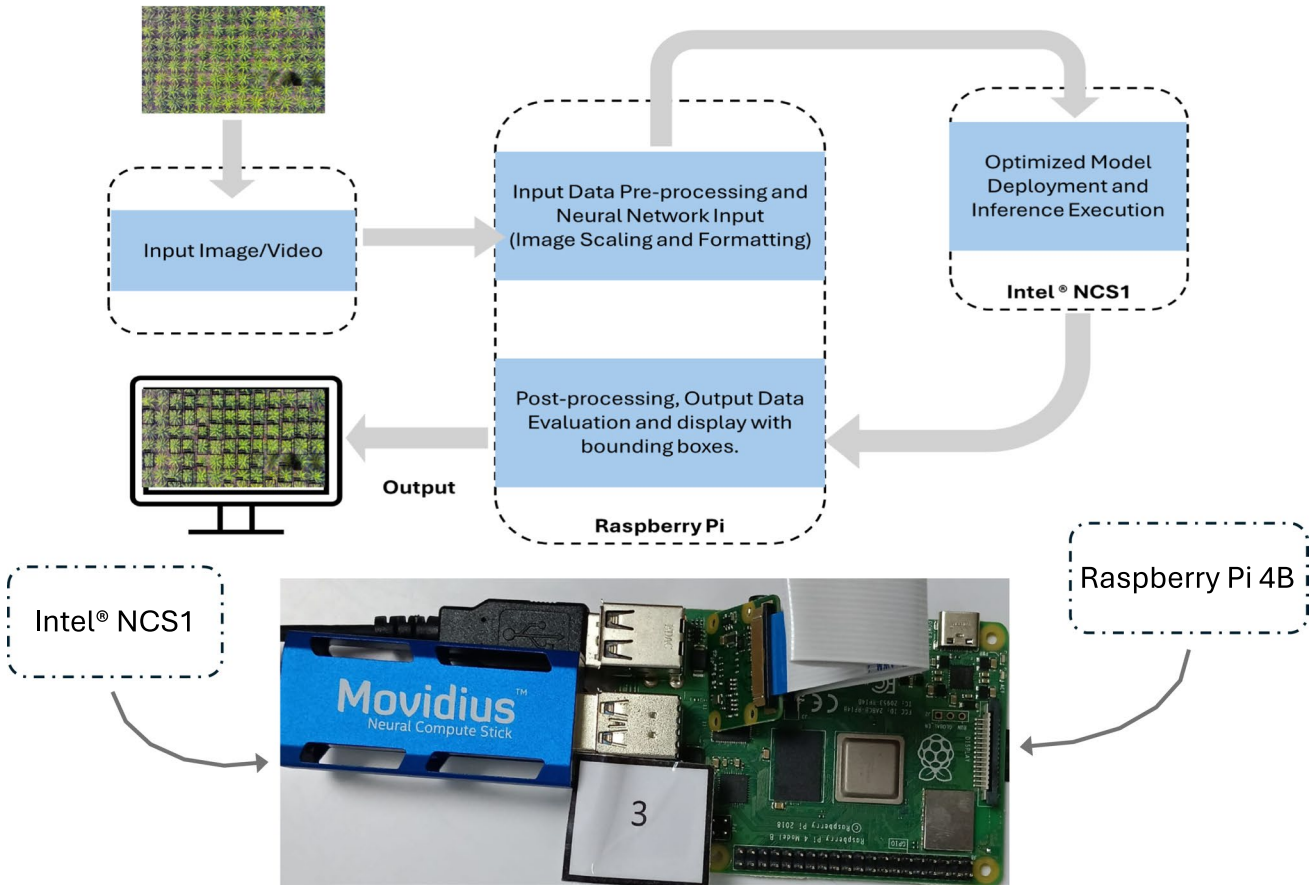


Fig. 5 Experimental setup for real-time palm oil tree detection and counting, featuring a Raspberry Pi 4B @ 1.5 GHz with 4 GB RAM connected to an Intel® NCS1 via USB

A True Positive (TP) refers to relevant objects correctly identified, while a False Negative (FN) represents relevant objects that the model failed to detect. The recall value is calculated using Eq. (2).

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{2}$$

The F1-Score represents the harmonic mean of precision and recall. The predicted model has solid precision and recall values. As a result, the optimal metric for measuring the model is F1-Score. Mathematically, it can be written as Eq. (3).

$$F1 - score = \frac{2(Precision \times Recall)}{Precision + Recall} \tag{3}$$

In addition to these metrics, mean Average Precision (mAP) provides an in-depth analysis of the precision-recall curve. It is calculated by taking an average of the Average Precision (AP) of overall classes (in this case, only palm oil tree) and different Intersection over Union (IoU) thresholds. It can be calculated using Eq. (4).

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i \tag{4}$$

For YOLOv8, mAP is usually reported at various IoU thresholds. Standard measures include mAP@0.5 (mean average precision at 0.5 IoU threshold) and mAP@0.5–0.95 (mean average precision averaged over IoU thresholds ranging from 0.5 to 0.95 in 0.05 increments). In the experimental phase of the training process, all variants of the YOLOv8 network were compared. The training was conducted for 300 epochs with a batch size of 16. The palm oil tree detection results using the trained model can be seen in Fig. 6.

The results are summarized in Table 3, which compares the evaluation metrics and inference time for all YOLOv8 models and inference time may vary based on hardware. These measurements were taken on intel Core i7-8665U CPU @ 1.90 GHz with 16 GB RAM. The YOLOv8 nano model outperformed all other versions, with a precision of 95.6%, recall of 93.3%, F1-score of 94%, and mAP of 98%. Despite having a lighter network, the YOLOv8n model outperformed the deeper versions by offering a better combination of precision and recall. The YOLOv8x model,

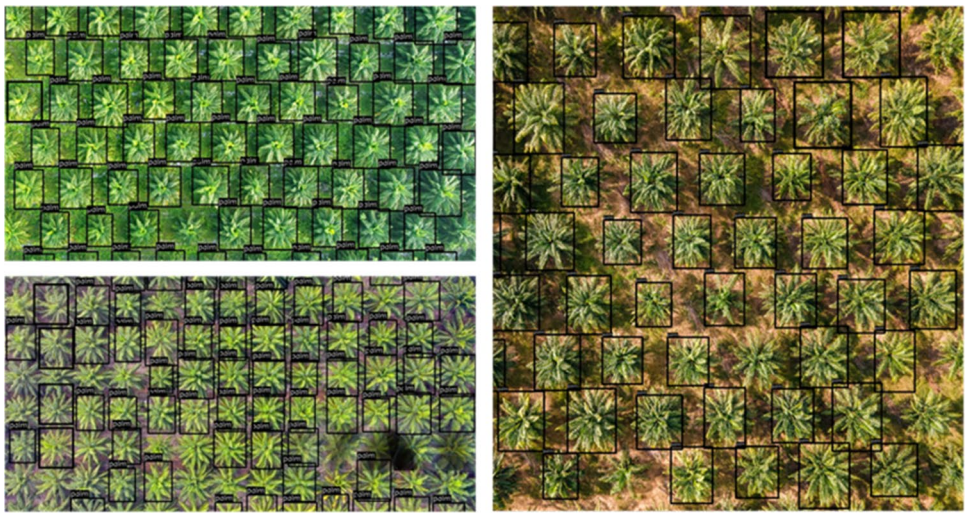


Fig. 6 Detection results of palm oil tree using trained model

while having a deeper network architecture, did not significantly enhance the outcomes, with an average F1-score of 94%, which is same as the nano version.

In terms of training duration, the YOLOv8n, YOLOv8s, and YOLOv8m models completed in around 1 h, while the YOLOv8l and YOLOv8x models took about 2 h for 300 epochs. Given the hardware limitations of the RPi 4B and Intel® NCS1 AI accelerator, the nano variant was the preferred choice for deployment due to the ideal balance of accuracy, efficiency, and computational demand. While maintaining high performance metrics, the YOLOv8n model offers the fastest inference time at 111.9 ms size shown in Fig. 7 and requires only 450 MB of RAM, making it uniquely suitable for the resource constraints of edge devices. This combination of factors like accuracy, speed, and memory efficiency positions the nano variant as the optimal choice over larger variants that would exceed the computational capabilities of RPi.

The detailed results of various aspects of YOLOv8 nano are shown in Fig. 8. The training results show that the model has progressive development across various metrics. The training loss curves for bounding boxes, classification, and distribution all show continuous decreases over time, indicating effective learning. Validation losses follow a similar downward trend, but with more fluctuation, indicating successful generalization to previously unseen data. Precision and recall metrics are both trending upward, reaching high levels (over 0.95 for both), showing improving accuracy of the model in object identification and comprehensive detection. The mAP curves indicate the improved performance of the model. The mAP50 metric immediately stabilizes at a high level, while the more stringent mAP50-95 metric improves consistently, showing that the model not only identifies but also localizes objects with increasing accuracy. Overall, the results indicate to a well-trained model with high performance in palm oil tree detection.

When running YOLOv8n on a RPi with an Intel® NCS1, additional performance metrics must be considered to assess real-world efficacy. These include inference time, which indicates the speed at which the model processes an image, and Frames Per Second (FPS), which reflects the number of images processed in real-time. Power consumption is also a critical factor for embedded systems. In our experiments, the YOLOv8n model achieved an average inference time of 0.4 s per image on the RPi with NCS1, corresponding to approximately 2.4 FPS (Table 4). During inference, the power consumption peaked at 1.6 W.

The model maintained its mAP of 97% and best balance of other variable metrices as shown in Fig. 9, with only a slight reduction in performance compared to more capable hardware, despite the hardware limitations. The YOLOv8n model is well-suited for real-time oil palm detection and counting on edge devices like the RPi with Intel® NCS1 due to its balance of speed, efficiency, and accuracy.

Table 3 Precision, recall, F1-score, mAP, and inference time of all the variants of the YOLOv8 model

	Precision	Recall	F1-score	mAP@0.5	Inference time (in ms)
YOLOv8n	0.95	0.93	0.94	0.98	111.9
YOLOv8s	0.94	0.84	0.87	0.89	320.8
YOLOv8m	0.94	0.88	0.90	0.93	920.4
YOLOv8l	0.95	0.88	0.92	0.92	1737.5
YOLOv8x	0.94	0.93	0.94	0.97	2391.5

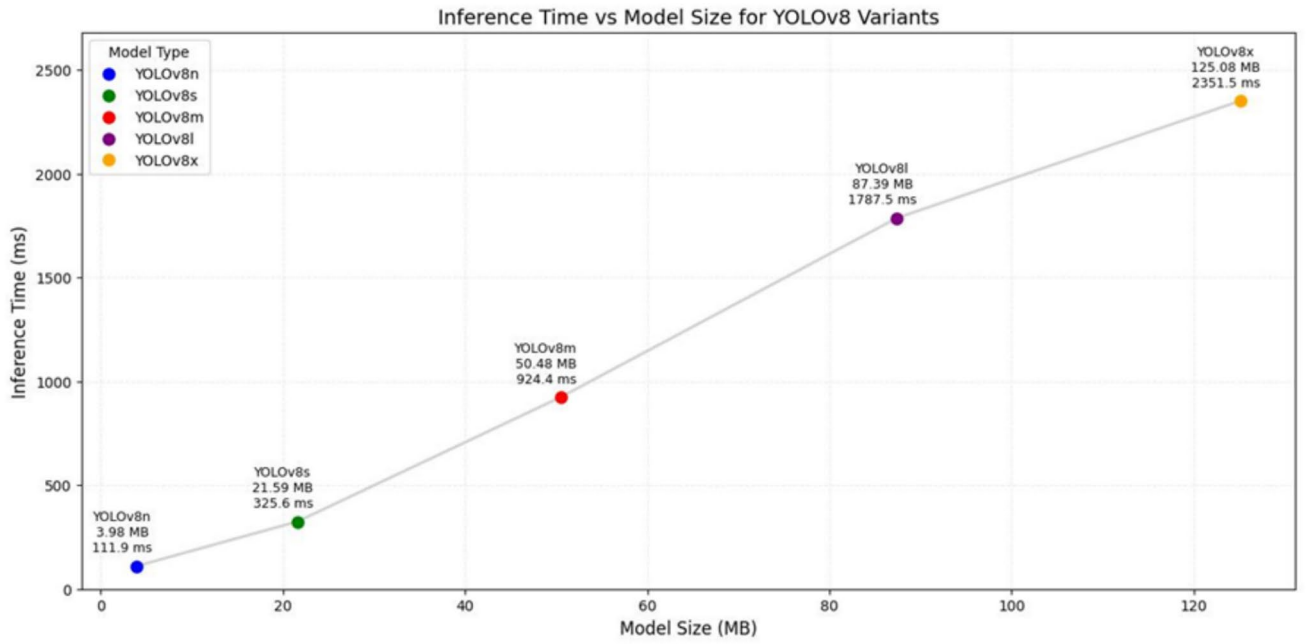


Fig. 7 Inference time vs model size for YOLOv8 Variants

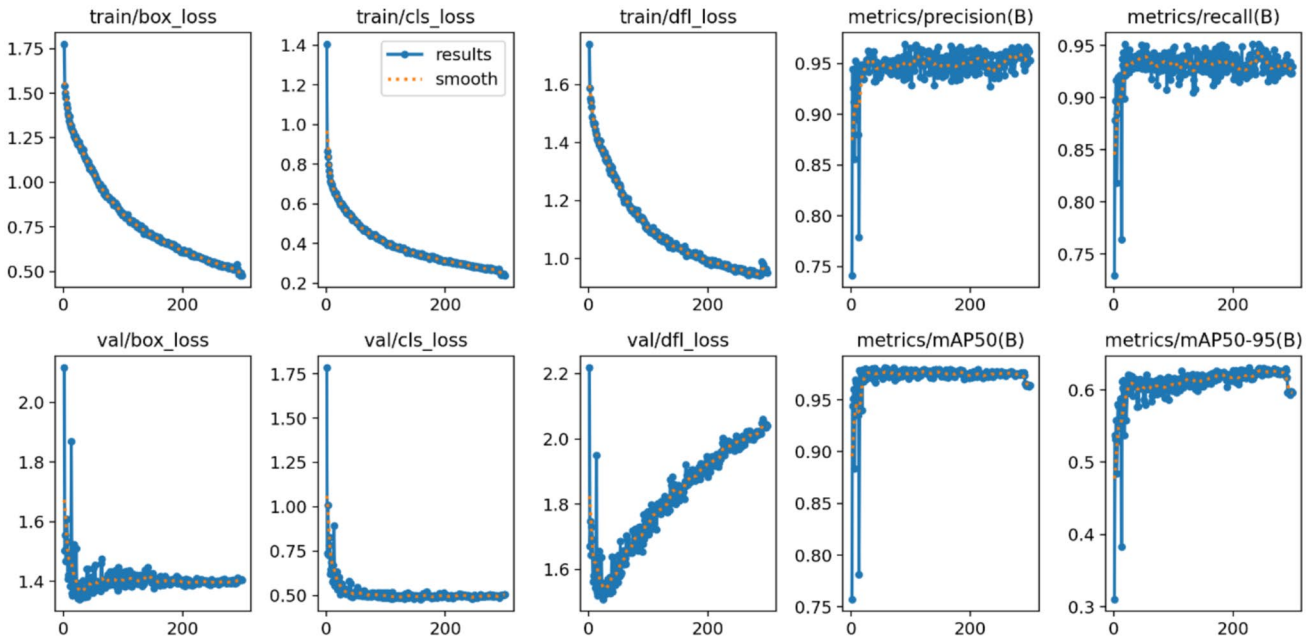


Fig. 8 Training and validation loss curves, precision-recall metrics, and mAP scores for YOLOv8n object detection model over training iterations

Table 4 Metrics details between intel i7-8665U and RPi + NCS1

Metrics	intel Core i7-8665U CPU @ 1.90 GHz with 16 GB RAM	RPi + NCS1
FPS	10	2.4
Power consumption	170 W	1.6 W

4 Discussion

This research shows that the YOLOv8n model is useful for detecting oil palm trees, with excellent accuracy across several metrics. The performance of model, with 95.6%

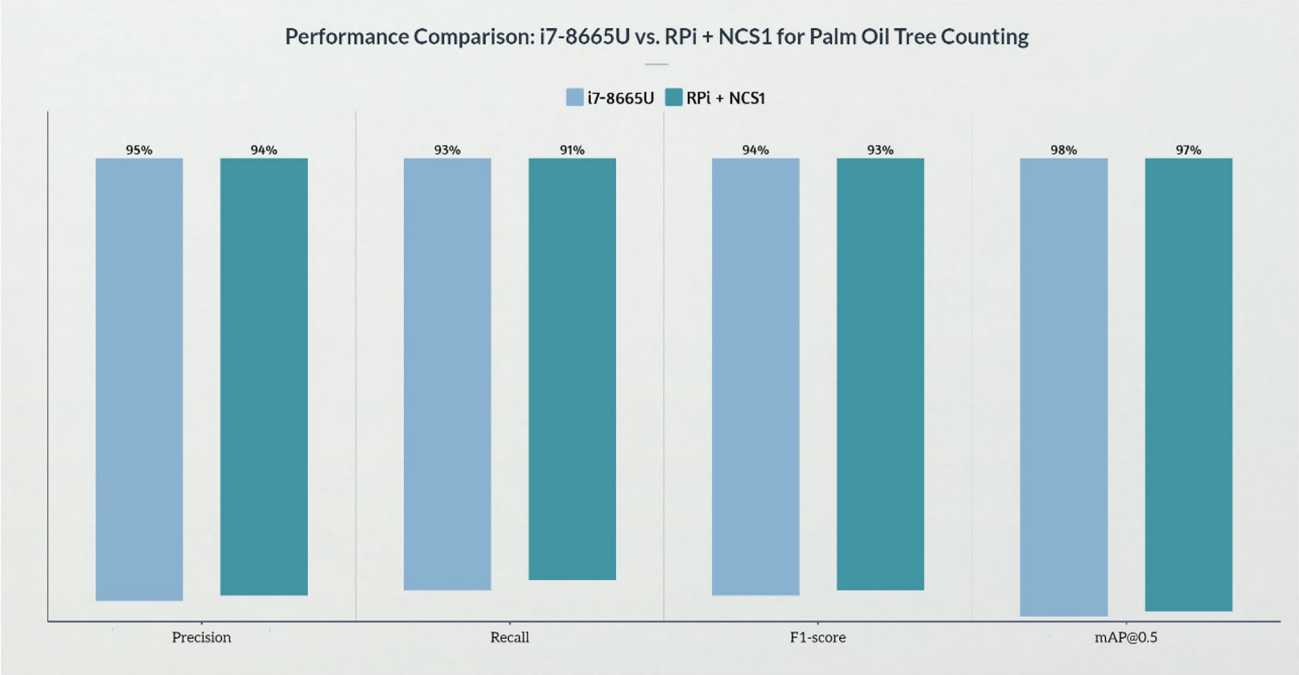


Fig. 9 Performance comparison between i7-8665U and RPi + NCS1

precision, 93.3% recall, and 98% mAP@0.5, demonstrates its strong capability to identify and locate oil palm trees in drone-captured imagery. Moreover, the successful deployment of model on a Raspberry Pi 4B with an NCS1 accelerator, which achieved real-time detection with minimal accuracy loss, shows a significant step towards practical application of deep learning in precision agriculture.

To evaluate the findings of this study within the wider context of oil palm tree detection research, the results were collected and compared with the findings of recent research using various YOLO architectures. Table 5 highlights these comparisons, including precision, recall, F1-score, and mAP@0.5 metrics as available. The YOLOv8n model employed in this research performs effectively across all metrics, particularly mAP@0.5. While several studies have shown various levels of performance, these can be attributed to differences in datasets, image resolutions, and specific implementation details. Wibowo et al. [46] reported high precision and recall values for their models, showing the potential benefits of YOLO architecture in this sector.

Shaikh et al. (2024) used the YOLOv8m model to achieve results essentially equivalent to those in this study, including a comparable mAP@0.5 [47]. Although their findings are significant, the current study provides more information about the performance of lighter architectures, especially with respect to edge computing applications.

The model's performance for oil palm tree detection can be seen by its strong mAP@0.5 and balanced performance across precision, recall, and F1-score. Furthermore, the effective implementation on resource-limited devices such as the Raspberry Pi highlights the potential for real-world applications of this technology in precision agriculture.

The successful deployment of the YOLOv8 nano model on edge devices, such as the Raspberry Pi 4B, demonstrates a flexible framework that can be adapted for detecting various plantation crops beyond oil palm. For crops with similar canopy structures like coconut, rubber, or mango, the existing methodology can be applied with minimal changes by retraining the model using crop-specific datasets. In cases where crops have different structural characteristics, such as the smaller and denser arrangements found in tea or coffee plantations, adjustments to detection thresholds and anchor box configurations may be necessary to accommodate variations in scale and density. The edge computing approach, utilizing devices like the Raspberry Pi and Intel® NCS1, remains particularly valuable for all plantation types, especially in remote areas with limited connectivity. Moreover, the low power consumption of this setup, peaking at approximately 1.6W, enables the possibility of solar-powered deployments, facilitating extended field operations. The modular architecture of the system also allows for seamless integration with existing agricultural management systems, providing opportunities to customize the model for detecting not only different types of plantation crops but also their growth stages or health conditions, given appropriate training data.

Table 5 Comparison of present study with previous published Literature

No	Research Title	Dataset	Yolo Versions	Precision (%)	Recall (%)	F-1 Score (%)	mAP@0.5 (%)	References
1	Deep neural network-based date palm tree detection in drone imagery	UAV—125 images	YOLOv3 YOLOv5m	95 91	85 92	88.31 -	- 92.34	[23]
2	Automatic Detection of Oil Palm Growth Rate Status with YOLOv5	UAV—2303 images	YOLOv5l	96	98	97	90	[48]
3	Recognition and counting of oil palm tree with deep learning using satellite image	Citra satellite images	YOLOv5	98.9	86.6	-	-	[49]
4	Large-Scale Oil Palm Trees Detection from High-Resolution Remote Sensing Images Using Deep Learning	UAV—401 images	YOLOv3 YOLOv4 YOLOv5m	99.96 99.86 99.69	94.75 95.72 90.62	97.28 97.74 94.94	- - -	[46]
5	Enhancing sustainability in the production of palm oil: creative monitoring methods using YOLOv7 and YOLOv8 for effective plantation management	UAV—9667 images	YOLOv7-D6 YOLOv8m	95 94.9	94 95.2	94 95	98 98.6	[47]
6	Efficient Real-Time Palm Oil Tree Detection and Counting Using YOLOv8 Deployed on Edge Devices	UAV—1158 images	YOLOv8n	95.6	93.3	94	98	Present Research

5 Conclusion

This study demonstrates the successful application of YOLOv8 object detection models for oil palm tree identification and counting in plantation management, focusing on edge device deployment. The research evaluated various YOLOv8 architectures, with the YOLOv8 nano model demonstrating superior performance, achieving 95.6% precision, 93.3% recall, and 98% mAP@0.5. The successful deployment of the optimized model on a RPi 4B with an Intel NCS1 accelerator represents a significant advancement in edge computing for precision agriculture.

Post-deployment performance analysis revealed an average inference time of 0.4 s per image (2.4 FPS) with minimal accuracy loss (97% mAP@0.5). The system's low power consumption (1.6W peak) and memory usage (450 MB RAM) underscore its suitability for field conditions. These results highlight the potential for compact, efficient deep-learning models on resource-constrained devices for real-time plantation monitoring. The proposed system offers a portable, energy-efficient oil palm tree detection and counting solution, potentially revolutionizing plantation management practices. By enabling data-driven decision-making in remote or challenging environments, this technology could significantly improve resource allocation, yield forecasting, and overall plantation efficiency.

5.1 Limitations and future research directions

Despite the promising results, this study presents several limitations that warrant consideration. The model was trained and evaluated on data collected from a specific geographic region under relatively controlled environmental conditions, which may limit its generalizability to diverse plantation environments with different terrain characteristics, climate conditions, or palm varieties. The drone imagery was captured only during specific hours (9:00 a.m. to 1:00 p.m.) and seasons (January–February 2024), potentially not accounting for variations in lighting, seasonal changes, or adverse weather conditions that could affect

detection accuracy. While the YOLOv8n model performs well for tree detection and counting, its lightweight architecture may present challenges when scaling to more complex tasks requiring finer-grained feature extraction. Additionally, the current implementation focuses solely on tree detection and counting without addressing other critical aspects of plantation management such as disease identification or yield estimation.

Building on these limitations, several specific directions for future research emerge:

Multi-condition robustness: Future studies should evaluate and enhance model performance across diverse environmental conditions including different lighting situations, adverse weather (rain, fog), seasonal variations, and geographical regions to improve robustness and generalizability.

Disease and stress detection: Extending the current architecture to identify early signs of common palm diseases such as Ganoderma basal stem rot, nutrient deficiencies, or drought stress by incorporating multispectral imaging capabilities that can detect subtle changes in vegetation indices before visible symptoms appear.

Yield estimation integration: Developing complementary models that can detect and count fruit bunches, estimate fruit ripeness stages, and predict yield potential based on tree crown characteristics, potentially using a dual-model approach where the first model detects trees and the second analyzes detected trees for yield factors.

Authors' contributions Mohammad Farhan: Conceptualization; Investigation; Data collection; Experiments; Writing-original draft; Writing—review & editing. Mohammad Nishat Akhtar: Conceptualization; Investigation; Funding Acquisition; Project administration; supervision, Writing—review & editing. Elmi Abu Bakar: Conceptualization; Investigation; Visualization; Writing—review & editing.

Funding No funding available.

Availability of data and materials The data supporting this article is included in the article itself; no supporting information is included. Additionally, the raw data files for this study can be made available by contacting the corresponding author.

Declarations

Competing interests The authors have no competing interests.

References

1. Razak MIM, Abas NM, Yaacob NJA, Rodzi S et al (2015) An overview of the primary sector in Malaysia. Int J Econs Mgmt 3:1–13. https://doi.org/10.3390/su15108283
2. Statistics S (2023) Malaysian economy in figures. Retrieved from https://ekonomi.gov.my/ms/statistik-sosioekonomi/malaysian-economy-Fig.s
3. Frank NEG, Albert MME, Laverdure D, Paul K (2011) Assessment of the quality of crude palm oil from smallholders in Cameroon. J Stored Prod Postharvest Res 2(3):52–58
4. Nations FAO (2023) Food and Agriculture Data. Retrieved from https://www.fao.org/faostat/en/#home
5. (DOSM) DSM (2023) Domestic palm oil production of Malaysia 2023 Retrieved from https://www.dosm.gov.my/portal-main/search?theme=&keyword=agriculture
6. Yohansyah WM, Lubis I (2014) Analisis produktivitas Kelapa Sawit (Elaeis guineensis Jacq.) di PT. Perdana Inti Sawit Perkasa I. Riau. Buletin Agrohorti 2(1):125–131. https://doi.org/10.29244/agrob.2.1.125-131
7. Whelan B, Taylor J (2013) Precision agriculture for grain production systems: Csiro Publishing
8. Bramley RG (2009) Lessons from nearly 20 years of Precision Agriculture research, development, and adoption as a guide to its appropriate application. Crop Pasture Sci 60(3):197–217. https://doi.org/10.1071/CP08304
9. Akhtar MN, Ansari E, Alhady SSN, Abu Bakar E (2023) Leveraging on advanced remote sensing-and artificial intelligence-based technologies to manage palm oil plantation for current global scenario: a review. Agriculture 13(2):504. https://doi.org/10.3390/agriculture13020504
10. Rokhmana CA (2015) The potential of UAV-based remote sensing for supporting precision agriculture in Indonesia. Procedia Environ Sci 24:245–253. https://doi.org/10.1016/j.proenv.2015.03.032
11. Adam F, Mönks M, Esch T, Datcu M (2018) Cloud removal in high resolution multispectral satellite imagery: comparing three approaches. Paper Present Proc. https://doi.org/10.3390/ecrs-2-05166
12. Colefax AP, Butcher PA, Kelaher BP (2018) The potential for unmanned aerial vehicles (UAVs) to conduct marine fauna surveys in place of manned aircraft. ICES J Mar Sci 75(1):1–8. https://doi.org/10.1093/icesjms/fsx100
13. Singh A, Patil D, Omkar S (2018) Eye in the sky: real-time drone surveillance system (dss) for violent individuals identification using scatternet hybrid deep learning network. Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition workshops.
14. Saqib M, Khan SD, Sharma N, Scully-Power P, Butcher P, Colefax A, Blumenstein M (2018) Real-time drone surveillance and population estimation of marine animals from aerial imagery. Paper presented at the 2018 Int Conf Image Vis Comput N Z (IVCNZ). https://doi.org/10.1109/IVCNZ.2018.8634661.
15. Mishra B, Garg D, Narang P, Mishra V (2020) Drone-surveillance for search and rescue in natural disaster. Comput Commun 156:1–10. https://doi.org/10.1016/j.comcom.2020.03.012
16. Oré G, Alcântara MS, Góes JA, Oliveira LP, Yepes J, Teruel B, Luebeck D (2020) Crop growth monitoring with drone-borne DInSAR. Remote Sens 12(4):615. https://doi.org/10.3390/rs12040615
17. Yang J, Kang Z, Cheng S, Yang Z, Akwensi PH (2020) An individual tree segmentation method based on watershed algorithm and three-dimensional spatial distribution analysis from airborne LiDAR point clouds. IEEE J Sel Top Appl Earth Obs Remote Sens 13:1055–1067. https://doi.org/10.1109/JSTARS.2020.2979369

18. Yun T, Jiang K, Li G, Eichhorn MP, Fan J, Liu F, Cao L (2021) Individual tree crown segmentation from airborne LiDAR data using a novel Gaussian filter and energy function minimization-based approach. Remote Sens Environ 256:112307. https://doi.org/10.1016/j.rse.2021.112307
19. Harikumar A, D'Odorico P, Ensminger I (2020) A fuzzy approach to individual tree crown delineation in uav based photogrammetric multispectral data. Paper presented at the IGARSS 2020–2020 Int Geosci Remote Sens Symp https://doi.org/10.1109/IGARSS39084.2020.9324303.
20. Dalla Corte AP, Souza DV, Rex FE, Sanquetta CR, Mohan X et al (2020) Forest inventory with high-density UAV-Lidar: machine learning approaches for predicting individual tree attributes. Comput Electron Agric 179:105815. https://doi.org/10.1016/j.compag.2020.105815
21. Aliandra SR, Prasvita DS (2022) Application of median filter method for classification of oil palm tree on LiDAR images. Paper presented at the 2022 international conference on informatics, multimedia, cyber and information system (ICIMCIS). https://doi.org/10.1109/ICIMCIS56303.2022.10017880.
22. Puliti S, Astrup R (2022) Automatic detection of snow breakage at single tree level using YOLOv5 applied to UAV imagery. Int J Appl Earth Obs Geoinf 112:102946. https://doi.org/10.1016/j.jag.2022.102946
23. Jintasuttisak T, Edirisinghe E, Elbattay A (2022) Deep neural network based date palm tree detection in drone imagery. Comput Electr Agric 192:106560. https://doi.org/10.1016/j.compag.2021.106560
24. Vasavi S, Likhitha AL, Premchand VS, Yasaswini J (2024) Object classification by effective segmentation of tree canopy using U-Net model. J Adv Inform Technol. https://doi.org/10.12720/jait.15.3.422-434
25. Azam B, Khan MJ, Bhatti FA, Maud ARM, Hussain SF, Hashmi AJ, Khurshid K (2022) Aircraft detection in satellite imagery using deep learning-based object detectors. Microprocess Microsyst 94:104630. https://doi.org/10.1016/j.micpro.2022.104630
26. Babulal KS, Das AK (2022) Deep learning-based object detection: an investigation. Paper presented at the futuristic trends in networks and computing technologies: select proceedings of fourth international conference on FTNCT. https://doi.org/10.1007/978-981-19-5037-7_50.
27. Zhao ZQ, Zheng P, Xu S, Wu X (2019) Object detection with deep learning: A review. IEEE Trans Neural Netw Learn Syst 30(11):3212–3232. https://doi.org/10.1109/TNNLS.2018.2876865
28. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition
29. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. Paper presented at the Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit
30. Ren S, He K, Girshick R, Sun J (2016) Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 39(6):1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031
31. Girshick R (2015) Fast r-cnn. Paper presented at the Proc IEEE Int Conf Comput Vis
32. Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. Adv Neural Inf Process Syst. https://doi.org/10.48550/arXiv.1506.01497
33. Redmon J, Farhadi A (2017) YOLO9000: better, faster, stronger. Paper presented at the Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit https://doi.org/10.48550/arXiv.1612.08242.
34. Bochkovskiy A, Wang CY, Liao HYM (2004) Yolov4: Optimal speed and accuracy of object detection. Conf Comput Vis Pattern Recognit. https://doi.org/10.48550/arXiv.2004.10934
35. Joseph N, Jacob S (2020) YOLOv5 is here: state-of-the-art object detection at 140 FPS In
36. Hussain M (2023) YOLO-v1 to YOLO-v8, the rise of YOLO and its complementary nature toward digital manufacturing and industrial defect detection. Mach 11(7):677. https://doi.org/10.3390/machines11070677
37. Cao L, Zheng X, Fang L (2023) The semantic segmentation of standing tree images based on the Yolo V7 deep learning algorithm. Electronics 12(4):929. https://doi.org/10.3390/electronics12040929
38. Jocher G, Chaurasia A, Qiu J (2023) YOLO V8. Retrieved from https://github.com/ultralytics/ultralytics
39. Yao S, Chen Y, Tian X, Jiang R, Ma S (2020) An improved algorithm for detecting pneumonia based on YOLOv3. Appl Sci 10(5):1818. https://doi.org/10.3390/app10051818
40. Ma H, Liu Y, Ren Y, Yu J (2019) Detection of collapsed buildings in post-earthquake remote sensing images based on the improved YOLOv3. Remote Sens 12(1):44. https://doi.org/10.3390/rs12010044
41. Zhang H, Qin L, Li J, Guo Y, Zhou Y, Zhang J, Xu Z (2020) Real-time detection method for small traffic signs based on Yolov3. Ieee Access 8:64145–64156. https://doi.org/10.1109/ACCESS.2020.2984554
42. Liu G, Nouaze JC, Touko Mbouembe PL, Kim JH (2020) YOLO-tomato: A robust algorithm for tomato detection based on YOLOv3. Sens 20(7):2145. https://doi.org/10.3390/s20072145
43. Wang CY, Liao HYM, Wu YH, Chen PY, Hsieh et al (2020) *CSP-Net*: a new backbone that can enhance learning capability of CNN. Paper presented at the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. https://doi.org/10.1109/CVPRW50498.2020.00203.
44. Liu S, Qi L, Qin H, Shi J, Jia J (2018) Path aggregation network for instance segmentation. Paper presented at the Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit. https://doi.org/10.48550/arXiv.1803.01534.
45. Zagitov A, Chebotareva EV, Toschev AS, Magid E (2024) Comparative analysis of neural network models performance on low-power devices for a real-time object detection task. Компьютерная оптика 48(2):242–252. https://doi.org/10.18287/2412-6179-CO-1343
46. Wibowo H, Sitanggang IS, Mushthofa M, Adrianto HA (2022) Large-scale oil palm trees detection from high-resolution remote sensing images using deep learning. Big Data Cogn Comput 6(3):89. https://doi.org/10.3390/bdcc6030089
47. Shaikh IM, Akhtar MN, Aabid A, Ahmed OS (2024) Enhancing sustainability in the production of palm oil: creative monitoring methods using YOLOv7 and YOLOv8 for effective plantation management. Biotechnol Rep 44:e00853. https://doi.org/10.1016/j.btre.2024.e00853
48. Prasvita DS, Chahyati D, Arymurthy AM (2023) Automatic detection of oil palm growth rate status with YOLOv5. Int J Adv Comput Sci Appl. https://doi.org/10.14569/IJACSA.2023.0140361
49. Nurhabib I, Seminar K (2022) Recognition and counting of oil palm tree with deep learning using satellite image. Paper presented at the IOP Conference Series: Environ Earth Sci. https://doi.org/10.1088/1755-1315/974/1/012058.

Authors and Affiliations

Mohammad Farhan[1] · Mohammad Nishat Akhtar[1] · Elmi Abu Bakar[1]

✉ Mohammad Nishat Akhtar
nishat@usm.my

Mohammad Farhan
md_farhan@student.usm.my

Elmi Abu Bakar
meelmi@usm.my

[1] School of Aerospace Engineering, Universiti Sains Malaysia, Engineering Campus, Seri Ampangan, 14300 Nibong Tebal, Seberang Perai Selatan, Pulau Penang, Malaysia