IAES International Journal of Robotics and Automation (IJRA)
Vol. 15, No. 4, December 2026, pp. 1062~1084
ISSN: 2722-2586, DOI: 10.11591/ijra.v15i4.pp1062-1084

Oil palm fresh fruit bunch quality assessment system based on YOLOv8

Pelangi Soko Aji, Danang Lelono, Aufaclav Zatu Kusuma Frisky
Department of Computer Science and Electronics, Universitas Gadjah Mada, Yogyakarta, Indonesia

Article Info

Article history:

Received Jan 19, 2026
Revised Oct 6, 2026
Accepted Nov 10, 2026

Keywords:

Agriculture
Grading
Machine vision
Oil palm
YOLO

ABSTRACT

Oil palm (*Elaeis guineensis Jacq.*) was recognized as the primary source of palm oil because of its high oil productivity. Due to its low production cost, palm oil was projected to meet the increasing global demand and was produced extensively in Indonesia, particularly in Riau and Central Kalimantan. The palm oil industry had undergone rapid development with an increasing focus on product quality. Grading was essential for determining crude palm oil quality because the classification of fresh fruit bunches (FFB) influenced both the oil extraction rate and free fatty acid content. Artificial intelligence (AI), particularly computer vision and machine learning, was employed to address challenges arising from perspective variation, human error, weather conditions, and grading inconsistency. The AI-based system improved the efficiency, accuracy, and consistency of the grading process, thereby supporting sustainable palm oil production. YOLOv5 and YOLOv8 were implemented to evaluate FFB quality using conveyor-based scraping for object counting and visibility. The system replaced human grading operators and has been validated under industrial-scale conditions. Unlike prior studies evaluated under controlled laboratory conditions, this work reframes fresh fruit bunch grading as a real-time robotic-perception task and optimizes the detector for conveyor-speed industrial inspection, supplying reliable input for downstream robotic sorting and smart plantation management.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Danang Lelono
Department of Computer Science and Electronics, Gadjah Mada University
Bulaksumur, Kec. Depok, Kab. Sleman, D. I. Yogyakarta 55281, Indonesia
Email: danang@ugm.ac.id

1. INTRODUCTION

Agriculture has always evolved in response to the demands placed upon it, and few sectors illustrate this more clearly than palm oil production. *Elaeis guineensis Jacq.* is among the most productive oil-bearing crops in the world, capable of yielding approximately 10 tons of oil per hectare—far exceeding comparable crops such as soybean, canola, and sunflower [1], [2]. With global demand projected to reach 240 million tons by2050 [3], the pressure on palm oil operations to produce consistently and efficiently has never been greater. Meeting that pressure, however, requires more than simply expanding plantation areas; it demands a fundamental rethinking of how quality is assessed at every stage of production, beginning with the grading of Fresh Fruit Bunches (FFB).

FFB grading sits at the heart of palm oil quality control. The maturity of incoming bunches directly determines the oil extraction rate and free fatty acid content of crude palm oil (CPO), making accurate grading one of the most consequential steps in the entire milling process [4]. In practice, operators assess each bunch by examining mesocarp color, the presence and count of loose fruitlets, overall bunch weight,

spikelet coverage, and signs of parthenocarpy—a combination of visual and tactile judgments that requires at least a year of hands-on experience to perform reliably [5]. Under Standard Operating Procedure WI-PKS.GN-002, only 10% of daily incoming FFB is sampled per division, meaning that the majority of bunches are never formally inspected at all. Even within that sample, outcomes vary considerably depending on the operator's fatigue, experience, and working conditions. These are not minor inefficiencies; they translate directly into inconsistent oil quality, elevated operating costs, and, in some cases, deliberate grading fraud.

It is this combination of high economic stakes and operational fragility that makes FFB grading a compelling target for robotic automation. Rather than replacing human expertise wholesale, the goal is to provide a reliable, unbiased perceptual system that can function continuously at the throughput speeds demanded by industrial conveyor lines—one capable of feeding real-time quality data to downstream robotic sorters, autonomous plantation vehicles, and smart management platforms. Achieving this requires a detection model that is both accurate and fast enough to keep pace with physical material flow, and over the past decade, the YOLO family of architectures has emerged as the most consistently capable framework for meeting both requirements simultaneously [6].

Research into vision-based FFB assessment has grown substantially in recent years. YOLOv4 was shown to achieve a mean average precision of 87.9% for real-time mature bunch detection [7], and subsequent comparisons confirmed YOLO-based models outperforming SSD and EfficientDet alternatives across multiple evaluation criteria [8], [9]. Studies employing YOLOv8 have reported mAP50-95 values as high as 0.92 under controlled conditions [6], while broader investigations using CNN architectures, Faster R-CNN, SVM with HOG features, and artificial neural networks have collectively mapped out the landscape of what is technically feasible [10]–[18]. Each of these contributions has added a piece to the puzzle, yet most have been designed and evaluated in isolation from the robotic systems they would ultimately need to serve.

Building on these foundations, recent advancements in 2024 and 2025 have driven a shift towards more robust solutions. Zulkhoiri *et al.* [19] highlighted the robustness of color-based machine learning classifiers in dynamic agricultural environments. A 2024 systematic review by Alfatni *et al.* [20] confirmed that vision-based automated grading is the most scalable solution for overcoming the limitations of manual inspection. Furthermore, emerging research in 2025 has pushed the boundaries of detection models; Nur'aini and Rahardi [21] demonstrated the potential of the YOLO12s algorithm for classifying FFB ripeness with precision exceeding 93%, while other studies have successfully integrated fuzzy logic with YOLOv8 for nuanced harvest decision-making [22]. Moreover, comparisons with other models like AlexNet [23] and SVM [24] continue to validate the superior real-time performance of the YOLO architecture. Recent work has also demonstrated the versatility of YOLOv4 in autonomous mobile robot collectors arguably the clearest demonstration to date that vision-based detection can be embedded within a working robotic system rather than evaluated in isolation [25] and the effectiveness of lightweight YOLOv8 models for general fruit ripeness [26] and disease detection [27]. Taken together, these developments point toward a future in which intelligent vision is not merely a quality-checking tool but an integral component of autonomous plantation operations.

What remains missing, however, is a dedicated system that addresses the specific demands of conveyor-based FFB inspection at scale-one designed from the outset to provide the reliable, unbiased, high-throughput detection that robotic automation requires. This paper presents such a system, built around the YOLOv8 architecture and evaluated under conditions representative of real milling environments. The contributions of this work are threefold: first, it reframes FFB grading explicitly as a robotic perception problem, situating it within the broader context of precision agriculture and autonomous plantation management; second, it describes the design and implementation of a YOLOv8-based detection pipeline optimized for conveyor-speed operation and compatible with robotic inspection infrastructure; and third, it presents experimental results demonstrating that the system achieves grading accuracy and consistency that manual methods cannot reliably match, offering a practical and scalable foundation for the next generation of smart plantation systems.

2. RESEARCH METHOD

2.1. Dataset preparation

The data used in this study was carefully assembled with FFB obtained from a commercial palm oil plantation estate, covering around 2,692 hectares. The plantations include Nigrescent oil palm progenies, which were obtained from London Sumatra and Sucofindo, planted in the period ranging from 2000 to 2011. The plantations have diverse soil types, consisting of clay-mineral soil, swamp-mineral soil, sulfuric acid soil, in addition to shallow peat with medium depth and medium peat. The topography in the plantations was generally flat to wavy.

The production parameters specify an average weight of about 19.54 kg for every FFB bunch, contributing to a total annual produce of around 19.29 tons per hectare. The FFBs used in this study were strictly categorized according to their level of ripeness, among various other factors. The fundamental categories designed to define the classification include Unripe, Ripe, Overripe, Empty Bunch, and Abnormal, in addition to various miscellaneous classes that include Long Stalk and Small Bunch as shown in Figure 1. Such an extensive classification technique provided the rudimentary groundwork on which the object detection classification was to be designed in this study.



Figure 1. Object detection classes used in this research

The raw data collected was produced by post-processing video shots taken through a Hikvision DS-2CD2086G2-I CCTV camera. The camera has a resolution of 8 megapixels (1920x1080 pixels) and a frame rate of 25 frames per second (fps). The video data was stored in a 128 GB SD card. The image data set was generated by post-processing the video shots using VLC Media Player to extract all the individual images. This generated a set of 1,765 images, which contained a total of 13,904 objects. Another set of 300 images also was produced to reflect only the background without FFB.

For the purpose of labeling, the LabelImg software (Figure 2) was used to mark the FFB objects in the images created. In order to ensure that the images have uniform visual quality, the concept of controlled lighting was applied. The controlled lighting involved the use of warm white light (3000K) and cool daylight (5000K) LED light sources, which consisted of five LED light sources (Figure 3) amounting to a total power output of 350 watts. The controlled lighting applied in the experiment removed the effect caused by images taken during the night compared to those taken during the day. A conveyor scraper (Figure 4) was also applied during the data gathering process to enable the controlled presentation of the FFB during imaging. Finally, the simple architecture diagram was shown in Figure 5, all the devices were connected via gigabit ethernet interface.

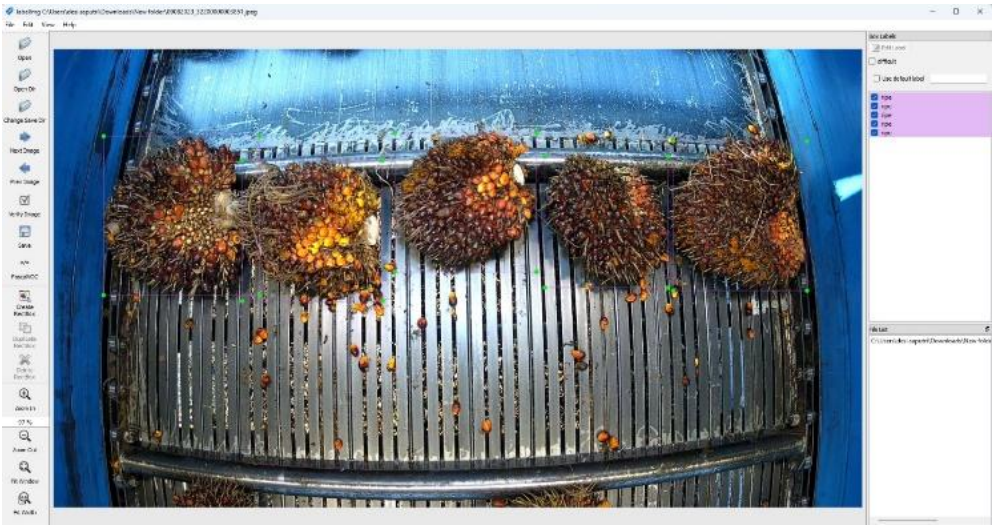


Figure 2. LabelImg Software for object detection annotation



Figure 3. Layout and installation of LEDs



Figure 4. Conveyor scraper utilized to organize the FFBs, ensuring optimal visibility for CCTV recording

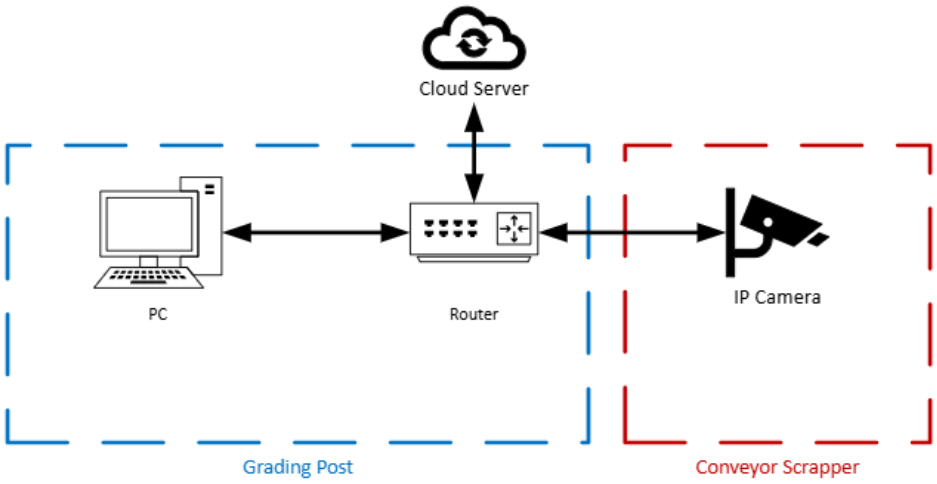


Figure 5. System architecture diagram

To ensure annotation quality and reproducibility, every bounding box was drawn against a fixed class definition derived from the plantation standard operating procedure, and the labeled set was cross-checked by quality-control personnel to resolve ambiguous or borderline bunches before training. Imaging conditions were standardized through the controlled 3000 K and 5000 K LED arrangement described above, which fixed illumination across daytime and nighttime acquisition and removed the colour and shadow variability that would otherwise bias mesocarp-based ripeness labels.

2.2. Data pre-processing and augmentation

Data preparation procedures were begun with a split of the labeled dataset into separate training and validation datasets. This splitting specifically allocated 80% of the images to training model development and the other 20% to validation. To increase the size and quality of the dataset, data augmentation procedures were performed. The augmentation methods involved vertical flip, horizontal flip, and a combination of

vertical and horizontal flip techniques as shown in Figure 6. After the completion of this augmentation procedure, the total images were significantly increased to 7,060 images, including a total of 55,616 objects in all the specified classes. This extensive preparation procedure was imperative in establishing object detection and classification models using a significantly rich dataset.

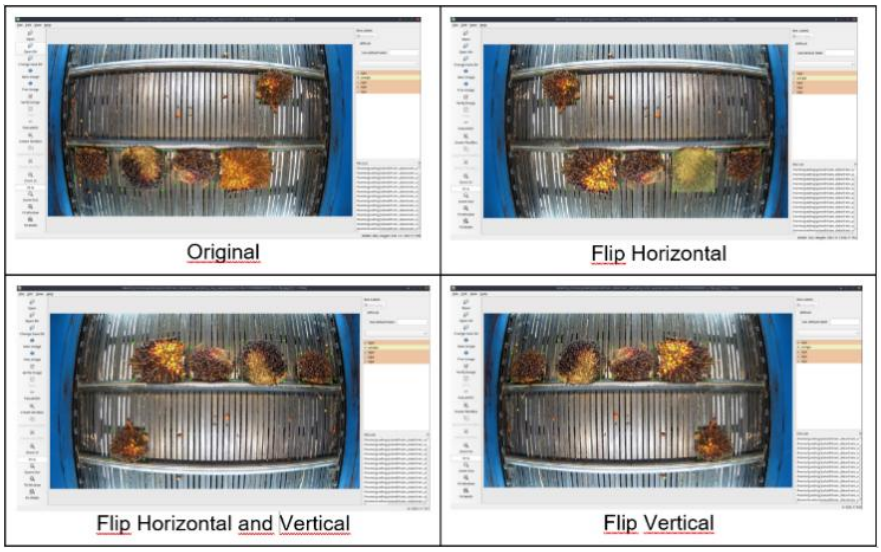


Figure 6. Augmentation methods used in this study to ensure varied dataset

Horizontal, vertical, and combined flips were selected deliberately because fresh fruit bunches arrive on the conveyor in arbitrary orientations; flip transformations expose the detector to this orientation variability while preserving label validity and, unlike photometric or geometric distortions, they introduce no colour or shape artefacts that could corrupt ripeness cues. This orientation-invariance strategy is consistent with the flip-robustness results reported later and improves generalization without inflating the dataset with unrealistic samples.

2.3. YOLO

The starting point for the development of our approach is the family of computer vision models referred to as the you only look once (YOLO) models. Developed by Joseph Redmon *et al.* 2016, the YOLO algorithm brought a massive shift to the field of object detection as it was the first real-time algorithm that adopted the end-to-end approach for a single pass scan, thus becoming the most notable development in the field of computer vision by increasing the accuracy and the speed of object detection [28].

YOLOv5 was introduced by Glenn Jocher in 2020 and is maintained as an open-source platform by Ultralytics. It is well-known for its usability and easy training and inference. YOLOv5 is built on top of PyTorch. It has various upgrades compared to its previous versions. One of the major components of YOLOv5 is the AutoAnchor function. With the help of this function, the anchor boxes are dynamically adjusted according to the characteristics of the training data being used. This model uses other stabilization functions like Mosaic and MixUp. YOLOv5 models are available in scalable models (nano, small, medium, large, and extra-large) depending on the hardware required [28].

YOLOv8, the newest version released in January 2023, advances not just the object detection function but encompasses additional modules such as image segmentation, pose estimation, and image classification. Architectural advancements include the introduction of the C2f module and the anchor-free architecture Figure 7. Both components are responsible for the increase in accuracy and the increase in speed. This model utilizes the complete intersection over union (CIoU) Loss function and the distributional focal loss function. Additionally, the binary cross-entropy loss function is utilized for classification. Its image segmentation model, the YOLOv8-Seg, achieved state-of-the-art performance on the benchmark with the absence of an impact on inference speed or efficiency [28].

This training involved the use of the lighter and faster models of both architectures: YOLOv5nu, YOLOv5su, YOLOv8n, and YOLOv8s models. Additionally, all image resolutions during training were tested at fixed sizes of 640 pixels and 1280 pixels respectively. Additionally, the maximum training of 2,000

epochs was ensured with an Early stopping mechanism and patience of 20 epochs to avoid overfitting during training. Additionally, the computational setup utilized in training had an NVIDIA RTX 3070 card along with an Intel Core i7-11700 processor and 32GB of DDR4 RAM operating at 2900MHz. To ensure rational use of resources during training, scripts in the Jupyter Lab environment were utilized to iteratively check and fine-tune the highest possible training batch sizes starting with larger sizes and incrementally decreasing them whenever necessary. Additionally, this training utilized Ultralytics 8.2.79.

For reproducibility, all models were trained with the Ultralytics default optimizer and learning-rate schedule; the deliberately tuned hyperparameters were the input resolution (640 and 1280 px) and the batch size, which was maximized iteratively for the available GPU memory. Early stopping monitored the validation mean average precision and terminated training after 20 epochs without improvement, so the reported best epoch corresponds to peak validation performance rather than a fixed epoch budget.

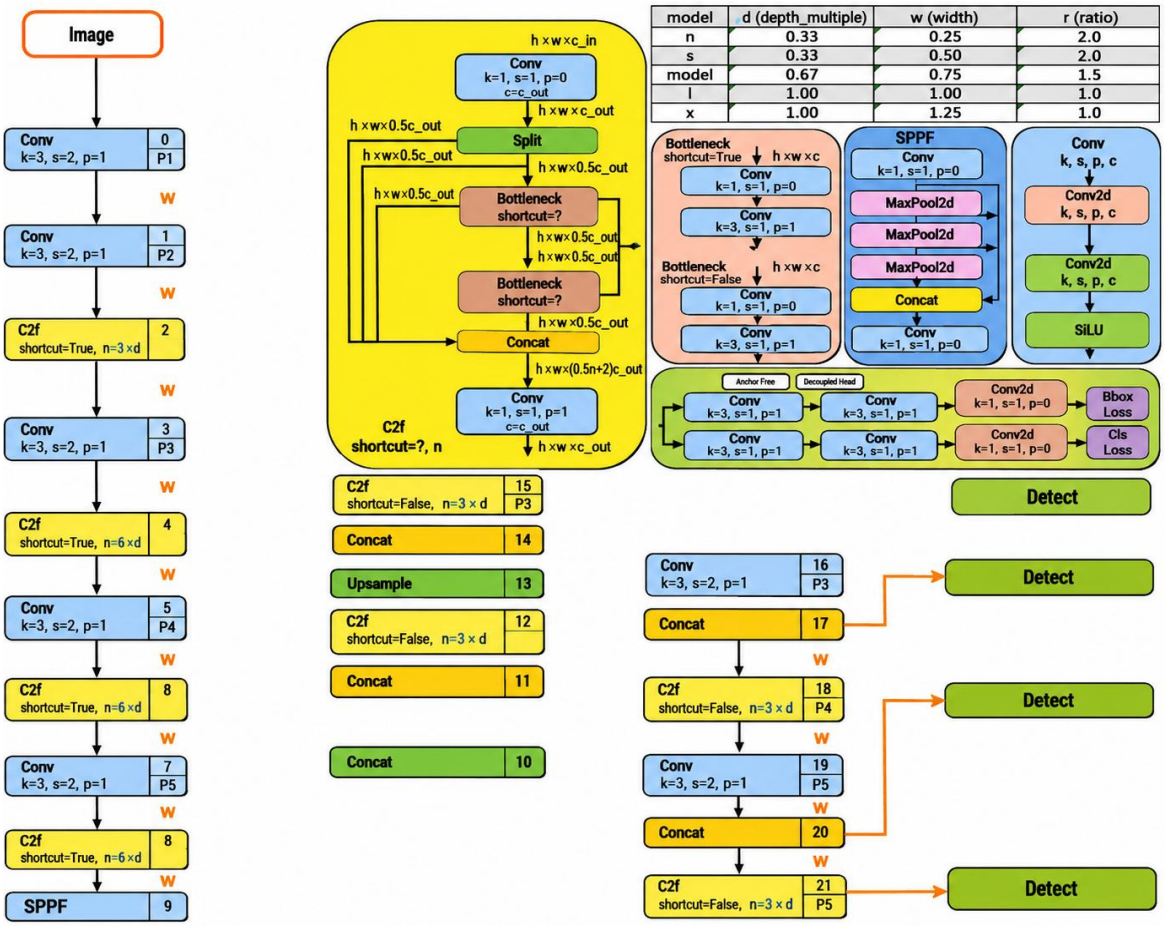


Figure 7. YOLOv8's architecture

2.4. Evaluation method

In the course of the research, the assessment was done by a thorough comparison of the three different FFB grading systems, namely the human grade done by the grade operators, the visual grade done by the quality controllers based on the recorded CCTV footage, and the computer vision grade.

The manual grading involved the grading operators examining the FFB (Figure 8) as soon as they were discharged from the truck to the floor at the loading ramp. The grading operators were supposed to manually turn, count, and grade each FFB, which was recorded using a hardcopy reception form. To facilitate the video shooting of the FFB, which had previously undergone manual grading, the FFBs were loaded into a conveyor scraper using a wheeled loader (Figure 9). The FFBs required the conveyor, which ensured that the FFBs were stretched apart and in full view of the CCTV camera, thus eliminating any chances of overlapping and stacking. While it was possible to make the assessment in real time using YOLO, it was intentionally recorded into a microSD card for further controlled analysis.



Figure 8. The FFB are graded manually



Figure 9. The wheeled loader is used to load FFB into the conveyor

The video footage obtained from the recording process was downloaded and manually checked by quality control personnel aimed at conducting the visual grading. This created a second truth. It is essential to note the use of the same video footage to thoroughly examine a number of computer vision models and parameters. This examination focused on the essential parameters, which included image dimensions (800px, 1024px, and 1280px), IoU (0.5), the confidence level (0.2 and 0.45), and the RoI (0.3 and 0.4) parameters.

The video recordings were systematically segregated according to plantation regions (Division OA, OB, OC, and OD). As a cluster of plantations covering approximately 700 hectares is contained within every division, such division was used to standardize variations regarding FFB progenies, soil, topography, and microclimate conditions. Thus, different models and optimal parameters were used for every division. The result was a comprehensive testing scheme that enabled a thorough comparison between conventional and automated systems used to determine quality FFB.

To measure the error in classification and detection output, Mean Absolute Percentage Error, or MAPE, was used. MAPE is a commonly accepted mathematical formula for determining accuracy in data mining algorithms, like other formulas for prediction, classification, and other algorithms that entail forecasting and detection. The formula calculates the difference between estimated and actual values, and it is one of the most accurate methods used in determining accuracy levels [29]. The formula for MAPE is as:

$$MAPE = \left(\frac{\sum_{t=1}^{n}\left|\frac{a-b}{a}\right|}{n} \times 100\%\right) \quad (1)$$

This formula was then employed for the identified key factors, namely the number of bunches, the entire ripeness stage, and the classification of both ripe and unripe FFB productions. Through the use of MAPE, the research authors were able to assess the error-consistency of the classification performance and identify areas for potential improvement.

MAPE was computed independently for each ripeness class and for the total bunch count within every plantation division, using the manual reception-form tallies as ground truth, so that class-level and aggregate errors could be reported separately (Table 2). Because each configuration was evaluated in a single pass over the recorded footage, the confidence intervals reported in Table 5 describe variability across plantation divisions and parameter settings rather than repeated-run sampling error; this constraint is noted among the study limitations.

3. RESULTS AND DISCUSSION

3.1. YOLO training result

This particular study assessed the performance of two popular object detection models, YOLOv5 and YOLOv8, using their optimized versions respectively. This included an examination of the YOLOv5 models: YOLOv5nu, YOLOv5su, YOLOv5x; as well as an exploration of YOLOv8n-YOLOv8s in the YOLOv8 model testing. Training parameters such as image size, batch size, as well as the number of epochs, had been carefully selected in accordance with the GPU capability as well as considerations for efficient model training as outlined in model documentation. For a detailed understanding of the training outcomes as outlined in the particular study along with detailed parameter selections, refer to Table 1.

The training result of the YOLO model, as shown in Table 1 and Figures 10 to 15, indicates a large discrepancy between these two model families when tested under different sizes of images as well as

different batch sizes. Each of these six figures is presented as a pair of panels: panel (a) restricts the horizontal axis to the first 33 epochs, the point at which the YOLOv8s run was stopped, so that the early behaviour of all four models is compared on a common scale, while panel (b) shows the complete training history to convergence. Training curves for object detection models show a smooth decrease in loss values with epochs, indicating model stability. Figure 10(a) shows the box regression loss falling steeply from above 1.0 to between 0.758 (YOLOv5nu) and 0.878 (YOLOv8s) within the first 33 epochs, with YOLOv5nu descending fastest, whereas Figure 10(b) extends the same curves over the full run, where YOLOv8s settles lowest at 0.400 and YOLOv5nu, YOLOv5su and YOLOv8n converge to 0.511, 0.485 and 0.508 respectively. The class loss behaves similarly: Figure 11(a) captures the sharp initial drop from about 2.25 to between 0.590 and 0.806 over the first 33 epochs, and Figure 11(b) shows the subsequent slow decline to a plateau between 0.239 and 0.365. The distribution focal loss follows the same pattern, with Figure 12(a) showing the early decrease from roughly 1.37 to between 1.092 and 1.158, and Figure 12(b) showing convergence to between 0.892 and 0.955. Also, the fact that there are no sudden spikes in loss values for training as well as validation curves signifies no overfitting. All the models are optimizing on training data without sacrificing generalization performance, ending up with minimized bounding box prediction error values, which are highly accurate as illustrated by detection accuracy.

Table 1. Comparison of YOLO model training results

No	Model	Best Epoch	Batch Size	Img Size	Cls Loss	Box Loss	Model Size (mb)
1	YOLOv5nu	252	8	1280	0.4143	0.6249	5,24
2	YOLOv8s	261	8	1280	0.2347	0.3618	22,07
3	YOLOv5su	423	8	1280	0.2335	0.3483	18,16
4	YOLOv8n	658	10	1280	0.3079	0.4704	6,24

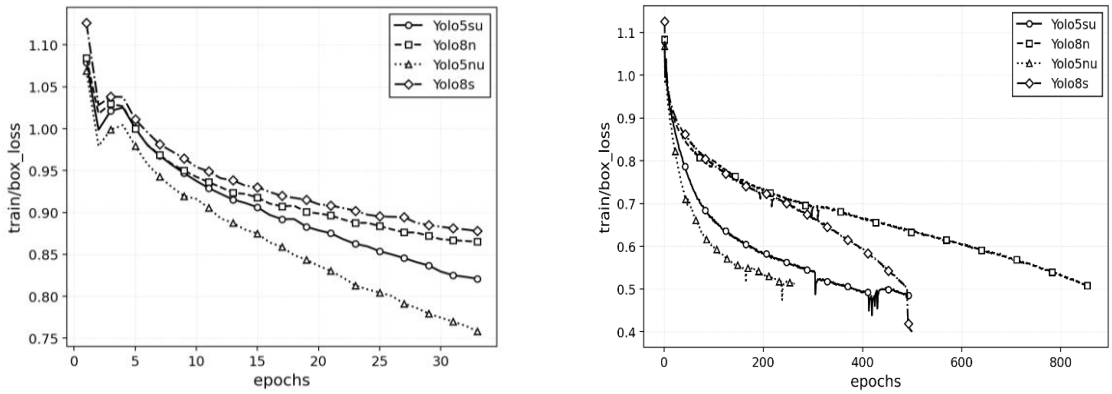


Figure 10. Graph of train/box_loss (a) 33 epochs and (b) max epochs

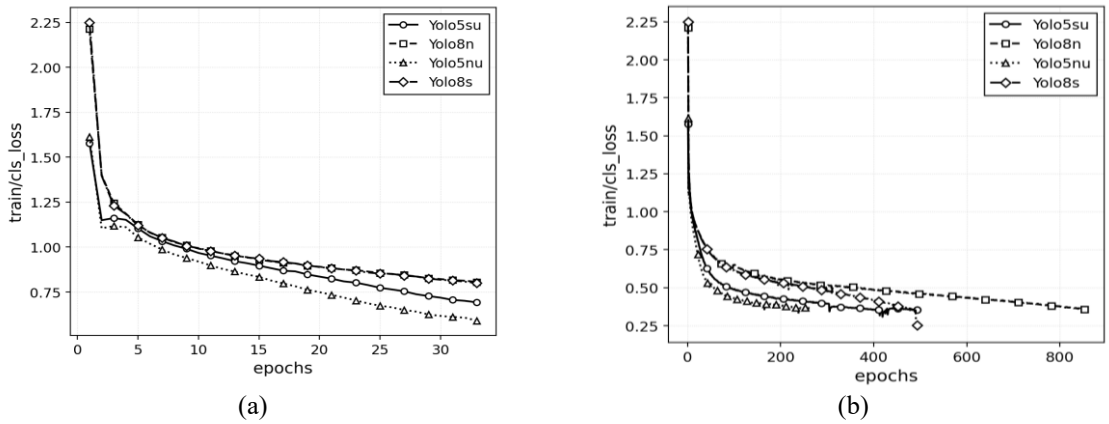


Figure 11. Graph of train/cls_loss (a) 33 epochs and (b) max epochs

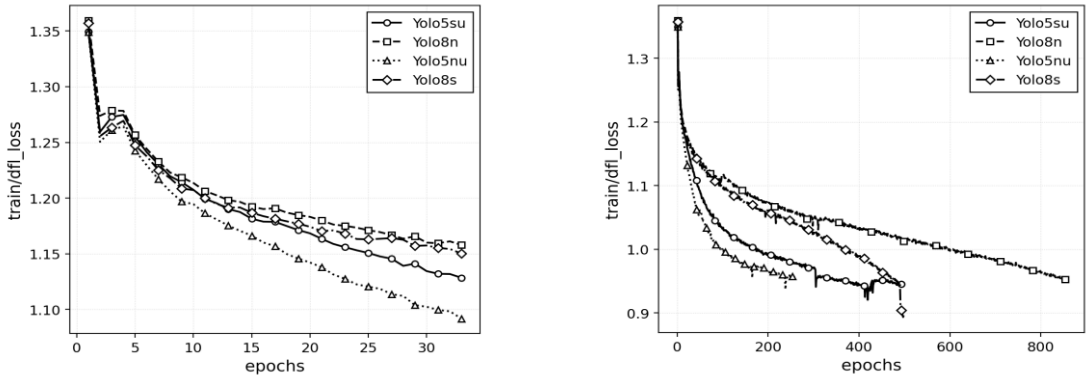


Figure 12. Graph of train/dfl_loss (a) 33 epochs and (b) max epochs

The metrics for the accuracy of object detection improved significantly throughout the training process. Precision (Figure 13) and recall (Figure 14) gradually increased with each progressing epoch and reached a plateau close to their maximal values. Figure 13(a) shows that within the first 33 epochs precision is already widely separated, ranging from 0.828 for YOLOv8n to 0.943 for YOLOv5nu, while Figure 13(b) shows that this gap closes almost entirely once training completes, with all four models finishing between 0.946 and 0.957. Recall follows the same convergence: Figure 14(a) shows values between 0.834 and 0.901 at epoch 33, and Figure 14(b) shows the fully trained models grouped between 0.895 and 0.904. The mean average precision is reported over the complete training run in Figure 15, where Figure 15(a) presents mAP_0.5 and Figure 15(b) the stricter mAP_0.5-0.95; the former converges between 0.949 and 0.965 and the latter, being far more sensitive to localisation quality, settles lower between 0.896 and 0.931. At the point of convergence, the most accurate model (YOLOv8-nano) performed very well: It held precision at around 0.9525 and recall at around 0.8949, alongside class mean Average Precision at 50% Intersection over Union (mAP50) of around 0.9591 and the more stringent mAP50-95 of around 0.9158 (Figure 15). These numbers suggest that this model was capable of detecting and distinguishing correctly a very high percentage of actual objects (fresh fruit bunches) while reporting only a negligible number of false alarms. For comparison, a YOLOv8-small model performed significantly poorer on this dataset (as would be expected from an inadequately trained model) although precision was around 0.616 and recall was around 0.653 (mAP50 ~0.656, mAP50-95 ~0.456) after being trained for 33 epochs, which is precisely the truncated window isolated in panel (a) of Figures 13 and 14. The large difference in the small model's performance in early epochs relative to the fully trained nano model's performance serves the need for thorough training of the model. For all models that had fully completed their training processes, results showed high-performance levels, which implies that this model (specifically YOLOv8-nano) is highly appropriate for this fruit bunch detection and grading application.

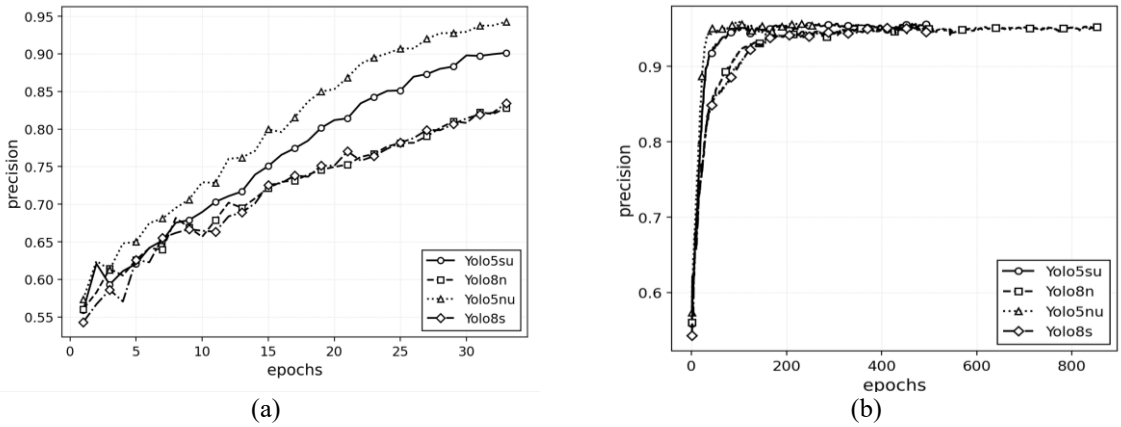


Figure 13. Graph of precision (a) 33 epochs and (b) max epochs

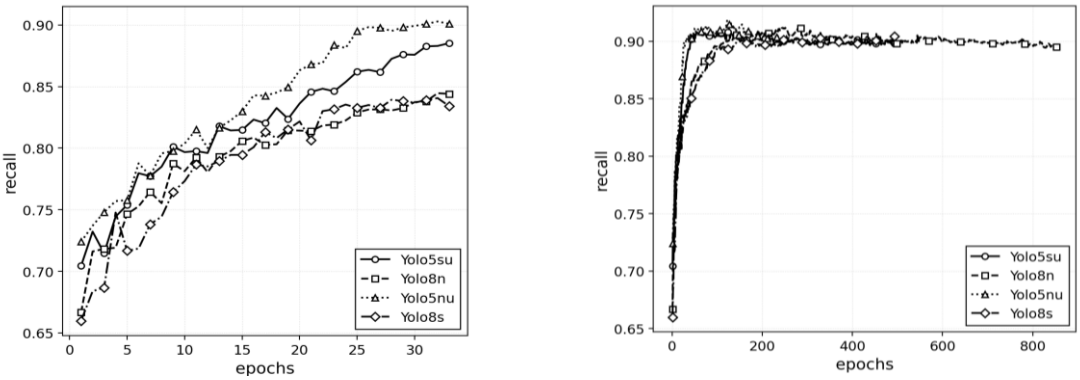


Figure 14. Graph of recall (a) 33 epochs and (b) max epochs

Model comparisons and experimental variants confirm these observations. Specifically, YOLOv5 baselines, in their small/nano variants, could produce accuracy and mAP comparable to YOLOv8-nano models when fully trained (Figure 15). For example, YOLOv5-small could reach a precision of about 0.957 and an mAP@50 of 0.9599 at about epoch 500, reaching mAP@50-95 values of 0.9280, slightly better than YOLOv8-nano models. In another YOLOv5-nano experiment, precision levels of about 0.954 and recall values of 0.904 (along with mAP@50 values of 0.9648, and mAP@50-95 values of 0.9307) could be reached at about epoch 265, showing better recall and mAP values than YOLOv8-nano models, but still in a marginally better manner. These comparisons show that all high-capacity models assessed in this research possess excellent accuracy in FFB object detection when well-trained, in which case YOLOv5 models possess a slight advantage in average precision, but in a setting not significantly different in practice, YOLOv8-nano also performs equally well, possibly adopting a notably lighter model. In another arrangement of various experimental models, including "sampling-only" and "sampling-only revisions" subsets, detection qualities in high ranges were also achieved (about 0.95+ in both precision and mAP, at termination), suggesting that differences in data preprocessing impacted to conclude, from the analysis of training dynamics and accuracy metrics, it is clear that YOLO-based object detectors are very effective in the task of quality assessment in FFB. The loss curves tend to be low in all cases, which is indicative of an optimal parameter fit, besides precision/recall curves and map scores tend to be high. The best-performing model, YOLOv8-nano, is very accurate in object detection (precision ≈ 95%, recall ≈ 89-90%, MAP50 ≈ 96%), comparable to other YOLOv5 variants with similar parameter counts. The above observations validate the object detection strategy, where the models are capable of correctly identifying ripe or unripe/defective regions in fruits with negligible misses/false alarms, thus validating YOLOv8 for the purpose of automatic quality assessment of FFB.

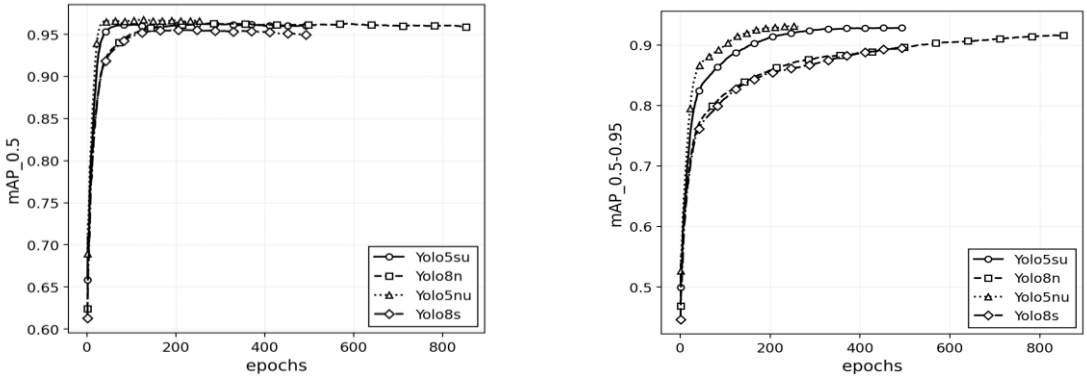


Figure 15. Graphs of (a) mAP_0.5 and (b) mAP_0.5-0.95

In terms of the size of the models, the variations were considerable, ranging from 5.18 MB (YOLOv5nu) to 22.07 MB (YOLOv8s). This also shows the balance between the complexity of the models and the performance of the object detection tasks. In spite of the size, the YOLOv8 models showed better accuracy

and lower losses. This makes the models appropriate for the FFB grading tasks requiring high accuracy but with the cost of additional computational requirements. It is clear that the YOLOv8 models generally performed better than the YOLOv5 models in terms of accuracy and losses, especially when using larger images.

Interpreting these results, the anchor-free C2f design of YOLOv8 reaches competitive accuracy with far fewer parameters than the anchor-based YOLOv5 variants, which explains why the lightweight YOLOv8-nano matches larger models while remaining attractive for edge deployment. The residual misclassifications concentrate in the Overripe, Empty Bunch, and Abnormal categories, consistent with their severe under-representation in the data (2.00%, 0.24%, and 1.03% of samples, respectively) and their high morphological variability; for deployment this implies that rare-class reliability, rather than overall accuracy, is the binding constraint on a fully automated sorting line.

3.2. Implementation result

The YOLOv5nu model was tested on several flipped versions of the original images to verify its orientation independency in detecting FFBs. This process included horizontal flip, vertical flip, and combined flip transformations. As seen in Figure 16, FFBs were detected in all flipped images with high accuracy regardless of orientation. This experiment revealed that there was no effect of image orientation on the accuracy of the detections, and FFBs could be identified in all orientations. The choice of the YOLOv5nu model in this testing was due to its simplicity and low processing power, making it an appropriate choice for initial testing.

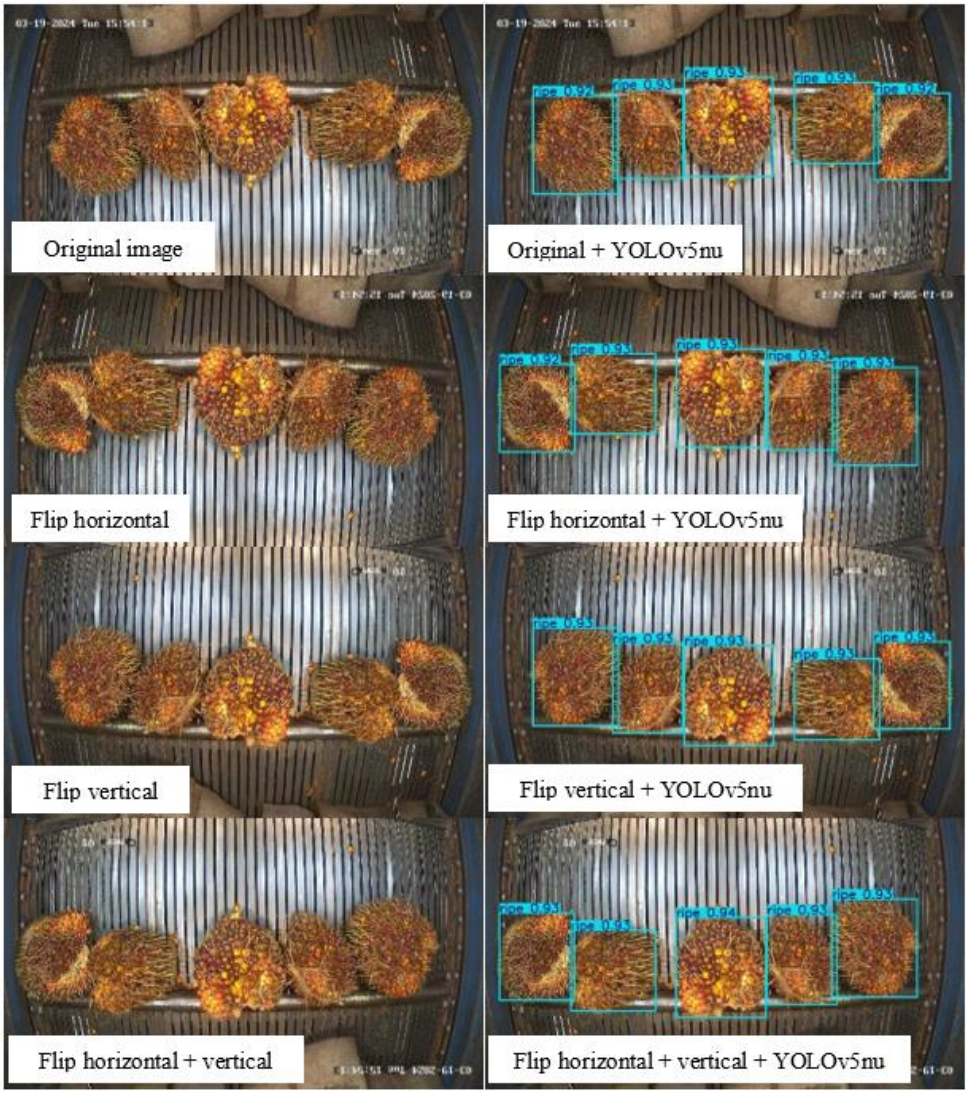


Figure 16. Various augmentation methods tested on YOLOv5nu

Other evaluations involved all the FFB categories used for the training dataset, including unripe, ripe, overripe, empty bunch, abnormal, and long-stalk images. Four separate models were also validated using different parameters like the size and confidence level and/or IoU. While all the models worked well for the identification of the unripe FFB (Figure 17), some models and settings showed cases of misclassification for ripe FFB. Models like YOLOv5nu and settings including the 1280px size and 0.43 confidence level, as well as the YOLOv5su model and settings including the 800 px size and 0.20 confidence level, showed the tendency to misclassify ripe FFB as unripe.

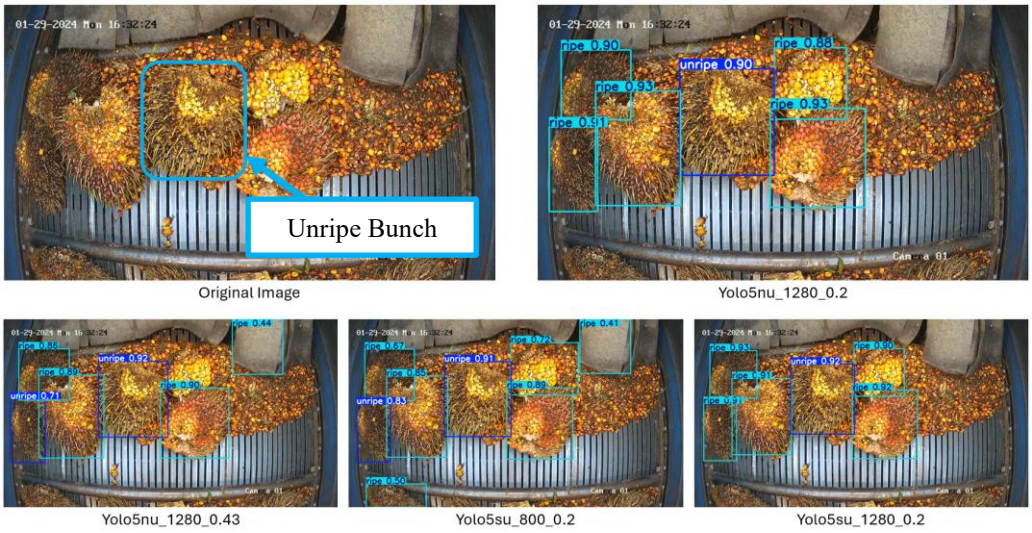


Figure 17. Unripe bunch tested on various models and parameters

Overripe bunches: The results on overripe bunches show that all models performed well, with one exception (Figure 18). The critical failure was in the YOLOv5su settings (800 px/0.2), which incorrectly delineated an overripe bunch into two different things: an overripe bunch and a ripe bunch. This setting sometimes also mistakenly marked ripe bunches as unripe.

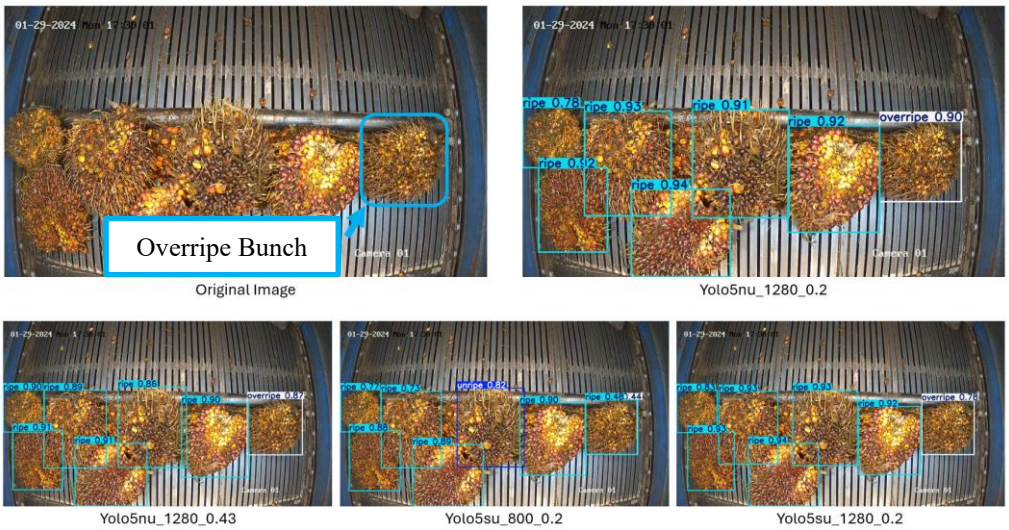


Figure 18. Overripe bunch tested on various models and parameters

The results for the empty bunch recognition test (Figure 19) showed a positive performance, except for the model YOLOv5su (1280 px/0.2), which tended to misclassify empty bunches as overripe ones. At the

same time, the model did occasionally mistake the overripe bunches for the empty ones, suggesting a likely confusion presumably because of the closeness of the two classes due to the small number of loose fruitlets.

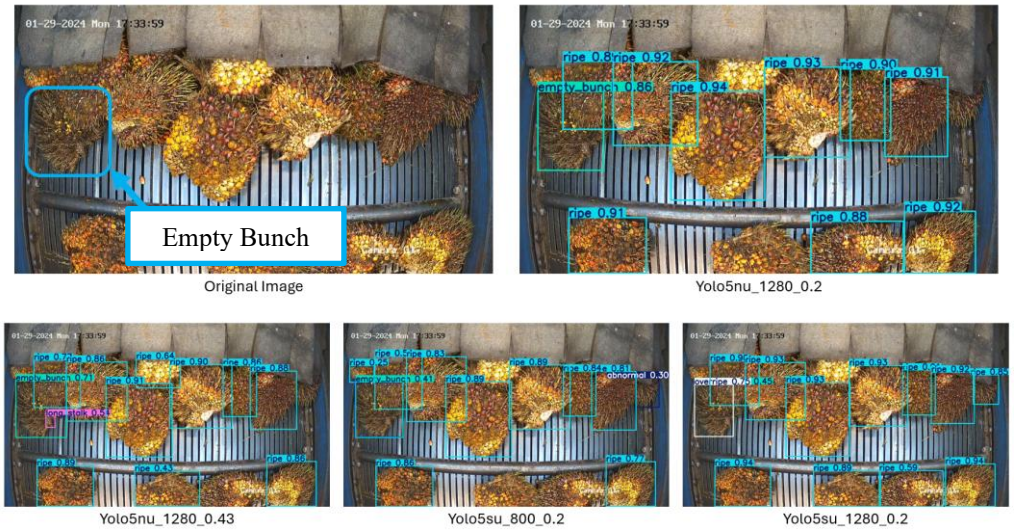


Figure 19. Empty bunch tested on various models and parameters

The detection of an abnormal bunch is the most difficult task. Only the YOLOv5nu with the setting 1280 px/0.2 has shown a specific error of misclassification when it detected the abnormal bunch as ripe with the highest confidence of 0.87 (Figure 20). The diversity and irregular shapes of the abnormal category are identified as the main cause of this problem.

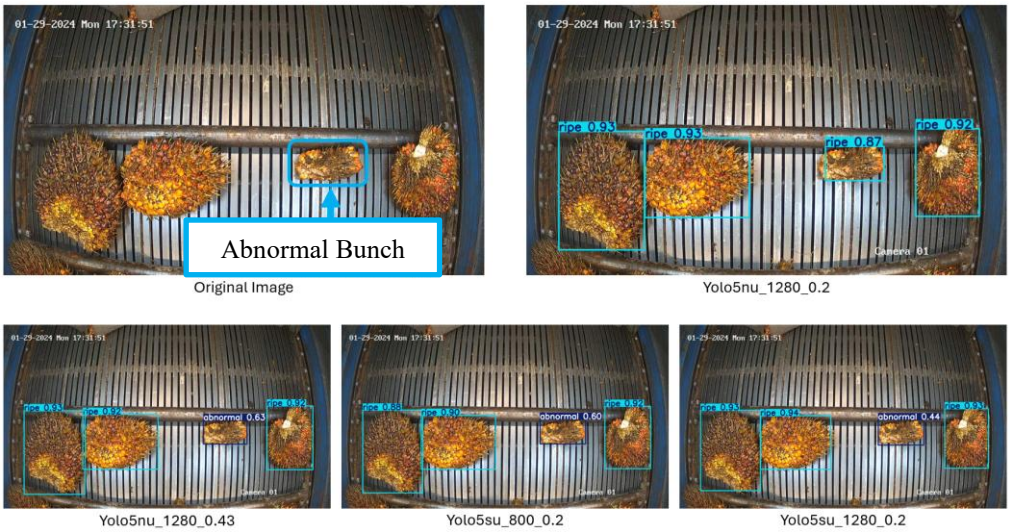


Figure 20. Abnormal bunch tested on various models and parameters

The last class considered was the long stalk class. As illustrated in Figure 21, the YOLOv5nu (1280 px/0.2) and YOLOv5su (1280 px/0.2) models were successful in identifying long stalks within the test images. The YOLOv5nu (1280 px/0.43) and YOLOv5su (800 px/0.2) models were, however, unsuccessful in accurately identifying long stalks. The model being unsuccessful is attributed to two possible reasons: the size of the long stalk objects within the test images may be too small, or long stalk objects may lack enough representations within the training data.

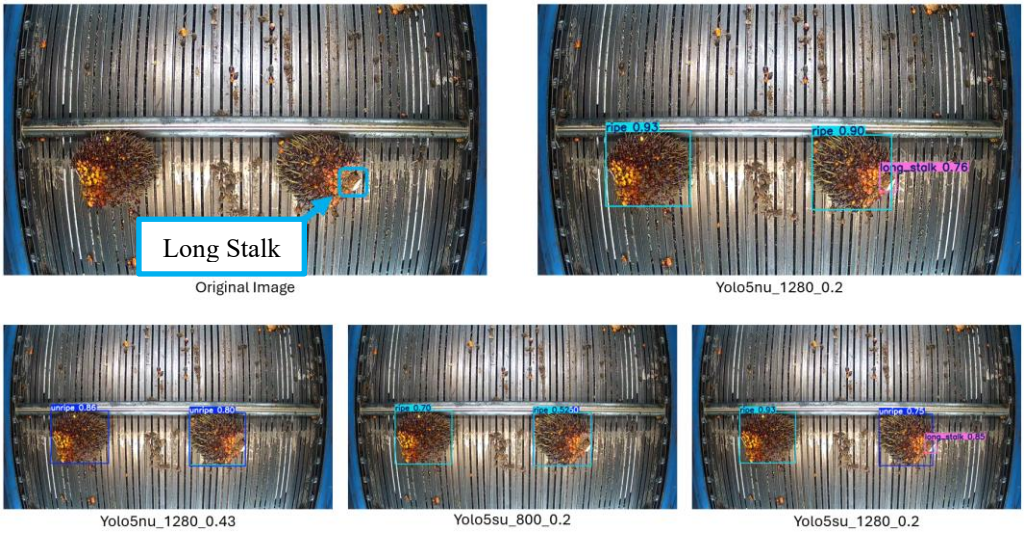
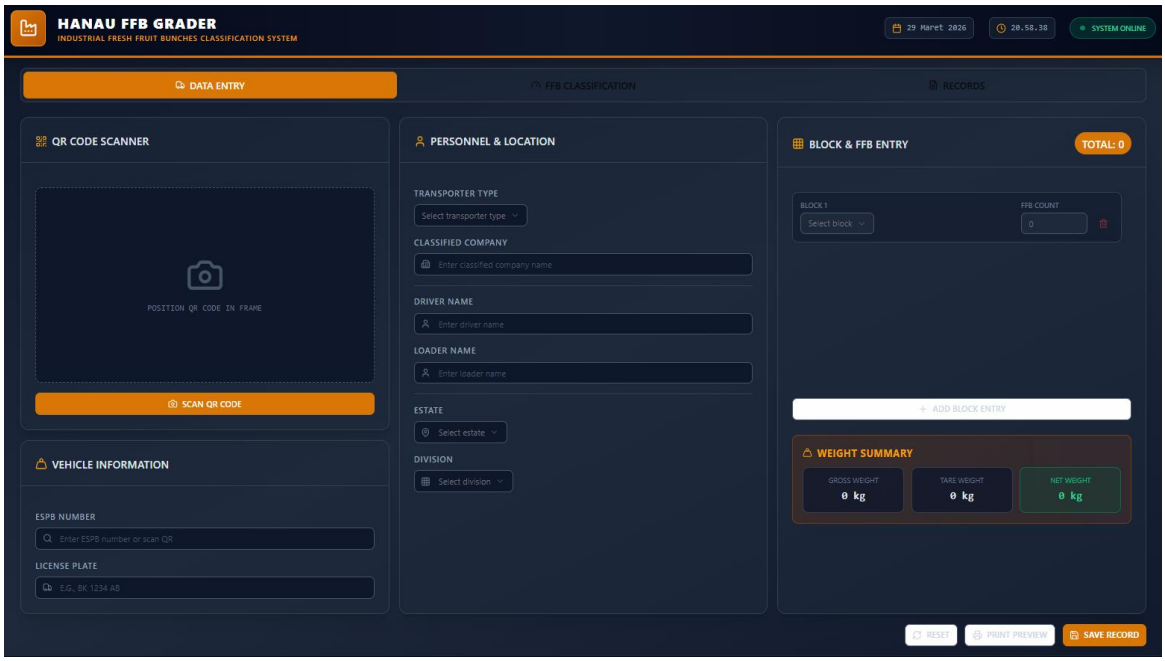


Figure 21. Long stalk tested on various models and parameters

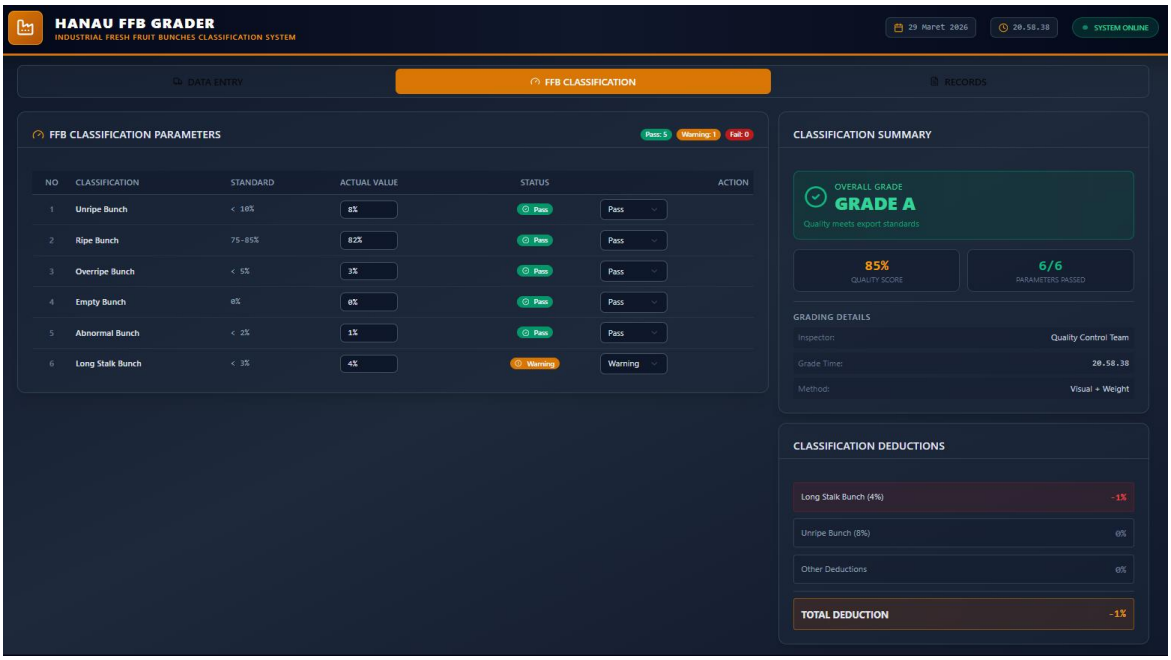
The user interface (UI) of the developed application is designed in a manner that facilitates intuitive quality assessment as shown in Figure 22. The user interface comprises counters for each of the identified classes, dynamic bounding boxes for FFBs, a specified Region of Interest (RoI) line, and a manual completion button for quality assessment. The completion button can alternatively be replaced with a timer, whereby when a certain time elapses without new FFB identifications, the system automatically completes the assessment and displays a final report. In the top-left corner of the interface, the total number of identified and classified FFBs is shown prominently, along with their respective class numbers. The numbers comprising the text of these counters are color-coded to enable easy identification with rapid identification of their respective bounding boxes on display. Each bounding box contains the name of its respective object class along with a unique ID of each object on display. These boxes do not contain confidence percentage values to avert obstruction of visible display of FFB objects. The green-colored RoI line indicates the motion of FFBs along with an increased counter, with an automatic change of counter values for each new object that traverses the region of interest. In situations whereby an already tracked object reverses and traverses again, the counter assumes its initial values without allowing double counting. Figure 23 demonstrates the user interface of the proposed system. Specifically, it details the modules used for logging the FFB's point of origin and logistical carrier (Figure 23(a)), as well as the interface utilized for recording grading outcomes (Figure 23(b)).



Figure 22. The user interface of system inference



(a)



(b)

Figure 23. The user interface of (a) harvest origins and delivery vehicles of FFB and (b) grading result

The results of the oil palm FFB quality assessment system implementation, using both YOLOv5 and YOLOv8 models, are shown in three different tables. Table 2 highlights the MAPE performance based on the object detection outcomes. In highlighting the best-performing setups, only the top three parameter settings that produce the lowest MAPE for each division are picked, based on 48 different parameter settings tested systematically. The classes being analyzed via MAPE are ripe, unripe, overripe, empty bunch, and abnormal bunch classes. To ensure industry relevance and to validate industry-level reliability, this comparison analysis focuses on MAPE performance for total FFB, ripe, and unripe classes, which are identified to be of highest priority to the industry.

Table 2. The top 3 MAPE for each division

No	Division and Parameter	MAPE (%)					
		Total FFB	Ripe	Unripe	Over ripe	Empty Bunch	Abnormal
1	OA 1	0.25	0.18	4.17	33.33	83.33	73.58
2	OA 2	0.80	0.38	4.17	11.11	58.33	18.87
3	OA 3	1.15	0.53	5.56	18.89	50.00	16.98
4	OB 1	0.73	1.40	0.00	9.80	85.71	90.00
5	OB 2	0.58	0.55	3.03	17.65	100.00	6.67
6	OB 3	0.41	0.37	12.12	7.84	42.86	10.00
7	OC 1	0.95	1.85	3.23	14.93	0.00	89.47
8	OC 2	1.87	0.96	3.23	102.99	100.00	63.16
9	OC 3	0.17	0.07	67.74	10.45	60.00	89.47
10	OD 1	0.79	2.42	5.56	15.38	80.00	88.89
11	OD 2	0.59	2.73	0.00	43.59	60.00	73.33
12	OD 3	0.56	2.66	5.56	39.74	100.00	75.56

Table 3 shows the entire list of parameter settings derived from the results illustrated in Table 2. The table contains the YOLO model type (for example, YOLOv5nu and YOLOv8s), the size of the input images during inference, confidence threshold values, IoU threshold levels, and the size of the RoI. The parameter settings were tested and explored to examine 48 possible settings. This is crucial for identifying the best parameter setting for the FFB classification and detection techniques. The sensitivity of the accuracy was tested for all possible settings on each division of the plantation. For ease of interpretation, the table indicates the setting where the parameter values are the same (for example, setting OA 1 and OB 1, setting OA 3 and OD 3, and setting OC 1 and OD 1). Table 4 contains the total counts of FFB that were classified for each division. The counts were used as the ground truth against which the MAPE values in Table 2 were determined. The counts offer critical information on the performance of the automated system and its capability to produce accurate results under diverse conditions.

Table 3. Best parameter settings for top 3 in each division

No	Division and Parameter	Model	Image size	Confidence	IOU	RoI	FPS
1	OA 1	YOLO5nu	1280	0.20	0.5	0.3	79
2	OA 2	YOLO5nu	800	0.20	0.5	0.3	79
3	OA 3	YOLO5nu	800	0.43	0.5	0.3	79
4	OB 1	YOLO8s	1280	0.20	0.5	0.3	51
5	OB 2	YOLO5nu	1280	0.20	0.5	0.3	79
6	OB 3	YOLO5nu	1280	0.43	0.5	0.3	79
7	OC 1	YOLO5su	1280	0.20	0.5	0.3	48
8	OC 2	YOLO8n	1280	0.20	0.5	0.3	61
9	OC 3	YOLO5su	800	0.20	0.5	0.3	67
10	OD 1	YOLO5su	1280	0.20	0.5	0.3	48
11	OD 2	YOLO5nu	1024	0.43	0.5	0.3	57
12	OD 3	YOLO5nu	800	0.43	0.5	0.3	79

Table 4. Numbers of FFB for each class and division

Division	Total					
	FFB	Ripe	Unripe	Over ripe	Empty Bunch	Abnormal
OA	3637	3410	72	90	12	53
OB	4689	4568	33	51	7	30
OC	2938	2811	31	67	10	19
OD	3043	2897	18	78	5	45

Concerning the distribution of the data sets or samples (Table 4), it was observed that the number of empty bunches was remarkably small, at 0.24%. This was mainly because harvesting difficulties due to long harvest cycle rotation, missed harvest schedules, and obstructed harvest paths due to flooding led to the loss of most of the fruitlets, thereby leaving below 10% intact. There were also few overripe bunches at 2.00%. The main distinguishing features of overripe bunches are their remaining intact fruitlet percent ranging from 50% to 10%. Ripe bunches constituted the major portion of the total at 95.66%, which was a standard range suitable for oil palm harvesting. Unripe bunches develop from inexperienced harvesters. Abnormal bunches comprised 1.03% of the total, which included bunches containing genetic abnormalities or unpollinated female flowers. The data set was obtained from 57 trucks that carried FFB from plantations to the oil palm mill each day (Figure 24). Each truck normally carried 300 FFBs that weighed 7 tons.



Figure 24. The FFB transporter trucks are queueing in front of POM

Analysis in Table 2 sheds light on the MAPE for different parameter sets and plantation division combinations. For different parameter sets and plantation division pairs, it is found that for total FFB and Ripe Bunch classes, parameter division pair OA 1 has the lowest MAPE at 0.25% and 0.18%, respectively. For total FFB class again, parameter division pair OC 3 has the lowest MAPE value of 0.17%, indicating high efficiency in estimating the class.

In regard to class-wise accuracy of detection, the perfect accuracy (MAPE 0.00%) is obtained by Unripe Bunch detection with configurations OB 1 & OD 2. On the other hand, OC 3 shows the greatest error magnitude (MAPE 67.74%) among these classes. In case of Overripe Bunch detection, OB 3 indicates the lowest error magnitude (7.84%), whereas the greatest error magnitude is shown by OC 2 with 102.99%. For Empty Bunch detection, perfect accuracy (MAPE 0.00%) is obtained by OC 1 & OD 2, while OB 2 & OC 2 demonstrate the greatest error magnitude (100.00%). For Abnormal Bunch detection, accuracy maximizes for OB 2 with an error magnitude of 6.67%, but enormous error magnitudes are demonstrated by OB 1 & OC 1 with MAPEs of 89.47%. In spite of these class-wise fluctuations, the overall performance assures that the proposed system is competent within industry-grade quality evaluation for FFB.

From the perspective of robotic automation, these findings hold substantial practical value. As detailed in Table 3, the system's real-time processing speeds—ranging from 48 frames per second (FPS) for the YOLOv5su and YOLOv8s models (at 1280px) to 79 FPS for YOLOv5nu (at 800–1280px)—comfortably exceed the 25 FPS threshold necessary for seamless decision-making in conveyor-based sorting systems. Given the typical conveyor speeds at palm oil mill's loading ramps, a FFB remains within the camera's field of view for several seconds. Consequently, even the most computationally demanding configuration operating at 48 FPS captures approximately 100 to 150 detection frames per bunch. This provides an abundance of redundant data, ensuring highly reliable inputs for robotic actuation.

Furthermore, the system aligns exceptionally well with industrial automation standards. The most effective configurations yielded a total FFB counting error of just 0.17% and a Ripe Bunch error of 0.07% (OC 3), both operating well within the widely accepted ±2% tolerance for automated sorting lines. These metrics confirm the system's viability as a robust, real-time perception framework for smart plantation management, robotic harvesting assistance, and automated conveyor sorters. Importantly, the model maintained its reliability across four distinct plantation divisions—each characterized by varying soil compositions, crop progenies, topography, and microclimates—highlighting its strong potential for large-scale, multi-estate deployment. While lighting inconsistencies were successfully mitigated using a controlled 350W LED setup, challenges regarding bunch occlusion and physical deformation remain; these specific limitations for robotic integration are explored in the subsequent section.

3.3. Identify the cause of the error

The inaccuracies in detecting and distinguishing oil palm FFB are contributed by a variety of reasons. The following are some important challenges involved in tracking objects within the framework for assessing quality in the oil palm FFB:

a. Misclassification of a single FFB into two distinct classes. This is where one FFB is mistakenly identified as consisting of two classes, for instance, when one FFB is mistaken for both ripe and overripe, or for both ripe and unripe classes (Figure 25). This problem is presumed to arise due to the improper setting of IoU and confidence thresholds.
b. Splitting of a single FFB into multiple detections of the same class as a result of a bunch fragmentation, which can be produced in case of physical impact during transport (Figure 26). Integrity of the object gets

broken, and thus there's overcounting. Additionally, morphological irregularities of some FFB (Figure 27) can affect perception. In all models and parameter sets, it was identified that only the YOLOv5su model, working on an 800 px resolution image, with a 0.2 confidence threshold, could detect a single FFB as a single entity. IoU parameter adjustments could help improve this problem through tightening the boundaries of the predicted bounding box.



Figure 25. One FFB (yellow marked) was detected as two FFBs

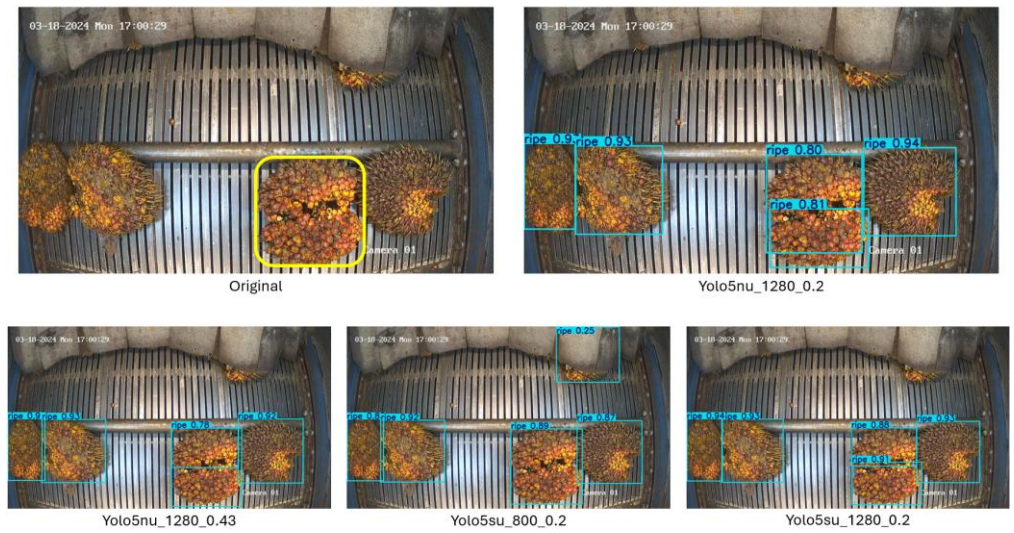


Figure 26. A splitted FFB (yellow marked) was detected as two FFBs

c. Clusters of fruitlets or individual fruits identified as a unit of FFB because the model fails to distinguish between dispersed and aggregated fruits (Figure 28, red). The problem might be mitigated with a higher confidence level, but this will likely cause errors to go undetected. A possible improvement could be achieved through optimized model adjustments and additional samples.
d. Other failures and misclassifications in the detection because of visual obstruction and external elements such as soil, mud, accumulations of fruitlets, overlap among several FFBs, and mechanical damage (Figure 28, yellow). These affect visual attributes such as color, texture, and structure. Rectification of the above-mentioned error sources can help contribute significantly to increased detection accuracy and the reliability of the automated quality assessment system for palm oil.

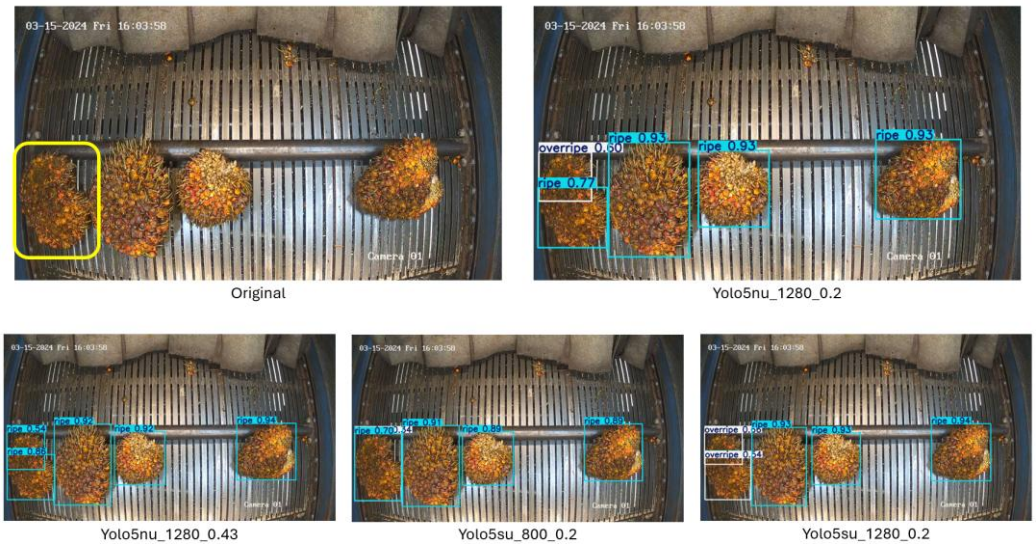


Figure 27. A yellow marked FFB with morphological anomaly was detected as two FFBs

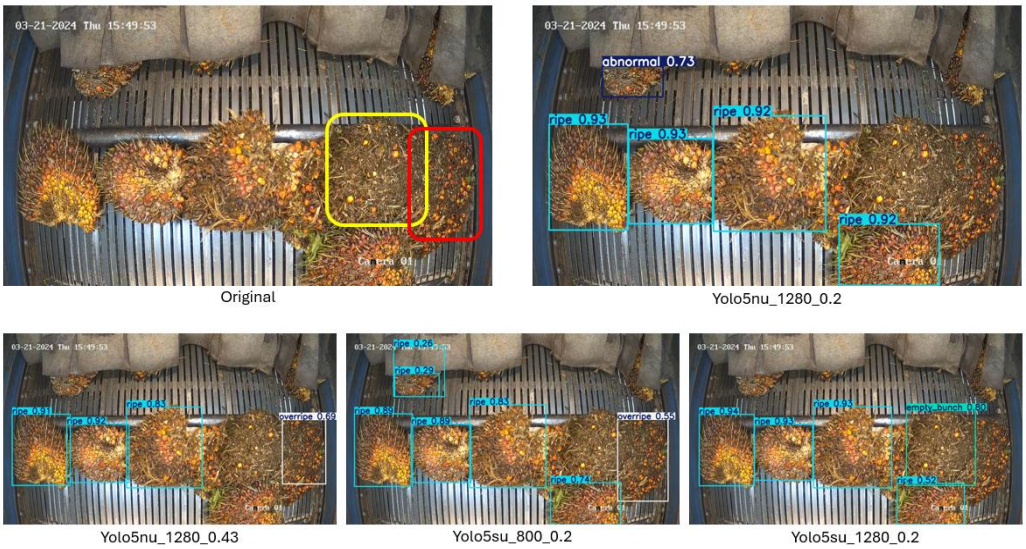


Figure 28. A yellow marked FFB was buried under the dirt and fruitlets causing it undetected, a pile of fruitlets, marked by red, was detected as an FFB

3.4. Comparative performance and statistical reliability

To make the comparison between architectures explicit and to quantify the reliability of the reported accuracy, the converged detection metrics of the four trained models are summarized in Figure 29, and the dispersion of counting error across configurations is summarized in Figure 30 and Table 5.

Figure 29 shows that the four variants are separated by narrow margins: precision spans only 0.952–0.959 and mAP@0.5 spans 0.956–0.968. YOLOv5nu attains the highest values on all four metrics, yet it is also the smallest model (5.24 MB) and the fastest in deployment (79 FPS, Table 3). The practical implication for automation is that accuracy is not the discriminating factor between these candidates; model size and inference speed are, which is why the lightweight variants were preferred for conveyor-speed operation.

Figure 30(a) shows that the classes on which mill payment depends are estimated with narrow uncertainty: Total FFB averages 0.74% error and Ripe 1.18%, with upper confidence limits of 1.02% and 1.81% respectively, both inside the ±2% tolerance normally accepted for automated sorting lines. Figure 30(b) shows the opposite situation for the rare classes, where the intervals are wide enough that the mean error cannot be estimated precisely at all.

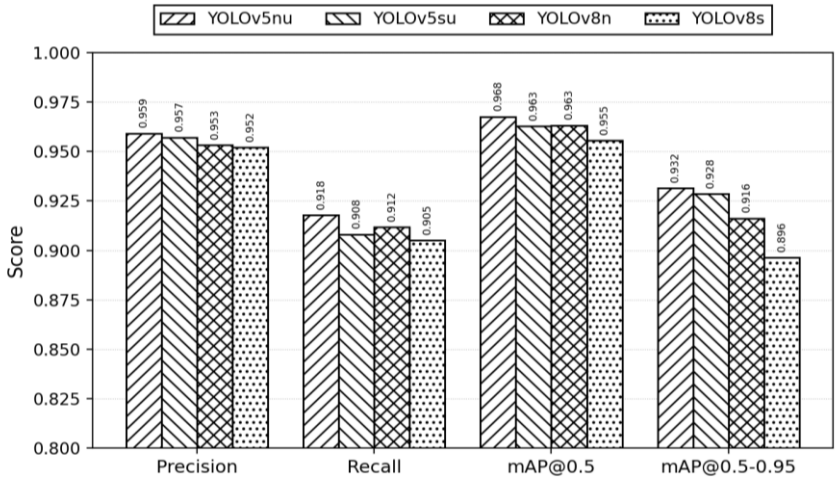


Figure 29. Converged detection performance of the four YOLO variants, showing that all models exceed 0.95 precision and 0.95 mAP@0.5 and are therefore suitable candidates for robotic conveyor inspection

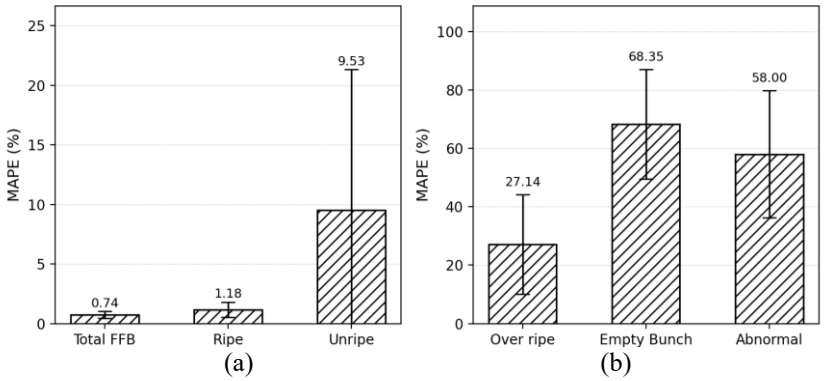


Figure 30. Mean absolute percentage error per class with 95% confidence intervals across the twelve top-performing configurations of Table 2: (a) industry-priority classes and (b) rare classes

Table 5 quantifies this contrast. The confidence interval for total FFB spans less than one percentage point, whereas the interval for empty bunch spans nearly forty. This dispersion is a direct consequence of class imbalance: Empty bunch and abnormal represent only 0.24% and 1.03% of the dataset, so a handful of missed detections moves the percentage error substantially. The intervals therefore reflect variability between plantation divisions and parameter settings rather than repeated-run sampling error, since each configuration was evaluated once over the recorded footage. They should be read as an indication of operational consistency across estates, and they support deploying the system for aggregate and ripe-unripe grading while retaining human verification for the rare categories.

Table 5. Mean MAPE and 95% confidence intervals by class across the twelve configurations of Table 2

Class	Mean MAPE (%)	95% CI (%)	SD (%)
Total FFB	0.74	0.45 – 1.02	0.45
Ripe	1.18	0.54 – 1.81	1.00
Unripe	9.53	0.00 – 21.34	18.59
Over ripe	27.14	10.14 – 44.14	26.76
Empty Bunch	68.35	49.62 – 87.09	29.49
Abnormal	58.00	36.22 – 79.78	34.28

3.5. Ethical considerations

Deploying an automated grading system in a commercial mill carries ethical obligations beyond detection accuracy. Because grading outcomes affect supplier payments, the system remains transparent and auditable: every counted bunch is logged with its class and a unique identifier (Figure 22), allowing decisions to be reviewed and contested rather than treated as an opaque black box. The system is intended to augment

rather than summarily replace human graders, relieving them of fatiguing repetitive counting while retaining human oversight for disputed or abnormal cases, and its consistent, bias-resistant scoring can reduce the deliberate grading fraud noted in the introduction. Responsible deployment should therefore be paired with worker reskilling toward supervisory and maintenance roles rather than abrupt displacement.

3.6. Limitations

Several limitations should be acknowledged. First, the dataset originates from a single estate of four divisions; although the divisions differ in soil, progeny, topography, and microclimate, external validity to other regions and cultivars remains to be demonstrated. Second, evaluation was performed on recorded conveyor footage rather than a live closed-loop deployment, so real-time latency, mechanical synchronization, and actuation effects were not measured. Third, manual grading was used as ground truth and may itself carry human bias, and the severe class imbalance limits the statistical reliability of the rare-class error estimates. These constraints directly motivate the future work outlined below.

4. CONCLUSION

The adopted computer vision technique proves its effectiveness in satisfying the oil palm industry's criteria concerning the quality assessment of total FFB, Ripe, and Unripe categories. Sustainability is also assured, with considerable eradication of potential human mistakes, enabling the program's viability. Furthermore, the system has been deployed at industrial scale: operational trials on real processing lines show that it can be integrated with existing grading workflows and sorting equipment, operate continuously under variable field conditions, and deliver accuracy and consistency comparable to experienced human graders while increasing throughput and reducing reliance on manual labour. Remaining mistakes are traced chiefly to challenging morphological matters (*e.g.*, distorted FFB) in addition to factors not part of the prediction range of the suggested technique. For increased efficiency, future studies are advised to incorporate more variables into the analysis stage, including FFB progenies, soil varieties, plantation topography, and micro-climatic variations, specifically involving more challenging categories such as overripe, empty bunch, and abnormal bunches.

In addition, to fill the gap that exists with current detection performance, there is a need to look extensively at alternate approaches that can be applied to detecting objects, including segmentation and oriented bounding box approaches, to improve performance and precision. Another challenge that still exists is that of data collection, specifically being able to obtain a bigger set of data that can be applied to classes that are rarely observed (overripe, abnormal, empty bunch). Other approaches, such as using artificial data that can be achieved through physically separating fruitlets to form artificial data, are considered unfeasible and unrealistic.

Future work should prioritize closed-loop validation on a live conveyor, coupling the detector with control logic for belt synchronization and with robotic sorters or diverters so that latency and throughput can be measured end-to-end. Multi-sensor fusion, for example combining RGB detection with hyperspectral or near-infrared sensing, could improve discrimination of the visually ambiguous overripe and empty bunch classes, while lightweight and quantized YOLO variants deployed on embedded edge hardware would enable in-line inference without a workstation-class GPU, moving the system closer to full integration within autonomous plantation and milling operations.

FUNDING INFORMATION

This work was supported by the Department of Computer Science and Electronics, Universitas Gadjah Mada under the Publication Funding Year 2026.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Pelangi Soko Aji	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Danang Lelono	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓		
Aufaclav Zatu Kusuma Frisky	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST

Authors state no conflict of interest.

DATA AVAILABILITY

The data presented in this study are available on request from the corresponding author.

REFERENCES

[1] E. Barcelos *et al.*, "Oil palm natural diversity and the potential for yield improvement," *Frontiers in Plant Science*, vol. 6, no. MAR, pp. 1–16, 2015, doi: 10.3389/fpls.2015.00190.

[2] A. Rival and P. Levang, *Palms of controversies oil palm and development challenges*. Center for International Forestry Research, 2014. doi: 10.17528/cifor/004860.

[3] R. H. V Corley, "How much palm oil do we need?," *Environmental Science and Policy*, vol. 12, no. 2, pp. 134–139, 2009, doi: 10.1016/j.envsci.2008.10.011.

[4] G. R. Iskarlia and M. R. Wardhana, "Determination of Oil Palm Grading (*Elaeis guineensis Jacq.*) at PT Hasnur Citra Terpadu," (in Indonesia: Penentuan Grading pada Kelapa Sawit (*Elaeis guineensis Jacq.*) di PT. Hasnur Citra Terpadu) *AGRISAINS*, vol. 2, pp. 85–90, 2015.

[5] H. Winkler, *Parthenogenesis und Apogamie im Pflanzenreiche*, vol. 26, no. 4. Jena: G. Fischer, 1908. doi: 10.1007/bf02161684.

[6] T. S. Gunawan, M. Kartiwi, H. Mansor, and N. M. Yusoff, "Palm fruit ripeness detection and classification using various YOLOv8 models," in *ICSIMA 2023 - 9th IEEE International Conference on Smart Instrumentation, Measurement and Applications*, 2023, pp. 193–198. doi: 10.1109/ICSIMA59853.2023.10373435.

[7] 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, doi: 10.1109/ACCESS.2022.3204762.

[8] M. R. A. Priyadi and Suharjito, "Comparison of YOLOv8 and efficientDet4 algorithms in detecting the ripeness of oil palm fruit bunch," in *10th International Conference on ICT for Smart Society, ICISS 2023 - Proceeding*, 2023, pp. 1–6. doi: 10.1109/ICISS59129.2023.10291928.

[9] M. Y. M. A. Mansour, K. D. Dambul, and K. Y. Choo, "Object detection algorithms for ripeness classification of oil palm fresh fruit bunch," *International Journal of Technology*, vol. 13, no. 6, pp. 1326–1335, 2022, doi: 10.14716/ijtech.v13i6.5932.

[10] N. Khamis, H. Selamat, S. Ghazalli, N. I. Md Saleh, and N. Yusoff, "Comparison of palm oil fresh fruit bunches (FFB) ripeness classification technique using deep learning method," in *ASCC 2022 - 2022 13th Asian Control Conference, Proceedings*, 2022, pp. 64–68. doi: 10.23919/ASCC56756.2022.9828345.

[11] Z. Chen, S. Elsaid, D. Y. Y. Sim, T. Maul, and I. Y. Liao, "Detection of oil palm fresh fruit bunches (FFBS) with computer vision models," in *Proceedings - 2022 International Conference on Mechanical, Automation and Electrical Engineering, CMAEE 2022*, 2022, pp. 86–91. doi: 10.1109/CMAEE58250.2022.00023.

[12] N. A. Mohd Basir Selvam, Z. Ahmad, and I. A. Mohtar, "Real time ripe palm oil bunch detection using YOLO V3 algorithm," in *19th IEEE Student Conference on Research and Development: Sustainable Engineering and Technology towards Industry Revolution, SCOReD 2021*, 2021, pp. 323–328. doi: 10.1109/SCOReD53546.2021.9652752.

[13] T. S. Hong, F. Hanim Hashim, T. Raj, and A. B. Huddin, "Classification of oil palm fruit ripeness using artificial neural network," in *2021 IEEE International Conference on Automatic Control and Intelligent Systems, I2CACIS 2021 - Proceedings*, 2021, pp. 358–363. doi: 10.1109/I2CACIS52118.2021.9495857.

[14] A. Septiarini, H. R. Hatta, H. Hamdani, A. Oktavia, A. A. Kasim, and S. Suyanto, "Maturity grading of oil palm fresh fruit bunches based on a machine learning approach," in *2020 5th International Conference on Informatics and Computing, ICIC 2020*, Nov. 2020, pp. 1–4. doi: 10.1109/ICIC50835.2020.9288603.

[15] Suharjito, E. P. Wonohardjo, D. Pratama, T. R. Sahroni, R. A. August, and Marimin, "Effect of pre-processing dataset on classification performance of deep learning model for detection of oil palm fruit ripe," in *9th International Conference on ICT for Smart Society: Recover Together, Recover Stronger and Smarter Smartization, Governance and Collaboration, ICISS 2022 - Proceeding*, 2022, pp. 1–5. doi: 10.1109/ICISS55894.2022.9915269.

[16] S. Sae-Tang, "Estimation of oil content in oil palm fresh fruit bunch by its surface color," in *Proceedings - 2020 15th International Joint Symposium on Artificial Intelligence and Natural Language Processing, iSAI-NLP 2020*, 2020, pp. 1–6. doi: 10.1109/iSAI-NLP51646.2020.9376834.

[17] 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 (Switzerland)*, vol. 15, no. 2, 2023, doi: 10.3390/su15020901.

[18] Suharjito *et al.*, "Annotated datasets of oil palm fruit bunch piles for ripeness grading using deep learning," *Scientific Data*, vol. 10, no. 1, 2023, doi: 10.1038/s41597-023-01958-x.

[19] M. A. Zulkhoiri *et al.*, "Investigation of oil palm fruit bunch ripeness classification using machine learning classifiers," in *E3S Web of Conferences*, 2024, vol. 595, pp. 1–6. doi: 10.1051/e3sconf/202459502010.

[20] M. S. M. Alfatni, S. Khairunniza-Bejo, M. H. B. Marhaban, O. M. B. Saaed, A. Mustapha, and A. R. M. Shariff, "Towards a real-time oil palm fruit maturity system using supervised classifiers based on feature analysis," *Agriculture (Switzerland)*, vol. 12, no. 9, 2022, doi: 10.3390/agriculture12091461.

[21] L. P. Nur'aini and M. Rahardi, "Detection of ripeness in oil palm fresh fruit bunches using the YOLO12S algorithm on digital images," *Journal of Applied Informatics and Computing*, vol. 9, no. 4, pp. 1633–1638, 2025, doi: 10.30871/jaic.v9i4.10250.

[22] N. Y. D. Rahmawati and F. P. Rochim, "Chili ripeness level detection with YOLOv8 and fuzzy logic for harvest decision making," *Jurnal Teknik Pertanian Lampung (Journal of Agricultural Engineering)*, vol. 14, no. 5, pp. 1807–1818, 2025, doi:

10.23960/jtepl.v14i5.1807-1818.
[23] R. Kurniawan, Samsuryadi, F. S. Mohamad, H. O. L. Wijaya, and B. Santoso, "Classification of palm oil fruit ripeness based on AlexNet deep convolutional neural network," *Sinergi (Indonesia)*, vol. 29, no. 1, pp. 207–220, 2025, doi: 10.22441/sinergi.2025.1.019.
[24] R. E. Saragih, A. R. Purnajaya, I. Syafrinal, Y. Pernando, and Yodi, "Mango and banana ripeness detection based on lightweight YOLOv8," *Jurnal Buana Informatika*, vol. 15, no. 2, pp. 79–88, 2024, doi: 10.24002/jbi.v15i2.8895.
[25] 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, doi: 10.1109/ACCESS.2024.3446890.
[26] Y. Shen, Z. Yang, Z. Khan, H. Liu, W. Chen, and S. Duan, "Optimization of improved YOLOv8 for precision tomato leaf disease detection in sustainable agriculture," *Sensors*, vol. 25, no. 5, 2025, doi: 10.3390/s25051398.
[27] B. Oktaviana and S. Suriati, "Implementation of the support vector machine (SVM) method for classifying the maturity level of oil palm fruit," *Journal of Applied Informatics and Computing*, vol. 9, no. 5, pp. 2288–2295, 2025, doi: 10.30871/jaic.v9i5.10783.
[28] J. Terven, D. M. Córdova-Esparza, and J. A. Romero-González, "A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, Nov. 2023, doi: 10.3390/make5040083.
[29] Al-Khowarizmi, S. Efendi, M. K. M. Nasution, and M. Herman, "The role of detection rate in MAPE to improve measurement accuracy for predicting FinTech data in various regressions," in *ICCoSITE 2023 - International Conference on Computer Science, Information Technology and Engineering: Digital Transformation Strategy in Facing the VUCA and TUNA Era*, 2023, pp. 874–879. doi: 10.1109/ICCoSITE57641.2023.10127820.

BIOGRAPHIES OF AUTHORS



Pelangi Soko Aji is Digital Architect Assistant Manager at PT Sawit Sumbermas Sarana Tbk. He earned his Bachelor's degree in Electronics and Instrumentation from Universitas Gadjah Mada (UGM), Yogyakarta, Indonesia, in 2017, and is currently pursuing a Master's degree in the same field at UGM. His research interests include IoT, robotics, and machine learning applications in agriculture, particularly in palm oil plantation management. He has worked in international companies before joining PT Sawit Sumbermas Sarana Tbk. His recent work focuses on integrating AI-driven solutions for agricultural efficiency. He can be contacted at pelangisokoajissi@mail.ugm.ac.id.



Danang Lelono is received the Ph.D. degree in physics from Universitas Gadjah Mada Yogyakarta, Indonesia, in 2017. His Bachelor's degree in Electronics and Instrumentation and Master degree in Signals and Control from Universitas Gadjah Mada (UGM). He is currently part of the Faculty of Mathematics and Natural Sciences at UGM and a member of various research initiatives. He was appointed as an assistant professor of Electronics and Instrumentation, in 2020. He has published more than 50 peer-reviewed articles. His current research interests are mainly sensor and sensor systems, embedded systems, controls and smart instruments. He can be contacted at danang@ugm.ac.id.



Aufaclav Zatu Kusuma Frisky received the bachelor degree in Department of Computer Science and Electronics from Universitas Gadjah Mada, Indonesia at 2012 and M.Sc. degree in Department of Computer Science and Information Engineering from National Central University, Taiwan at 2015 and Doctoral at TUWien, Austria, He is currently Assistant Professor at Embedded System and Robotic Lab, Department of Computer Science and Electronics, UGM. His research interest includes computer vision, artificial intelligence, single image reconstruction and its applications in cultural heritage preservation. He can be contacted at aufaclav@ugm.ac.id.