IJCCS (Indonesian Journal of Computing and Cybernetics Systems)
Vol.x, No.x, July xxxx, pp. 1~5
ISSN (print): 1978-1520, ISSN (online): 2460-7258
DOI: 10.22146/ijccs.xxxx

A Comparative Deep Learning Approach for Classifying Oil Palm Fruit Ripeness Levels Using YOLOv8s and Faster R-CNN

Rasmila[1*], Sakbanullah Dwi Oktavian[2], Rahmat Novrianda[3], Rahayu Amalia[4]
[1,2,3,4]Program Studi Teknik Informatika, Universitas Bina Darma, Indonesia Indonesia
e-mail: [*1]rasmila@binadarma.ac.id, [2]okta45168@gmail.com,
[3]rahmat_novrianda@binadarma.ac.id, [4]rahayu_amalia@binadarma.ac.id.

Abstrak

Penilaian tingkat kematangan buah kelapa sawit penting untuk mengoptimalkan waktu panen dan meningkatkan nilai jual. Di banyak wilayah berkembang, panen masih dilakukan setiap 10–15 hari melalui inspeksi visual manual yang rentan kesalahan dan sering menyebabkan panen prematur. Hal ini menurunkan kualitas buah dan nilai jual hingga 50%, menimbulkan kerugian ekonomi bagi petani. Penelitian ini mengkaji penerapan deteksi objek berbasis pembelajaran mendalam untuk klasifikasi otomatis tandan buah segar (TBS) kelapa sawit. Dataset berisi 4.578 citra beresolusi tinggi yang telah dianotasi ke dalam enam kelas kematangan: Kosong, Mentah, Setengah Matang, Abnormal, Matang, dan Lewat Matang. Dua model deteksi terkini, YOLOv8s dan Faster R-CNN dengan backbone ResNet-50, dievaluasi menggunakan precision, recall, dan mean Average Precision (mAP). Hasil menunjukkan YOLOv8s mencapai precision dan recall di atas 99% dengan mAP 0.5:0.95 sebesar 0.9254, efisien untuk penggunaan real-time. Faster R-CNN mencapai mAP 0.5 sebesar 0.9964 dengan akurasi lebih tinggi namun waktu komputasi lebih lama. YOLOv8s menawarkan keseimbangan optimal antara akurasi dan kecepatan. Penelitian ini berkontribusi pada pertanian presisi melalui penerapan kecerdasan buatan untuk meningkatkan produktivitas dan keberlanjutan.

Kata kunci: *Object Detection, YOLOv8s, Computer Vision, Faster R-CNN,Precision Agriculture*

Abstract

Assessing oil palm fruit ripeness is essential for optimizing harvest timing and maximizing market value. In many developing regions, harvesting is still performed every 10–15 days through manual visual inspection, a process prone to human error that often causes premature harvesting and reduces selling value by up to 50%. This study explores deep learning-based object detection for automatic classification of oil palm fruit bunches. A dataset of 4,578 annotated high-resolution images was prepared and categorized into six ripeness classes: Empty, Immature, Underripe, Abnormal, Ripe, and Overripe. Two advanced detection models, YOLOv8s and Faster R-CNN with a ResNet-50 backbone, were evaluated under identical conditions using precision, recall, and mean Average Precision (mAP) metrics. YOLOv8s achieved precision and recall above 99%, with a mAP 0.5:0.95 of 0.9254, demonstrating strong reliability and efficiency for real-time use. Faster R-CNN achieved a higher mAP 0.5 of 0.9964, indicating superior localization accuracy but slower computation. Overall, YOLOv8s provides a better trade-off between accuracy and speed, making it more practical for automated harvesting. This research supports precision agriculture by emphasizing AI driven solutions that improve productivity, minimize losses, and promote sustainable palm oil management.

Keywords: *Object Detection, YOLOv8s, Computer Vision, Faster R-CNN,Precision Agriculture*

Received June 1st,2012; Revised June 25th, 2012; Accepted July 10th, 2012

1. INTRODUCTION

Indonesia has historically been among the globe's top producers and exporters of crude palm oil (CPO), a product that is vital for the national economy and the well-being of millions of Indonesians.[1], [2]. However, the area encounters various operational obstacles that impede maximum productivity. A primary challenge is the precise determination of the ripeness stage of fresh fruit bunches (FFBs), which directly influences oil yield, processing quality, and total commercial worth. Traditionally, plantation workers manually assess FFB ripeness through visual inspection, relying on color and texture indicators. Although this technique has been utilized for many years, it remains very subjective, susceptible to variations, and frequently results in errors, particularly when performed under tight deadlines or in suboptimal conditions. Manual inspection becomes increasingly inefficient on large plantations, where thousands of trees must be evaluated daily. Additionally, environmental elements like differing lighting situations, intricate backgrounds, and obstructions from nearby vegetation can complicate the ripeness assessment process, hindering even skilled workers from achieving consistent accuracy (Lai et al., 2023).

In many rural and semi-urban regions, including the authors' village, growers are increasingly interested in oil palm farming because of its significant economic opportunities. However, gathering operations in these regions continue to be conducted by hand and without technological support. Usually, fruit clusters are examined regularly every 10 to 15 days. Farmers assess ripeness by observing color variations: unripe fruits look dark black, semi-ripe fruits exhibit a blackish-reddish tint, and fully ripe fruits become vibrant reddish-orange[4]. These visual signals are the main determinants for choosing the right time to harvest. This manual method is subjective and often inaccurate. Based on field experience, about 15% of fruits are harvested prematurely, despite appearing ripe externally. This error in judgment may result in substantial monetary loss. During the study, fully ripe fruit bunches that exceeded 6 kilograms were priced at approximately IDR 2,940 for each kilogram. Nonetheless, clusters weighing merely 3 to 5 kilograms or any fruit deemed unripe regardless of weight were sold at a much lower price of only IDR 1,450 per kilogram. This indicates a possible decrease of 50% in market value, directly affecting farmer profit margins. The insufficient training and the non-existence of effective, budget-friendly detection tools exacerbate the issue, emphasizing the urgent need for an intelligent, objective, and accessible solution.

In recent times, Deep learning and artificial intelligence (AI) developments have surfaced as potential remedies for these farming problems. Convolutional neural networks' (CNNs') advancements have transformed object detection into an efficient and versatile technology that supports multiple applications, including autonomous driving, surveillance systems, medical diagnostics, and intelligent agriculture. In agriculture, object detection algorithms have shown considerable success in areas like pest detection, disease identification, yield forecasting, and fruit localization, providing both accuracy and swiftness[5]. Prominent object detection methods consist of YOLO (You Only Look Once) and Faster R-CNN (Region-based Convolutional Neural Network), both demonstrating unique architectural approaches. YOLOv8s, a recent version launched by Ultralytics, is an efficient, single-stage detector that evaluates the entire image in one pass and forecasts object bounding boxes alongside class probabilities simultaneously. Its efficient design allows for real-time processing and low-latency inference, making it especially ideal for use in field settings where swift decisions are necessary. The YOLOv8s model features advancements like an anchor-free detection system, separate heads for classification and regression [6], and design enhancements that boost training efficiency and detection precision. Conversely, Faster R-CNN employs a two-phase detection approach. It initially employs a Region Proposal Network (RPN) to locate potential object areas and subsequently utilizes a second stage to classify and enhance these proposals through a convolutional network. This method, while more computationally intensive and typically slower than YOLO based models, usually attains greater accuracy in situations with cluttered scenes, small or overlapping objects, and intricate backgrounds. Utilizing ResNet-50 as its backbone enables Faster R-CNN to obtain rich hierarchical features from images, thereby enhancing its performance in intricate visual tasks[7].

Earlier studies have investigated CNN-driven classification for palm fruit ripeness, typically without utilizing comprehensive object detection techniques. For instance, [8]. and [9]. employed simple CNN classifiers to forecast maturity phases, missing object localization.. introduced EfficientDet-Lite aimed at lightweight classification, but did not assess inference efficiency among detectors. Furthermore, [10]. emphasized the increasing importance of object detection in agriculture while mentioning the absence of comparative studies specific to palm oil plantations. This research tackles these gaps by experimentally comparing YOLOv8s and Faster R-CNN for identifying six maturity classifications of oil palm FFBs: Empty, Immature, Underripe, Abnormal, Ripe, and Overripe. Using a diverse and annotated dataset of 4,578 images, the models are evaluated on accuracy, recall, and mean Average Precision (mAP) over various Intersection over Union (IoU) thresholds. The goal is to determine the most effective object detection model for automated harvesting systems and to provide practical suggestions for its application in smart agriculture[11].

2. METHODS

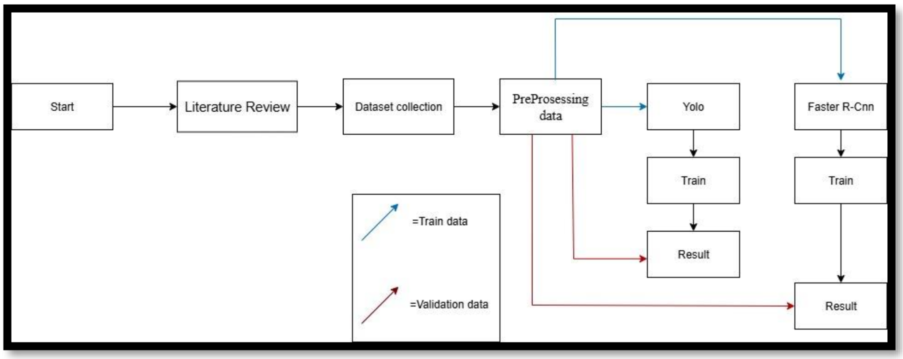

Figure 1 Research process

2.1 Dataset Preparation

This study employs an extensive dataset consisting of 4,578 high-resolution images of oil palm fruit bunches, obtained from both open-access sources and manual field gathering. A sum of 4,078 images was obtained from Kaggle and Roboflow two popular platforms in computer vision while 500 more images were taken manually with a standard digital camera in actual plantation settings[12]. These images gathered manually promote class balance and better depict real-world variability. Every image was marked with bounding boxes and classified into one of six maturity categories: Empty Bunch, Underripe, Abnormal FFB, Ripe FFB[13], Immature FFB, and Overripe. Labeling relied on visual characteristics like fruit hue, surface texture, and the existence of loose fruits criteria typically employed in palm oil harvesting to assess ripeness. To improve model robustness, the dataset encompasses diverse environmental circumstances like varying lighting (sunlight, shadow, overcast), backgrounds (soil, foliage, sky), and camera perspectives. These changes replicate real-life difficulties and assist in enhancing generalization[14]. The dataset was divided into training (70%, 3,204 images), validation (20%, 916 images), and testing (10%, 458 images) through stratified sampling to maintain balanced class representation. This separation guarantees fair assessment, as the test set remains independent from training and validation activities. Merging curated online information with field-collected images creates a more varied and authentic dataset, facilitating the creation of models that can detect accurately in real-world plantation situations, aiding applications like automated harvesting, fruit classification, and yield assessment[15].

2.2 Preprocessing and Labeling

All images in the dataset were scaled to 640×640 pixels to standardize input dimensions and maintain compatibility across both detection models. This resizing was implemented uniformly to uphold aspect ratios and ensure consistency throughout training and evaluation. The selected resolution achieves a compromise between processing efficiency and the preservation of important visual elements, like fruitlet texture and color variations, which are vital for differentiating between various maturity stages of oil palm fruit. To maintain the original distribution and features of the dataset, no data augmentation methods like flipping, rotation, brightness modification, or noise addition were utilized. This decision was made to avoid introducing artificial distortions that could bias the model or obscure subtle but significant visual cues associated with specific maturity classes. The models were trained exclusively using naturally captured images across diverse real-world conditions to adapt to authentic changes in lighting, background clutter, and fruit orientation.

The dataset was annotated using Roboflow's annotation tool, a cloud-based labeling platform that enables collaborative and efficient bounding box creation. Every image was carefully annotated by creating bounding boxes around the desired fruit clusters and designating the correct class labels according to established maturity categories. To uphold labeling consistency, annotation guidelines were created to guarantee that the bounding boxes precisely encompassed the pertinent fruit regions while reducing the capture of adjacent background. After the annotation process, the dataset was converted into two common annotation formats to meet the needs of each detection architecture. For YOLOv8s, the dataset was saved in YOLO format, featuring normalized bounding box coordinates and class indices in text files, tailored for YOLO based models. The dataset for Faster R-CNN was exported in COCO (Common Objects in Context) format, utilizing a structured JSON format that accommodates multiple objects per image, class hierarchy, and segmentation suiting the requirements of two-stage detection pipelines effectively.

This dual-format export approach allows for direct integration with both model architectures, eliminating the necessity for extra preprocessing or format conversion, thus simplifying the training process. Additionally, upholding uniform image sizes, annotation quality, and labeling criteria across both formats guarantees an equitable and regulated evaluation of the models' performance under the same data conditions.

2.3 Preprocessing and Labeling

To guarantee a fair and consistent evaluation under the identical experimental settings, the training procedures for the two object identification models YOLOv8s and Faster R-CNN with a ResNet-50 backbone were meticulously planned. Using the same dataset and training parameters whenever possible, the primary objective was to assess and compare how well the two architectures identified the oil palm fruit's developmental phases.

2. 3.1 YOLOv8s Training Setup

The official Ultralytics implementation was utilized to train the YOLOv8s model, providing a simplified and accessible interface for setting up and deploying YOLO-based models. The training utilized the standard hyperparameter settings supplied by Ultralytics, as these configurations are fine-tuned for general object detection purposes. Essential training parameters for YOLOv8s encompassed:

Table 1. YOLOv8s Training Setup

Number of epochs	150
Batch size	8
Initial learning rate	0.01
Optimizer	Stochastic Gradient Descent (SGD)

How many times the entire training dataset is input into the model throughout the training phase is indicated by the number of epochs. To guarantee sufficient learning iterations and improve generalization in this situation, the model was trained for 150 epochs[16]. Batch size determines the number of images handled concurrently during one iteration. A batch size of 8 was selected to optimize memory efficiency and model convergence, enabling the training to effectively use available GPU resources without exceeding limits[17]. Initial learning rate, established at 0.01, dictates the size of the update steps made while modifying the model's internal weights. An appropriate learning rate is crucial to guarantee that the model converges effectively without overshooting or becoming stagnant[18]. Optimizer, Stochastic Gradient Descent (SGD), is a method employed to adjust the model's parameters according to the gradients obtained during backpropagation. SGD is recognized for its reliability and is ideal for models such as YOLOv8s that emphasize quickness and straightforwardness. Training took place on a workstation featuring an NVIDIA RTX 3060 GPU (12GB VRAM), 32GB RAM, and an Intel Core i9 CPU. In this arrangement, training was finished in around 2.5 hours, demonstrating the computational efficiency and rapid convergence of the YOLOv8s model.

2. 3.2 Faster R-CNN Training Setup

Conversely, the Faster R-CNN model was created utilizing the torchvision framework based on PyTorch, which provides strong tools for creating two-stage object detection models[19]. A ResNet-50 backbone, pre-trained on the ImageNet dataset, was utilized to harness transfer learning, allowing the model to effectively extract high-level semantic features. For experimental consistency, the Faster R-CNN model was trained for 150 epochs, enabling the model to progressively adjust its weights via several complete cycles over the dataset.Owing to the complexity of the architecture, particularly during its region proposal phase and deep feature extraction, the batch size was decreased to 4 to avoid memory overflow on the identical hardware setup. This batch size enables the model to work with smaller data subsets while maintaining training stability[20]. Initial learning rate was set at 0.01 to remain consistent with the YOLOv8s configuration, guaranteeing that the training step sizes were similar. Adam optimizer was utilized for optimization. Adam is a flexible learning algorithm that modifies the learning rate for each parameter separately, utilizing both first- and second-order gradients. This renders it especially useful for intricate and advanced models such as Faster R-CNN, which gain from the adjustment of the learning rate dynamically. Even though the identical hardware was utilized an NVIDIA RTX 3060 GPU (12GB VRAM), 32GB RAM, and an Intel Core i9 CPU the training duration for Faster R-CNN was much longer, around 20 hours, because of the model's computational requirements and its multi-stage processing architecture.

Table 2. Faster R-CNN Training Setup

Number of epochs	150
Batch size	4 (due to higher memory usage)
Initial learning rate	0.01
Optimizer	Adam

3. RESULTS AND DISCUSSION

3.1 Training Performance

The YOLOv8s model was developed utilizing 4,578 annotated training images and 916 validation images across 150 epochs, using the official Ultralytics implementation. The training procedure was carried out locally on a machine with an NVIDIA RTX 3060 GPU (12GB VRAM), allowing for effective computation while ensuring memory consistency. The training utilized the standard hyperparameters of the YOLOv8 framework, incorporating the Stochastic Gradient

Descent (SGD) optimizer A batch size of 8 was chosen to optimize computational efficiency while considering hardware constraints. During training, no data augmentation methods were utilized, maintaining the dataset's original structure and feature distribution. The whole training procedure finished in roughly 2.5 hours, while GPU memory usage stayed steadily near 8GB. The model demonstrated consistent convergence traits with negligible variations in loss values, signifying efficient learning and generalization abilities throughout the epochs.

Table 3. YOLOv8s Training Results

Epoch	Box Loss	Class Loss
50	0.54162	0.35071
100	0.45247	0.26629
150	0.27669	0.13542

As shown in Table 3, the Box Loss and Class Loss metrics both exhibited a steady decline as training advanced. This demonstrates a consistent improvement in the model's capacity to identify object edges and correctly allocate the relevant labels. The downward trend suggests that YOLOv8s was efficiently reducing prediction errors and improving its internal representations progressively. At the last epoch, the Box Loss decreased by almost 50% from its average value, and the Class Loss saw a comparable decline, validating the model's capacity to learn distinguishing object features effectively even without augmentation techniques. Simultaneously, the Faster R-CNN model was trained on the identical training and validation dataset, hardware setup, and total epochs (150). The model was developed with the PyTorch torchvision library and employed a ResNet-50 backbone that was pre-trained on ImageNet to enhance feature extraction speed. In contrast to YOLOv8s, Faster R-CNN uses a two-phase detection technique where a classification and regression step comes after the Region Proposal Network (RPN) first finds possible object areas. The Adam optimizer and a learning rate scheduler that dynamically reduced the learning rate as the model got closer to convergence were used to train the model. The total training time grew to about 20 hours due to its complex architecture and the sequential processing of regional proposals.

Table 4. Faster R-CNN Training Result

Epoch	Box Loss	Class Loss
50	0.0531	0.0338
100	0.0338	0.0168
150	0.0440	0.0328

Table 4 shows that the Faster R-CNN model realized significant decreases in both Box Loss and Class Loss in the early phases of training. At epoch 100, the model achieved its minimum recorded Box Loss and sustained Class Loss values at a consistently low range. Even though a minor rise in Class Loss was noted toward the last epoch, the figures stayed within a satisfactory range and did not suggest any overfitting or instability. These outcomes demonstrate the model's ability to achieve detailed localization and classification, a significant benefit of its two-stage detection approach. The performance trends indicate that the Faster R-CNN model successfully captured deeper semantic features early in the training phase, enabling it to generalize effectively on the validation set, even with the extended training duration.

3.2 Model Testing Results

The testing phase utilized 458 images that were absent from both the training and validation subsets. These images were intentionally reserved to evaluate the generalization ability of both YOLOv8s and Faster R-CNN when faced with new data. This stage is essential to confirm that the models do not overfit the training distribution and can consistently operate in practical applications that include real-world variability. Four common metrics for object detection were used in a thorough assessment: precision, recall, mAP 0.5, and mAP 0.5:0.95. Precision measures the model's capacity to reduce false positives by calculating the percentage of accurately predicted items compared to all detections. Recall highlights the model's effectiveness in reducing false negatives by assessing its ability to identify all pertinent objects. When there is adequate overlap between the predicted and real bounding boxes, the average detection precision is shown by the metric mAP 0.5 (mean average precision at an intersection over union threshold of 0.5). However, mAP 0.5:0.95 provides a more comprehensive evaluation by calculating average performance across several IoU thresholds (0.5 to 0.95).and acts as a reliable measure of detection quality under tougher standards. The two models were assessed under the same conditions to guarantee fairness. No external calibration or post-processing was performed, and evaluations were made directly based on the model outputs. The assessment outcomes showed unique attributes of every model. YOLOv8s, featuring a lightweight and real-time-focused design, demonstrated notably better performance in terms of inference speed and consistent accuracy. Conversely, Faster R-CNN exhibited superior recall and mAP values, particularly in situations with intricate object overlapping or nuanced class distinctions due to its region-based proposal method and enhanced feature extraction through the ResNet-50 backbone.

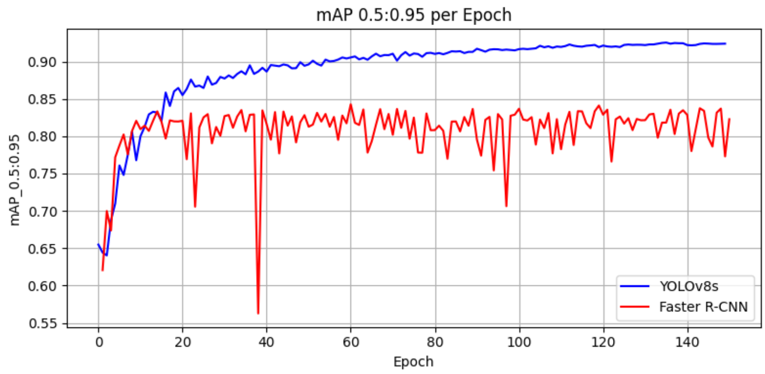

Figure 2 Comparison of YOLOv8s and Faster R-CNN based on mAP 0.5:0.95

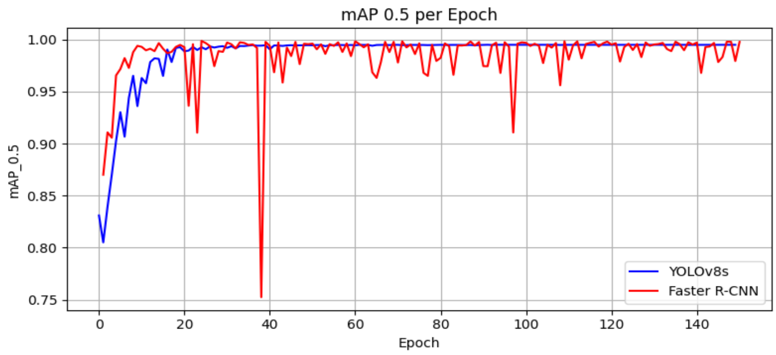

Figure 3 Comparison of YOLOv8s and Faster R-CNN based on mAP 0.5

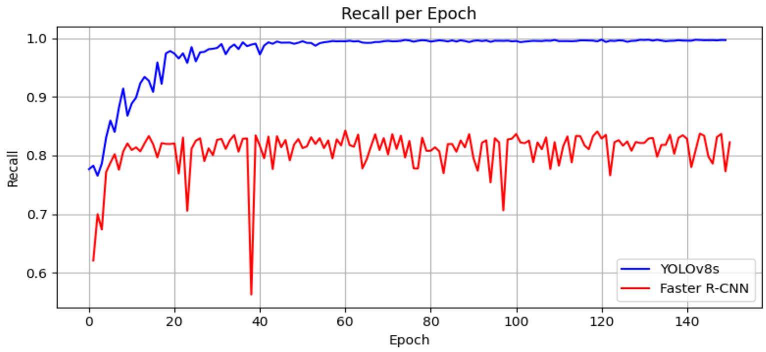

Figure 4 Comparison of YOLOv8s and Faster R-CNN based on Recall

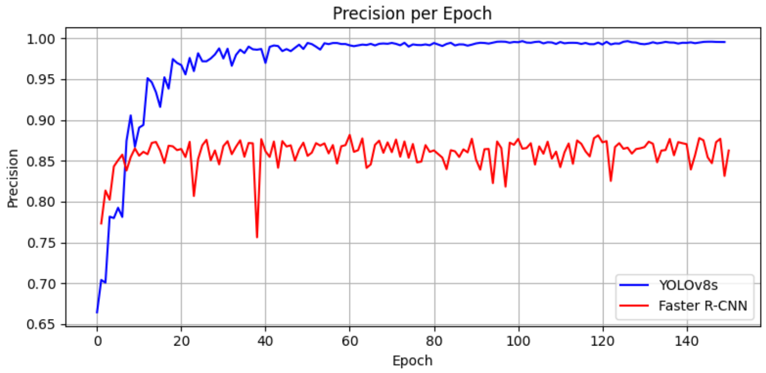

Figure 5 Comparison of YOLOv8s and Faster R-CNN based on Precision

As shown in Fig. 3, the YOLOv8s model exhibits remarkable performance with a mAP 0.5:0.95 score that surpasses 90%. This shows that the model is very precise in detecting and categorizing objects across various IoU (Intersection over Union) thresholds. Conversely, Faster R-CNN also shows strong performance, exceeding 80% and coming close to 85% in the identical metric. Although both models show impressive performance, YOLOv8s distinctly excels in consistency across evaluation thresholds, while Faster R-CNN reaches its peak score at a particular threshold.

In Fig. 4, YOLOv8s once more shows outstanding performance, achieving a recall score that is close to 100%. This indicates a strong ability to recognize all pertinent objects in the test images. On the other hand, Faster R-CNN also produces favorable results, with a recall rate surpassing 80%, but it doesn't quite match YOLOv8's level of comprehensiveness.

In the meantime, Fig. 5 illustrates that YOLOv8s reaches a precision score exceeding 95%, demonstrating a robust capability to reduce false positives in detection. Even though it is slightly reduced, Faster R-CNN maintains an impressive precision score exceeding 85%, showcasing its competitive accuracy in object detection. In general, although Faster R-CNN achieves the top score in mAP 0.5 (refer to Fig. 3), YOLOv8s reliably outperforms it in various metrics. Its durability and ability to generalize render it better suited for practical applications that require high precision and consistency under diverse circumstances. Throughout the training process, testing was conducted after each epoch to monitor model performance. Based on these evaluations, the best-performing model for each algorithm was selected and is presented in Table 5. This table summarizes the most optimal results achieved by YOLOv8s and Faster R-CNN on the test dataset.

Table 5. Overall Performance of the Best Model

Best Model	Precision	Recall	mAP 0.5	mAP 0.5:0.95
YOLOv8s	0.9954	0.9951	0.9948	0.9254
Faster R-CNN	0.8740	0.8355	0.9964	0.8355

According to the test outcomes, the YOLOv8s model exhibited exceptional performance, achieving precision and recall over 99%, along with a mAP 0.5:0.95 score of 0.9254. These outcomes suggest that the model can effectively identify and categorize objects at various Intersection over Union (IoU) thresholds, demonstrating notable generalization and resilience. While the high recall verifies the model's ability to consistently identify all important elements in the input photographs, the extraordinarily high precision indicates that the model generates few false positives. Both the broader mAP 0.5:0.95 metric and its high performance on mAP 0.5 are enhanced by this balance between recall and precision. The Faster R-CNN model, on the other hand, achieved an even better mAP 0.5 score of 0.9964 reflecting almost flawless localization and classification precision at the 0.5 IoU threshold. This exceptional performance at one threshold shows the power of its two-stage design, where region proposals are improved prior to the final classification. Nevertheless, the performance of Faster R-CNN in terms of precision (87.40%), recall (83.55%), and mAP 0.5:0.95 (0.8355) was somewhat inferior to that of YOLOv8s, particularly when assessed with more stringent IoU evaluation standards. This indicates that although Faster R-CNN can efficiently enhance detections at a particular threshold, its performance across different IoU levels is less consistent compared to YOLOv8s.

3.3 Discussion

The experimental results highlight the distinct trade-offs between YOLOv8s and Faster R-CNN when applied to oil palm fruit maturity detection. Both models achieved strong performance, yet their suitability for practical applications diverges depending on the operational context. YOLOv8s demonstrated an exceptional balance of accuracy and computational efficiency, with precision and recall surpassing 99% and an mAP 0.5:0.95 score above 0.92. These results indicate the robustness of its single-stage anchor-free detection mechanism, which enables rapid inference without significantly sacrificing accuracy. This finding aligns with the work of Jocher who emphasized the real-time adaptability of YOLO architectures for edge devices and field-based deployments.

In contrast, Faster R-CNN achieved an outstanding mAP 0.5 of 0.9964, underscoring its strength in precise localization under moderate IoU thresholds. However, the decrease in performance under stricter evaluation (mAP 0.5:0.95 = 0.8355) suggests that its two-stage region proposal mechanism, while effective for fine-grained detection, is less consistent across diverse IoU ranges. Similar observations have been reported by Ren , where Faster R-CNN excels in benchmark conditions but exhibits latency and scalability challenges in resource-limited environments.

From a practical standpoint, the training and inference efficiency of YOLOv8s completed in approximately 2.5 hours presents significant advantages for real-world agricultural use. In comparison, Faster R-CNN required nearly 20 hours of training, posing limitations for iterative retraining or adaptation to dynamic field conditions. The shorter computation time and lighter architecture of YOLOv8s enable its deployment on portable devices such as drones or edge-based processors, directly supporting smart farming initiatives that demand immediacy and scalability.

The generalization ability of YOLOv8s was shown through its stable performance under varying conditions such as lighting, background clutter, and partial occlusion. This robustness is vital in agriculture, where environmental unpredictability is common. Conversely, Faster R-CNN proved more sensitive to such variations, making its superior detection accuracy more suitable for controlled or semi-automated facilities than direct field use. Overall, the choice between YOLOv8s and Faster R-CNN depends on operational priorities. For large-scale plantations requiring real-time decision support and rapid harvesting, YOLOv8s offers a practical solution. Meanwhile, for applications where detection precision at standard IoU thresholds is critical, Faster R-CNN remains competitive. This study highlights the complementary strengths of both models and provides empirical evidence to support the integration of deep learning into precision agriculture, advancing AI-driven palm oil management.

However, this study has several limitations that should be acknowledged. The dataset used, although diverse, was limited to 4,578 images, which may not fully represent the wide variability of real-world plantation conditions. The experiments only compared two object detection models (YOLOv8s and Faster R-CNN), without considering other recent architectures that could offer different trade-offs. In addition, the evaluation was conducted on static images under controlled conditions, without direct field deployment. These limitations suggest that the results, while promising, may not yet fully reflect performance in operational harvesting environments.

4. CONCLUSIONS

This study provides a unique comparative assessment of YOLOv8s and Faster R-CNN models for identifying six ripeness stages of oil palm fruit, establishing a strong benchmark for applications in precision agriculture. The main advancement is the utilization of a varied, high-resolution, and practical dataset specifically designed for classifying oil palm maturity, which has been significantly overlooked in earlier research.Experimental findings reveal that YOLOv8s exceeds projections by attaining an unusual blend of very high accuracy and nearly immediate inference speed, achieving precision and recall rates above 99%, alongside a mAP 0.5:0.95 of 0.9254. These results represent a noteworthy progress in instantaneous detection for resource-limited settings like smallholder farms.

In the meantime, Faster R-CNN attained an impressive mAP 0.5 of 0.9964, underscoring its capability for precise detection at a standard IoU threshold, essential for high-fidelity tasks. Nonetheless, its comparatively poorer performance at tougher thresholds and longer training duration highlight its limitations for field use without advanced computing resources. The outstanding performance indicators for both models demonstrate a distinct compromise between real-time efficiency and detection detail. YOLOv8s emerges as the ideal choice for scalable, automated fruit maturity assessment in practical agricultural environments, where speedy, precise, and efficient processing is crucial. This research establishes a foundation for the future advancement of AI-powered harvesting tools and provides a verified dataset along with a baseline comparison for additional studies in precision agriculture for palm oil.

ACKNOWLEDGEMENTS

The authors would like to express their gratitude to Universitas Bina Darma for the support and facilities provided throughout the completion of this research. Special thanks are also extended to the academic supervisors and colleagues whose guidance and insights greatly contributed to the success of this study.

REFERENCES

[1] N. Yuslaini, R. Widya Setiabudi Sumadinata, M. Fedryansyah, A. Abdillah, A. L. Prianto, and D. Febriyanti, "Sustainable investment strategies in the palm oil industry in Indonesia," *Journal of Infrastructure, Policy and Development*, vol. 7, no. 3, 2023, doi: 10.24294/jipd.v7i3.2288.

[2] N. Ngadi and J. Nagata, "Oil Palm Land Use Change and Rice Sustainability in South Sumatra, Indonesia," *Land (Basel)*, vol. 11, no. 5, May 2022, doi: 10.3390/land11050669.

[3] J. W. Lai, H. R. Ramli, L. I. Ismail, and W. Z. Wan Hasan, "Oil Palm Fresh Fruit Bunch Ripeness Detection Methods: A Systematic Review," Jan. 01, 2023, *MDPI*. doi: 10.3390/agriculture13010156.

[4] Y. A. A. Soetrisno, E. Handoyo, Sumardi, and E. W. Sinuraya, "OIL PALM LEVEL OF RIPENESS CLASSIFICATION USING EFFICIENTDET-LITE CNN

ARCHITECTURE," *J Oil Palm Res*, vol. 36, no. 4, pp. 618–629, Dec. 2024, doi: 10.21894/jopr.2023.0059.

[5] M. Dalal and P. Mittal, "A Systematic Review of Deep Learning-Based Object Detection in Agriculture: Methods, Challenges, and Future Directions," 2025, *Tech Science Press*. doi: 10.32604/cmc.2025.066056.

[6] M. Ma and H. Pang, "SP-YOLOv8s: An Improved YOLOv8s Model for Remote Sensing Image Tiny Object Detection," *Applied Sciences (Switzerland)*, vol. 13, no. 14, Jul. 2023, doi: 10.3390/app13148161.

[7] I. Bonet, M. Gongora, F. Acevedo, and I. Ochoa, "Deep Learning Model to Predict the Ripeness of Oil Palm Fruit," in *International Conference on Agents and Artificial Intelligence*, Science and Technology Publications, Lda, 2024, pp. 1068–1075. doi: 10.5220/0012434600003636.

[8] S. Ashari, G. J. Yanris, and I. Purnama, "Oil Palm Fruit Ripeness Detection using Deep Learning," *Sinkron*, vol. 7, no. 2, pp. 649–656, May 2022, doi: 10.33395/sinkron.v7i2.11420.

[9] J. Zulkarnain, Kusrini, and T. Hidayat, "Klasifikasi Tingkat Kematangan Tandan Buah Segar Kelapa Sawit Menggunakan Pendekatan Deep Learning," *JST (Jurnal Sains dan Teknologi)*, vol. 12, no. 3, Jan. 2024, doi: 10.23887/jstundiksha.v12i3.59140.

[10] Z. Khan, Y. Shen, and H. Liu, "ObjectDetection in Agriculture: A Comprehensive Review of Methods, Applications, Challenges, and Future Directions," *Agriculture*, vol. 15, no. 13, p. 1351, Jun. 2025, doi: 10.3390/agriculture15131351.

[11] S. Lipiński, S. Sadkowski, and P. Chwietczuk, "Application of AI in Date Fruit Detection—Performance Analysis of YOLO and Faster R-CNN Models," *Computation*, vol. 13, no. 6, Jun. 2025, doi: 10.3390/computation13060149.

[12] P. Alirezazadeh, F. Rahimi-Ajdadi, Y. Abbaspour-Gilandeh, N. Landwehr, and H. Tavakoli, "Improved digital image-based assessment of soil aggregate size by applying convolutional neural networks," *Comput Electron Agric*, vol. 191, p. 106499, Dec. 2021, doi: 10.1016/J.COMPAG.2021.106499.

[13] Suharjito *et al.*, "Annotated Datasets of Oil Palm Fruit Bunch Piles for Ripeness Grading Using Deep Learning," *Sci Data*, vol. 10, no. 1, Dec. 2023, doi: 10.1038/s41597-023-01958-x.

[14] M. Mohinur Rahaman and M. Azharuddin, "Wireless sensor networks in agriculture through machine learning: A survey," *Comput Electron Agric*, vol. 197, p. 106928, Jun. 2022, doi: 10.1016/J.COMPAG.2022.106928.

[15] Q. Su, J. Tang, M. Zhai, and D. He, "An intelligent method for dairy goat tracking based on Siamese network," *Comput Electron Agric*, vol. 193, p. 106636, Feb. 2022, doi: 10.1016/J.COMPAG.2021.106636.

[16] F. Rehman, M. Rehman, M. Anjum, and A. Hussain, "Optimized YOLOV8: An efficient underwater litter detection using deep learning," *Ain Shams Engineering Journal*, vol. 16, no. 1, p. 103227, Jan. 2025, doi: 10.1016/J.ASEJ.2024.103227.

[17] T. Yang *et al.*, "YOLO-SGD: Precision-Oriented Intelligent Detection of Seed Germination Completion," *Agronomy*, vol. 15, no. 9, p. 2146, Sep. 2025, doi: 10.3390/agronomy15092146.

[18] O. G. Ajayi, P. O. Ibrahim, and O. S. Adegboyega, "Effect of Hyperparameter Tuning on the Performance of YOLOv8 for Multi Crop Classification on UAV Images," *Applied Sciences (Switzerland)*, vol. 14, no. 13, Jul. 2024, doi: 10.3390/app14135708.

[19] S. Lu *et al.*, "CodeXGLUE: A Machine Learning Benchmark Dataset for Code Understanding and Generation," Mar. 2021, doi: https://doi.org/10.48550/arXiv.2102.04664.

[20] H. Bai and S. C. Hui, "A crowdsourcing-based incremental learning framework for automated essays scoring," *Expert Syst Appl*, vol. 238, p. 121755, Mar. 2024, doi: 10.1016/J.ESWA.2023.121755.