Evaluating YOLO Variants for Real-Time Multi-Object Detection of Strawberry Quality and Ripeness

1st Rifki Rosada
School of Electrical Engineering
Telkom University
Bandung, Indonesia
ORCID : 0009-0002-9661-4762

2nd Zidane Muhammad Hussein
School of Electrical Engineering
Telkom University
Bandung, Indonesia
ORCID : 0009-0006-7526-1856

3rd Ledya Novamizanti
School of Electrical Engineering
CoE of AILO
Telkom University
Bandung, Indonesia
ledyaldn@telkomuniversity.ac.id

***Abstract*—Automated fruit grading plays a pivotal role in modern agriculture by enabling timely harvesting and maintaining quality standards, especially for high-value crops such as strawberries. This paper presents an end-to-end approach for the real-time detection and classification of strawberry ripeness using state-of-the-art YOLO-based models: YOLOv7, YOLOv8, and YOLOv11. A comprehensive dataset of 3,055 strawberry images is compiled from three distinct sources. Each image is meticulously annotated into five classes—Unripe (UNR), Fully Ripe Grade A (AFR), Fully Ripe Grade B (BFR), Half Ripe Grade A (AHR), and Half Ripe Grade B (BHR)—with additional complexity introduced by images containing multiple strawberries per frame. Data preprocessing and augmentation are performed using Roboflow, and model training is executed on Google Colab with a uniform protocol to ensure a fair comparison among the YOLO variants. Experimental results reveal a steady performance improvement from YOLOv7 to YOLOv11, with YOLOv11 achieving the highest detection accuracy (precision: 0.874, recall: 0.855, mAP@50: 0.942, and mAP@50–95: 0.820). The superior performance of YOLOv11 is attributed to its incorporation of dynamic attention modules and self-adaptive layer-wise fusion, which significantly enhance the detection of subtle ripeness variations and mitigate occlusion challenges. These findings underscore the potential of advanced YOLO architectures for deployment in real-time agricultural applications and automated harvesting systems.**

Keywords—Strawberry, real-time detection, computer vision, YOLO, object detection, deep learning

I. INTRODUCTION

Recent years advancements in agriculture have seen the use of deep learning for image annotation, which efficiently extracts features from rapidly growing image data and enables the successful analysis of large datasets [1]. The ability to accurately assess fruit ripeness is critical for optimizing harvesting processes, ensuring high-quality yields, and reducing post-harvest losses [2]. Strawberries, in particular, are a high-value crop whose market value is highly dependent on the precise determination of ripeness. However, the variability in fruit size, color gradation, and occlusions from leaves and branches pose significant challenges to reliable detection and classification as noted in Guo et al.'s study, cited by Chai et al. [3].

Several studies have explored the application of YOLO-based models for fruit ripeness detection, each offering unique enhancements to address specific challenges. While the individual YOLO models are publicly available, this paper uniquely contributes by offering a structured, head-to-head comparison of YOLOv7, v8, and v11 using a rigorously balanced strawberry dataset under real-time constraints. The novelty lies in the practical deployment insights, detailed architectural benchmarking, and performance under occlusion and lighting variations—elements not previously analyzed in one unified study. For instance, a YOLOv7-based model demonstrated promising results in accurately classifying grape maturity [4]. In another study, Azizah et al. [5] stated that combining YOLOv7 with EfficientNetV2S significantly improved the classification accuracy of strawberry ripeness, achieving up to 99.0% in F1-score, precision, and recall. Although the hybrid model required a longer training time, it outperformed the standalone YOLOv7 in overall performance. These findings highlight the potential of model hybridization for improved detection, laying the foundation for real-time applications in agriculture, which this study further explores. Additionally, in complex agricultural environments, accurate object detection remains a challenge due to occlusions, lighting variability, and overlapping fruits. Zhang et al. [6] addressed pear detection using an improved YOLO model, demonstrating its capability to handle orchard-level complexity effectively. Similarly, this study extends the comparison of YOLO variants specifically to strawberry ripeness detection under real-time constraints. Furthermore, recent advancements in YOLO variants have led to high-precision models capable of handling agricultural tasks. Xiao et al. [7] leveraged YOLOv8 to classify general fruit ripeness, achieving 99.5% accuracy while maintaining ultra-fast detection speeds, demonstrating the model's potential for lightweight, real-time detection. A notable GitHub project also demonstrated the feasibility of using YOLOv7 for strawberry counting and ripeness detection, reinforcing the practical utility of these models in agricultural settings [8].

This study aims to evaluate and compare YOLOv7, YOLOv8, and YOLOv11 using a robust and diverse strawberry dataset compiled from previous study [5], open-source repositories, and on-site collections at Ichigo Farm in Ciwidey, West Java, Indonesia. We intentionally selected the three major, publicly-released YOLO versions—v7, v8, and v11—for comparison. Versions labeled v9 and v10 were internal or incremental updates whose core innovations (e.g., improved anchors and attention refinements) were rolled directly into the publicly released YOLOv11 [14]. Hence, including v9/v10 separately would neither add new insights nor be practically reproducible. By systematically analyzing performance metrics—including precision, recall, mAP@50, and mAP@50–95—under controlled experimental conditions, we provide valuable insights into the optimal YOLO architecture for real-time strawberry ripeness

detection. This work contributes to the advancement of smart agricultural solutions by demonstrating the effectiveness of advanced architectural features, such as dynamic attention modules and self-adaptive layer-wise feature fusion, in addressing the inherent challenges of fruit detection in complex environments.

The subsequent sections of this paper are arranged as follows. Section II provides a comprehensive review of existing literature pertinent to this research. Section III describes the proposed method, including a detailed overview of the dataset, YOLO architecture variants, and the training process implemented on cloud-based platforms. Section IV discusses experiments and results, comparing the performance of YOLOv7, YOLOv8, and YOLOv11 using key evaluation metrics and analyzing confusion matrices to highlight their respective strengths and limitations. The final section, Section V, offers concluding remarks and proposes prospective directions for future scholarly inquiry.

II. Related Work

The integration of deep learning in agricultural image analysis has significantly advanced precision farming, particularly in the classification and detection of fruit ripeness. Object detection algorithms, especially those in the YOLO (You Only Look Once) family, have proven effective due to their ability to process images in real time while maintaining high accuracy. These capabilities are especially crucial in agricultural contexts where timely decisions impact yield quality and operational efficiency. As such, numerous studies have explored various YOLO architectures and hybrid models to address challenges in detecting fruit under natural conditions, such as occlusions, lighting variability, and overlapping foliage.

Azizah et al. [5] proposed an integrated approach using deep learning techniques that combined YOLOv7 with EfficientNetV2S to classify the ripeness of strawberries. Their approach achieved high performance, with an F1-score, precision, and recall reaching up to 99.0%. The integration of EfficientNetV2S as the backbone improved the model's ability to differentiate subtle variations in strawberry maturity, particularly under complex visual conditions. Although the model required longer training time, it consistently outperformed the standalone YOLOv7 model, indicating that hybrid architectures can significantly enhance accuracy and robustness in agricultural applications.

Xiao et al. [7] proposed a lightweight YOLOv8-based architecture to detect general fruit ripeness with high precision. The model achieved 99.5% accuracy while maintaining low computational latency, making it suitable for real-time applications in edge devices. Key improvements included the use of decoupled detection heads and an anchor-free mechanism, which streamlined the model structure and reduced detection time. These optimizations position YOLOv8 as a strong candidate for mobile and UAV-based smart farming systems that require both speed and accuracy in diverse environments.

Y. Chen et al. [9] introduced CES-YOLOv8, an enhanced YOLOv8 variant, for strawberry ripeness detection under challenging field conditions. The model incorporated ConvNeXt V2 modules and the Efficient Channel Attention (ECA) mechanism to improve feature representation and channel weighting. Experimental results showed a precision of 88.2% and a mAP@50 of 92.10%, confirming the model's effectiveness in identifying strawberries at different maturity stages. The added attention mechanisms allowed the network to focus on relevant features even when fruits were partially occluded or appeared under inconsistent lighting.

X. Chen [10] presented an advanced YOLOv8-Pose model for Fast identification of ripe strawberries and pinpointing ideal harvest locations in three dimensions. The model was enhanced using a Bidirectional Feature Pyramid Network (BiFPN) for multi-scale feature fusion and MobileViTv3 as the backbone to reduce computational complexity.

Gamani et al. [11] evaluated Various YOLOv8 model setups were explored to perform instance segmentation of strawberry developmental phases in outdoor agricultural settings. Their study identified YOLOv8n as the most efficient model, achieving a mean Average Precision (mAP) of 80.9% and processing time of only 12.9 milliseconds per frame. The segmentation approach allowed for detailed analysis of different strawberry development stages, aiding in phenological monitoring. This research demonstrates the practical benefits of lightweight, high-speed models for continuous monitoring in outdoor agricultural settings.

III. Proposed Method

A. Dataset

The strawberry image dataset used in this study is compiled from three distinct sources to ensure a comprehensive and balanced representation. First, a subset of images was obtained from previous research titled “Identifying the Ripeness and Quality Level of Strawberries Based on YOLOv7-EfficientNet” [5], which provided a well-documented benchmark with detailed annotations and a rigorous acquisition protocol. Second, additional images were collected from open sources to enhance the dataset’s diversity and volume. Third, a dedicated on-site collection was carried out at our partner strawberry farm, Ichigo Farm, located in Ciwidey, West Java, Indonesia, on 6 December 2024 between 7:00 AM and 9:00 AM under clouded weather conditions, using multiple smartphone devices (Realme 9 Pro Plus, Samsung S21 5G, and Redmi Note 9 Pro).

The dataset is meticulously annotated into five classes: Unripe (UNR), Fully Ripe Grade A (AFR), Fully Ripe Grade B (BFR), Half Ripe Grade A (AHR), and Half Ripe Grade B (BHR), Each class comprises exactly 611 images, ensuring balanced representation across all categories. Fig. 1 presents example images for each class, demonstrating the visual distinctions among the different ripeness and quality levels.

It is important to note that while some images contain a single strawberry, others feature multiple strawberries within the same frame. This increases the complexity of object detection, as models must correctly identify and classify each individual instance. Such variability can influence recall and precision, particularly in cases where occlusions or overlapping objects occur. The data allocation is organized with the dataset partitioned into a total of 489 images were allocated for model training, while 61 images each were reserved for validation and testing purposes. This structured approach to data collection and annotation enhances

reproducibility and provides a solid foundation to assess how effectively the suggested detection systems function.

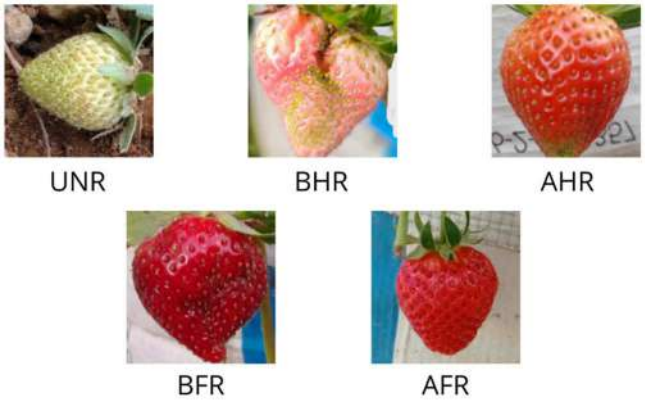


FIGURE 1
Example strawberry images for each class, including Unripe (UNR), Grade B – Half Ripe (BHR), Grade A – Half Ripe (AHR), Grade B – Fully Ripe (BFR), and Grade A – Fully Ripe (AFR) (from left to right).

For quick reference on the distinctive visual attributes associated with the quality of each strawberry class, a simplified in Table I. The quality attributes were derived from expert visual assessments and complemented by quantitative analysis where applicable.

TABLE I.
QUALITY CHARACTERISTICS

Class	Color	Texture	Size/Shape	Blemishes
UNR	Pale green	Firm	Small, irregular	None/Minimal
BHR	Red/green mix	Uneven	Varying	Prominent
AHR	Red/green mix	Slightly uneven	Varying	Noticeable
BFR	Bright red	Smooth	Regular	Few
AFR	Bright red	Smooth	Regular	Minimal

Table I provides a detailed breakdown of the strawberry dataset allocation, categorizing the images based on ripeness and quality. The table classifies strawberries into five categories—UNR, AFR, BFR, AHR, and BHR—with 489 images allocated for training, 61 for validation, and 61 for testing in each category, resulting in a total of 611 images per class. Overall, the complete dataset comprises 3055 images, with 2445 used for training, 305 for validation, and 305 for testing purposes. Additionally, the dataset underwent augmentation using three methods: horizontal flipping, 90-degree clockwise and counterclockwise rotations, and random rotations within the range of –15° to +15°. This augmentation further increases data variability and helps improve model robustness.

B. Architecture

To perform real-time multi-object detection of strawberry ripeness, we compare three state-of-the-art YOLO variants: YOLOv7, YOLOv8, and YOLOv11. Each of these models represents successive advancements in object detection technology, with improvements in feature extraction, computational efficiency, and robustness under challenging conditions. YOLOv7 (2022) is recognized for its robust feature extraction and fast inference; it employs a deep convolutional backbone, an FPN-like neck for multi-scale feature fusion, and a detection head that outputs bounding box predictions along with class probabilities [12]. This architecture is designed to balance accuracy and speed, making it highly suitable for real-time applications.

Building on YOLOv7, YOLOv8 (2023) streamlines the convolutional layers and incorporates dynamic receptive fields, which help capture small object details while reducing computational overhead. In addition, YOLOv8 utilizes improved loss functions that enhance prediction accuracy and further optimize the model's performance in terms of precision and recall [13]. These refinements ensure that the model is more efficient without compromising its detection capabilities, making it an excellent candidate for deployment in resource-constrained environments. This version represents a significant step forward in both architectural design and practical application.

The most recent iteration, YOLOv11 (2024), incorporates dynamic attention modules and self-adaptive layer-wise feature fusion to boost detection performance under challenging conditions, such as variable lighting and occlusion [14]. This enhancement enables YOLOv11 to dynamically prioritize salient features at multiple scales, in contrast to YOLOv7's static CSP backbone and YOLOv8's refined CSPDarknet, thereby improving detection robustness under complex conditions. YOLOv11 maintains the standard three-part structure—comprising the backbone, neck, and detection head—while integrating these advanced modules to refine feature representation and improve accuracy. The dynamic attention mechanisms enable the model to focus on salient features across multiple scales, which is critical for distinguishing subtle differences in strawberry ripeness. Each architecture is systematically evaluated for speed, accuracy, and computational efficiency, ensuring that the best-performing model for strawberry ripeness detection is identified.

Table III provides a comparative summary of the key architectural components, including input size, backbone type, neck design, detection head configuration, and the approximate total number of layers for YOLOv7, YOLOv8, and YOLOv11. This detailed comparison not only highlights the evolution of design principles across these models but also serves as a reference for understanding how each architectural enhancement contributes to overall performance. The comprehensive evaluation of these architectures forms the basis for selecting the optimal model for our specific application.

TABLE 2
SUMMARY OF YOLO ARCHITECTURES

Component	YOLOv7	YOLOv8	YOLOv11
Input Size	640×640	640×640	640×640
Backbone	modules	improved small-object detection	modules
Neck	scale fusion	efficient multi-scale fusion	with self-attention
Detection Head	predefined anchors		and attention-
Layers	configuration)	(approximate)	variant)

It is important to note that each YOLO variant introduces specific modifications to enhance overall detection performance. YOLOv7 leverages an Extended ELAN module combined with CSP and SPP components in its backbone to ensure robust feature extraction, while the PAN neck facilitates effective multi-scale fusion [12]. In contrast, YOLOv8 refines the backbone by adopting a streamlined CSPDarknet design and replaces the traditional PAN with an enhanced PAN++ to capture small objects more efficiently and reduce computational load [13]. YOLOv11 takes these improvements further by integrating dynamic attention modules within a modified CSP backbone and an enhanced PAN that incorporates self-adaptive layer-wise fusion and attention mechanisms, resulting in an optimized detection head with dynamic anchor refinement [14]. These incremental advancements are aimed at improving speed, accuracy, and robustness, making the architectures highly suitable for the real-time detection of strawberry ripeness.

C. *Training Process*

The training process is executed on cloud-based platforms to ensure reproducibility and scalability. The entire dataset is pre-processed and augmented using Roboflow [15], which standardizes image resizing, normalization, and applies various augmentation techniques—such as rotations, scaling, and brightness adjustments—to increase data diversity. The augmented dataset is then uploaded to Google Colab, which provides GPU acceleration for efficient model fine-tuning. A uniform training protocol is adopted for all YOLO variants to facilitate a fair comparison.

For YOLOv7 (2022), training was conducted with 8 workers, a batch size of 8, a 640×640 input image size, and a total of 10 epochs. The training configuration included a custom YAML file defining the network architecture and hyperparameters, along with pre-trained weights to initialize the model. Similarly, YOLOv8 (2023) and YOLOv11 (2024) were trained on the same dataset with identical input sizes and epoch counts, using their respective lightweight pre-trained models (yolov8n.pt and yolov11n.pt). Both models leverage refined loss functions and architectural improvements—such as dynamic receptive fields in YOLOv8 and integrated attention modules in YOLOv11—to enhance prediction accuracy and computational efficiency. Validation checkpoints are regularly monitored throughout training to prevent overfitting, and final model performance is evaluated on a held-out test set. Detailed training scripts and commands are provided in the supplementary material and our associated GitHub repository.

Fig. 2 provides an overview of the proposed real-time multi-object strawberry ripeness detection system based on the YOLO model. The process includes image annotation and augmentation using Roboflow, data export, and fine-tuning on Google Colab with pre-trained YOLOv7, YOLOv8, and YOLOv11 models.

D. *Evaluation Matrix*

To assess how well the model performs, commonly used evaluation indicators include precision, recall, and the mean Average Precision (mAP) metric. The formulas for precision and recall are provided in Eqs. (1) and (2), respectively [16-18].

$$\text{Precision} = \frac{TP}{TP+FP} \quad (1)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (2)$$

Here, TP denotes true positives, FP stands for false positives, and FN refers to false negatives.

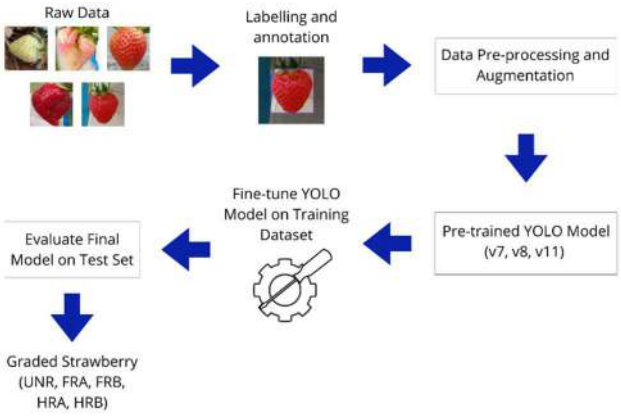


Fig. 2.
Overview of the proposed real-time multi-object detection of strawberry ripeness based on the YOLO model.

The mean Average Precision (mAP) is calculated by averaging the Average Precision (AP) across all classes, where AP corresponds to the area under the precision-recall curve. In this study, we report mAP at an Intersection over Union (IoU) threshold of 0.50 (mAP@50) and also averaged over IoU thresholds from 0.50 to 0.95 in increments of 0.05 (mAP@50–95), as standardized in modern evaluation protocols. These metrics offer an all-encompassing assessment of a model's performance in terms of both object localization and classification accuracy. Such metrics are well-established and have been widely adopted in recent literature for benchmarking object detection models [19].

IV. Experiments and Results

This section presents a comparative evaluation of the YOLOv7, YOLOv8, and YOLOv11 models based on key performance metrics: precision, recall, mAP@50, and mAP@50–95. First, the final performance of each model is summarized, followed by an analysis of the training and validation curves. Finally, the confusion matrices are discussed to highlight areas of misclassification and opportunities for further improvement.

A. *Experiment Setup*

The experiments were carried out on the Google Colab platform, utilizing hardware that included a 16 GB memory setup with an AMD Ryzen 5 4600H CPU, NVIDIA GeForce GTX 1650 Ti GPU. Each YOLO model (YOLOv7, YOLOv8, YOLOv11) was trained for 10 epochs using PyTorch and Ultralytics libraries. The dataset, sourced from Roboflow, includes annotated strawberry images across five classes, including UNR, AFR, BFR, AHR, BHR).

B. *Comparative Evaluation Based on Final Metrics*

Tables III to V show the performance of the YOLOv7, YOLOv8, and YOLOv11 baseline models. The YOLOv7 baseline model, summarized in Table III, achieved an average precision of 0.833, a recall of 0.794, a mAP@50 of 0.898, and a mAP@50-95 of 0.769. It demonstrated strong precision in classifying ripe strawberries, particularly in the AHR class (0.981).

However, it showed a lower recall for AHR (0.668), indicating a higher rate of false negatives. The highest recall was observed in the BFR class (0.904), reflecting effective detection of fully ripe strawberries. Overall, YOLOv7 delivered solid precision but struggled with recall, especially for certain classes, leading to missed detections.

TABLE 3
PERFORMANCE OF YOLOV7 BASELINE MODEL TRAINING

Class	Precision	Recall	mAP50	mAP50-95
UNR	0.807	0.794	0.846	0.641
AFR	0.674	0.848	0.904	0.791
BFR	0.796	0.904	0.915	0.807
AHR	0.981	0.668	0.905	0.809
BHR	0.908	0.758	0.917	0.799
Average	0.833	0.794	0.898	0.769

The YOLOv8 model, presented in Table IV, improved across all metrics, with an average precision of 0.883, recall of 0.834, mAP@50 of 0.938, and mAP@50-95 of 0.813. Precision for unripe strawberries (UNR) rose to 0.92, up from YOLOv7's 0.807, indicating better class differentiation. BHR achieved the highest recall at 0.924, demonstrating better detection of ripe berries. YOLOv8 offered a more balanced trade-off between precision and recall, reducing misclassification and false negatives.

TABLE 4
PERFORMANCE OF YOLOV8 BASELINE MODEL TRAINING

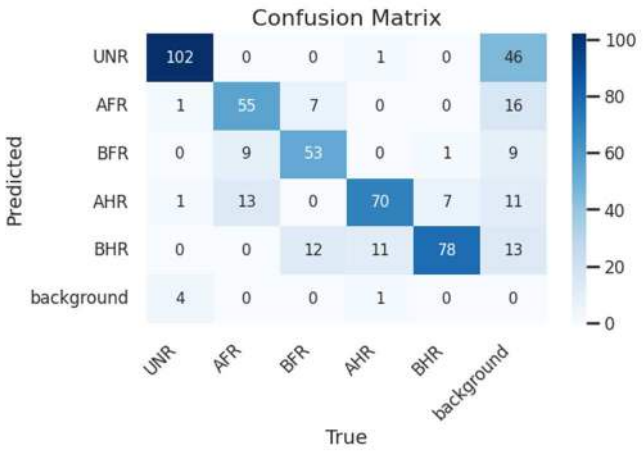
Class	Precision	Recall	mAP50	mAP50-95
UNR	0.92	0.795	0.902	0.679
AFR	0.851	0.833	0.942	0.837
BFR	0.878	0.832	0.947	0.851
AHR	0.917	0.785	0.937	0.838
BHR	0.848	0.924	0.964	0.858
Average	0.883	0.834	0.938	0.813

TABLE 5
PERFORMANCE OF YOLOV11 BASELINE MODEL TRAINING

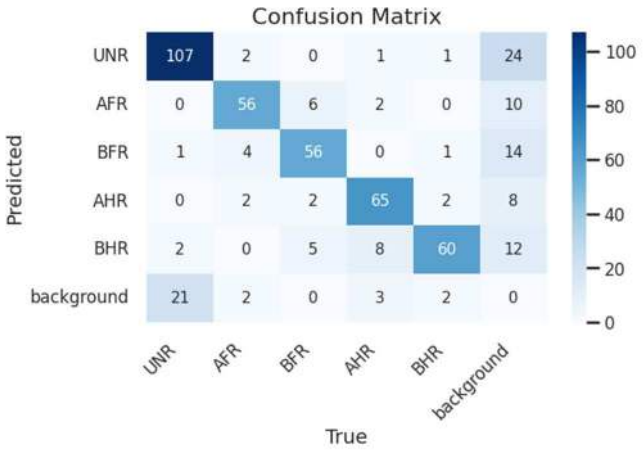
Class	Precision	Recall	mAP50	mAP50-95
UNR	0.926	0.77	0.899	0.675
AFR	0.876	0.924	0.954	0.834
BFR	0.877	0.826	0.961	0.861
AHR	0.909	0.823	0.939	0.853
BHR	0.784	0.934	0.958	0.874
Average	0.874	0.855	0.942	0.82

TABLE 6
COMPARISON OF OVERALL PERFORMANCE

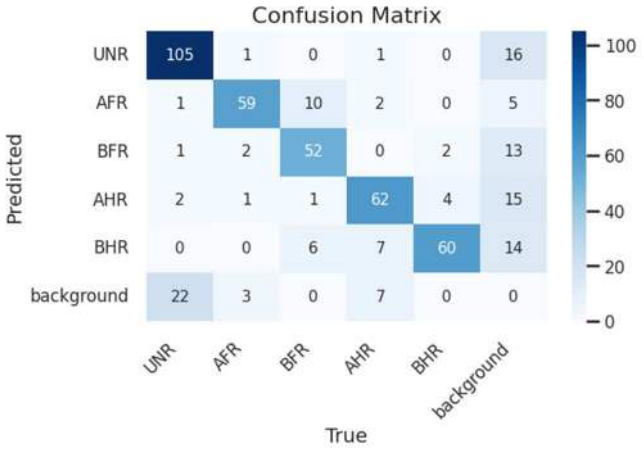
Model	Precision	Recall	mAP50	mAP50-95
YOLOv7	0.833	0.794	0.898	0.769
YOLOv8	**0.883**	0.834	0.938	0.813
YOLOv11	0.874	**0.855**	**0.942**	**0.82**



(a)



(b)



(c)

FIGURE 3
Confusion matrices (a) YOLOv7 (b) YOLOv8, and (c) YOLOv11. Illustrating classification performance across six classes: UNR, AFR, BFR, AHR, BHR, and background.

The YOLOv11 model, shown in Table V, further enhanced performance, achieving an average precision of 0.874, recall of 0.855, mAP@50 of 0.942, and mAP@50-95 of 0.820. Although its precision was slightly lower than YOLOv8's, it had the highest recall, showing improved sensitivity to object presence. UNR had the highest precision (0.926) and BHR had the highest recall (0.934), indicating strong classification and detection across stages. YOLOv11

also achieved the best mAP@50-95, confirming its robustness across confidence thresholds.

As summarized in Table VI, YOLOv7 performed moderately well but lagged in recall and mAP@50-95. YOLOv8 delivered the highest average precision and showed a balanced improvement in all metrics. YOLOv11 slightly trailed in precision but led in recall and mAP@50-95, making it the most effective at minimizing missed detections. Overall, YOLOv11 offered the best generalization and detection performance, especially in real-world scenarios requiring fewer false negatives.

C. Confusion Matrix

Confusion matrices were generated by counting true positives (TP), false positives (FP), and false negatives (FN) for each class. For example, if TP = 90, FP = 10, FN = 15 for the UNR class, precision = TP/(TP+FP) = 0.900 and recall = TP/(TP+FN) ≈ 0.857. Mean Average Precision at IoU=0.50 (mAP@50) and averaged across IoU=0.50–0.95 (mAP@50–95) were computed by integrating the precision–recall curve at the respective thresholds.

To assess classification performance in greater detail, confusion matrices for YOLOv7, YOLOv8, and YOLOv11 are presented in Fig. 3. The YOLOv7 confusion matrix (Fig. 3(a)) reveals frequent misclassifications between AFR and BFR, as well as between AFR and UNR. This observation is consistent with the lower precision of AFR shown in Table III, indicating challenges in distinguishing early ripeness stages. Misclassifications in BHR further suggest that the model struggles with high-ripeness boundaries, highlighting the need for improved feature separation.

(a) (b)

(c)

FIGURE 4
Prediction result image for each YOLO version (a) YOLOv7, (b) YOLOv8, (c) YOLOv11.

The confusion matrix for YOLOv8 (see Fig. 3(b) demonstrates improved class distinction, particularly reducing confusion between AFR and BFR. This improvement reflects the higher recall values seen in Table IV, with fewer false negatives across most classes. Nonetheless, some misclassifications persist, especially between BHR and background elements. These indicate that while the model is more accurate, background variation still poses a challenge.

The confusion matrix for YOLOv8 (see Fig. 3(b) demonstrates improved class distinction, particularly reducing confusion between AFR and BFR. This improvement reflects the higher recall values seen in Table IV, with fewer false negatives across most classes. Nonetheless, some misclassifications persist, especially between BHR and background elements. These indicate that while the model is more accurate, background variation still poses a challenge.

The confusion matrix of YOLOv11 (see Fig. 3(c)) demonstrates the most distinct class separation among all three models. Misclassifications between AFR and BFR are notably reduced, contributing to YOLOv11's improved recall and mAP scores. This enhanced performance is likely attributed to advanced features such as attention mechanisms and adaptive receptive fields. Although some misclassifications persist in the BHR class, they are fewer and more localized, aligning with the model's slightly lower precision as reported in Table V. These matrices underscore the improvement in classification accuracy and the reduction of errors, particularly between challenging class pairs like AFR–BFR and BHR–background. The advancements in YOLOv11 reflect the effectiveness of enhanced attention and receptive field strategies.

The confusion matrix for YOLOv8 (see Fig. 3(b) demonstrates improved class distinction, particularly reducing confusion between AFR and BFR. This improvement reflects the higher recall values seen in Table IV, with fewer false negatives across most classes. Nonetheless, some misclassifications persist, especially between BHR and background elements. These indicate that while the model is more accurate, background variation still poses a challenge.

The confusion matrix of YOLOv11 (see Fig. 3(c)) demonstrates the most distinct class separation among all three models. Misclassifications between AFR and BFR are notably reduced, contributing to YOLOv11's improved recall and mAP scores. This enhanced performance is likely attributed to advanced features such as attention mechanisms and adaptive receptive fields. Although some misclassifications persist in the BHR class, they are fewer and more localized, aligning with the model's slightly lower precision as reported in Table V. These matrices underscore the improvement in classification accuracy and the reduction of errors, particularly between challenging class pairs like AFR–BFR and BHR–background. The advancements in YOLOv11 reflect the effectiveness of enhanced attention and receptive field strategies.

Overall, the confusion matrices illustrate a clear progression from YOLOv7 to YOLOv11, with each successive model reducing error rates and improving class differentiation. However, further refinement in background detection and high-ripeness classes such as BHR could lead to even greater accuracy. These findings support the quantitative performance gains observed in the evaluation metrics.

Fig. 4 shows the predicted output images from each YOLO version. The image compares the performance of three versions of the YOLO object detection models—YOLOv7 (a), YOLOv8 (b), and YOLOv11 (c)—on the task of detecting strawberries. Each subfigure shows a grid of test images, some annotated with bounding boxes and class predictions, providing a visual comparison of model accuracy and confidence. YOLOv7 underperforms significantly and may require retraining or hyperparameter tuning. YOLOv8 performs adequately but has room for improvement. YOLOv11 is clearly the most effective for this task, showing robust detection and accurate class labeling.

V. CONCLUSION

In this study, we presented a framework for real-time strawberry ripeness detection using three YOLO-based models—YOLOv7, YOLOv8, and YOLOv11. Our experiments showed that YOLOv7 achieved moderate results (AP=0.833, recall=0.794, mAP@50–95=0.769), while YOLOv8 improved overall performance (AP=0.883, recall=0.834, mAP@50=0.938). Notably, YOLOv11, which incorporates dynamic attention modules and self-adaptive layer-wise feature fusion, delivered the highest recall (0.855) and mAP@50–95 (0.820) despite a slight drop in precision (0.874), demonstrating enhanced sensitivity and robustness in complex agricultural conditions. Real-world deployment challenges such as lighting variability and resource-limited hardware constraints have also been considered, necessitating adaptive preprocessing and model quantization strategies for practical implementation. These findings indicate that further fine-tuning and additional data augmentation could address the precision trade-offs, and future work will focus on integrating these models into automated harvesting systems to optimize crop yield and quality control.

ACKNOWLEDGEMENT

The authors wish to convey their profound appreciation to Telkom University for the internal research grant No. 340/LIT06/PPM-LIT/2024 and to Ichigo Farm, Ciwidey, Bandung, West Java, Indonesia, for providing the strawberries dataset and sharing their expertise.

REFERENCES

[1] N. Mamat, M. F. Othman, R. Abdoulghafor, S. B. Belhaouari, N. Mamat, and S. F. M. Hussein, "Advanced technology in agriculture industry by implementing image annotation technique and deep learning approach: A review," *Agriculture*, vol. 12, no. 7, p. 1033, 2022, doi: 10.3390/agriculture12071033.

[2] Z. Mi and W. Q. Yan, "Strawberry ripeness detection using deep learning models," *Big Data and Cognitive Computing*, vol. 8, no. 8, p. 92, 2024, doi: 10.3390/bdcc8080092.

[3] J. J. K. Chai, J.-L. Xu, and C. O'Sullivan, "Real-time detection of strawberry ripeness using augmented reality and deep learning," *Sensors*, vol. 23, p. 7639, 2023, doi: 10.3390/s23177639.

[4] E. Badeka, E. Karapatzak, A. Karampatea, E. Bouloumpasi, I. Kalathas, C. Lytridis, E. Tziolas, V. N. Tsakalidou, and V. G. Kaburlasos, "A deep learning approach for precision viticulture, assessing grape maturity via YOLOv7," *Sensors*, vol. 23, no. 19, p. 8126, 2023, doi: 10.3390/s23198126.

[5] S. Azizah, M. Wahidin, M. Padang, L. Novamizanti, and S. Sa'idah, "Identifying the ripeness and quality level of strawberries based on YOLOv7-EfficientNet," in *Proc. Int. Conf. Data Sci. Appl. (ICoDSA)*, Kuta, Indonesia, 2024, pp. 451–456, doi: 10.1109/ICoDSA62899.2024.10651988.

[6] M. Zhang, S. Ye, S. Zhao, W. Wang, and C. Xie, "Pear object detection in complex orchard environment based on improved YOLO11," *Symmetry*, vol. 17, no. 2, p. 255, 2025, doi: 10.3390/sym17020255.

[7] B. Xiao, M. Nguyen, and W. Q. Yan, "Fruit ripeness identification using YOLOv8 model," *J. Real-Time Image Process*., Aug. 2023, doi: 10.1007/s11554-023-01345-6.

[8] GitHub Repository: "Strawberry Counting and Ripeness Detection using YOLOv7," 2024. [Online]. Available: https://github.com/amitamola/Strawberry-Counting-and-Ripeness-detection (Accessed: May 24, 2025).

[9] Y. Chen et al., "CES-YOLOv8: Strawberry Maturity Detection Based on the Improved YOLOv8," *Agronomy*, vol. 14, no. 7, p. 1353, 2024.

[10] X. Chen, "Rapid Strawberry Ripeness Detection and 3D Localization of Picking Point Based on Improved YOLO V8-Pose with RGB-Camera," *J. Electr. Syst*., 2024.

[11] A.-R. A. Gamani, I. Arhin, and A. K. Asamoah, "Performance Evaluation of YOLOv8 Model Configurations for Instance Segmentation of Strawberry Fruit Development Stages in an Open Field Environment," *arXiv* preprint arXiv:2408.05661, 2024. [Online]. Available: https://arxiv.org/abs/2408.05661 (Accessed: May 24, 2025).

[12] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *arXiv* preprint arXiv:2207.02696, 2022. [Online]. Available: https://arxiv.org/abs/2207.02696 (Accessed: April 10, 2025).

[13] Ultralytics, "YOLOv8," 2023. [Online]. Available: https://github.com/ultralytics/yolov8 (Accessed: May 24, 2025).

[14] X. Li, Y. Chen, and K. Zhang, "YOLOv11: Latest Advances in Real-Time Object Detection," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2024, pp. 1453–1461.

[15] Roboflow Inc., "Roboflow: Data Preparation for Computer Vision," 2023. [Online]. Available: https://roboflow.com (Accessed: May 24, 2025).

[16] A.K. Aziz, M.D. Maulana, R.F. Adawiyah, R.F. Firdaus, L. Novamizanti, and F. Ramdhon, "Comparative analysis of YOLOv8 models in skipjack fish quality assessment system," in *2023 3rd International Conference on Intelligent Cybernetics Technology & Applications (ICICyTA)*, December 2023, (pp. 237-242). IEEE.

[17] F. Akhyar, L. Novamizanti, K. Usman, G.M. Aditya, F.N. Hakim, M.Z. Ilman, F. Ramdhon, and C.Y. Lin, "A Comparative Analysis of the Yolo Models for Intelligent Lobster Surveillance Camera," in *2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, October 2023, (pp. 2131-2136). IEEE.

[18] F. Akhyar, L. Novamizanti, T. Putra, E.N. Furqon, Furqon, M.C. Chang, and C.Y. Lin, "Lightning YOLOv4 for a surface defect detection system for sawn lumber," in *2022 IEEE 5th International Conference on Multimedia Information Processing and Retrieval (MIPR)*, August 2022, (pp. 184-189). IEEE.

[19] R. Padilla, W. L. Passos, T. L. B. Dias, S. L. Netto, and E. A. B. da Silva, "A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit," *Electronics*, vol. 10, no. 3, p. 279, 2021, doi: 10.3390/electronics10030279.