Journal of Oil Palm Research
DOI: https://doi.org/10.21894/jopr.2026.0033

# REAL-TIME OIL PALM FRESH FRUIT BUNCH DRONE IMAGE CLASSIFICATION USING YOLO11 ARCHITECTURE

CHE AQIL ZULHAZIM CHE HASSAN[1]; MUHAMMAD RUSYDI MUHAMMAD RAZIF[1]*; MOHD NORZALI HJ MOHD[2]; AQILAH ABD HALIM[1]; NURUL HASYIMAH MOHD MUSTAPHA[1] and MOHD NORAZYSYAM AZMAN[3]

## ABSTRACT

*Conventional monitoring of oil palm fresh fruit bunches (FFBs) relies heavily on manual inspection, which is labour-intensive and often fails to deliver timely information on fruit maturity and yield potential required for precision harvesting. This study proposes a drone-based, real-time FFB classification system using deep learning to support efficient plantation monitoring, harvesting management, and yield estimation. A custom dataset comprising 11,613 annotated images of oil palm trees with visible FFBs was developed, categorising bunches into six distinct ripeness classes: Abnormal, empty bunch, underripe, ripe, unripe, and overripe. The dataset was divided into training (9,121 images), validation (1,657 images), and testing (835 images) subsets. Object detection models based on the Ultralytics YOLO11 architecture were trained for 100 and 200 epochs and deployed within a Python application developed using Streamlit, OpenCV, and the Ultralytics API. The system processes live unmanned aerial vehicle (UAV) video streams, performs frame-level FFB detection, tracks FFB across consecutive frames, and records per-tree detection results in comma-separated values (CSV) format using QR-code-based tree identification. Experimental evaluation shows that the YOLO11 model achieved a mean average precision (mAP) of 98.3%, precision of 94.0%, recall of 97.2%, and fast inference speed of approximately 7.7 ms/frame. The results demonstrate that the proposed system offers a scalable and practical solution for automated, real-time FFB monitoring, supporting ripeness-based harvest planning and operational efficiency in oil palm plantations.*

**Keywords:** fresh fruit bunch, oil palm, real-time monitoring, streamlit, YOLO.

**Received:** 17 January 2026; **Accepted:** 4 June 2026; **Published online:** 17 August 2026.

## INTRODUCTION

Oil palm (*Elaeis guineensis*) is one of the world's most important oil-producing crops, with global production exceeding 78 million tonnes in 2023 (Food and Agriculture Organization of the United Nations, 2024). Malaysia and Indonesia contribute a substantial proportion of this output, and timely monitoring of fresh fruit bunches (FFBs) is critical for optimising harvest scheduling and maximising oil yield (Rosbi et al., 2024). Despite its importance, conventional monitoring practices in many plantations continue to rely on manual scouting and periodic ground surveys. These approaches are labour-intensive, time-consuming, and often impractical over large plantations (Arpyanti, 2025; Murphy et al., 2021). In addition, manual assessments are prone to delays and human error, which may result in missed ripe bunches, suboptimal harvest timing, and reduced operational efficiency. These challenges are further exacerbated by labour shortages and the increasing scale of modern plantations (Musa et al., 2021).

[1] Electrical Engineering Technology Department, Faculty of Engineering Technology, Universiti Tun Hussein Onn Malaysia, 84600 Muar, Johor, Malaysia.

[2] Electronic Engineering Department, Faculty of Engineering Electrical and Electronic, Universiti Tun Hussein Onn Malaysia, 84600 Muar, Johor, Malaysia.

[3] Victory Enghoe Plantations Sdn. Bhd., Southern Malay Palm Oil Mill, Miles 41, Jalan Johor Bahru, 86200 Simpang Renggam, Kluang, Johor, Malaysia.

* *Corresponding author e-mail: rusydi@uthm.edu.my*

In the Malaysian context, recent achievements in precision agriculture have focused on palm counting and health mapping. However, real-time ripeness classification remains a significant challenge due to dense canopy occlusion and extreme lighting variability in tropical climates (Goh et al., 2025; Norhashim et al., 2024). While two-stage detectors, such as Faster R-CNN, offer high accuracy, their high computational latency makes them impractical for live unmanned aerial vehicle (UAV) telemetry. This study utilises YOLO11 because its single-stage architecture enables sub-10 ms inference, which is critical for real-time flight synchronisation and safety.

Recent advances in UAVs and computer vision have introduced promising alternatives for large-scale plantation monitoring. UAVs enable rapid acquisition of high-resolution imagery across extensive areas, while deep learning-based object detection models support automated interpretation of visual data. Furthermore, this research defines the operational parameters, including a ground sampling distance (GSD) of 0.13 cm/pixel at an average flight altitude of 4 m, to address localised failure modes such as specular glare. Among one-stage detectors, the You Only Look Once (YOLO) architecture lineage has evolved through various iterations, with each version introducing optimisations in feature extraction and computational efficiency. It also has gained wide adoption due to its ability to balance detection accuracy and inference speed by performing object detection in a single forward pass of a convolutional neural network (CNN) (Redmon et al., 2016; Ren et al., 2017; Wang et al., 2024). Continuous improvements in YOLO architectures have enhanced backbone design, feature fusion, and deployment efficiency. The Ultralytics YOLO11 model introduces further architectural and training refinements that improve detection accuracy and inference throughput, making it well-suited for real-time UAV-based applications (Ma & Chen, 2024; Ultralytics YOLO docs, 2024).

The application of YOLO-based detectors in agriculture and remote sensing has expanded rapidly, including tasks such as crop detection, tree mapping, and pest monitoring using UAV imagery (Naghipour et al., 2024). In oil palm research, previous studies have demonstrated that YOLO variants, including YOLOv5 and YOLOv8, can effectively detect palm trees and fruiting structures from aerial images, achieving high precision and Recall under controlled experimental settings (Hidayat et al., 2024; Yipeng & Junwu, 2024). While these studies highlight the potential of deep learning for plantation monitoring, detecting FFBs under real field conditions remains challenging. Factors such as occlusion by fronds, variable illumination throughout the day, heterogeneous canopy backgrounds, scale variation caused by changes in UAV altitude, and motion blur from UAV movement significantly affect detection reliability. The dense, overlapping canopy of mature palms also creates significant low-light zones (occlusion) and high-contrast 'specular glare' on fruit surfaces. These variables often lead to feature loss in traditional detection algorithms, requiring a more robust architecture capable of extracting deep features from low-resolution or shadowed imagery (Liang et al., 2024; Thapa et al., 2025).

From an operational standpoint, these challenges directly influence the practicality of automated detection systems in commercial plantations. Inaccurate or delayed detection may lead to inefficient harvest scheduling, inconsistent yield estimation, and reduced decision-making confidence. Moreover, many existing studies focus primarily on offline analysis of pre-collected imagery, limiting their applicability to time-sensitive field operations (Suharjito et al., 2023). The lack of real-time output, tree-level association, and integrated logging mechanisms further restricts the integration of detection results into plantation management workflows (Kaukab et al., 2024; Lai et al., 2023). Addressing these limitations is therefore essential to enable AI-driven systems that can support on-site harvest planning, labour optimisation, and data-driven plantation management.

This study addresses these gaps by developing and deploying a drone-based, real-time FFB detection system using the Ultralytics YOLO11 framework, with YOLOv5 and YOLOv8 included for comparative evaluation. While the Malaysian Palm Oil Board (MPOB) and existing literature have explored UAVs for palm counting and broad health mapping, a critical research gap remains in real-time, tree-specific maturity grading. Most existing studies focus on post-processing imagery or batch analysis of fruit piles (Sharma et al., 2023). The novelty of this study lies in the development of a live-telemetry classification framework that utilises QR-code-based session initialisation to link real-time YOLO11 inference directly to individual tree identifiers. This allows for immediate, on-site harvest decision-making, providing a level of operational granularity not currently available in standard aerial surveying protocols. In detail, the trained models were embedded within a Streamlit-based Python application that processes live UAV video streams via Open Broadcaster Software (OBS) Studio, performs frame-level inference, visualises detection outputs, and records per-tree session summaries in comma-separated values (CSV) format. To ensure a reliable association between detections and individual palm trees, QR-code-based tree identification was implemented. In addition, a lightweight tracking mechanism combined with IoU-based deduplication was applied to prevent double-counting across consecutive frames.

This research adopts the YOLO11 architecture, which introduces several key enhancements over its predecessors, YOLOv5 and YOLOv8. Specifically, YOLO11 integrates Cross Stage Partial with C3 (C3k2) and Position-Sensitive Attention (C2PSA) modules. These architectural refinements allow for deeper feature extraction with a significant reduction in computational parameters compared to YOLOv8. For UAV-based real-time deployment, where the hardware must process high-frequency video downlink with sub-10 ms latency, YOLO11 offers a superior speed-to-accuracy ratio. By leveraging these enhancements, the proposed system achieves high-precision classification in complex tropical plantation environments where earlier iterations often struggled with real-time stability (Hassan et al., 2026).

By integrating a high-performance detection model, a field-representative dataset, and a real-time user interface, the proposed system aims to provide plantation managers with a scalable and practical tool for FFB monitoring and harvest planning. In addition to presenting model training and evaluation results, this study discusses deployment considerations, including UAV flight altitude, ground sampling distance, inference latency, and common failure modes related to occlusion and lighting variability. Overall, the work advances oil palm monitoring from offline analysis toward an operational, field-deployable solution for precision agriculture applications.

## MATERIALS AND METHODS

This section describes the procedures for image detection and system integration. A detection and localisation framework was developed using the YOLO11 model from Ultralytics to monitor and identify FFBs in oil palm plantations. The YOLO11 architecture is designed to autonomously extract, learn, and evaluate distinct visual features of FFBs from input images (Mansour et al., 2023), enabling accurate real-time recognition and localisation. This capability supports operational efficiency by providing fast, consistent, and data-driven outputs for harvesting-related decision-making.

The system deployment began with data collection, where a total of 11,613 images were obtained from drone footage and ground-based captures. These images were subsequently divided into training, validation, and testing datasets. The training workflow included image annotation, label map generation, and model configuration prior to training for 100 and 200 epochs, producing optimised .pt model weights and corresponding performance graphs.

To enhance model robustness under varying lighting and background conditions, data preprocessing was applied to standardise the input images. Both YOLO11 and YOLOv8 models were trained for performance comparison, after which the most suitable configuration was selected for system deployment based on evaluation results.

Following the training phase, the selected model was integrated into a Streamlit-based Python application for testing and deployment. The application received live video streams from drone footage via OBS Studio and performed real-time inference. Detected FFBs are displayed with bounding boxes and class labels, while an integrated QR recognition module identifies individual trees for record-keeping purposes. Each detection session is automatically logged into a CSV file, generating a per-tree summary of detected FFB counts (Mansour et al., 2022; Suharjito et al., 2021).

To ensure detection reliability during continuous video processing, the system incorporates a tracking mechanism combined with Intersection-over-Union (IoU) filtering to eliminate duplicate detections across consecutive frames (Zheng et al., 2020). This approach supports accurate counting and stable real-time monitoring throughout UAV operations. The field experiments were conducted at Kampung Jawa, Panchor, Johor. The study site consists of 10-year-old *Tenera* (DxP) palms planted on sedentary soil. While this study was conducted on firm ground, the UAV-based approach is particularly practical for peat soil areas, where ground-level mechanised scouting is limited by soft terrain and accessibility. Ripeness classification was standardised according to the MPOB Oil Palm Fruit Grading Manual, categorised into six classes (Ghazalli et al., 2023). Overall, the integration of YOLO-based object detection, UAV imagery, and a user-oriented dashboard provides a practical solution for large-scale, real-time FFB monitoring in oil palm plantations.

### Experimental Setup

The experimental evaluation of the YOLO11-based FFB detection system was conducted using a Python environment with CUDA 12.4 support, running on an NVIDIA GeForce RTX 4060 GPU with 8 GB of VRAM and 16 GB of system memory. GPU acceleration was employed to support efficient training and inference. The YOLO11 model consists of approximately 125 layers and 20 million trainable parameters, with an estimated computational complexity of 67.7 GFLOPs, enabling real-time processing of drone-acquired imagery.

As illustrated in *Figure 1*, the experimental and real-world workflow utilised the custom trained YOLO11 model to detect and classify FFBs according to maturity level. Training and validation followed the predefined dataset partitions, with all input images resized to 640 × 640 pixels. Real-time inference was performed using drone video streams to closely simulate operational deployment in plantation environments. Performance metrics, including precision, recall, mAP, and frame per second (FPS), were collected to evaluate both detection accuracy and inference speed.

The inference process was managed by a StreamlitDetectorController integrated within the Streamlit interface. This controller implemented a two-stage operational protocol designed for field deployment. The process is initiated by a QR-code-based Tree Identifier that scans incoming video frames for a designated QR tag positioned at the base of each palm tree. Once a QR code is detected, a TreeSession is activated, and all subsequent detections are associated with the identified tree.

During an active TreeSession, the YOLO11 model processed each frame, while a CentroidTracker assigned unique identifiers to detected FFB to prevent double-counting across consecutive frames. Two outputs are generated simultaneously: A real-time annotated video feed displaying bounding boxes, class labels, and tracking identifiers, and a session logging mechanism that records cumulative detection results. When the QR code exits the frame, the TreeSession is automatically finalised, and all detection data is written to a master CSV file (tree_sessions.csv). This workflow reflected practical UAV-based field operations, where continuous video acquisition is combined with real-time analysis and systematic data recording.

**Image Acquisition**

Image acquisition is a critical component of this study, as the performance and reliability of the detection model depend heavily on the quality and diversity of the dataset. The dataset was constructed by collecting oil palm images from multiple reliable sources (Omar et al., 2024; Suharjito et al., 2023). A substantial portion of the images was captured manually using drones equipped with high-resolution cameras during field surveys conducted at varying altitudes, lighting conditions, and viewing angles. Additional images were obtained from ground-level captures and internal archives to ensure that the dataset reflected realistic conditions encountered in commercial plantations.

A DJI Neo UAV was used, featuring a 12 MP resolution sensor. The drone followed a manual orbiting flight path at a steady speed of 0.5 m/s. The system supports a maximum operational duration of 28 min per battery session, with live video data transmitted wirelessly to a ground station via OBS Studio. The image data acquired via the DJI Neo UAV consisted of standard visible-spectrum Red-Green-Blue (RGB) imagery. No multispectral or thermal sensors were utilised, as the study aimed to develop a cost-effective solution using consumer-grade optical sensors' capabilities.

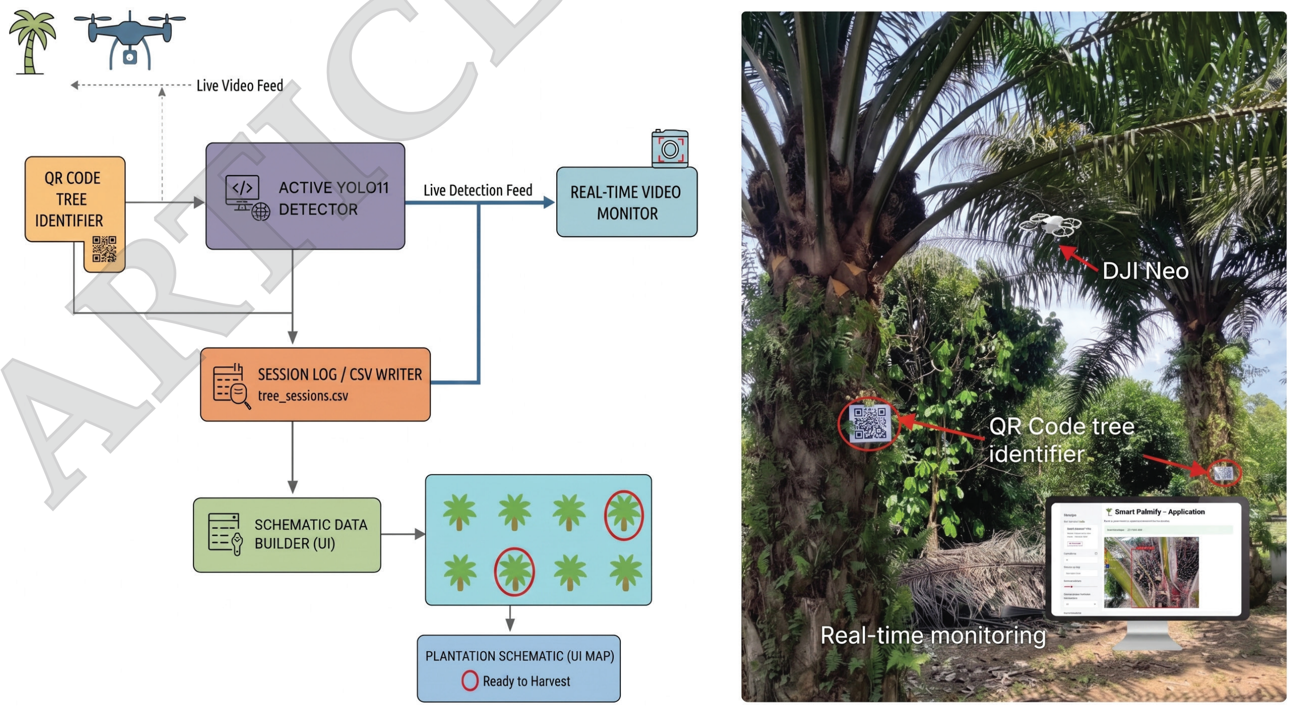


*Figure 1. Operational workflow of the experimental and real-world setup, from QR-code tree initialisation to real-time YOLO11 inference and session logging.*

To enhance dataset diversity and mitigate the risk of over-fitting to a specific localised environment, the training data was curated from a strategic mixture of primary and external sources. Primary imagery was captured on-site at the study location via UAV telemetry to provide high-resolution local context, while high-quality annotated datasets from the Roboflow Universe repository were integrated to incorporate variations in palm age and fruit density. Furthermore, a curated selection of open-source web imagery was utilised to include FFB samples from diverse geographical regions and various palm health conditions. These supplementary sources specifically addressed critical edge cases, such as extreme weather variability and rare abnormal fruit structures, which were less frequent in the primary study site, thereby ensuring a more robust and generalised detection model.

To further improve model robustness and classification capability, supplementary images were refined and adapted from datasets previously used in related oil palm studies. All images were carefully reviewed to ensure accurate representation of FFBs across different ripeness levels and physical conditions, including occluded, partially visible, and clustered bunches. The final dataset comprised 11,613 annotated images representing a wide range of canopy densities, fruit maturity stages, and environmental variations. Each image was manually labelled using the YOLO format and categorised into six classes: Abnormal, empty bunch, underripe, ripe, unripe, and overripe. These classes provided comprehensive visual examples for distinguishing fruit maturity levels, including abnormal conditions.

For training and evaluation purposes, the dataset was divided into three subsets: 60% for training, 30% for validation, and 10% for testing. This distribution supported balanced learning, objective evaluation, and reliable generalisation to unseen data. *Figure 2* illustrates representative images from the dataset, highlighting variations in FFB conditions and environmental settings. This structured dataset forms the foundation for effective YOLO11 model training and real-world FFB detection.

**Dataset Preparation**

To construct a robust dataset for oil palm FFB detection, images were collected from multiple plantation environments, with a primary focus on drone-captured aerial imagery from real field conditions. The dataset represented diverse palm canopy structures, tree heights, and fruit maturity stages to support broad generalisation. Image acquisition involved both manual ground-based photography and aerial footage captured using UAVs equipped with high-resolution cameras. Video footage was extracted frame-by-frame to obtain clear views of FFBs at different crown positions and viewing angles.

The dataset included diverse visual representations of FFB conditions, namely abnormal, empty bunch, underripe, ripe, unripe, and overripe classes. This intentional variation enabled the model to learn subtle maturity characteristics and to recognise bunches under partial occlusion by fronds. Each image was manually verified to ensure correct class assignment and adequate visibility of the target FFB.

Environmental variability commonly encountered in plantation operations was also considered during data collection. Images were captured under different lighting conditions, including bright sunlight, overcast skies, and low-light conditions near dusk, to improve model adaptability. Variations in UAV altitude and camera

*Figure 2. Representative samples from the custom-developed fresh fruit bunch (FFB) dataset, illustrating the various ripeness categories captured under varying plantation lighting conditions and canopy densities.*

angle were included to simulate realistic flight patterns and canopy perspectives. Most of the images were captured at resolutions of 1,080p where the suffix 'p' denotes progressive scanning or higher to preserve detailed visual information.

The collected images were processed and annotated using the Roboflow labelling platform. Each FFB instance was enclosed within a bounding box and assigned to one of the predefined classes. The resulting dataset of 11,613 annotated images provided a comprehensive representation of visual diversity in oil palm plantations, contributing to improved robustness and consistent detection performance during real-time UAV deployment.

### Data Preprocessing and Annotation

Prior to model training, data preprocessing and annotation were performed to standardise the dataset and ensure compatibility with the YOLO-based framework. All images were resized to a fixed resolution of 640 × 640 pixels, maintaining aspect ratios where possible to minimise visual distortion. This standardisation supported consistent feature learning related to fruit shape, size, and texture during training.

Annotation was conducted using the Roboflow labelling platform, which provided direct compatibility with YOLO annotation formats. Bounding boxes were used to identify and label all visible FFBs within each image, with each instance assigned to one of the six predefined classes. Upon completion, Roboflow automatically generated annotation text files in YOLO format, containing class identifiers and normalised bounding box coordinates. These files enable seamless integration with the Ultralytics YOLO training pipeline.

*Figure 3a* presents the data preprocessing flow, *Figure 3b* illustrates the image labelling workflow, and *Figure 3c* shows examples of annotated images highlighting different FFB classes and bounding box placements.

Following annotation and quality verification, the dataset was divided into training (60%), validation (30%), and testing (10%) subsets. This structured division ensured effective model learning while preserving sufficient unseen data for evaluation. With preprocessing and annotation completed, the dataset was prepared for YOLO11 model training and subsequent system development.

### Model Development and Training

YOLO11, one of the recent iterations in the YOLO family, introduces several architectural enhancements that improve feature extraction, spatial representation, and inference efficiency



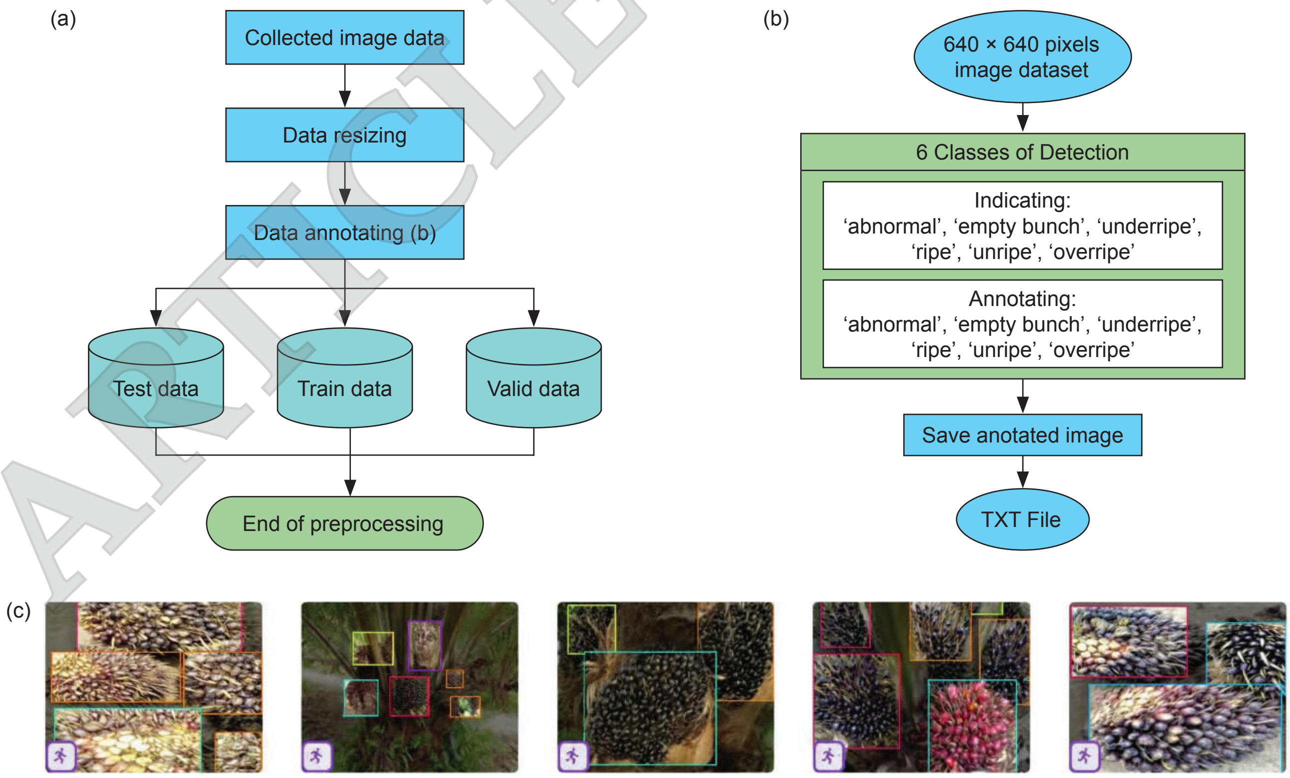


*Figure 3. (a) Data preprocessing pipeline featuring image resizing to 640 × 640 pixels and format conversion for YOLO compatibility; (b) image labelling workflow using the Roboflow platform; and (c) samples of annotated images showing bounding box placements for various FFB classes in complex backgrounds.*

compared to earlier versions. The model incorporates an improved feature pyramid structure and advanced modules, including the C3k2 module, Spatial Pyramid Pooling Fast (SPPF), C2PSA, and Upsample. These components enable higher detection precision while preserving real-time processing capability. With detection speeds ranging from approximately 5 to 200 FPS, depending on model scale and hardware configuration, YOLO11 demonstrates strong suitability for real-time applications such as aerial FFB detection in oil palm plantations. Its reported performance on benchmark datasets, including Microsoft Common Objects in Context (MS COCO), further supports its effectiveness as a high-accuracy object detection framework (Ultralytics YOLO docs, 2024).

The YOLO11 architecture comprises three main components: The input network, backbone network, and head network. The input network preprocesses images by resizing them to 640 × 640 × 3 prior to feature extraction. The backbone network serves as the core feature learning stage and integrates advanced modules, including C3k2 and C2PSA, to enhance representation capability. The C3k2 module employs dual convolutional paths with cross-stage connections, facilitating improved gradient flow and multi-scale feature learning. This design enables more effective detection of partially occluded or overlapping FFB. In parallel, the C2PSA module introduces position-sensitive attention to improve spatial awareness, allowing the network to focus on relevant regions within complex canopy environments. The overall YOLO11 network structure is illustrated in *Figure 4.*

To aggregate features across different spatial resolutions, the neck component of YOLO11

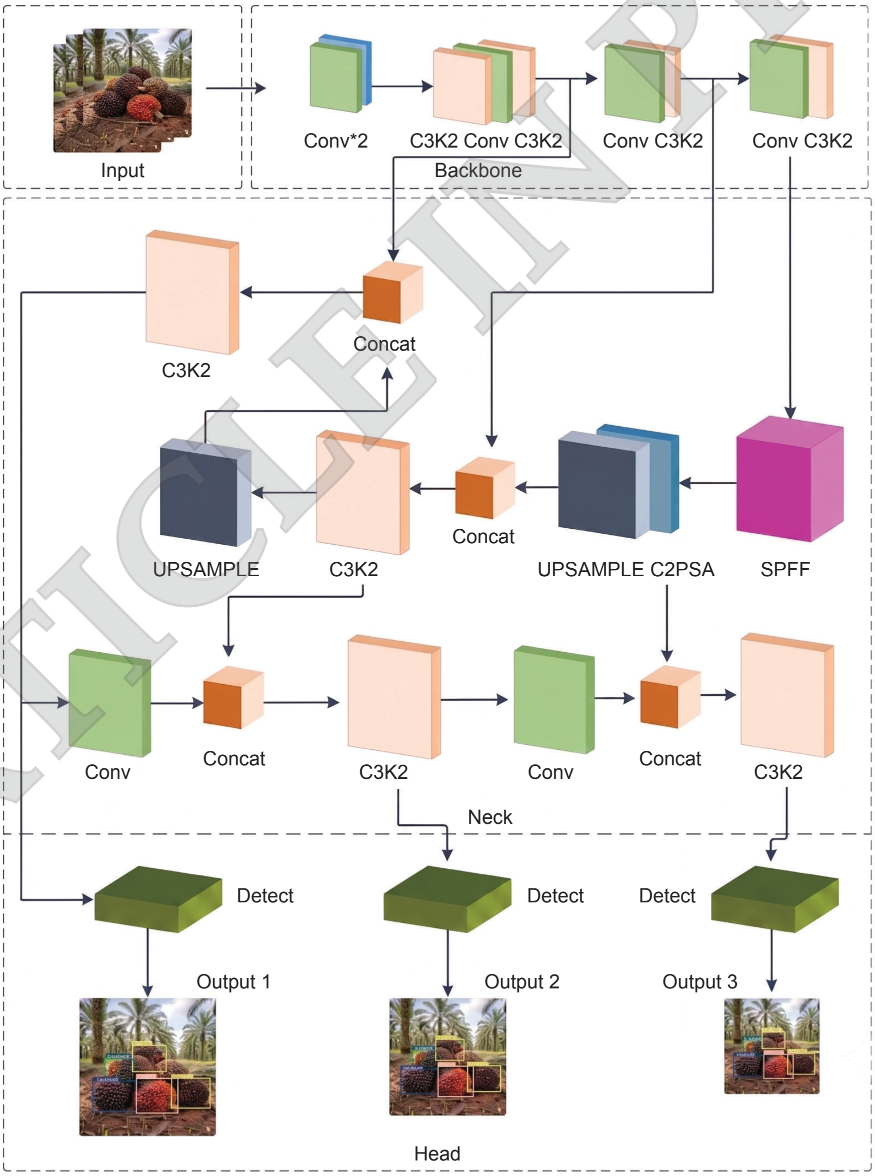


*Figure 4. Detailed architecture of the proposed YOLO11 network, highlighting the integration of advanced modules for optimised real-time feature extraction and multi-scale detection.*

integrates SPPF and Upsample modules. The SPPF module enhances the conventional Spatial Pyramid Pooling (SPP) approach by applying max pooling with kernel sizes of 5 × 5, 9 × 9, and 13 × 13 pixels, effectively expanding the receptive field while preserving spatial information. This design improves the detection of FFBs across varying sizes and flight altitudes. The Upsample module further increases feature map resolution prior to concatenation, ensuring effective information fusion between low-level and high-level features and improving localisation accuracy for small or partially visible objects.

The head network performs object detection and classification at three different scales, producing bounding box coordinates, confidence scores, and class probabilities within a single forward pass. Through the combined use of C3k2, SPPF, C2PSA, and Upsample modules, YOLO11 achieves a balanced trade-off between inference speed, model complexity, and detection accuracy. This architectural design enables reliable real-time inference on drone-acquired imagery, making the model suitable for FFB detection and maturity classification in oil palm plantation environments.

## System Integration

The integration of the YOLO11 model with Streamlit was carried out to develop an interactive, real-time platform for oil palm FFB detection and monitoring. Streamlit is an open-source Python library for building web-based applications in data science and artificial intelligence, offering a lightweight framework for deploying machine learning models with minimal interface development effort. Its flexibility and ease of use make it suitable for agricultural applications that require rapid deployment and intuitive visualisation. Through this integration, the YOLO11-based detection system supported real-time monitoring via a web interface without the need for complex graphical interface programming.

During the system integration stage, the trained YOLO11 model, developed using the custom FFB dataset, was embedded within the Streamlit application for inference and visualisation. Users can configure detection parameters dynamically, including confidence threshold, IoU limit, and video input source, through a control panel located in the Streamlit sidebar, as shown in *Figure 5a*. Live video input can be sourced from drone footage via OBS Studio, a webcam, or a recorded video file. Each frame is processed by the YOLO11 inference engine, and detected FFB are displayed with bounding boxes, class labels, and confidence scores in real-time. *Figure 5b* presents the interface section displaying the live detection output.

Streamlit's interactive features were further utilised to organise detection outputs through a Per-Tree Session Summary table. Rather than reporting per-frame statistics, the system aggregates detection results into sessions corresponding to individual palm trees. Each session automatically recorded the tree identifier, detection date and time, session duration, and the number of detected FFB categorised by class. A download link is generated for each session, allowing users to export the corresponding CSV file for offline analysis. The real-time session logs recorded in the CSV files utilise the following data headers: Timestamp, Tree_ID, Abnormal_Count, Empty_Count, Underripe_Count, Ripe_Count, Unripe_Count, Overripe_Count, and Confidence_Score. This session-based reporting approach enables plantation operators to review detection results on a tree-by-tree basis

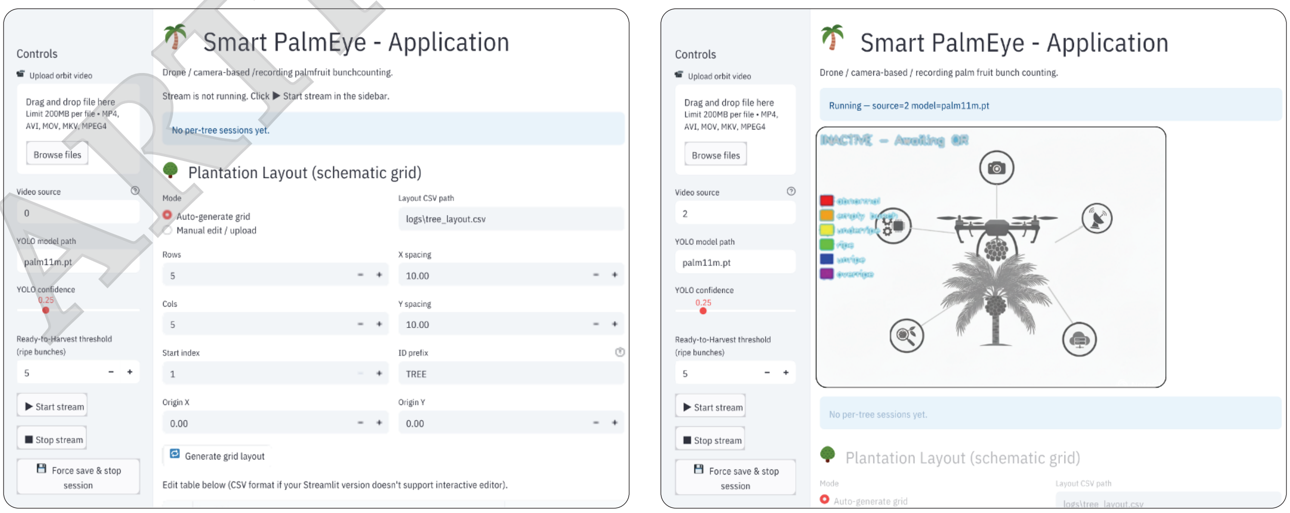


*Figure 5. (a) User interface of the Streamlit-based application showing the sidebar control panel for parameter configuration and (b) real-time fresh fruit bunch (FFB) detection window displaying live inference results with bounding boxes and confidence scores synchronised with the unmanned aerial vehicles (UAV) video stream.*

and maintain a structured digital record of field activities, as illustrated in *Figure 6a.*

In addition, a master log export function consolidated all per-tree sessions into a single CSV file. This Master Session Log recorded detection information for all trees, including tree ID, session duration, and FFB counts, providing a unified dataset for long-term yield analysis and plantation monitoring. A dedicated download button within the Streamlit interface enabled easy access to the consolidated file, as shown in *Figure 6b.*

To ensure accurate association between detections and individual palm trees, the system incorporates QR-code-based tree identification. When a QR tag appears within the video frame, the system automatically records the corresponding tree ID and assigns all subsequent detections to the same session. Upon completion of the session, a per-tree summary CSV file is generated, containing cumulative FFB counts by class and associated timestamps. This mechanism supports reliable record-keeping and facilitates yield estimation and plantation data management.

A dedicated palm tree layout configuration tool was also developed within the Streamlit interface to visualise plantation structure and assist with tree-level mapping. This tool allows users to specify the number of tree rows and columns, generate unique QR codes for each tree, and preview the plantation layout in real-time. The visual layout supports monitoring of scanning progress and verification of trees assessed during UAV operations. An example of this interface is shown in *Figure 6c.*

To support smooth real-time operation, the system architecture leverages Streamlit functionalities together with Python threading to manage concurrent processes, including video streaming, YOLO11 inference, QR detection, and CSV logging. This design maintains stable frame rates and responsive user interaction without interrupting the detection workflow. The integrated system combines YOLO11-based detection, UAV image acquisition, and an interactive web interface to provide a scalable and efficient solution for oil palm FFB monitoring.



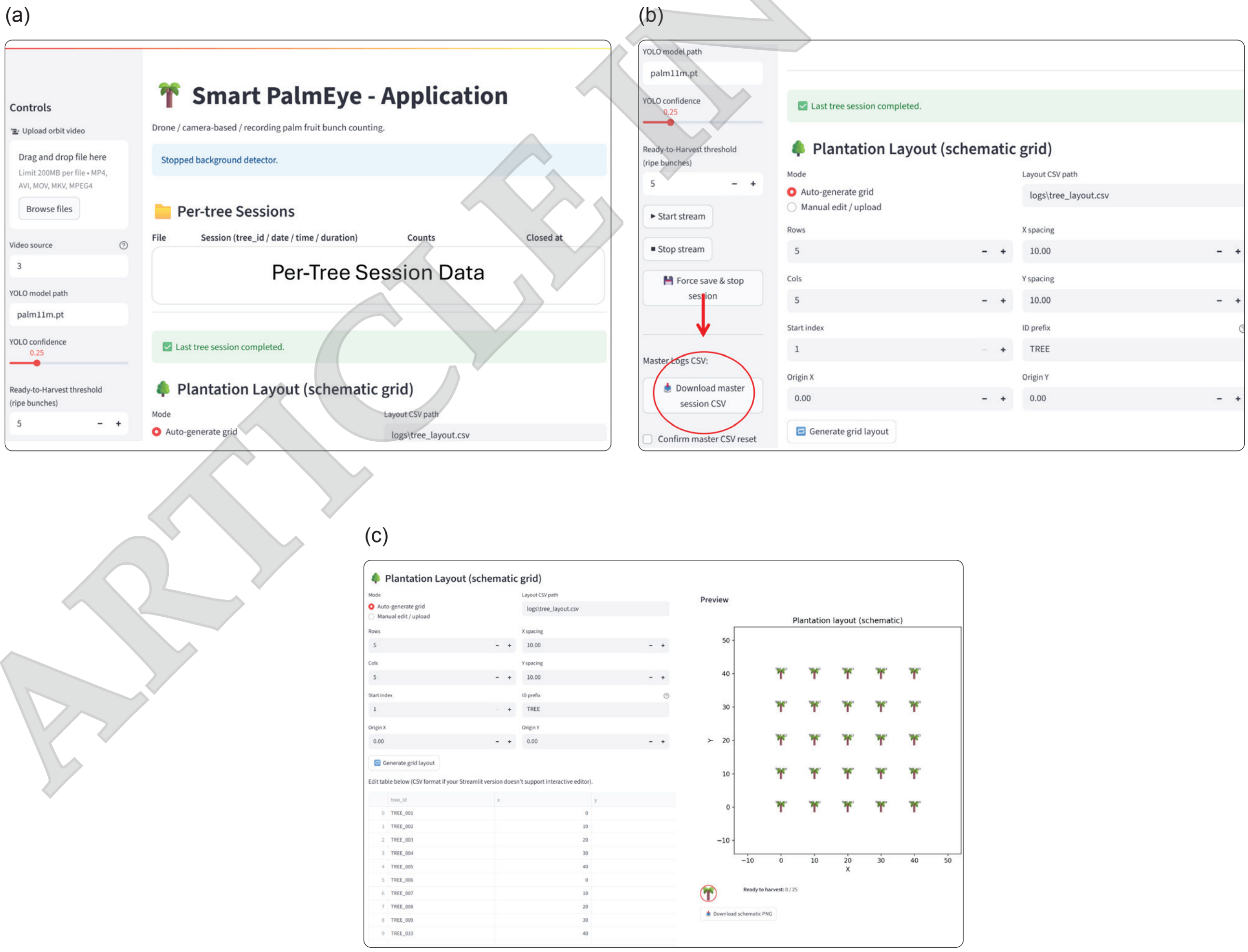


*Figure 6. Plantation data management modules: (a) Automated per-tree session summary recording detection counts; (b) Master Session Log export interface for consolidated comma-separated values (CSV) reporting; and (c) digital plantation layout tool for QR-code-based tree mapping and flight progress verification.*

**Testing and Evaluation**

The detection models were evaluated using standard object detection metrics, including precision (*P*), recall (*R*), mean average precision (*mAP*), and F1 Score. These metrics collectively assess the ability of the models to detect and classify FFBs across different maturity stages. Precision represents the proportion of correct detections among all predicted instances, while recall measures the proportion of relevant objects successfully detected. The mAP metric evaluates detection accuracy across varying confidence thresholds, and the F1 score, calculated as the harmonic mean of precision and recall, provides an overall indicator of detection reliability. The evaluation formulas are presented in Equations (1) to (5):

$$P = \frac{TP}{TP + FP} \times 100\% \tag{1}$$

$$R = \frac{TP}{TP + FN} \times 100\% \tag{2}$$

$$AP = \int_0^1 P(r)\, dr \tag{3}$$

$$mAP = \frac{1}{n} \Sigma_{i=1}^{n} AP_I \tag{4}$$

$$F1 = 2 \times \frac{P \times R}{P + R} \tag{5}$$

In these evaluations, True Positive (TP) refers to correctly detected FFBs, False Positive (FP) denotes incorrect detections, and False Negative (FN) represents missed detections. The metrics were applied to assess model performance under varying operational conditions, including changes in lighting, canopy density, and partial occlusion. For comparative analysis, three YOLO architectures YOLOv5, YOLOv8, and YOLO11 were trained and evaluated using the same dataset, training configuration, and hardware environment. The dataset consisted of six detection classes, and training was conducted for 100 and 200 epochs to examine the influence of training duration on detection accuracy and model stability.

All experiments were implemented in a Python environment using the Ultralytics framework with GPU acceleration via PyTorch. Identical hyperparameters were applied across all models to ensure fair comparison. Model checkpoints were saved automatically, and training progress was monitored using loss curves, precision-recall plots, and mAP progression graphs. Upon completion, performance comparisons were conducted based on mAP@0.5, mAP@0.5:0.95, precision, recall, and F1 score. The best-performing model checkpoints were exported as .pt files and deployed within the Streamlit-based system for real-time UAV testing.

## RESULTS AND DISCUSSION

The YOLO11-based model for oil palm FFB detection was trained using carefully tuned hyperparameters to achieve optimal performance. Training was conducted for 100 and 200 epochs using the stochastic gradient descent (SGD) optimiser with a learning rate of 0.01, momentum of 0.937, and weight decay of 0.0005. Model initialisation utilised pre-trained weights from the MS COCO dataset to accelerate convergence and improve feature generalisation. An early stopping strategy was applied to mitigate overfitting, where training was automatically terminated if no improvement in validation mAP was observed over 50 consecutive epochs.

*Figure 7* presents the YOLO11 F1-confidence curve and training performance over 200 epochs. As shown in *Figure 7a*, all six classes (abnormal, empty bunch, underripe, ripe, unripe, and overripe) achieved stable recall values approaching 0.96 at lower confidence thresholds, indicating high sensitivity in detecting FFBs, including partially occluded instances. *Figure 7b* illustrates the comparison between validation loss and mAP, which demonstrates model convergence and stability at 180 epochs. The model reached optimal stability around the 180th epoch, where the precision-recall curve plateaued. To ensure rigorous evaluation beyond simple detection, we utilised mAP@0.5:0.95. This metric calculated the average mAP across 10 IoU thresholds (0.5 to 0.95), proving the model's high localisation accuracy under variable canopy density.

Following training, the YOLO11 model achieved a precision of 94.0%, recall of 97.2%, and mAP@0.5 of 98.3%, outperforming both YOLOv8 and YOLOv5 under identical experimental conditions. These results demonstrate the effectiveness of YOLO11's enhanced architectural modules, namely C3k2, C2PSA, SPPF, and Upsample, in improving feature extraction and localisation accuracy, particularly in complex outdoor plantation environments.

*Figure 8* illustrates the confusion matrix obtained after 200 training epochs, providing a detailed assessment of per-class detection performance. The TP and FN distributions reveal both model strengths and remaining challenges. Among the six maturity classes, empty bunch and unripe achieved the highest TP rates of 100% and 99%, respectively, attributable to their distinctive colour and texture characteristics. In contrast, the underripe and ripe classes exhibited moderate confusion, with TP rates of approximately 97%, primarily due to similar visual appearance and lighting variations under dense canopy conditions. The overripe and abnormal classes both achieved TP accuracies of 98%, indicating reliable classification even in cases of minimal or irregular fruit presence.

Overall, confidence values across the confusion matrix ranged from 0.01 to 1.00, reflecting strong class discrimination capability. The YOLO11 model attained an overall classification accuracy of 98.3%, confirming its robustness for UAV-based FFB monitoring. These findings highlight YOLO11's suitability for real-time, tree-level fruit detection, demonstrating improved precision, inference stability, and adaptability to challenging lighting and occlusion conditions compared to earlier YOLO architectures.

## Model Comparison

To further evaluate detection performance, a comparative analysis was conducted among three YOLO architectures: YOLOv5, YOLOv8, and YOLO11. All models were trained and evaluated using the same dataset, hyperparameter settings, and experimental conditions. Training was performed for 100 and 200 epochs on the six-class oil palm FFB dataset. Performance evaluation focused on mAP, precision, recall, and F1 score, while the number of trainable parameters was examined to assess computational efficiency. *Table 1* summarises the comparative performance of each model.

The comparison indicates a clear progression in detection performance from YOLOv5 to YOLOv8 and subsequently to YOLO11. YOLOv5, while

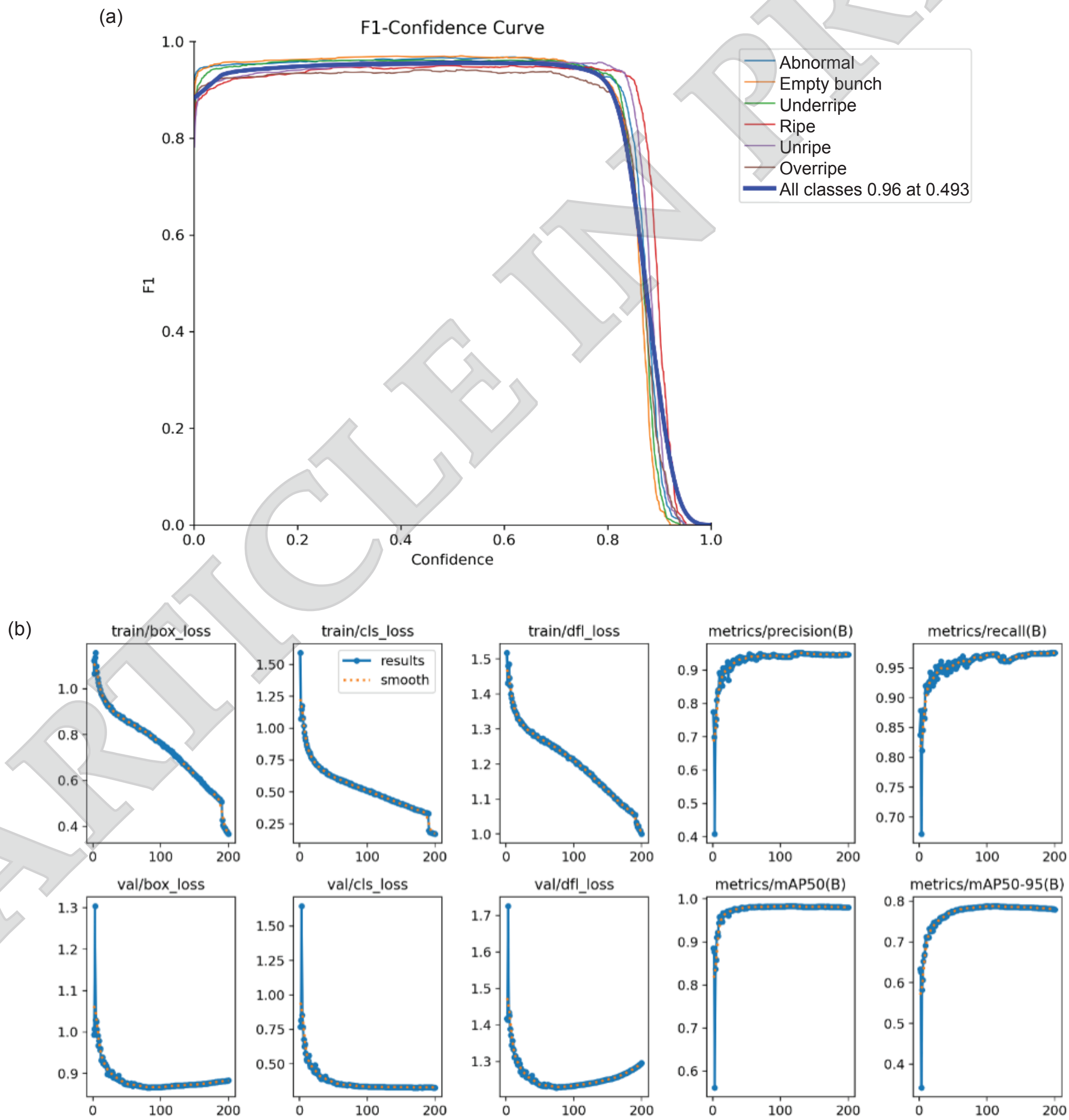


*Figure 7. Model training evaluation: (a) F1-confidence curve showing high stability across all six ripeness classes; and (b) training progression curves for precision, recall, and mAP, indicating model convergence and optimal performance stability around the 180th epoch.*

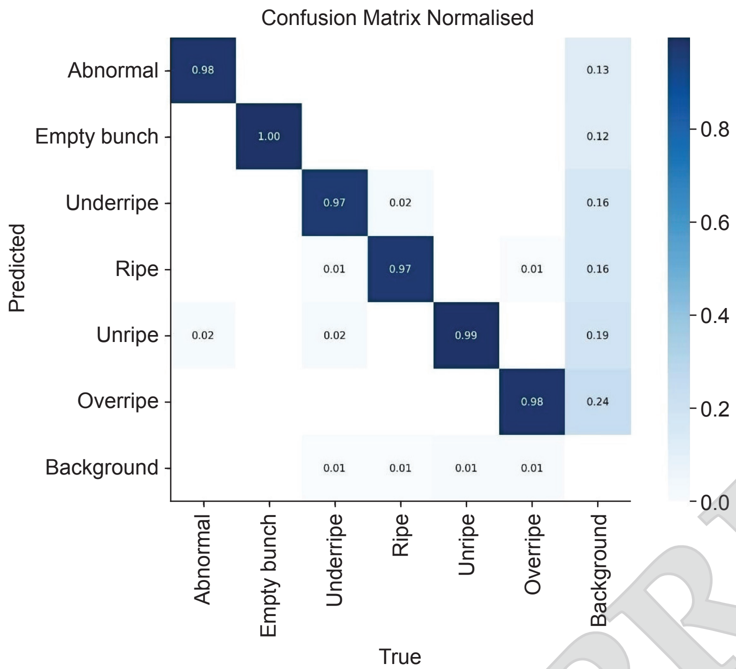


*Figure 8. Normalised confusion matrix of the YOLO11 model, demonstrating high classification accuracy.*



**TABLE 1. SUMMARY OF THE PERFORMANCE OF EACH MODEL UNDER IDENTICAL TRAINING CONDITIONS**

| Model | Epoch | mAP (%) | Precision (%) | Recall (%) | Trainable parameters | Inference time (ms) |
|---|---|---|---|---|---|---|
| Proposed YOLO11 | 100 | 78.4 | 94.2 | 96.3 | 13.5M | 7.7 |
| Proposed YOLO11 | 200 | 78.8 | 94.0 | 97.2 | 13.5M | 7.7 |
| YOLOv8 | 100 | 77.9 | 92.9 | 97.1 | 11.2M | 8.4 |
| YOLOv8 | 200 | 77.8 | 94.4 | 96.0 | 11.2M | 8.4 |
| YOLOv5 | 100 | 76.9 | 93.6 | 96.4 | 7.2M | 6.2 |
| YOLOv5 | 200 | 76.9 | 93.7 | 96.7 | 7.2M | 6.2 |

computationally lightweight, showed limitations in distinguishing visually similar maturity classes, particularly between Underripe and Unripe categories. YOLOv8 demonstrated notable improvements due to its enhanced feature extraction strategy, achieving higher mAP and balanced precision-recall performance. However, YOLO11 achieved the highest statistical performance across all evaluated metrics, demonstrating superior mAP and Precision values at both training durations while maintaining the lowest inference latency. In terms of operational efficiency, the YOLO11 model achieved an average inference time of 7.7 ms/frame on the NVIDIA RTX 4060 GPU. This represents a significant improvement over YOLOv8 (8.4 ms), despite YOLO11 having a larger parameter set (13.5M compared to 11.2M). While YOLOv5 was the fastest at 6.2 ms, its lower detection accuracy made it less suitable for precise ripeness classification in complex plantation environments.

The performance gains of YOLO11 can be attributed to the integration of advanced modules, including C3k2, C2PSA, SPPF, and Upsample, which collectively enhance spatial awareness, attention mechanisms, and multi-scale feature representation. Although YOLO11 has a slightly higher parameter count than YOLOv8, it maintained efficient GPU inference performance, confirming its suitability for real-time UAV-based deployment.

Overall, while YOLOv5 remains appropriate for lightweight applications and YOLOv8 offers a balanced trade-off between speed and accuracy, YOLO11 provides the most robust and scalable solution for drone-based oil palm FFB detection. Its combination of high detection accuracy, stable convergence behaviour, and efficient feature utilisation makes it the preferred model for operational deployment in large-scale plantation monitoring systems.

**Integration into the Streamlit Platform**

The integration of the YOLO11-based detection model within the Streamlit environment demonstrated strong potential for developing a scalable, responsive, and user-friendly platform for real-time FFB monitoring in oil palm plantations. The system successfully embedded the custom-trained YOLO11 model into a web-based interface capable of processing live UAV video streams in real-time (Hassan et al., 2026). As shown in *Figure 9a*, the Streamlit interface enables users to configure operational parameters such as confidence thresholds, IoU filtering, video input sources, and visualisation settings through an intuitive sidebar control panel.

Prior to each detection session, the system enters a QR-code initialisation stage, during which the live video feed scans for a QR tag associated with a specific palm tree. Once detected, real-time inference and session logging are activated, ensuring that all subsequent detections are linked to the corresponding tree identifier. A visual countdown is displayed to confirm stable positioning before detection begins. This workflow was validated under controlled laboratory conditions, as illustrated in *Figure 9b*.

Beyond live detection, the Streamlit platform includes a Per-Tree Session Summary module that organises detection records at the individual tree level. Each session log records the Tree ID, timestamp, detection duration, and FFB counts categorised by maturity class. These records are displayed in an interactive table, allowing efficient review and traceability, as shown in *Figure 9c*. Individual session summaries can be downloaded for offline analysis and long-term record management, supporting data-driven decision-making in plantation operations.

Finally, *Figure 9d* provides a qualitative analysis of the detection challenges encountered during real-time deployment. These failure modes are primarily categorised into false negatives caused by occlusion, where FFBs are partially obscured by leaves, and misclassifications resulting from specular glare. The latter occurs when high solar intensity creates bright reflections on the surface of underripe bunches, occasionally causing the model to misidentify them as ripe. Documenting these limitations is crucial for future model refinement and highlights the technical environmental variables present in tropical plantation settings.

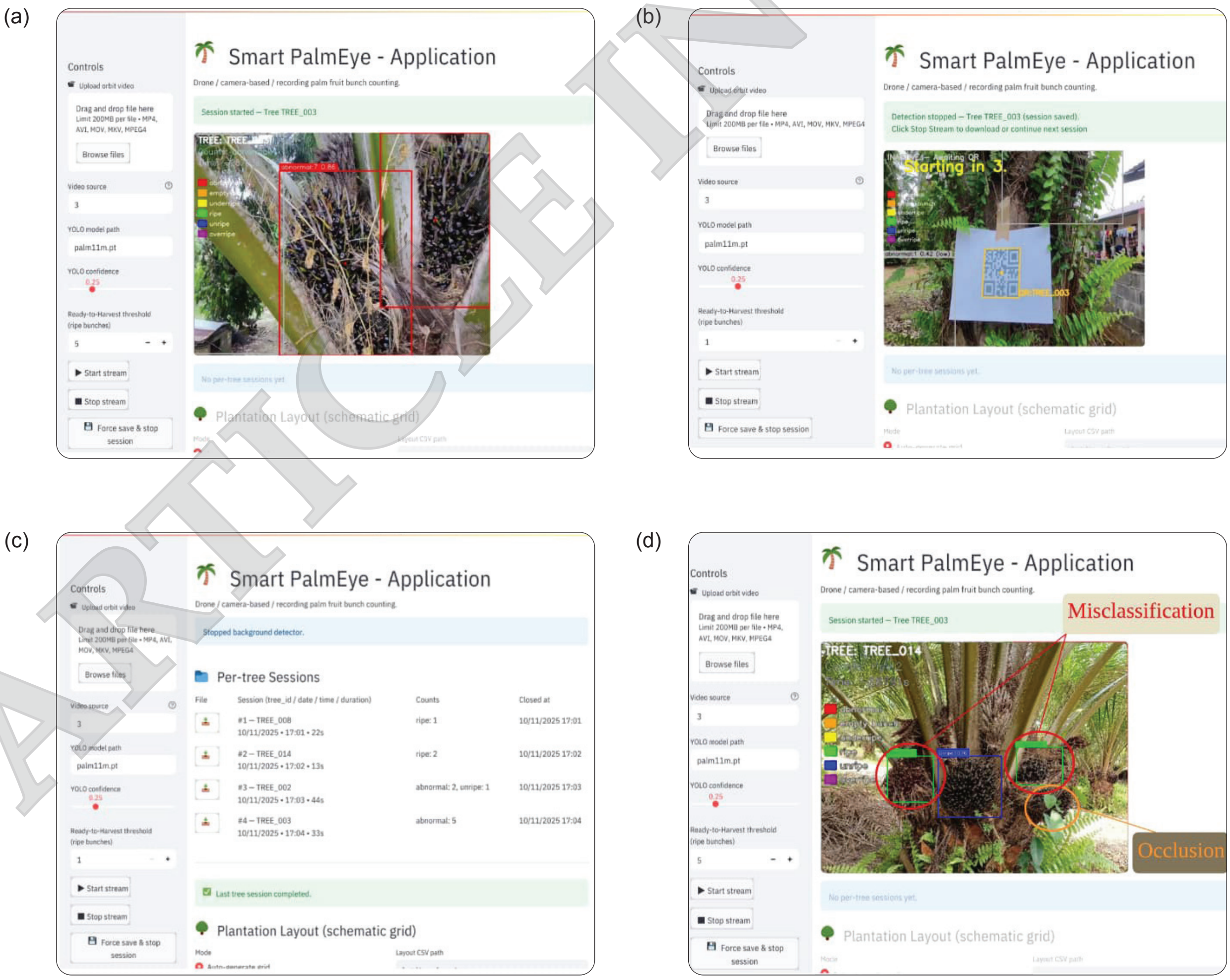


*Figure 9. (a) Streamlit interface setup for real-time unmanned aerial vehicles (UAV)-based fresh fruit bunch (FFB) detection using the YOLO11 model; (b) QR code initialisation process at the start of a detection session; (c) Per-Tree Session Summary table displaying tree-level detection logs and (d) qualitative samples of detection failure modes featuring frond occlusion and specular glare misclassification.*

## CONCLUSION

This study successfully developed an AI-driven system for real-time detection and classification of FFBs in oil palm plantations, integrating YOLO11 architectures for comparative performance evaluation. The proposed YOLO11-based model demonstrated strong detection accuracy and real-time responsiveness, enabling effective classification of FFBs into six maturity categories: Abnormal, empty bunch, underripe, ripe, unripe, and overripe, using drone-acquired video streams. When evaluated against YOLOv5 and YOLOv8 under identical training conditions, YOLO11 achieved competitive precision, recall, and mAP values, confirming its robustness and suitability for dynamic aerial monitoring environments. The results confirm that the YOLO11 based system achieved a mAP of 78.4% and a precision of 94.2%. With a processing latency of 7.7 ms/frame, the system is capable of high-speed monitoring at a flight speed of 0.5 m/s, providing a robust tool for real-time harvest decision-making in oil palm plantations.

The Streamlit-based interface provided an interactive and operationally practical platform for plantation monitoring, supporting live video inference, per-tree detection logging, and session-based data export. The incorporation of QR-code-based tree identification and master session log management enhanced data traceability, allowing detection results to be directly associated with individual trees for yield monitoring and operational planning. Initial deployment tests indicated stable real-time performance with low latency across varying environmental and lighting conditions.

Overall, the findings demonstrated the potential of combining deep learning-based object detection, UAV imaging, and real-time application frameworks to support precision agriculture practices. The developed system represented a transition from manual estimation toward an automated, data-driven, and scalable approach for fruit maturity assessment and yield estimation in oil palm plantations. Future work will focus on extending the framework to include ripeness quantification, GPS-based spatial mapping, and large-scale field validation across multiple plantation sites, with the aim of supporting more efficient harvesting strategies and sustainable palm oil production. To assist harvesters in the field, it is recommended that the CSV-generated logs be integrated into a mobile geographic information system (GIS) interface. This would provide harvesters with a real-time heat map of ripe bunches, optimising their collection route and reducing manual scouting time.

## ACKNOWLEDGEMENT

This research was supported by the Ministry of Higher Education (MOHE) through the *Geran Inovasi Sosial* (K582) and Universiti Tun Hussein Onn Malaysia (UTHM) through *Geran Penyelidikan Pascasiswazah* (GPPS) (vot J075).

## REFERENCES

Arpyanti, N. (2025). Detection of ripeness level of oil palm fresh fruit bunches using YOLOv4 model in automated harvesting system: A review. *Journal of Integrated Artificial Intelligence Science and Engineering, 1*(2), 29–34. https://doi.org/10.59190/jiaise.v1i2.328

Food and Agriculture Organization of the United Nations. (2024). *World production of palm oil 2023–2024*. Food and Agriculture Organization of the United Nations. Retrieved August 8, 2025, from https://www.fao.org/faostat/en/#data/QCL

Ghazalli, S. A., Selamat, H., Khamis, N., & Haniff, M. F. (2023). Short review on palm oil fresh fruit bunches ripeness and classification technique. *Journal of Advanced Research in Applied Mechanics*, *106*(1), 37–47. https://doi.org/10.7934/aram.106.1.3747

Goh, J. Y., Mohamed Ali, M. S., Md Yunos, Y., Sheikh, U. U., & Khan, M. S. (2025). Outdoor RGB and point cloud depth dataset for palm oil fresh fruit bunch ripeness classification and localization. *Scientific Data*, *12*(1), 687.

Hassan, C. A. Z. C., Razif, M. R. M., Sulaiman, M. N., Mustaphaa, N. H. M., Saad, M. A., & Azman, M. N. (2026). AI-Driven Real-Time PPE Compliance Monitoring in the Palm Oil Industry Using YOLOv8. *Jurnal Teknologi (Sciences & Engineering), 88*(2), 223–233. https://doi.org/10.11113/jurnalteknologi.v88.24334

Hidayat, M. R., Sulistyo, P. A., Oktario, H., Gunawan, A. A. S., & Mimboro, P. (2024). Establishing a standard operating procedure (SOP) for palm oil plantation FFB image capture: Utilizing YOLOv8 for counting and ripeness classification. *2024 6th International Conference on Cybernetics and Intelligent System, ICORIS 2024*, 01–06. https://doi.org/10.1109/ICORIS63540.2024.10903724

Kaukab, S., Komal, Ghodki, B. M., Ray, H., Kalnar, Y. B., Narsaiah, K., & Brar, J. S. (2024). Improving real-time apple fruit detection: Multi-modal data and depth fusion with non-targeted background removal. *Ecological Informatics, 82*, 102691. https://doi.org/10.1016/j.ecoinf.2024.102691

Lai, J. W., Ramli, H. R., Ismail, L. I., & Wan Hasan, W. Z. (2023). Oil palm fresh fruit bunch ripeness detection methods: A systematic review. *Agriculture, 13*(1), 156. https://doi.org/10.3390/agriculture13010156

Liang, W., Tan, J., He, H., Xu, H., & Li, J. (2024). Detection of small objects from UAV imagery via an improved Swin Transformer. *2024 IEEE International Geoscience and Remote Sensing Symposium*, 9134–9138. https://doi.org/10.1109/igarss53475.2024.10640832

Ma, J., & Chen, J. (2024). Pix2PixSSR: Spatial super-resolution synthesis and visualization for time-varying volumetric data. *IET Image Processing, 18*(1), 59–76. https://doi.org/10.1049/ipr2.12933

Mansour, M. Y. M. A., Dambul, K. D., & Choo, K. Y. (2022). Object detection algorithms for ripeness classification of oil palm fresh fruit bunch. *International Journal of Technology, 13*(6), 1326–1335. https://doi.org/10.14716/ijtech.v13i6.5932

Mansour, M. Y. M. A., Dambul, K. D., & Yeep, C. K. (2023). A review of non-destructive ripeness classification techniques for oil palm fresh fruit bunches. *Journal of Oil Palm Research, 35*(4), 543–554. https://doi.org/10.21894/jopr.2022.0063

Murphy, D. J., Goggin, K., & Paterson, R. R. M. (2021). Oil palm in the 2020s and beyond: Challenges and solutions. *CABI Agriculture and Bioscience, 2*(1), 39. https://doi.org/10.1186/s43170-021-00058-3

Musa, M., Abdullah, A. M., Ismail, M. M., Saari, M. Y., & Abdurofi, I. (2021). Labour efficiency in the Malaysian oil palm plantations: A data envelopment analysis (DEA) approach. *Journal of Agribusiness Management and Development*, *2*(1), 01–10.

Naghipour, M., Ling, L. S., & Connie, T. (2024). YOLO-based oil palm FFB ripeness detection. *2024 International Conference on Electrical, Communication and Computer Engineering (ICECCE)*, 1–5. https://doi.org/10.1109/ICECCE63537.2024.10823630

Norhashim, N., Kamal, N. M., Shah, S. A., Sahwee, Z., & Ruzani, A. A. (2024). A review of unmanned aerial vehicle technology adoption for precision agriculture in Malaysia. *Unmanned Systems, 12*(04), 707–725.

Omar, Z., Abdul Majeed, A.P. P., Rosbi, M., Ghazalli, S. A., & Selamat, H. (2024). Outdoor oil palm fruit ripeness dataset. *Data in Brief, 55*, 110667. https://doi.org/10.1016/j.dib.2024.110667

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016 - December*, 779–788. https://doi.org/10.1109/CVPR.2016.91

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*(6), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

Rosbi, M., Omar, Z., Khairuddin, U., P. P. A. Majeed, A., & Bakar, S. A. R. S. A. (2024). Machine learning for automated oil palm fruit grading: The role of fuzzy C-means segmentation and textural features. *Smart Agricultural Technology, 9*, Article 100691. https://doi.org/10.1016/j.atech.2024.100691

Sharma, K., Sahu, T. K., Bhatt, P., Singh, A. K., & Bhatia, M. (2023). Real-time video object detection using deep learning. *2023 International Conference on Advances in Computation, Communication and Information Technology, ICAICCIT 2023*, 601–606. https://doi.org/10.1109/ICAICCIT60255.2023.10466207

Suharjito, Elwirehardja, G. N., & Prayoga, J. S. (2021). Oil palm fresh fruit bunch ripeness classification on mobile devices using deep learning approaches. *Computers and Electronics in Agriculture, 188*, 106359. https://doi.org/10.1016/j.compag.2021.106359

Suharjito, Junior, F. A., Koeswandy, Y. P., Debi, Nurhayati, P. W., Asrol, M., & Marimin. (2023). Annotated datasets of oil palm fruit bunch piles for ripeness grading using deep learning. *Scientific Data, 10*(1), 72. https://doi.org/10.1038/s41597-023-01958-x

Thapa, A., Horanont, T., Neupane, B., Klaylee, J., & Witayangkurn, A. (2025). Classification of oil palm tree conditions from UAV imagery using the YOLO object detector. *Big Earth Data, 9*(4), 860–886. https://doi.org/10.1080/20964471.2025.2491881

Ultralytics YOLO docs. (2024, September 30). *Ultralytics YOLO11*. https://docs.ultralytics.com/models/yolo11

Wang, Z., Zhu, Y., Ji, Z., Liu, S., & Zhang, Y. (2024). An efficient YOLOv8-based model with cross-level path aggregation enabling personal protective equipment detection. *IEEE Transactions on Industrial Informatics*, *20*(11), 1–12. https://doi.org/10.1109/tii.2024.3431045

Yipeng, L., & Junwu, W. (2024). Personal protective equipment detection for construction workers: A novel dataset and enhanced YOLOv5 approach. *IEEE Access*, *12*, 47338–47358. https://doi.org/10.1109/ACCESS.2024.3382817

Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., & Ren, D. (2020). Distance-IoU loss: Faster and better learning for bounding box regression. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(7), 12993–13000. https://doi.org/10.1609/aaai.v34i07.6999