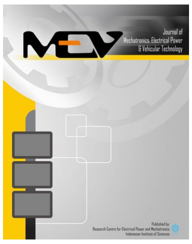


Journal of Mechatronics, Electrical Power, and Vehicular Technology 16 (2025) 305-317

Journal of Mechatronics, Electrical Power, and Vehicular Technology

CrossMark

e-ISSN: 2088-6985
p-ISSN: 2087-3379

mev.brin.go.id

Real-time FFB ripeness detection using IoT-enabled YOLOv8n on Raspberry Pi 4 edge devices for precision agriculture

Nurul Hazlina Noordin *, Rosdiyana Samad, Abdul Haikal Abdul Malek

UMPSA STEM LAB, Faculty of Electrical and Electronics Engineering Technology, Universiti Malaysia Pahang Al-Sultan Abdullah Pekan, Pahang, 26600, Malaysia

Abstract

This paper presents the development of an edge device for cost-effective implementation in agricultural environments. Experimental evaluations demonstrate accuracy and real-time performance, showcasing its potential for adoption in the industry. The proposed system provides a reliable tool for timely and accurate monitoring of fresh fruit bunch (FFB) ripeness, facilitating optimized crop management practices. The system employs the YOLOv8n model, renowned for its efficiency in real-time object detection tasks, and is adapted to run on the resource-constrained Raspberry Pi 4. To ensure seamless operation on edge devices, model optimization techniques such as quantization and hardware acceleration are implemented, enabling rapid decision-making based on live data feeds. A dataset comprising 4,194 annotated FFB images was utilized, with a [3,681:348:165] training-validation-test split. Performance evaluation demonstrated an average precision of 0.898 and a mean average precision (mAP) of 0.952. The system potentially enhances yield quality and sustainability while supporting data-driven decision-making in precision agriculture.

Keywords: edge devices; fresh fruit bunch ripeness detection; precision agriculture; Raspberry Pi 4; YOLOv8n object detection.

I. Introduction

Detecting palm fruit ripeness is crucial for optimizing oil quality and yield. Traditional manual methods are labour-intensive and prone to errors. Integrating Internet of Things (IoT) and edge computing with deep learning, such as the You Only Look Once version 8 (YOLOv8n) object detection model on a Raspberry Pi, offers a solutions that enables real-time, accurate ripeness detection, overcoming computational limitations, and enhancing efficiency [1][2][3][4].

Deep learning utilizes neural networks with many layers to model complex patterns in large data sets [5]. It has revolutionized fields such as image recognition, natural language processing, and autonomous driving [6]. Deep learning models, particularly convolutional neural networks (CNNs), are highly effective in image processing tasks, including detecting fruit ripeness by learning and recognizing intricate features from images [7].

In palm fruit ripeness detection, deep learning models are trained on datasets of palm fruit images at various ripeness stages. These models identify visual cues like colour, texture, and shape to predict fruit ripeness [8][9]. CNNs, designed for processing grid-like data, are particularly suitable for this task due to their ability to learn hierarchical representations from visual data [8].

* Corresponding Author. hazlina@umpsa.edu.my (N.H. Noordin)

https://doi.org/10.55981/j.mev.2025.1220

Received 29 July 2025; revised 30 November 2025; accepted 20 December 2025; available online 30 December 2025; published 31 December 2025

How to Cite: N. H. Noordin *et al.*, "Real-time FFB ripeness detection using IoT-enabled YOLOv8n on Raspberry Pi 4 edge devices for precision agriculture," *Journal of Mechatronics, Electrical Power, and Vehicular Technology*, vol. 16, no. 2, pp. 305-317, Dec. 2025.

Previous studies demonstrate the efficacy of CNNs in this domain. For instance, a study comparing DenseNet and AlexNet models found DenseNet outperformed AlexNet by 8.5 % in accuracy and 8 % in F1 score when classifying palm fruit ripeness levels [8].

Another approach combined deep neural networks (DNNs) and machine learning methods to determine ripeness stages in fruits like strawberries and tomatoes, achieving excellent classification performance [9]. The study implemented a deep learning-based method to classify oil palm fresh fruit bunches (FFB) into raw, ripe, and rotten categories.

In [10], a machine learning (ML) model is proposed utilizing a dataset of 400 images. The CNN model achieved 92 % classification accuracy on test data, although performance dropped to 76 % during model testing, likely due to variations in background color and shape. Another possible reason for the decline is overfitting, where the model learned the training data too well but failed to generalize to unseen test data. This issue is primarily attributed to the small dataset size, with only 400 images, which may have caused the model to memorize patterns instead of learning meaningful features.

The importance of ripeness classification in preventing overripe oil palm FFBs is emphasized as overripe fruits lead to higher free-fatty acid levels, reducing the quality of extracted oil [11][12]. The work in [11] focuses on the application of deep learning for object detection and classification, particularly using CNNs. The study proposes EfficientDet-Lite2, a specialized CNN model for oil palm FFB ripeness detection, which is optimized for real-time applications in plantation environments. EfficientDet-Lite2 utilizes a compound scaling approach, adopting the D2 configuration for its input resolution. This configuration enhances the input resolution to 448×448 pixels, enabling the model to capture finer details in images. In applications such as ripeness detection of palm FFBs, the higher resolution improves the differentiation of color gradients and texture variations, resulting in more accurate classification. The model integrates a bi-directional feature pyramid network (Bi-FPN) with five layers and three box classes per layer. The proposed model achieved an accuracy of 84 % when tested in Indonesian plantations, demonstrating its effectiveness in real-world scenarios. The potential of leveraging EfficientDet-Lite2 for accurate and efficient ripeness classification could significantly enhance harvesting decisions and overall palm oil quality. The Bi-FPN structure, while efficient, may struggle with highly occluded or clustered FFBs, where overlapping objects and varying lighting conditions reduce precision detection. This could lead to misclassifications in real-world plantation settings, particularly in dense foliage or shadowed areas, affecting its reliability in practical harvesting applications.

YOLO deep learning model has gained popularity for its speed and accuracy in object detection, treats object detection as a single regression problem, mapping pixels to bounding box coordinates and class probabilities. This enables real-time processing, essential for applications like palm fruit ripeness detection in field conditions [13][14]. The evolution and key feature of YOLO detection models are shown in Table 1 [15][16][17][18][19][20][21].

Recent studies have demonstrated the effectiveness of YOLO-based models in palm oil ripeness detection. The work in [22] highlighted YOLOv3's real-time detection capabilities, while [23] emphasized its role in improving classification accuracy. The study in [24] showcased YOLO's robustness using video datasets and [13] found that YOLOv3 is superior to ResNet50 in both accuracy and speed. Further advancements include the development of a YOLOv4-based system with 87.9 % mAP in [14] and the exploration of YOLOv8, achieving high precision and recall in ripeness prediction. These findings underscore YOLO’s reliability for real-time agricultural applications.

Implementation on Raspberry Pi is an alternative approach to reduce costs and energy consumption in embedded system for object detection classification systems. The work in [25] demonstrate that YOLO-based object detection can be successfully deployed on resource-constrained embedded platforms, achieving near real-time performance suitable for monitoring and detection applications. This highlights the feasibility of low-power, cost-effective, edge-based vision systems as an alternative to GPU-dependent solutions [25].

Further evidence supporting efficient deep learning deployment on resource-constrained devices is provided in [26]. The study shows that transfer learning enables high-accuracy deep learning applications on Raspberry Pi despite limited computational resources. By comparing InceptionV3 and the lightweight MobileNetV2 architecture, the results indicate that MobileNetV2 achieves superior accuracy and F1 score while maintaining computational efficiency, making it more suitable for real-time edge applications. These findings emphasize the importance of lightweight architecture and model reuse strategies in achieving reliable performance on low-power embedded platforms.

YOLOBench systematically evaluates YOLO-based detectors on embedded platforms and shows that multiple YOLO variants achieve competitive accuracy–

Table 1.
Evolution and general key features of YOLO object detection models.

Version	Year	Key features	Performance	Architecture & techniques
YOLOv1	2015 – 2016	- Original YOLO model	- Real-time detection - Lower accuracy	- Single convolutional network predicting bounding boxes and class probs
YOLOv2	2017	- Batch normalization - Anchor boxes - Multi-scale training	- Improved accuracy and recall	- Predefined anchor boxes - Training on different image sizes
YOLOv3	2018	- Deeper network (Darknet-53) - Feature pyramid network (FPN)	- Better for small object detection	- 53-layer network - Multi-scale predictions
YOLOv4	2020	- CSPDarknet53 - Mish activation - Spatial pyramid pooing (SPP) - PANet	- Enhanced accuracy and speed - Bag of freebies and specials	- Advanced modules and data augmentation techniques
YOLOv5	2020	- Model variants (s, m, l, x) - Integration with modern tools	- Easier to use and accessible	- Variants balancing speed and accuracy
YOLOv6	2021	- Improved efficiency	- Further performance improvements	- Continued network and training strategy advancements
YOLOv7	2022	- Enhanced efficient layer aggregation networks (E-ELAN)	- Best balance of speed and accuracy	- New architectural changes
YOLOv8	2023	- Latest advancements - Further optimizations	- Highest accuracy and speed	- Continued improvements in architecture and training techniques

latency trade-offs on ARM CPUs, including Raspberry Pi. These findings support the use of lightweight YOLO models as practical alternatives to computationally intensive state-of-the-art detectors for edge deployment [27]. In [28] Deep Q-Learning was combined with YOLOv3 for real-time object detection and recognition on a Raspberry Pi, with accuracy enhanced through data augmentation. This implementation demonstrated adaptability to resource constraints, offering cost-effective, energy-efficient solutions for autonomous vehicles.

Deploying YOLO-based deep learning models in embedded systems for IoT applications, including palm fruit ripeness detection, presents several challenges, particularly related to hardware limitations, computational constraints, power consumption, and real-time processing requirements. The YOLOv8 model, with its millions of parameters, requires GPU acceleration, which most IoT devices lack. As a complex CNN-based model, it is computationally intensive and difficult to run on resource-constrained hardware. To address this challenge, a more efficient alternative is to use lightweight versions like YOLOv8n, which are optimized for edge deployment while maintaining reasonable accuracy. Another alternative is to implement edge devices, where the model runs locally on the device instead of relying on cloud servers, and to deploy hybrid cloud-edge models that transmit only essential data (e.g., detection results) rather than raw images, optimizing both performance and bandwidth efficiency. The subsequent section will explore an IoT-enabled palm fruit ripeness detection system on edge devices using YOLOv8n on a Raspberry Pi 4 Model B for real-time, precise detection, overcoming challenges associated with deploying advanced models on resource-constrained devices.

This work proposes an IoT-enabled palm fruit ripeness detection system using YOLOv8n on a Raspberry Pi 4 Model B. The system captures real-time images of oil palm fruits and classifies their ripeness stages, offering a scalable and cost-effective solution for precision agriculture. The novelty of this study lies in the adaptation and optimization of YOLOv8n for deployment on edge devices in tropical agricultural environments. The adaptation addresses real-world challenges such as environmental variations, model efficiency on low-power hardware, and real-time processing constraints. The study contributes to the field by demonstrating the feasibility of deploying high-accuracy deep learning models for palm fruit ripeness detection on edge devices, offering a practical solution for improving harvesting efficiency in the palm oil industry. The subsequent sections will discuss existing detection methods, detail the proposed system, and evaluate its performance in real-world conditions.

II. Materials and Methods

A. System architecture

The IoT-enabled palm fruit ripeness detection system is designed to enhance the efficiency and accuracy of palm oil fruit harvesting. This system leverages real-time visual inspection using advanced AI techniques to determine the ripeness of palm fruits, while also incorporating geolocation tracking for precise data logging and environmental monitoring. The system operates through two primary modes, which are camera mode for real-time detection and global positioning system (GPS) Mode for location tracking and data synchronization. A visual representation of the proposed system is shown in Figure 1.

B. Edge device setup

The core of the proposed system is an edge computing platform based on the Raspberry Pi 4 Model B, powered by the Broadcom BCM2711 system-on-chip (SoC). The BCM2711 integrates a quad-core ARM Cortex-A72 (64-bit) CPU operating at 1.5 GHz, which provides sufficient computational capability for running lightweight deep learning models on edge devices. The Raspberry Pi used in this study is equipped with 4 GB RAM, balancing performance and energy efficiency for real-time object detection tasks.

Image acquisition is performed using an 8-megapixel Raspberry Pi Camera Module V2, connected via the camera serial interface (CSI) interface to ensure low-latency data transfer. The system is powered using a 5 V, 3 A regulated power supply, suitable for both laboratory testing and portable field deployment using a power bank or battery pack. The detection and classification of the fruit's ripeness are carried out using the YOLOv8n deep learning model, which runs in real-time on the device. The hardware setup also includes a fan for thermal management, with the RPi4 monitoring its temperature and regulating the fan speed, accordingly, forming a closed-loop system.

The system integrates several sensors to support its functionality, namely camera module, GPS module, and BME 280 sensor. The camera module is 8MP and captures images of palm fruits for visual inspection and ripeness detection. The YOLOv8n model processes these images in real-time to classify the ripeness of the fruits. The GPS module (Neo 6M) is used for geolocation tracking, the GPS module logs the device's coordinates whenever a button is pressed in GPS Mode. This ensures that the location of each fruit detection event is accurately recorded. This BME280 sensor is used for environmental monitoring, providing data on temperature, humidity, and atmospheric pressure. These parameters are crucial for analyzing the conditions under which the palm fruits are growing and can impact the ripeness detection accuracy.

C. Communication protocols

Data transmission in the proposed system is managed to cope with the challenges of intermittent connectivity in plantation environments. The primary communication protocols include long range wide area network (LoRaWAN) communication, IoT gateway to cloud, offline data logging, and data synchronization.

LoRaWAN is employed for transmitting data from the handheld device, which includes the RPi4, camera, BME280 sensor, and GPS module, to an IoT gateway. LoRaWAN is chosen for its low power consumption and long-range communication capabilities, making it ideal for large plantation areas. The IoT gateway receives data from the handheld device via LoRaWAN and then connects to a cloud database. This setup ensures that data collected in the field is reliably transmitted to the cloud for further processing and storage.

In scenarios where immediate transmission is not possible, offline data logging mode is activated, where data are logged locally on the device in JavaScript object notation (JSON) format. This approach ensures that no

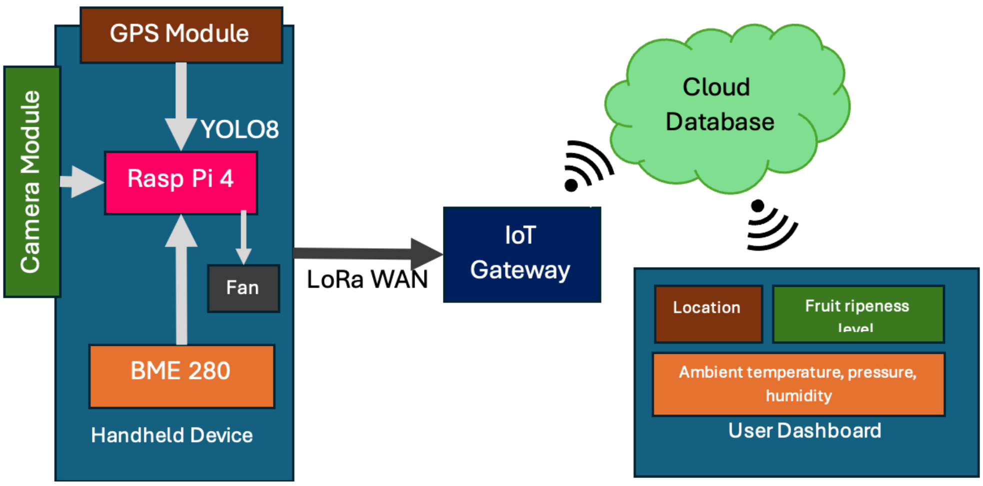


Figure 1. Block diagram of the proposed IoT-enabled palm fruit ripeness detection system.

data is lost due to connectivity issues. Once a stable connection is available, Data synchronization mode is activated, where the system uploads the logged data to the cloud database. The data can then be accessed and visualized using a dashboard.

D. Ripeness detection using YOLOv8n

The YOLOv8 implementation in the proposed system leverages pre-trained YOLOv8n models that are fine-tuned on a custom dataset of oil palm fruit images categorized by ripeness stages. Roboflow was used to annotate each image for palm oil fruit ripeness classification. Quality training data is crucial for accurate supervised deep learning models. Data augmentation techniques were applied to enhance model performance by creating variations in images through flipping, rotating, adjusting exposure, and brightness [29][30]. These methods aimed to expand the dataset, improve model generalization, and prevent overfitting. This approach ensures reliable classification of palm oil fruit ripeness, benefiting agricultural management practices.

The process begins with data collection and annotation, where a diverse set of images of oil palm fruits at various ripeness stages is collected and labelled (*kurang masak* – unripe, *masak* – ripe, *terlalu masak* – overripe) to create an accurate training dataset. This annotated dataset is then used to fine-tune the pre-trained YOLOv8n model, employing techniques such as data augmentation to enhance the model's robustness and improve its performance. An illustration of the model training and testing workflow is shown in Figure 2.

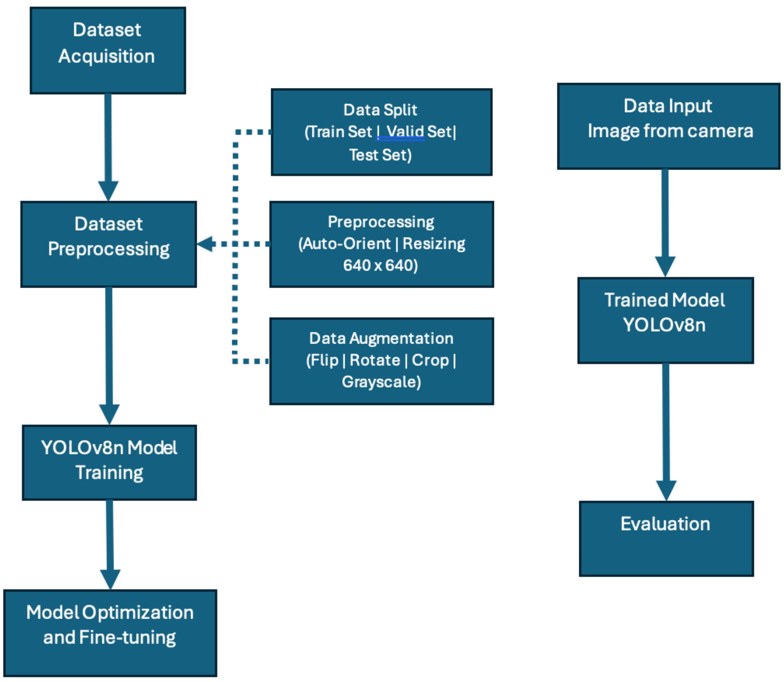
Table 2.
Dataset split and number of images.

Dataset Split	Percentage	Number of images
Training Set	88 %	3681
Validation Set	8 %	348
Test Set	4 %	165

The dataset was split into training (88 %), validation (8 %), and test sets (4 %), resulting in 3681, 348, and 165 images respectively, as shown in Table 2. Each image underwent critical preprocessing steps to standardize and enhance model robustness. Table 3 lists the preprocessing steps which incluce auto-orientation, 3024×4032 resizing to 640×640 pixels, and augmentation techniques such as horizontal flipping, 90° rotations (clockwise, counter-clockwise, and upside down), random cropping (0 % to 20 % zoom), and grayscale conversion for 15 % of images. These steps diversified the training data, improving the model's ability to generalize across different inputs.

E. System operation and data acquisition scenarios

When the handheld device is powered on, it initializes the screen and GPS module, as well as launches the device's graphical user interface (GUI).



(a) (b)

Figure 2. Workflow of YOLOv8n on IoT-enabled palm fruit ripeness detection system: (a) model training; and (b) model testing.

Table 3.
Preprocessing steps description.

Step	Description
Auto-orient	Correct image orientation
Resize	Resize to 640x640 pixels
Flip	Horizontal flipping
Rotate	90° rotations (clockwise, counter-clockwise, upside down)
Crop	Random cropping (0 % min zoom, 20 % max zoom)
Grayscale	Apply to 15 % of images

The user interface presents mode selection and a Dashboard. Pressing button 1 or 2 activates camera or GPS mode, respectively. The health monitoring features, which include battery and sensor status, of the device operates within a closed-loop system, tracking the handheld device temperature and adjusting the fan speed accordingly. Data synchronization occurs when the GPS mode is active.

During field deployment, data acquisition was conducted under two distinct operational scenarios, namely FFB detection on the tree and FFB detection on the ground. In the scenario of FFB on the tree, the handheld device is used to capture images of FFBs that are still attached to the palm tree. The operator positions the camera toward the palm canopy at an appropriate distance to ensure sufficient visibility of the FFB. This setup represents pre-harvest inspection, enabling ripeness assessment directly on the tree to support harvesting decisions.

In the FFB Detection on the ground scenario, the handheld device is used to scan harvested FFBs placed on the ground. The camera is oriented downward toward the FFB, representing post-harvest verification and grading, where reduced occlusion and closer proximity allow detailed ripeness evaluation.

Upon activation, the camera mode initiates a live feed with vision AI. The system awaits the presence of a palm oil fruit within the camera frame. Once detected, the YOLOv8n palm oil fruit model processes the image, performs detection, and visualizes a bounding box around the fruit, indicating its ripeness classification. The system then waits for button input to log the ripeness classification. In the GPS mode, the device attempts to obtain a position fix from satellites, parsing raw data into latitude and longitude, and displaying this information on the screen. The system then waits for button input to log GPS data.

During both acquisition scenarios, environmental parameters are simultaneously recorded using the BME280 sensor, including ambient temperature, relative humidity, and atmospheric pressure, to provide contextual information for each detection event. The handheld device saves location, environmental data and classification results (results of ripeness level, location, temperature and humidity) in JSON files, which are later synchronised to an IoT gateway using LoRaWAN. The IoT gateway then uploads the data to a cloud database using the message queuing telemetry transport (MQTT) protocol. The MQTT broker on the cloud server receives and stores the data. The dashboard retrieves the data from the cloud database to display real-time updates on palm fruit ripeness and location. Additionally, the handheld device continuously checks for Wi-Fi connections to synchronize and send data to the cloud when available.

III. Results and Discussions

A. Visual analysis of palm fruit ripeness model evaluation indicators

The efficacy of the developed model is evaluated through several key metrics illustrated in Figure 3, which depict its performance on both the training and validation datasets. The graphs bounding box loss (train/box_loss, val/box_loss) show the accuracy of the model in predicting bounding boxes around palm oil fruits [31]. Lower values in these metrics indicate better alignment between predicted and ground truth bounding boxes, as depicted by the downward trend in the graphs. The metrics classification loss (train/cls_loss, val/cls_loss) assesses how accurately the model predicts the ripeness classification of palm oil fruits. A decrease in these values across epochs signifies improved classification performance, as visually represented by the decreasing trend in the corresponding graphs [32]. The metrics distribution fitting loss (train/dfl_loss, val/dfl_loss) indicate the model's ability to refine object boundaries, crucial for precise localization of diseased areas within palm oil fruits. A decline in these metrics demonstrates enhanced boundary detection, as reflected by the decreasing trend in the graphs over training epochs.

The metric precision and recall (metrics/precision(B), metrics/precision(M), metrics/recall(B), metrics/recall(M)) evaluate the

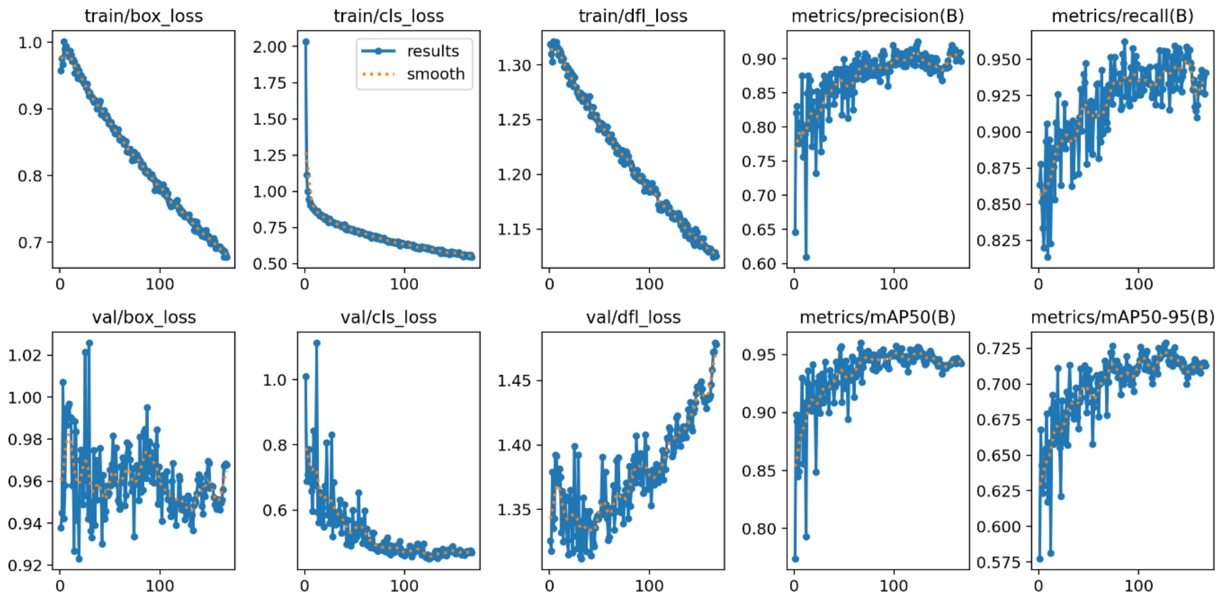


Figure 3. Model evaluation indicators.

model's precision in correctly identifying ripe, under ripe, and over ripe palm oil fruits (B for boxes and M for mass). Precision measures the accuracy of positive predictions, while recall assesses the model's ability to capture all relevant instances of ripe fruits [33]. The trends in precision and recall metrics over time, visible in the graphs, indicate the model's effectiveness in distinguishing between different ripeness levels of palm oil fruits.

The model demonstrates an average precision (AP) of 0.898, indicating its accuracy in predicting the correct class labels for the detected objects. Average precision is a crucial metric in object detection, reflecting the precision of the model across various thresholds. The model's average recall (AR) values are 0.945 for the "masak" (ripe), "kurang masak" (under ripe), and "terlalu masak" (overripe) classes. This high recall value signifies that the model effectively identifies a significant proportion of true positives for each class, ensuring that most of the correctly classifiable instances are detected. The model achieves a mean average precision (mAP) of 0.952. The mAP is a measure that averages the precision across all classes and detection thresholds, providing an overall performance evaluation of the model. A high mAP value indicates that the model performs consistently well across all categories of palm fruit ripeness, making it a reliable tool for real-world agricultural applications.

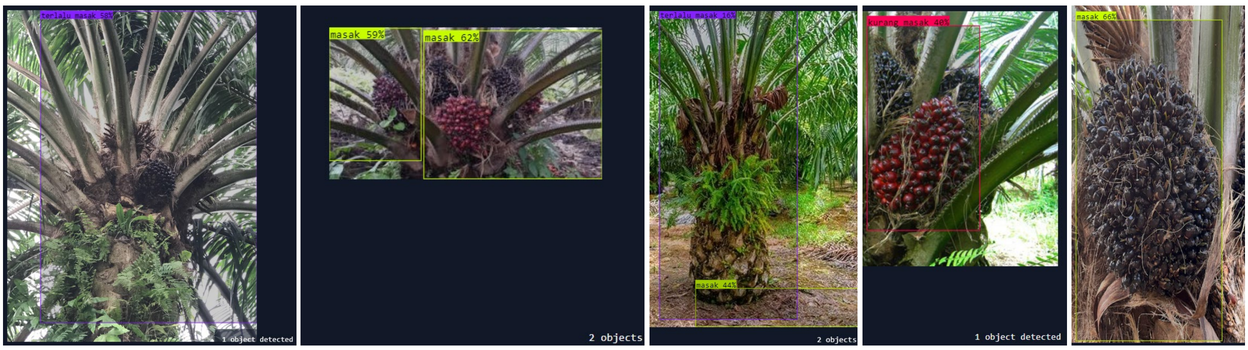
Human accuracy in visual ripeness detection for palm fruits typically ranges between 85-90 % in controlled conditions, varying with assessor experience and environmental factors. Compared to this benchmark, the YOLOv8n model's mAP of 0.952 demonstrates superior accuracy and reliability, especially in challenging conditions like poor lighting or high humidity [21]. Traditional methods, such as color thresholding, often falter under such conditions, whereas deep learning models, including CNNs, have achieved exceptional accuracy in similar tasks, reporting up to 99.89 % accuracy, with F-measure, precision, and recall values of 99.88 %, 99.90 %, and 99.85 %, respectively [34]. This highlights YOLOv8n's potential to enhance or replace manual assessments in ripeness detection, enabling precise and timely interventions in agricultural management.

B. Evaluation of the optimum model

The evaluation of the optimum model involved an onsite test of the IoT-enabled palm fruit ripeness detection system. The test was conducted with two groups, each scanning FFB both on the ground and on the trees. The system successfully detected and classified the fruits according to the trained clusters mentioned earlier. As shown in Figure 4, the detection and prediction results illustrate the system's capability to identify and classify palm fruits both as loose fruit on the ground and as fruit on palm trees. The figure highlights the successful application of the model in real-world conditions.

The detection and prediction results inferred the system's capability to identify and classify palm fruits both as loose fruit on the ground and as fruit on palm trees. For FFB on the ground, the system demonstrated high accuracy in detecting and classifying the fruits. The detection and classification process were straightforward, with the camera easily capturing clear images of the FFB. The model's performance in this

(a)



(b)

Figure 4. Detection and predictions on site: (a) loose fruit; and (b) fruit on palm tree.

scenario confirmed its reliability in identifying the ripeness levels of palm fruits at ground level.

Scanning FFB on trees presented additional challenges, particularly for tall trees. While the system was able to detect and classify fruits on trees up to an eye-level height of approximately six feet, the accuracy decreased for taller trees. This reduction in accuracy can be attributed to the difficulty in capturing clear images of the fruits, which are often obscured by foliage or situated at angles that are not ideal for the camera.

Despite these challenges, the model performed satisfactorily at eye-level height, accurately identifying and classifying the ripeness of the fruits. This indicates that while the system is effective for ground-level and lower tree-level scanning, improvements in image capture techniques or additional training data may be required to enhance performance for taller trees.

In conclusion, the onsite evaluation demonstrated that the IoT-enabled system is effective for detecting and classifying palm fruit ripeness at ground level and up to a certain tree height. Future work could focus on addressing the challenges associated with taller trees to further optimize the model's performance.

C. Performance comparison with previous studies

To evaluate the performance of the proposed system, a comparative analysis was conducted against previous studies utilizing various deep learning models for palm fruit ripeness detection, as shown in Table 4. The comparison includes key metrics such as precision and recall, which are critical for assessing the reliability and accuracy of the detection models. The proposed YOLOv8n model achieved a precision of 0.952 and a recall of 0.945, demonstrating a balanced performance

Table 4.
Performance comparison of palm fruit ripeness detection models using different algorithms.

Work	Algorithm	Precision	Recall	Method type
Herman *et al.* [8]	DesNet	0.87	0.86	PC-based simulation
Purba *et al.* [35]	YOLOv8	0.98	0.83	PC-based simulation
Harmiansyah *et al.* [36]	YOLOv5	0.87	0.95	PC-based simulation
A.F. Japar *et al.* [33]	YOLOv4	0.97	0.81	PC-based simulation
Mamat *et al.* [37]	YOLOv5	0.98	0.98	PC-based simulation
Proposed method	YOLOv8n	0.952	0.945	Edge AI implementation

in accurately detecting ripe palm fruit bunches. Compared to earlier works, the results indicate competitive accuracy and recall values.

In comparison to YOLO-based models, the study in [35] utilized YOLOv8 and reported the highest precision of 0.98, though its recall value was lower at 0.83, suggesting that while it achieves high confidence in positive detections, it may miss some instances of ripe fruit. Similarly, the work in [33] using YOLOv4, reported 0.97 precision and 0.81 recall, indicating strong performance but with slightly lower recall than the proposed model. The approach in [37] employing YOLOv5, achieved the highest recall (0.98) while maintaining a precision of 0.98, surpassing most other models. However, the trade-off between model complexity and edge-device implementation constraints must be considered. In contrast, [8] implemented a DenseNet-based approach, which yielded 0.87 precision and 0.86 recall, showing lower accuracy compared to YOLO-based methods, likely due to the model's complexity and its suitability for real-time applications. Compared to existing works, the proposed YOLOv8n-based model balances both high precision (0.952) and high recall (0.945) while maintaining real-time performance on an edge device (Raspberry Pi 4). The results indicate improved detection capabilities over previous YOLOv4 and YOLOv5 implementations, while achieving comparable performance to YOLOv8 variants. The slightly lower precision than [35] and [37] is counterbalanced by a more balanced recall, ensuring fewer missed detections in real-world applications.

The findings demonstrate the feasibility of implementing an edge computing solution for real-time palm fruit ripeness detection, optimizing model performance while maintaining deployment efficiency. The results indicate that the YOLOv8n model is well-suited for IoT-enabled agricultural applications, where real-time inference, low computational overhead, and high detection accuracy are required.

D. Dashboard for monitoring and analysis

The dashboard for monitoring and analysis in this study is shown in Figure 5. The interface is designed for easy navigation and interaction. Users can select different modes such as camera or GPS mode from the main interface. The dashboard also features real-time visualization of collected field data, including palm fruit ripeness classifications, GPS locations, temperature, humidity, and pressure. Key functionalities include the visualization of live feed data from the camera, which displays bounding boxes around detected palm fruits along with their ripeness classification. The dashboard enables data visualization that aids estate plantation management and decision-making by providing clear insights into palm fruit ripeness classification and associated GPS coordinates. It allows plantation managers to monitor real-time updates on palm fruit ripeness and the spatial distribution of scanned fruits across the plantation. With this information, better management and planning of harvesting can be achieved by understanding the ripeness levels across different locations. Additionally, the dashboard can help monitor the fertilization status and needs, ensuring optimal resource allocation and plantation health.

E. Scalability and deployment in large plantations

The feasibility of scaling the system across large plantations can be achieved by leveraging edge computing and sensor node networks. Rather than relying on a single Raspberry Pi 4 Model B unit, the system can be expanded into a distributed network of edge nodes, where each node performs local processing and communicates with a central system. This decentralized approach allows for more efficient handling of large areas by deploying multiple sensor nodes across the plantation. Each node can independently detect and classify ripeness, reducing the computational load on a single device and enabling real-time processing in various parts of the plantation.

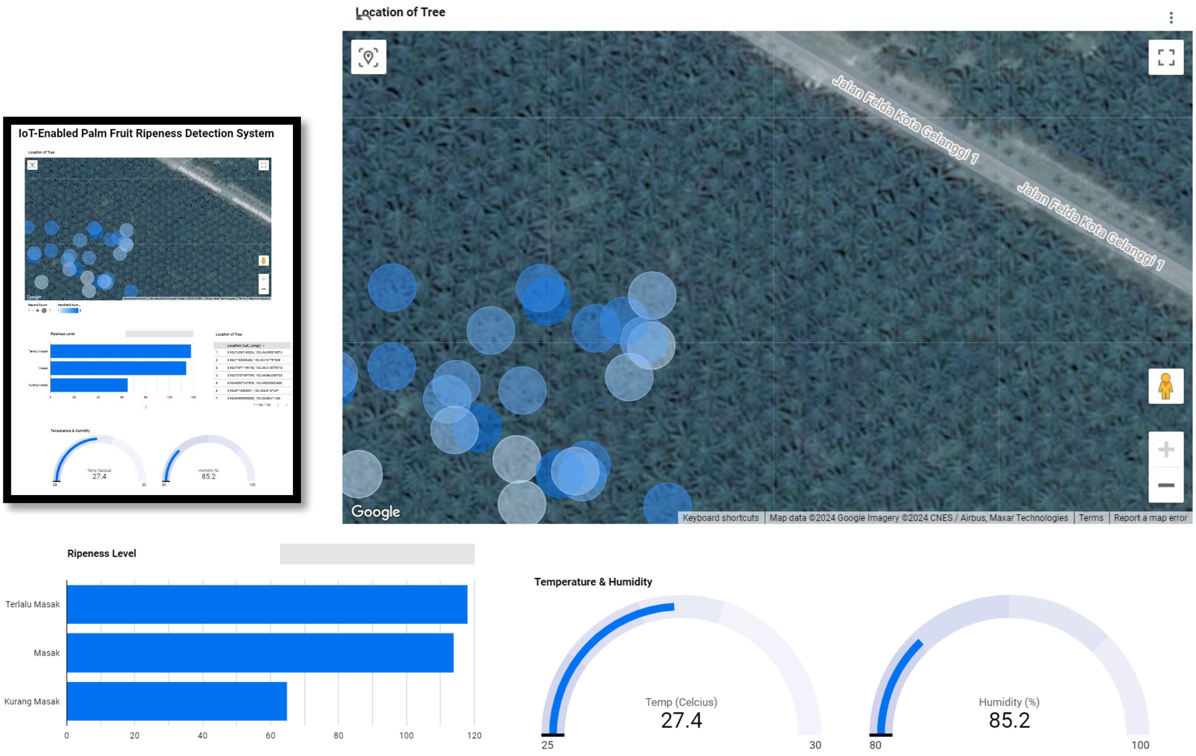


Figure 5. System dashboard.

Additionally, the use of wireless communication between nodes can ensure seamless data sharing and coordination, making the system adaptable for large-scale deployment in complex agricultural environments. This approach ensures that the system remains scalable, cost-effective, and adaptable to large plantations without compromising on performance.

F. Hardware limitations and future improvements

While the system demonstrated effective detection and classification capabilities during onsite evaluation, it is important to consider its performance under prolonged usage in tropical conditions. The Raspberry Pi, used as the edge device, may face processing limitations and heat management challenges due to the intensive computations required by the YOLOv8n model. High ambient temperatures in plantation environments could exacerbate these issues, potentially leading to overheating and reduced performance. Future iterations of the system should explore solutions such as passive or active cooling mechanisms, hardware optimization, or the use of more robust edge devices to ensure consistent performance in real-world conditions. In addition, the model's accuracy decreases when detecting fruits on taller trees, primarily due to the difficulty in capturing clear images of higher fruit clusters in complex environments. To address this limitation, future work will explore integrating telescopic cameras or developing a calibration method that accounts for varying tree heights. These advancements could significantly improve the model's performance in diverse agricultural settings, ensuring reliable detection across a broader range of plantation environments.

IV. Conclusion

The proposed IoT-enabled palm fruit ripeness detection system demonstrates significant potential in optimizing crop growth by monitoring key environmental parameters such as ripeness levels, temperature, humidity, and location. Leveraging IoT technologies, the system successfully collects real-time data that provides farmers with actionable insights for timely interventions to enhance crop health and productivity. The findings highlight the promise of integrating advanced technologies like YOLOv8n for FFB ripeness detection, despite challenges such as varying camera resolutions and the need for improved training datasets to address lighting conditions and diverse fruit types. The implementation of YOLOv8n on a Raspberry Pi edge device has proven feasible; however, continued research is essential to develop simplified yet reliable algorithms that are optimized for resource-constrained environments. Several implementation challenges emerged, particularly in achieving higher camera resolutions to improve image quality for more accurate FFB detection. Training models to recognize multiple fruit types under varied lighting conditions, including shadows, presented complexities. Additionally, managing tree heights effectively remains a critical limitation. Accurate measurement and monitoring of tree growth are essential for precision agriculture practices and demand robust sensor technologies capable of handling real-time data streams. To further enhance the robustness of the system, specific improvements can be made by integrating advanced image processing techniques to handle low-light and shadowed conditions commonly found in plantations. Developing higher-resolution cameras and enhancing training datasets to account for these conditions will significantly improve the accuracy and reliability of fruit detection. Moreover, future work should focus on optimizing object detection algorithms for edge computing platforms like Raspberry Pi, ensuring that the system remains efficient even in challenging environments. Expanding the system's scope beyond fruit ripeness detection to include disease monitoring is also crucial. By integrating disease detection algorithms into the IoT framework, farmers can proactively identify and manage plant diseases before they escalate, enhancing crop resilience and promoting sustainable farming practices. Advancing these algorithms, especially for use in variable lighting conditions, will further improve the system's effectiveness. Integrating this IoT system with precision agriculture practices will maximize agricultural productivity. Real-time monitoring of tree heights, canopy growth, and other metrics will enable farmers to implement targeted interventions such as optimized irrigation, fertilization, and pest management strategies. This integration not only fosters resource efficiency but also promotes sustainable farming practices tailored to the specific needs of crops.

Acknowledgements

The authors would like to thank Universiti Malaysia Pahang Al-Sultan Abdullah (UMPSA), the Faculty of Electrical and Electronics Engineering Technology, and the UMPSA STEM Lab, including its members, facilitators, and student mentors, for their participation and support in this research work.

Declarations

Author contribution

N.H. Noordin: Writing - Original Draft, Writing - Review & Editing, Conceptualization, Formal analysis, Investigation, Visualization, Supervision. **R. Samad:** Writing - Review & Editing, Formal analysis, Investigation, Validation. **A.H.A Malek:** Software, Simulation Execution (Roboflow), Visualization.

Funding statement

This work has been supported by the RDU232710 / UIC231519 grant, Universiti Malaysia Pahang Al-Sultan Abdullah (UMPSA), IBM Malaysia and the Ministry of Higher Education Malaysia (MoHE). This support enabled the successful completion of our research at the UMPSA STEM Lab, Faculty of Electrical and Electronics Engineering Technology, UMPSA.

Competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

The use of AI or AI-assisted technologies

Generative AI tools were used solely to enhance language clarity and improve the readability of the manuscript. No AI-generated content, data, or analysis was included in the research findings. The final version of the manuscript was thoroughly reviewed and verified to ensure accuracy and originality.

Additional information

Reprints and permission: information is available at https://mev.brin.go.id/.

Publisher's Note: National Research and Innovation Agency (BRIN) remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

[1] M. Nahiduzzaman *et al.*, "Deep learning-based real-time detection and classification of tomato ripeness stages using YOLOv8 on raspberry Pi," *Eng. Res. Express*, vol. 7, no. 1, p. 015219, Mar. 2025.

[2] N. Mishra and S. Pandya, "Internet of Things applications, security challenges, attacks, intrusion detection, and future visions: A systematic review," *IEEE Access*, vol. 9, pp. 59353-59377, 2021.

[3] T. Raj, F. H. Hashim, A. B. Huddin, A. Hussain, M. F. Ibrahim, and P. M. Abdul, "Classification of oil palm fresh fruit maturity based on carotene content from Raman spectra," *Sci. Rep.*, vol. 11, 2021.

[4] M. Farhan, M. N. Akhtar, and E. A. Bakar, "Efficient real-time palm oil tree detection and counting using YOLOv8 deployed on edge devices," *J. Umm Al-Qura Univ. Eng. Archit.*, vol. 16, no. 4, pp. 1293–1308, Dec. 2025.

[5] X. Shi, Y. Tao, and S.-C. Lin, "Deep neural network-based prediction of B-cell epitopes for SARS-CoV and SARS-CoV-2: Enhancing Vaccine Design through Machine Learning," in *2024 4th International Signal Processing,*

Communications and Engineering Management Conference (ISPCEM), IEEE, Nov. 2024, pp. 259–263.

[6] B. Xiao, M. H. Nguyen, and W. Q. Yan, "Apple ripeness identification using deep learning," *Geom. Vis.*, vol. 1386, pp. 53–67, 2021.

[7] N. O. M. Salim and A. K. Mohammed, "Comparative analysis of classical machine learning and deep learning methods for fruit image recognition and classification," *Trait. du Signal*, vol. 41, no. 3, pp. 1331–1343, Jun. 2024.

[8] H. Herman, T. W. Cenggoro, A. Susanto, and B. Pardamean, "Deep learning for oil palm fruit ripeness classification with DenseNet," in *Proc. 2021 Int. Conf. Inf. Manag. Technol. (ICIMTech)*, vol. 1, pp. 116–119, 2021.

[9] W. Cho, S. K. Kim, M. H. Na, and I. S. Na, "Fruit ripeness prediction based on DNN feature induction from sparse dataset," *CMC-Comput. Mater. Continua*, vol. 69, pp. 4003–4024, 2021.

[10] S. Ashari, G. J. Yanris, and I. Purnama, "Oil palm fruit ripeness detection using deep learning," *Sinkron*, vol. 7, no. 2, pp. 649–656, May 2022.

[11] Y. A. A. Soetrisno, E. Handoyo, Sumardi, and E. W. Sinuraya, "Oil palm level of ripeness classification using EfficientDet-Lite CNN architecture," *J. Oil Palm Res.*, vol. 36, no. 4, pp. 618–629, 2024.

[12] J. Y. Goh, Y. Md Yunos, and M. S. Mohamed Ali, "Fresh fruit bunch ripeness classification methods: A review," *Food Bioprocess Technol.*, vol. 18, no. 1, pp. 183–206, Jan. 2025.

[13] N. Khamis, H. Selamat, S. A. Ghazalli, N. I. M. Saleh, and N. Yusoff, "Comparison of palm oil fresh fruit bunches (FFB) ripeness classification technique using deep learning method," in *Proc. 2022 13th Asian Control Conf. (ASCC)*, pp. 64–68, 2022.

[14] I. Bonet, M. Gongora, F. Acevedo, and I. Ochoa, "Deep learning model to predict the ripeness of oil palm fruit," in *Proceedings of the 16th International Conference on Agents and Artificial Intelligence,* SCITEPRESS - Science and Technology Publications, 2024, pp. 1068–1075.

[15] 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.

[16] A. Vijayakumar and S. Vairavasundaram, "YOLO-based object detection models: A review and its applications," *Multimed. Tools Appl.*, vol. 83, no. 35, pp. 83535–83574, Mar. 2024.

[17] M. Hussain, "YOLO-v1 to YOLO-v8, the rise of YOLO and its complementary nature toward digital manufacturing and industrial defect detection," *Machines*, vol. 11, no. 7, 2023.

[18] J. R. Terven, D. M. C. Esparza, and J. A. Romero-González, "A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Mach. Learn. Knowl. Extr.*, vol. 5, pp. 1680–1716, 2023.

[19] M. Hussain, "YOLOv1 to v8: Unveiling each variant – a comprehensive review of YOLO," *IEEE Access*, vol. 12, pp. 42816–42833, 2024.

[20] C. Wang and G. Liang, "Overview of research on object detection based on YOLO," in *Proceedings of the 4th International Conference on Artificial Intelligence and Computer Engineering*, New York, NY, USA: ACM, Nov. 2023, pp. 257–265.

[21] 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 Int. Conf. Smart Instrum., Meas. Appl. (ICSIMA)*, Oct. 17–18, 2023.

[22] A. Murthy, Prathwini, and S. Kulkarni, "An exploration of deep learning-based object detection and classification in yakshagana imagery with YOLOv3," in *2024 5th International Conference on Electronics and Sustainable Communication Systems (ICESC),* IEEE, Aug. 2024, pp. 991–997.

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

[24] F. A. Junior and Suharjito, "Video based oil palm ripeness detection model using deep learning," Heliyon, vol. 9, no. 1, p. e13036, Jan. 2023.

[25] R. Gholami, H. D. Jahromi and S. Sedaghat, "Design and implementation of a near real-time human detection robot using YOLO framework and IoT technologies," in IEEE Access, vol. 13, pp. 175960-175983, 2025.

[26] A. Ahmed Ali Aboluhom and I. Kandilli, "Face recognition using deep learning on Raspberry Pi," *Comput. J.,* vol. 67, no. 10, pp. 3020–3030, Oct. 2024.

[27] I. Lazarevich, M. Grimaldi, R. Kumar, S. Mitra, S. Khan, and S. Sah, "YOLOBench: Benchmarking efficient object detectors on embedded systems," in 2023 *IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, IEEE, Oct. 2023, pp. 1161–1170.

[28] D. R. Reddy, C. Chella, K. B. Ravi Teja, H. Rose Baby, and P. Kodali, "Autonomous vehicle based on deep Q-learning and YOLOv3 with data augmentation," in *2021 International Conf. on Communication, Control and Information Sciences (ICCISc),* IEEE, Jun. 2021, pp. 1–7.

[29] T. Kumar, R. Brennan, A. Mileo, and M. Bendechache, "Image data augmentation approaches: A comprehensive survey and future directions," *IEEE Access*, vol. 12, pp. 187536–187571, 2024.

[30] K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," *Glob. Transitions Proc.*, vol. 3, no. 1, pp. 91–99, Jun. 2022.

[31] D. Ocharo, H. C. Nganga, and S. Kiambi, "Automatic PPE monitoring system for construction workers using YOLO algorithm based on deep reinforcement learning," *Int. J. Comput. Appl.*, vol. 186, no. 48, pp. 10–15, 2024.

[32] M. H. P. Rosa, P. S. De Jesus, T. G. De Campos, F. C. Amate, and C. E. Dos Santos, "AI-based system for threat detection in academic environments," *Int. J. Commun. Inf. Technol.*, vol. 5, no. 2, pp. 93–100, 2024.

[33] A. F. Japar, H. R. Ramli, N. M. H. Norsahperi and W. Z. W. Hasan, "Oil palm loose fruit detection using YOLOv4

for an autonomous mobile robot collector," IEEE Access, vol. 12, pp. 138582-138593, 2024.

[34] S. Puttinaovarat, S. Chai-Arayalert, and W. Saetang, "Oil palm bunch ripeness classification and plantation verification platform: leveraging deep learning and geospatial analysis and visualization," *ISPRS Int. J. Geo-Information*, vol. 13, no. 5, p. 158, May 2024.

[35] S. J. Purba, W. F. Tandion, and E. Irwansyah, "Comparison of the latest version of deep learning YOLO model in automatically detecting the ripeness level of oil palm fruit in PTPN IV, North Sumatra, Indonesia," in *Proc. Int. Conf. Inf. Technol. Comput. (ICITCOM)*, Yogyakarta, Indonesia, pp. 295–300, 2024.

[36] Harmiansyah, E. P. Sembiring, E. T. Oviana, and Supriyanto, "Detection of new oil palm fruit bunches based on deep learning," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 1379, no. 1, p. 012003, Aug. 2024.

[37] N. Mamat, M. F. Othman, R. Abdulghafor, A. A. Alwan, and Y. Gulzar, "Enhancing image annotation technique of fruit classification using a deep learning approach," *Sustainability*, vol. 15, no. 2, p. 901, Jan. 2023.