Journal of Applied Informatics and Computing (JAIC)
Vol.9, No.4, August 2025, pp. 1170~1178
e-ISSN: 2548-6861

Real-Time Detection of Coffee Cherry Ripeness Using YOLOv11

Anis Ilyana [1]*, Nurdin [2], Maryana [3]***
* Departement of Informatics, Universitas Malikussaleh, Lhokseumawe, Indonesia
** Departement of Information Technology, Universitas Malikussaleh, Lhokseumawe, Indonesia
anis.210170008@mhs.unimal.ac.id. [1], nurdin@unimal.ac.id [2], maryana@unimal.ac.id [3]

Article Info

Article history:

Received 2025-06-03
Revised 2025-06-23
Accepted 2025-07-03

Keyword:

Python,
Automated Detection System,
Coffee Fruit,
YOLOv11,
Machine Learning.

ABSTRACT

This study aims to develop a real-time detection system for determining the ripeness level of coffee cherries using the YOLOv11 algorithm, in order to assist farmers in identifying the optimal harvest time. The dataset used consists of 302 images categorized into three ripeness levels: ripe, semi-ripe, and unripe. The model was trained on Google Colab using data augmentation techniques to enhance dataset variability and prevent overfitting. After 20 epochs, the model demonstrated strong performance in the ripe category (mAP50: 0.774; Precision: 0.645; Recall: 0.812) and fairly good results for semi-ripe cherries (mAP50: 0.695; Precision: 0.624; Recall: 0.679). However, detection of unripe cherries remains low (mAP50: 0.4). The system achieves an inference time of 183.4 ms per image, with fast preprocessing and postprocessing (0.5 ms each), making it suitable for real-time applications. This research contributes to the development of agricultural automation and real-time detection systems to support precision farming in Indonesia.

I. INTRODUCTION

Agriculture and fruit plantations are important economic sectors, including in Indonesia with its tropical climate. In Aceh, agriculture is the backbone of the economy, producing various commodities such as rice, corn, soybeans, vegetables, and fruits, and employing the largest workforce. Horticultural products like fruits and vegetables are key commodities, with coffee playing a significant role in this industry[1][2][3]. Coffee is a commodity with great potential in the global market, where Indonesia ranks fourth as the largest producer and exporter after Brazil, Colombia, and Vietnam.The quality and market value of coffee are greatly influenced by the ripeness of the cherries at harvest. However, manual ripeness assessment is often time-consuming and prone to errors. With advanced technology, automatic detection significantly improves the efficiency and accuracy of ripeness evaluation, leading to more optimal harvest outcomes [4].

One of the artificial intelligence methods known as machine learning uses human behavior as a model to solve problems. A proven approach for object detection is the use of the YOLO (You Only Look Once) algorithm[5]. YOLO is an AI method based on Convolutional Neural Networks (CNN) designed for real-time object detection. An image is a representation or imitation of an object or picture. It can be produced through a data acquisition system in digital form (such as files stored in storage media), analog form (such as images on a monitor or video signals), or optical form (such as photographs)[6]. It utilizes a unified, pre-trained model that simultaneously predicts multiple bounding boxes and class probabilities. This algorithm is known for its high accuracy and fast frame rate, making it highly effective for detecting objects in both images and videos[7].

Several studies have explored the use of technology in object and fruit detection. Successfully developed the YOLOv8 algorithm to detect the ripeness level of mangosteen through a web-based system, significantly improving efficiency and accuracy compared to manual methods[8]. Evaluated the performance of YOLO11, YOLOv10, YOLOv9, and YOLOv8 in detecting fruits in orchard environments, with YOLOv9 gelan-c and gelan-m showing excellent precision, while YOLO11n achieved the best performance in terms of RMSE and MAE[9].

Developed a mobile application using CNN to detect defects in robusta coffee fruits, revealing that camera distance, resolution, and smartphone memory affect detection accuracy, with an ideal distance of around 15 cm[10]. A subsequent study employed the YOLOv8 algorithm to detect spices, achieving an accuracy of 86% using specific parameters: a batch size of 10, image size of 550x550 pixels, 100 epochs, a learning rate of 0.0001, and the Adam optimizer. Furthermore, the final model using YOLOv4 demonstrated superior performance in logo recognition, with a mean Average Precision (mAP) of 93.73%, indicating that the detection system is effective and reliable for various applications[11].

This study aims to develop a coffee ripeness detection model using the You Only Look Once (YOLO) algorithm to improve the efficiency of harvesting and fruit selection in real-time. YOLO was chosen for its speed and accuracy in object detection, enabling farmers to identify ripe coffee cherries more precisely. The research also contributes to the advancement of modern and sustainable agricultural technology.

Several previous studies have utilized various versions of YOLO (YOLOv5, YOLOv7, YOLOv8) for object detection in the agricultural domain, such as detecting diseases in rice plants, assessing mangosteen ripeness, and classifying spices. However, no research has been found that applies YOLOv11 for detecting the ripeness of coffee cherries. Therefore, this study aims to fill that gap by evaluating the performance of YOLOv11, which offers high efficiency and improved accuracy, despite being relatively new.

YOLOv11 is not yet widely recognized within the research community, as the most recent official YOLO version released by Ultralytics is YOLOv8. The use of YOLOv11 in this study is exploratory and is based on a recent study by Sapkota et al.[9], which examined various YOLOv11 configurations in agricultural environments. Therefore, the implementation of YOLOv11 in this research also provides a novel contribution to the application of object detection technology in the agricultural sector.

The author intends to implement this research as a system titled “Real-time Coffee Ripeness Detection Using YOLOv11.” By leveraging YOLO's strengths in object detection, the system is expected to automatically identify the ripeness of coffee cherries, assist farmers in determining the optimal harvest time, and contribute to the application of smart technology in agriculture to enhance the productivity and quality of coffee in Indonesia.

II. METHODOLOGY

The following is an explanation of the research procedure, starting with the source of the dataset, which is then annotated according to the classification. This is followed by the image preprocessing stage. The YOLO architecture model is trained and validated, followed by data testing for ripeness detection. Finally, a performance evaluation is conducted on the implemented model architecture.

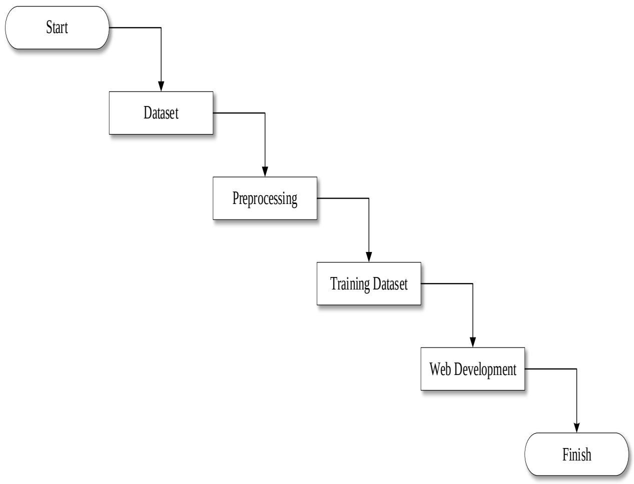

Figure 1. Research Stages

The dataset was collected directly from a local coffee plantation in Gayo Lues, Aceh, consisting of 302 images captured using a smartphone camera under various lighting conditions. Labeling was performed manually using the LabelImg application. The images were classified into three categories: ripe (red), semi-ripe (yellow), and unripe (green). The dataset was split into 70% for training, 20% for validation, and 10% for testing to prevent data leakage and enhance the reliability of the evaluation.

The class distribution was imbalanced, with fewer images of unripe cherries. Annotation format followed the YOLO standard, including class information and bounding box coordinates. The original 302 images were augmented to a total of 1,002 images to enrich data variation and improve the model's generalization capability in detecting coffee cherry ripeness levels.

In terms of architecture, YOLOv11 is a continuation of the YOLOv10 development and is not an official release from Ultralytics. According to Sapkota et al.[9], YOLOv11 features five configurations (YOLO11n, YOLO11s, YOLO11m, YOLO11l, YOLO11x), which were evaluated for fruit detection tasks under complex environmental conditions. The model demonstrated high efficiency in inference time (up to 2.4 ms per image) and competitive accuracy compared to YOLOv9 and YOLOv10. Therefore, YOLOv11 is considered suitable for real-time agricultural applications that require both speed and high efficiency.

A. YOLO Algorithm Scheme

This system diagram illustrates the workflow from image capture to displaying the predicted ripeness of coffee cherries. The image below provides a clearer visualization of each step involved in the process.

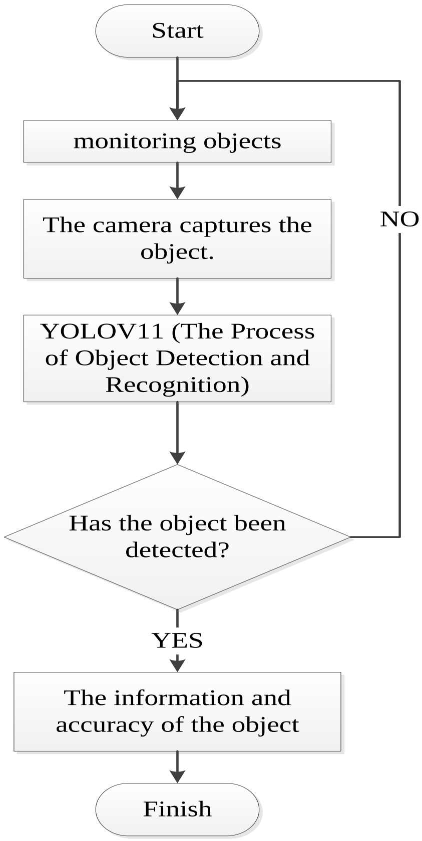

Figure 2. System Schematic

B. YOLO Algorithm

You Only Look Once is a method in the field of computer vision and object recognition used for real-time object detection in images and videos[12]. Released in 2015, YOLO's main advantage is its ability to detect objects quickly and accurately in a single processing step[13]. A key innovation of YOLO is its ability to make predictions at multiple scales by utilizing anchor boxes of varying sizes[14].

The YOLO architecture consists of 24 convolutional layers that extract features from the image, followed by two fully connected layers that predict the probability and coordinates[15]. YOLO features an integrated architecture that divides an image into a grid and directly predicts bounding boxes and class probabilities for each cell, enabling end-to-end learning. Utilizing a Convolutional Neural Network (CNN), YOLO first resizes the input image and partitions it into an S × S grid. Through the convolutional process, it generates bounding boxes containing information such as coordinates (x, y), width, height, and a confidence score. The confidence score is normalized within a range of 0 to 1, while the coordinates are adjusted relative to the top-left corner of the image, and the box dimensions are scaled according to the image size. Once the bounding boxes and confidence scores are obtained, further processing is conducted using CNN for each detected box[4]. An illustration of YOLO is presented as follows.

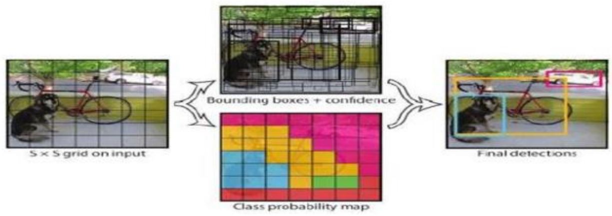

Figure 3. YOLO illustration

YOLOv11 is the latest version of the YOLO algorithm, released in 2024. This model introduces several improvements over its predecessors, including multi-scale prediction, a new anchor system, and a more efficient backbone network. The evolution of YOLO is illustrated in the following figure.

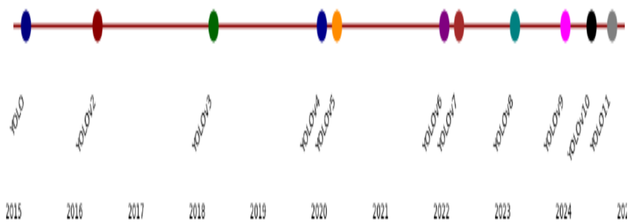

Figure 4. YOLO Evolution and model Performance from 2015-2024

Here is a summary of the development history of YOLOv1 to YOLOv11:

1. YOLOv1 (2015): The first single-stage object detection model that processes the entire image at once[16].
2. YOLOv2 (2016): Introduced anchor boxes and upsampling techniques to improve detection accuracy[16].
3. YOLOv3 (2018): Added a more efficient backbone, multiple anchors, and multi-scale feature extraction[17].
4. YOLOv4 (2020): Improved training with Mosaic data augmentation and enhanced detection accuracy[18].
5. YOLOv5 (2020): Developed by Glenn Jocher of Ultralytics as a flexible, Python-based framework[19].
6. YOLOv6 (2022): A lightweight, open-source model designed for edge devices[16].
7. YOLOv7 (2022): Further optimized for better accuracy and efficiency[20].
8. YOLOv8 (2023): Updated architecture from YOLOv7 with three main components: backbone, neck, and head. It introduces additional optimizations and new modules to enhance accuracy and efficiency[21].
9. YOLOv9 (2024): Introduced PGI and GELAN techniques to maintain deep feature performance[22].

10. YOLOv10 (2024): Developed by Tsinghua University, focusing on high efficiency using large kernels and partial self attention[9].
11. YOLOv11 (2024): The latest version, offering advanced innovations and optimizations for more complex computer vision task[9].

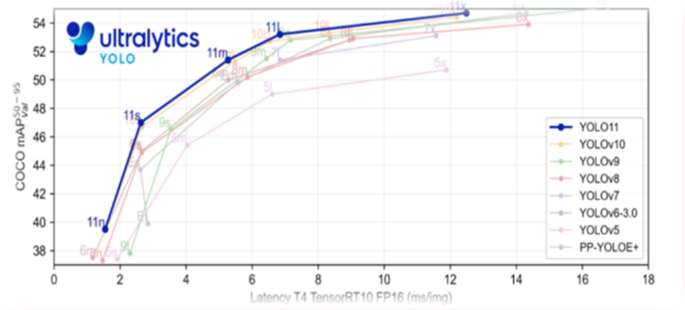

Figure 5. Performance results on benchmark datasets on YOLO

YOLOv11, the latest iteration in the Ultralytics YOLO series released in September 2024, integrates cutting-edge accuracy, speed, and efficiency for applications such as object detection, segmentation, classification, oriented bounding boxes, and pose estimation. Compared to its predecessors, YOLOv11 uses fewer parameters while delivering superior results, marking a significant advancement in the field of computer vision.

YOLOv11 achieves a mean Average Precision (mAP) score of 95.0% on the COCO dataset while utilizing 22% fewer parameters than YOLOv8m, demonstrating greater efficiency without compromising accuracy. With an average inference speed that is 2% faster than YOLOv10, YOLOv11 is optimized for real-time applications, ensuring rapid processing even in demanding environments.

These specifications position YOLOv11 as a powerful tool for advancing AI applications, particularly in sectors that require fast and accurate image analysis.

The performance evaluation of YOLOv11 aims to assess how well the model detects and classifies objects within images. Five key metrics are used to measure the system's performance: Precision, Recall, F1 Score, mean Average Precision (mAP), and Accuracy.

a. Recall is a metric used to evaluate the system's ability to retrieve relevant information from a given dataset. It measures how well the system can detect all relevant instances.

$$Recall = \frac{TP}{TP+FN} \quad (1)$$

b. Precision is a metric used to evaluate the accuracy of the system in providing relevant responses to user queries. It measures how many of the results returned by the system are truly relevant.

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

c. The F1 Score is a metric that combines precision and recall into a single value, calculated as the harmonic mean of the two. It provides an overall indication of the balance between precision and recall.

$$F1-Score = 2 . \frac{Precision .Recall}{Precision+Recall} \quad (3)$$

d. Accuracy is a metric used to evaluate how often the system makes correct predictions overall, both positive and negative, compared to the actual values.

$$Accuracy = \frac{TP+TN}{T+TN+FP+FN} \quad (4)$$

e. Mean Average Precision (mAP) is a commonly used metric to measure the accuracy of object detection predicted by models like YOLOv8. The Average Precision (AP) value ranges from 0 to 1, and mAP is calculated as the average of the AP values for each detected class or object. Here, represents the total number of AP values.

$$\text{mAP} = \frac{1}{N}\sum_{i=0}^{n} Api \quad (5)$$

TABLE I.
CONFUSION MATRIX

Confusion Matrix		**Actual**	
		TRUE	**FALSE**
Predicted	TRUE	TP (True Positif)	FP(False Positif)
	FALSE	FN (False Negatif)	TN (True Negatif)

This table provides a framework for evaluating the performance of a detection model using metrics such as precision, recall, F1 score, and accuracy, based on the combination of actual and predicted values.

III. RESULT AND DISCUSSION

Real-time detection of coffee fruit ripeness is a complex challenge influenced by various factors, such as lighting conditions, the type of camera used, and the natural variations in the appearance of the coffee cherries. This detection process requires a system capable of recognizing differences in ripeness with a high level of accuracy, so that the results can be reliably used to determine the optimal harvest time. Conventional approaches that rely on manual observation by farmers are often less effective, time-consuming, and prone to human error, which can reduce efficiency in the coffee industry.

The application of the YOLOv11 algorithm as a solution to this problem is highly relevant due to its ability to detect objects in real-time with high accuracy. YOLOv11 can automatically classify the ripeness levels of coffee cherries, reducing the reliance on manual processes and providing faster and more accurate results. In this study, YOLOv11 is used to detect and classify the ripeness of coffee fruits based on images captured by a camera, with the goal of improving efficiency and optimizing harvest timing for farmers.

A. Data Description

In this study, the data used to train the YOLOv11 model for detecting coffee fruit ripeness were obtained from images captured from various coffee trees in the field. The dataset consists of 302 images depicting coffee cherries at different stages of ripeness: green, yellow, and red. Each image was taken under varying lighting conditions and with diverse backgrounds, allowing the model to learn how to recognize coffee fruit ripeness in a wide range of real-world scenarios.

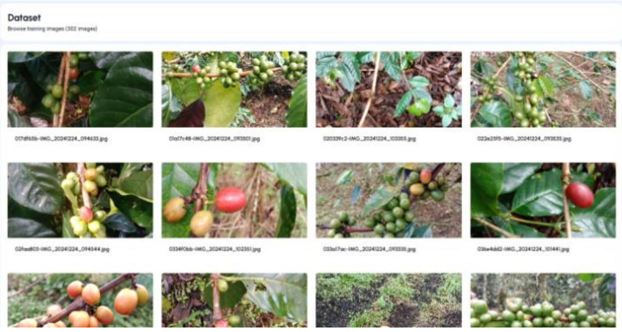

Figure 6. Research Dataset

The images include close-up shots of coffee cherries, highlighting differences in color and texture at each stage of ripeness. Some images show nearly ripe cherries (yellow), while others depict fully ripe cherries (red). This dataset is expected to provide sufficient variation to train the YOLOv11 model to accurately detect and identify the ripeness of coffee cherries in the field. The use of a dataset rich in variation is intended to enhance the model's generalization capability for real-time applications.

B. Algorithm Implementation

In this section, the researcher discusses the implementation of a real-time coffee fruit ripeness detection system using the previously developed YOLOv11 algorithm. This system allows users, such as farmers or field workers, to upload images of coffee cherries or capture images using a camera for analysis. Once the image is taken, the system processes it with the YOLOv11 model to detect coffee fruit objects within the image. The detection results are clearly displayed with bounding boxes around the detected cherries, along with information about the fruit's ripeness level whether unripe, half-ripe, or fully ripe.

The user interface is designed to be simple and easy to understand, with clearly visible detection results. Information about the ripeness level is also shown as text on the screen, making it easy for users to assess the condition of the fruit without manual inspection. This system implementation aims to facilitate more efficient monitoring of coffee fruit ripeness and assist farmers or workers in determining the optimal time to harvest based on accurate detection results.

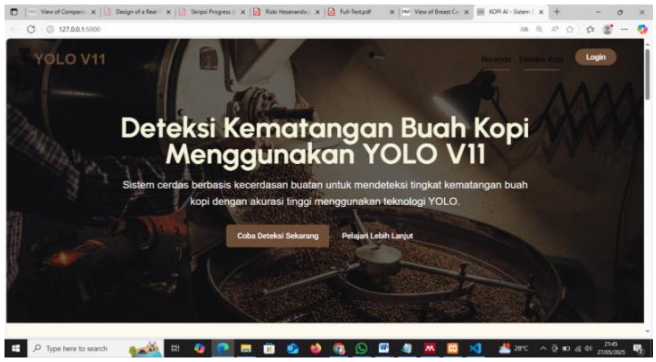

Figure 7. Homepage Display

The main page of the YOLOv11 coffee fruit ripeness detection system is designed to be simple and informative, featuring three key steps: upload image, AI processing, and view results. Users can upload coffee images to be analyzed quickly and accurately by YOLOv11. The page also includes interactive features such as FAQs and a button to view detection results with indicators for ripeness levels (ripe, half-ripe, or unripe). Additionally, there are links providing information about the system's advantages, such as over 95% accuracy and processing times under 3 seconds. This design ensures ease of use for users without requiring technical expertise.

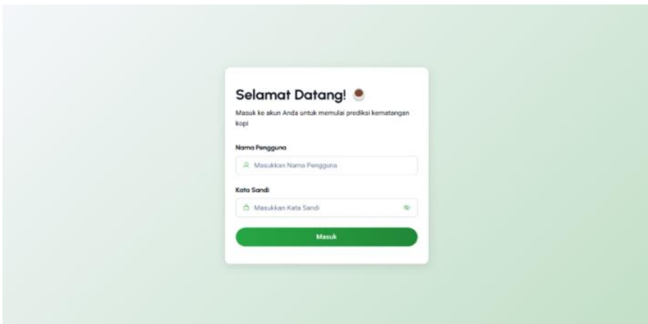

Figure 8. Login Page Display

The login page of the YOLOv11 coffee bean ripeness detection system is designed to be simple and user-friendly, featuring two input fields for Username and Password. After entering the correct information, users can click the green Login button to access the system and begin the detection process. The page design prioritizes comfort, ease of access, and data security, ensuring that only registered users can log in.

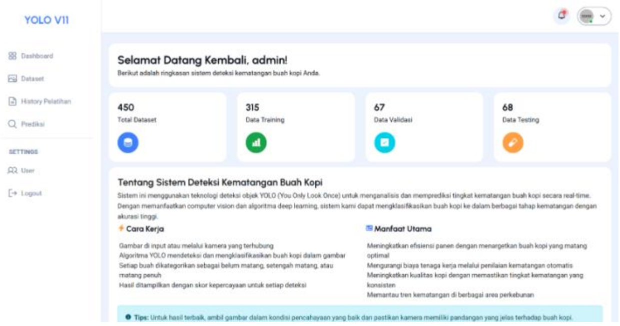

Figure 9. Dashboard Page View

The dashboard page of the YOLOv11 coffee bean ripeness detection system provides a concise and clear overview of the system's performance and status. It features a welcome message and important information about the dataset, including the total number of images used for training and testing the model. Below that, there is a brief explanation of how the system works and its main benefits, highlighting its ability to classify coffee bean ripeness quickly and accurately. The dashboard also offers tips to improve detection accuracy, such as ensuring good lighting and a clear camera view when capturing images. Overall, this page helps users monitor the system's real-time performance and enhance detection results.

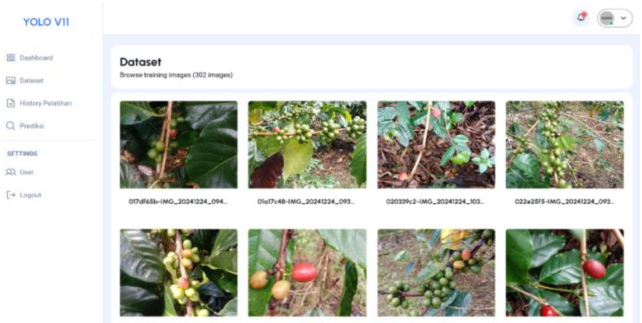

Figure 10. Dataset Page View

The Dataset page of the YOLOv11 coffee bean ripeness detection system allows users to view and manage the 302 images used for training. These images represent various stages of coffee bean ripeness and are neatly labeled for easy identification. The grid layout makes it simple for users to browse the dataset, ensuring the quality and diversity of the data meet the standards required for accurate ripeness detection. This feature helps guarantee that the training data adequately represents each ripeness category.

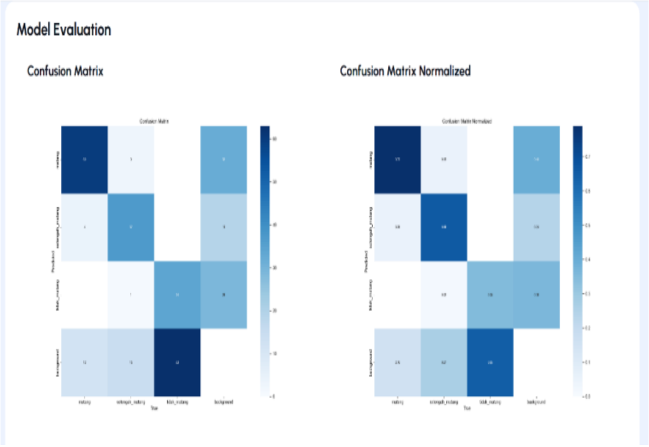

Figure 11. Confusion Matrix

Next, the Model Evaluation section presents the Confusion Matrix, which illustrates how well the model can classify objects in the images according to the predefined categories, such as ripe coffee beans, half-ripe, unripe, or even the background.

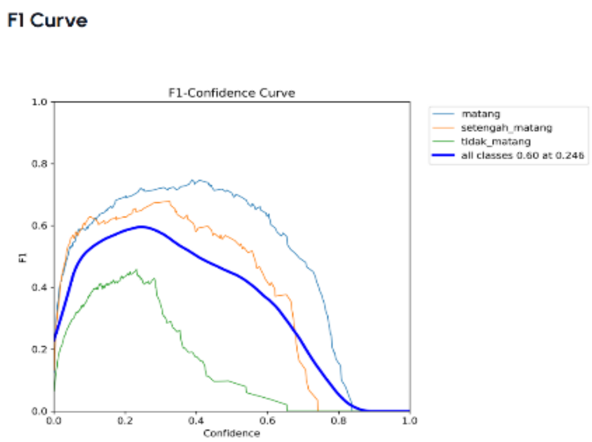

Figure 12. F1 Curve

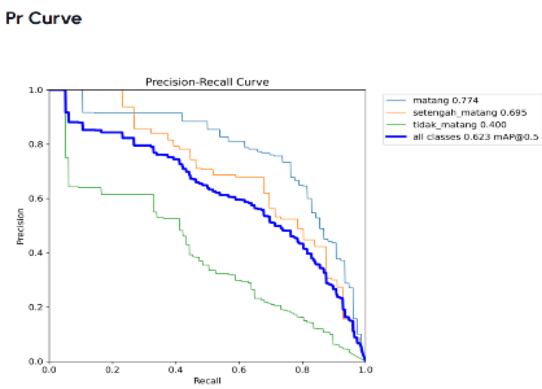

Figure 13. Pr Curve

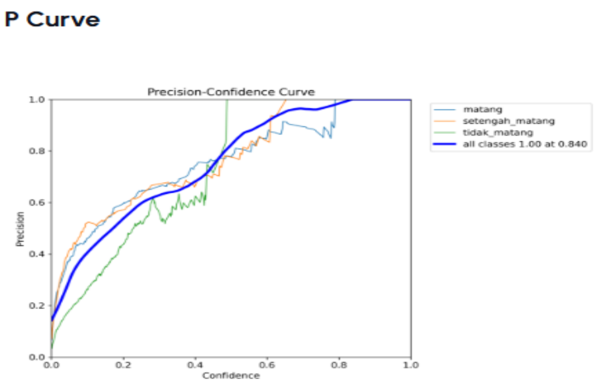

Figure 14. Precision Curve

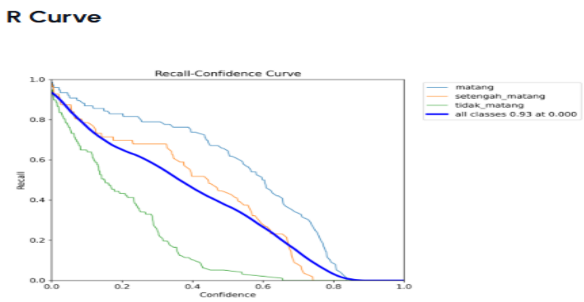

Figure 15. Recall Curve

Additionally, this page displays the Precision-Recall (PR) Curve, F1 Curve, Precision Curve, and Recall Curve, providing further insights into the model's performance in terms of precision, recall, and the balance between them. These curves illustrate how the model performs at different confidence levels and how its accuracy changes as the detection threshold varies. The F1 Curve, which represents the balance between precision and recall, is especially important for evaluating the overall performance of the model. The higher the F1 score, the better the model is at detecting objects with minimal errors.

C. *YOLOv11 Model Validation Process*

```
Validating runs\detect\train4\weights\best.pt...
Ultralytics 8.3.105  Python-3.10.8 torch-2.5.1+cu124 CPU (Intel Core(TM) i3-1005G1 1.20GHz)
YOLO11m summary (fused): 125 layers, 20,032,345 parameters, 0 gradients, 67.7 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 4/4 [00:21<00:00,  5.37s/it]
                   all         64        324      0.581      0.686       0.65      0.332
                matang         40         78       0.69      0.782      0.767      0.435
       setengah_matang         30         56      0.625       0.75      0.762      0.379
           tidak_matang         45        190      0.427      0.527      0.422      0.181
Speed: 0.5ms preprocess, 165.6ms inference, 0.0ms loss, 0.5ms postprocess per image
Results saved to runs\detect\train4
ultralytics.utils.metrics.DetMetrics object with attributes:
```

Figure 16. YOLOv11 Model Validation Process

Figure 16 illustrates the results of the validation process for the YOLOv11 model, which was trained using a dataset containing images of coffee bean ripeness. The model was tested on a validation dataset consisting of 64 images, covering three ripeness categories: ripe, semi-ripe, and unripe. The performance of the model was evaluated using several metrics, including Precision (P), Recall (R), mAP50, and mAP50-95.

The results indicate that the model performs reasonably well in detecting objects across all ripeness categories. Overall, the mAP50 value for all classes is 0.65, which reflects the model's object detection accuracy at an IoU threshold of 50%.

TABLE II.
YOLOV11 MODEL VALIDATION PROCESS RESULTS

Class	Images	Instances	*Precision* (P)	*Recall*(R)	mAP50	mAP50-95
all	64	324	0.581	0.686	0.65	0.332
ripe	40	78	0.69	0.782	0.767	0.435
half ripe	30	56	0.625	0.75	0.762	0.379
Un ripe	45	190	0.427	0.527	0.422	0.181

Additionally, the inference time per image is approximately 165.6 milliseconds, with exceptionally fast preprocessing and postprocessing times of just 0.5 milliseconds each. This demonstrates that the model is highly efficient and well-suited for real-time detection applications.

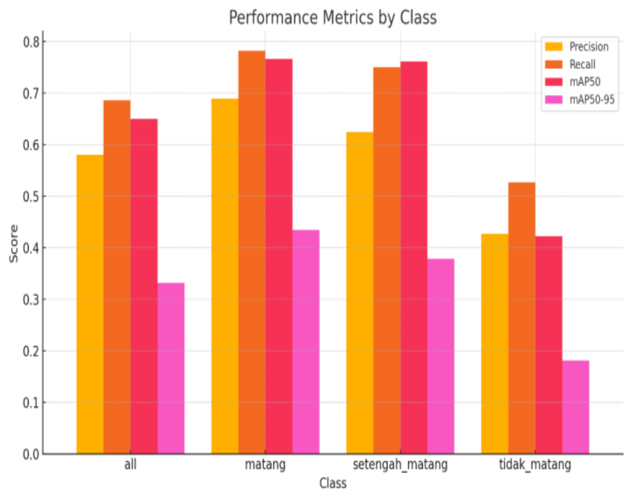

Figure 17. YOLOv11 Model Validation Process Results Graph

The chart compares the performance of the model in detecting four classes: "all," "ripe," "half-ripe," and "unripe" based on key evaluation metrics—Precision, Recall, mAP50, and mAP50-95. The "ripe" class shows the best performance across nearly all metrics, with a high mAP50 of 0.767, indicating strong object detection accuracy. The "half-ripe" class also performs reasonably well, though with slightly lower mAP scores. In contrast, the "unripe" class demonstrates the weakest performance, with the lowest values in all metrics, suggesting that the model struggles to accurately detect unripe coffee cherries. Overall, the chart highlights the model's effectiveness in identifying ripe cherries while also pointing out the need for improvement in detecting unripe ones.

D. Final Evaluation Process of YOLOv11 Model

Figure 18 presents the evaluation results of the YOLOv11 model after the training phase, using a validation dataset consisting of 64 images containing three categories of coffee fruit ripeness: ripe, semi-ripe, and unripe. The evaluation process was conducted using the model.val() command within the YOLOv11 framework, which generated key evaluation metrics used to assess the model's performance in detecting coffee fruit ripeness. The metrics include Precision, Recall, mAP50, and mAP50-95, providing a comprehensive overview of the model's object classification accuracy.

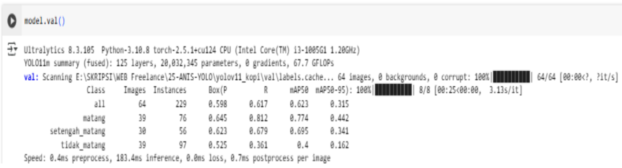

Figure 18.Model Validation Results After Training

The evaluation results indicate that the model performs well in certain ripeness categories. For the ripe category, the model achieved an mAP50 score of 0.774, demonstrating a high level of accuracy in detecting ripe coffee fruits. Precision and Recall for this category also reflect strong performance, recorded at 0.645 and 0.812 respectively. This suggests that the model can accurately identify ripe objects with minimal errors while successfully detecting the majority of ripe instances present in the dataset.

TABLE III.
MODEL EVALUATION RESULTS ON VALIDATION DATASET

Class	Images	Instances	*Precision* (P)	*Recall* (R)	mAP50	mAP50-95
All	64	229	0.598	0.617	0.623	0.315
Ripe	39	76	0.645	0.812	0.774	0.442
Half Ripe	39	56	0.624	0.679	0.695	0.341
Un Ripe	39	97	0.525	0.361	0.4	0.162

Nevertheless, the model demonstrates good efficiency in terms of inference time, with a processing duration of approximately 183.4 milliseconds per image. Both preprocessing and postprocessing times are also notably fast, each requiring only 0.5 milliseconds, indicating that the model is sufficiently efficient for use in real-time detection applications. These evaluation results provide a clear overview of the model's strengths and limitations in detecting the ripeness of coffee fruits in images.

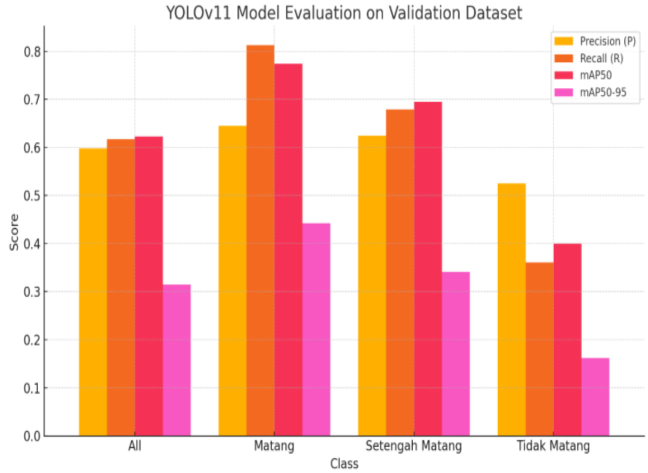

Figure 19. Model Evaluation Results Graph on Validation Dataset

This chart presents the evaluation results of the YOLOv11 model on four object classes: "All", "Ripe", "Half-Ripe", and "Unripe". The "Ripe" class demonstrates the best performance across all evaluation metrics, with a Precision of 0.645, Recall of 0.812, mAP50 of 0.774, and mAP50-95 of 0.442, indicating accurate and consistent object detection. The "Half-Ripe" class shows moderately good results, although slightly lower than "Ripe". On the other hand, the "Unripe" class has the lowest performance, with a mAP50 of only 0.4 and mAP50-95 of 0.162, suggesting that the model struggles to detect unripe objects accurately. Overall, the model performs better in detecting ripe and half-ripe coffee cherries, with room for improvement in identifying unripe ones.

The detection performance for unripe cherries (mAP50: 0.4; mAP50-95: 0.162) remains low compared to other classes. This is likely due to the low color contrast between the green cherries and the leafy background, as well as the imbalanced data distribution. Therefore, a more balanced dataset is needed for future research.

This study does not include comparisons with other algorithms such as SSD, Faster R-CNN, or previous YOLO versions like YOLOv5 and YOLOv8, making these results a starting point for further evaluation. Additionally, an ablation study has not yet been conducted to assess the impact of augmentation techniques and training configurations on the model's performance.

IV. CONCLUSION

Based on the research conducted, the coffee fruit ripeness detection system developed using the YOLOv11 algorithm has demonstrated good performance in identifying the ripeness levels of coffee cherries. While the detection results for ripe and half-ripe coffee cherries are fairly accurate, challenges remain in achieving the desired accuracy for detecting unripe cherries. The model's training and evaluation processes showed significant progress with each iteration; however, there is still room for improvement, particularly in enhancing the detection of unripe categories. On the other hand, the efficient inference time ensures that the system can be used in real-time, providing practical benefits for farmers in determining the optimal harvest time more accurately. The model demonstrated a relatively fast inference time of 183.4 ms per image, with preprocessing and postprocessing times of only 0.5 ms each, indicating good efficiency. Overall, while the model has shown reasonably good performance in detecting coffee fruit ripeness particularly in the ripe and half-ripe categories there is still room for improvement, especially in enhancing detection accuracy for the "unripe" category. Although the model shows promising results, particularly in the ripe and semi-ripe categories, its performance in detecting unripe cherries needs improvement. Furthermore, since the model has only been tested on Google Colab, the real-time implementation remains at an early-stage validation. Future plans include expanding the dataset to over 1,000 images and deploying the model on edge devices such as the Jetson Nano and Raspberry Pi to support field applications. This study serves as an initial step in applying YOLOv11 to smart agriculture systems and opens up opportunities for developing automated fruit ripeness classification systems for various other commodities.

REFERENCES

[1] M. S. Hawibowo and I. Muhimmmah, "Aplikasi Pendeteksi Tingkat Kematangan Pepaya menggunakan Metode Convolutional Neural Network (CNN) Berbasis Android," *J. Edukasi dan Penelit. Inform.*, vol. 10, no. 1, p. 162, 2024, doi: 10.26418/jp.v10i1.77819.

[2] Nurdin, Bustami, and Maryana, "Robust optimization approach for agricultural commodity supply chain planningg," *J. Theor. Appl. Inf. Technol.*, vol. 99, no. 2, pp. 304–315, 2021.

[3] J. Rusman and N. Pasae, "Prototype Sistem Penyortir Buah Kopi Arabika Berdasarkan Tingkat Kematangan Menggunakan Metode Support Vector Machine," *Teknika*, vol. 12, no. 1, pp. 65–72, 2023, doi: 10.34148/teknika.v12i1.602.

[4] S. R. Raysyah, Veri Arinal, and Dadang Iskandar Mulyana, "Klasifikasi Tingkat Kematangan Buah Kopi Berdasarkan Deteksi Warna Menggunakan Metode Knn Dan Pca," *JSiI (Jurnal Sist. Informasi)*, vol. 8, no. 2, pp. 88–95, 2021, doi: 10.30656/jsii.v8i2.3638.

[5] C. R. Gunawan, N. Nurdin, and F. Fajriana, "Deteksi Ikan Segar Secara Realtime dengan YOLOv4 menggunakan Metode Convolutional Neural Network," *J. Komtika (Komputasi dan Inform.*, vol. 7, no. 1, pp. 1–11, 2023, doi: 10.31603/komtika.v7i1.8986.

[6] J. Ulfah and N. Nurdin, "Implementasi Metode Deteksi Tepi Canny Untuk Menghitung Jumlah Uang Koin Dalam Gambar Menggunakan Opencv," *J. Inform. dan Tek. Elektro Terap.*, vol. 11, no. 3, pp. 420–426, 2023, doi: 10.23960/jitet.v11i3.3147.

[7] K. Azman, M. Arhami, and A. Azhar, "Metode You Only Look Once (YOLO) dalam Deteksi Physical Distancing dan Wajah Bermasker," *Pros. Semin. Nas. Politek. Negeri Lhokseumawe*, vol. 6, no. 1, pp. 107–113, 2022.

[8] A. Rifqi Akyas hifdzi Rahman, Asril Adi Sunarto, "Penerapan (You Only Look Once) V8 Untuk Deteksi Tingkat Kematangan Buah Manggis," vol. 8, no. 5, pp. 10566–10571, 2024.

[9] R. Sapkota, Z. Meng, M. Churuvija, X. Du, Z. Ma, and M. Karkee, "Comprehensive Performance Evaluation of YOLO11, YOLOv10, YOLOv9 and YOLOv8 on Detecting and Counting Fruitlet in Complex Orchard Environments," 2024.

[10] P. Hanifah, H. I. Antoni, S. R. Ramadhani, and Yuliska, "Pengembangan Aplikasi Mobile untuk Deteksi Cacat Biji Kopi Robusta Berdasarkan Standar Nasional Indonesia," pp. 17–26, 2018.

[11] M. F. Golfantara, "Penggunaan Algoritma YOLO V8 Untuk Identifikasi Rempah-Rempah," vol. 12, no. 3, pp. 3867–3873, 2024.

[12] L. Rahma, H. Syaputra, A. H. Mirza, and S. D. Purnamasari, "Objek Deteksi Makanan Khas Palembang Menggunakan Algoritma YOLO (You Only Look Once)," *J. Nas. Ilmu Komput.*, vol. 2, no. 3, pp. 213–232, 2021, doi: 10.47747/jurnalnik.v2i3.534.

[13] N. Khairunisa, . C., and A. Jamaludin, "Analisis Perbandingan Algoritma Cnn Dan Yolo Dalam Mengidentifikasi Kerusakan Jalan," *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 3, 2024, doi: 10.23960/jitet.v12i3.4434.

[14] R. Hesananda, D. Natasya, and N. Wiliani, "Cloth Bag Object Detection Using the Yolo Algorithm (You Only See Once) V5," *J. Pilar Nusa Mandiri*, vol. 18, no. 2, pp. 217–222, 2023, doi: 10.33480/pilar.v18i2.3019.

[15] C. R. Gunawan, N. Nurdin, and F. Fajriana, "Design of A Real-Time Object Detection Prototype System with YOLOv3 (You Only Look Once)," *Int. J. Eng. Sci. Inf. Technol.*, vol. 2, no. 3, pp. 96–99, 2022, doi: 10.52088/ijesty.v2i3.309.

[16] Y. Yanto, F. Aziz, and I. Irmawati, "Yolo-V8 Peningkatan Algoritma Untuk Deteksi Pemakaian Masker Wajah," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 7, no. 3, pp. 1437–1444, 2023, doi: 10.36040/jati.v7i3.7047.

[17] Z. Hong *et al.*, "Multi-Scale Ship Detection from SAR and Optical Imagery Via A More Accurate YOLOv3," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 14, pp. 6083–6101, 2021, doi: 10.1109/JSTARS.2021.3087555.

[18] S. Du, P. Zhang, B. Zhang, and H. Xu, "Weak and Occluded Vehicle Detection in Complex Infrared Environment Based on Improved YOLOv4," *IEEE Access*, vol. 9, pp. 25671–25680, 2021, doi: 10.1109/ACCESS.2021.3057723.

[19] A. Putra Pranjaya, F. Rizki, R. Kurniawan, and N. K. Daulay, "Penyakit Pada Daun Tanaman Padi Berbasis YoloV5 (You Only Look Once)," *Media Online)*, vol. 4, no. 6, pp. 3127–3136, 2024, doi: 10.30865/klik.v4i6.1916.

[20] Y. Zhou *et al.*, "Adaptive Active Positioning of Camellia oleifera Fruit Picking Points: Classical Image Processing and YOLOv7 Fusion Algorithm," *Appl. Sci.*, vol. 12, no. 24, 2022, doi: 10.3390/app122412959.

[21] H. Wang, X. Xu, Y. Liu, D. Lu, B. Liang, and Y. Tang, "Real-Time Defect Detection for Metal Components: A Fusion of Enhanced Canny–Devernay and YOLOv6 Algorithms," *Appl. Sci.*, vol. 13, no. 12, 2023, doi: 10.3390/app13126898.

[22] H. T. Vo, K. C. Mui, N. N. Thien, and P. P. Tien, "Automating Tomato Ripeness Classification and Counting with YOLOv9," *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 4, pp. 1120–1128, 2024, doi: 10.14569/IJACSA.2024.01504113.