Nucleus Journal
Jurnal Sains dan Teknologi
Website: https://ejournal.undar.or.id/index.php/Nucleus

e-ISSN: 2964-4100
p-ISSN: 2964-4097

Mango Leaf Detection: Comparison of YOLOv12n and YOLOv26n for Mangifera indica Disease

[1*] Setiyo Hari Atmojo, [2] Lilik Anifah
[1,2] Department of Electrical Engineering, Faculty of Engineering, Universitas Negeri Surabaya, Surabaya
[1] setiyo.23016@mhs.unesa.ac.id, [2] lilikanifah@unesa.ac.id

Article Info

Article history:

Received: 3 March 2026
Revised: 1 April 2026
Accepted: 27 April 2026

Keyword:

Mango Leaf Disease
Object Detection
YOLOv12
YOLOv26

ABSTRACT

Many deep learning-based automated detection systems for mango (Mangifera indica) leaf diseases have been developed. However, most previous studies have focused on older models and have not specifically evaluated the performance of the latest models in multi-class scenarios. Therefore, this study aims to fill this gap by comparatively evaluating two current lightweight YOLO models: YOLOv12n and YOLOv26n. This study developed an automated detection system to identify five types of mango leaf diseases: anthracnose, powdery mildew, cutting weevil, dieback, and sooty mold. Using a dataset of 1,970 images divided into training, validation, and testing data with proportions of 70%, 20%, and 10%, respectively. Both models were trained for 100 epochs and evaluated using accuracy, recall, precision, and F1 score. The results of the testing show that YOLOv26n shows higher performance with an accuracy of 97.06%, a precision of 98.25%, a recall of 98.76%, and an F1 score of 98.50%, as well as more consistent performance when compared to YOLOv12n.

DOI: https://doi.org/10.32492/nucleus.v5i1.5103

Corresponding Author:
[1] Setiyo Hari Atmojo, [2] Lilik Anifah
[1] Department of Electrical Engineering, Faculty of Engineering, Universitas Negeri Surabaya, Surabaya
Jl. Ketintang, Unesa, Gayungan District, Surabaya, East Java 60231.
[1] setiyo.23016@mhs.unesa.ac.id, [2] lilikanifah@unesa.ac.id

I. Introduction

Mango (Mangifera indica L.) is native to India and is classified in the Anacardiaceae family [1]. As a major export product, the mango (Mangifera indica L.) is widely cultivated in Indonesia. The Central Statistics Agency (BPS) recorded an increase in mango (Mangifera indica L.) production in Indonesia, from 2,835,442 tons in 2021 to 3,308,895 tons in 2022. However, in 2023, production decreased slightly to 3,302,620 tons [2]. Decreased productivity of mango (Mangifera indica L.) plants is one of the problems related to disease [1]. In some cases, the disease cannot be clearly diagnosed, and the response to the surrounding environment is delayed. However, certain diseases have clear signs based on their symptoms [3].

For example, leaf diseases such as anthracnose are often only identified when symptoms are already severe. This reduces the effectiveness of control and increases costs [4]. Using traditional methods to detect plant diseases is quite complex and prone to unavoidable errors [5].

Given these problems, this can be technically addressed by implementing disease detectors to assist farmers [6]. Therefore, a system for the automatic inspection and classification of disease-infected leaves

(Mangifera indica L.) using a Deep Learning (DL) model is proposed [7]. This intelligent system is designed to automatically detect and identify leaf diseases (Mangifera indica L.) using the YOLOv12n and YOLOv26n deep learning-based training models.

Many researchers have previously researched mango leaf disease detection. One of these is detection using YOLOv7. The results showed a high identification rate, reaching 92.1% [8]. Subsequent research using YOLOv8 showed a high success rate, reaching 99.29% [9]. However, the YOLOv7 model still has limitations when detecting small objects, while the YOLOv8 model is less than optimal when recognizing disease symptoms in the early stages that have less clear visual characteristics.

Furthermore, previous research has focused on a single YOLO model, without comparing it with other YOLO models, particularly with newer YOLO models. This has limited the ability to evaluate the performance improvements offered by the latest YOLO model, particularly within the multi-class mango leaf disease framework. Therefore, there is still a research gap in comparative analysis between the latest generation YOLO models to achieve better detection performance.

Based on this issue, this study aims to develop an automatic mango leaf disease detection system using the YOLOv12n and YOLOv26n models. The novelty of this study lies in the use of two latest-generation YOLO models, which are then directly compared on a multi-class dataset and a comprehensive performance evaluation using accuracy, precision, recall, and F1-score. In addition, this study also analyzes the model performance in classes that are difficult to detect, such as the cutting weevil, which has inconsistent shape characteristics, and anthracnose and sooty mold, which have visual similarities.

This study aims to analyze and compare the performance of YOLOv12n and YOLOv26n in detecting and developing a leaf disease detection system (Mangifera indica) using YOLOv12 and YOLOv26. This is expected to provide the best model for detecting leaf diseases (Mangifera indica) in real-time and offline by farmers.

II. Research Method

The stages in this research can be seen in Figure 1.

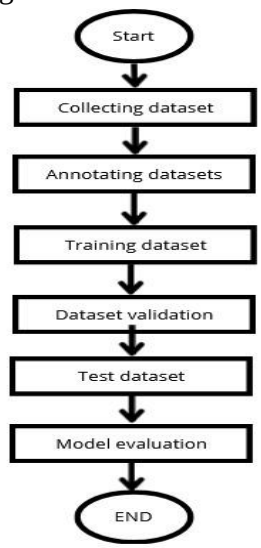


Figure 1. Research Method

The methods used are YOLOv12 and YOLOv26, with implementation using Ultralytics. YOLOv12 is an object detection model that can match the speed of previous CNN-based models. YOLOv12 outperforms all popular real-time object detectors in terms of accuracy with competitive speed [10]. Then YOLOv26 is the latest and most advanced generation of the YOLO family, specifically designed to provide accuracy, efficiency, and readiness for implementation on edge and low-power devices [11]. In addition, developments in YOLOv26 are aimed at improving the efficiency and stability of the model training process. And increased inference speed is expected to provide more optimal detection results [12].

A. Collecting Dataset

The data used in this study were obtained from datasets on the Kaggle platform. The number of datasets used was 1970 [13]. The number of each image was 394 images with an image resolution of 240×320 pixels. The types of leaf diseases (Mangifera indica) in this study were anthracnose, cutting weevil, dieback, powdery mildew, and sooty mold. Powdery mildew disease is characterized by the presence of white, powdery fungal growth on the leaf surface (Figure 2A) [14]. Anthracnose disease (Figure 2B) is characterized by brown to black lesions with unclear boundaries on the infected surface [15]. Then, for the cutting weevil, the leaves look like they have been neatly cut with scissors (Figure 2C) [16]. Sooty mold disease forms a black layer on the leaf surface (Figure 2D) [17]. Dieback disease was identified from the wilting that started from the tip of the branch [18] (Figure 2E).

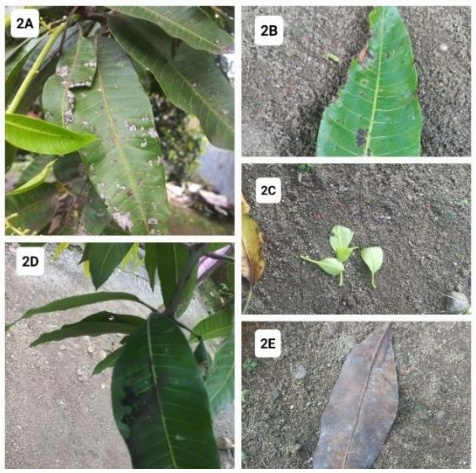


Figure 2. Classification of leaf diseases (Mangifera indica) (2A) Powdery mildew, (2B) Anthracnose, (2C) Cutting weevil, (2D) Sooty mold, (2E) Dieback

B. Annotating Dataset

Before generating the YOLO model, dataset annotation was performed. The dataset annotation process is a process to identify or provide markers to objects to be identified [19]. The dataset annotation process in this study was carried out using the Roboflow platform by one annotator with an annotation model using bounding boxes and polygons. However, for the needs of object detection model training, annotations were converted to YOLO format in the form of bounding boxes with normalized coordinates during the dataset export process from Roboflow. Then the preprocessing process used Auto-Orient, and the augmentation process used was Horizontal, Vertical Flip, 90° Rotation, Random Rotation in the range of −15° to +15°, Shear of ±10% in the horizontal and vertical directions, and no resizing was performed. Then, the dataset of 1970 images was divided into three categories, namely 70% for training, 20% for validation, and 10% for testing. This process can be seen in the image below. The annotation process can be seen in Figure 3.

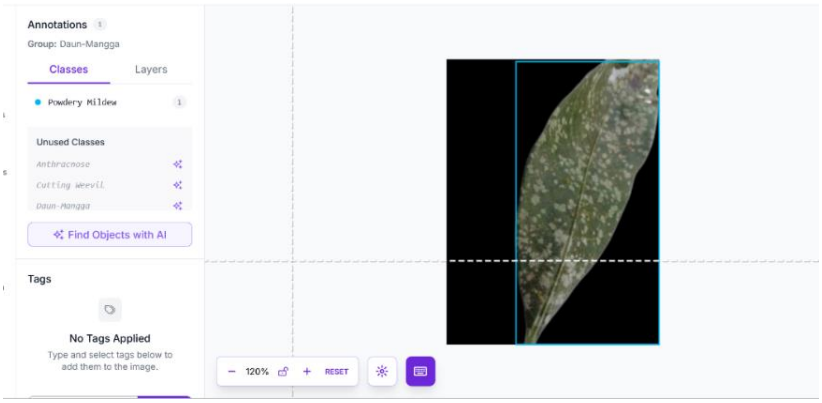


Figure 3. Annotation dataset

C. *Training Dataset*

Then, for the annotated dataset, a deep learning-based dataset training process was carried out to train the model used. The pre-trained models used in this study were YOLOv12 and YOLOv26. The annotation results from Roboflow were then trained for the YOLOv12n and Yolov26n models. The nano (n) variant was used because it is suitable for implementation on low-resource devices. The training process used the Google Colaboratory platform with a T4 GPU, with parameters for the number of epochs of 100, a batch size of 8, and an image size of 320 × 320 pixels. The optimizer, scheduler, learning rate, and random seed configurations used the default settings from Ultralytics YOLO. The use of this default configuration aims to ensure training stability and ease of reproducibility. In this process, 70% of the total dataset was used. The training process can be seen in Figure 4.

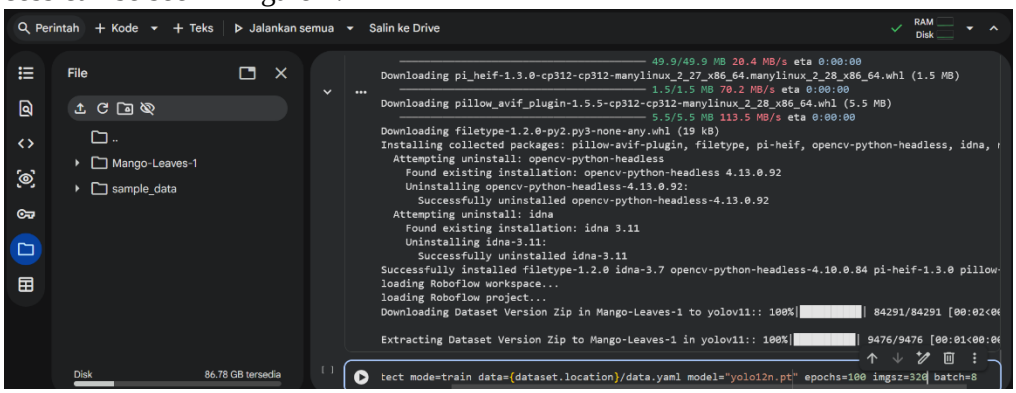


Figure 4. Training dataset

D. *Validation Dataset*

After the training process, the next step is model validation. This stage uses 20% of the total dataset, and no parameter changes are made. This stage is used to monitor model performance during the training process.

E. *Test Dataset*

After the training and validation process, the deep learning model is tested using a test dataset, a set of data used to assess the model's performance. The dataset used for testing represents 10% of the total dataset, which is 197 images. These images are then used to evaluate the model.

F. *Evaluation*

Then, model evaluation was conducted using two approaches: object detection-based for validation and classification-based for testing. In the validation phase, the evaluation used standard object detection methods, using precision, recall, and mean average precision (mAP) values at an IoU threshold of 0.5 and a range of 0.5-0.95. In addition to these values, a confusion matrix was also used as an additional analysis to evaluate and identify patterns of misclassification between classes. These values were obtained from the final results of the YOLO model on Google Collaboratory. This stage was used to measure the model's ability to accurately detect objects.

Then, in the testing phase, the object detection results were converted into a classification-based evaluation to calculate accuracy, precision, recall, and F1 score. The values for each category were obtained from equations (1), (2), (3), and (4). The training, validation, and testing processes used the same parameters for both models. This was done to ensure a fair performance comparison between the two models.

$$Accuracy = \frac{Tp+TN}{TP+TN+FN+FP} \times 100\% \qquad (1)$$

$$Precision = \frac{TP}{TP+FP} \times 100\% \qquad (2)$$

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (3)$$

$$F1\ Score = \frac{2*(Precision*Recall)}{(Precission+Recall)} \times 100\% \quad (4)$$

Table 1. Predicted Value

TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative

III. Results and Discussion

A. *Training performance*

In the training process, this study used 70% of the dataset for each YOLOv12n and YOLOv26n model with 100 epochs. The results of the training process are in the form of box loss values, classification loss, and distribution focal loss (dfl). The training results for each YOLO model are shown in Figures 5 and 6.

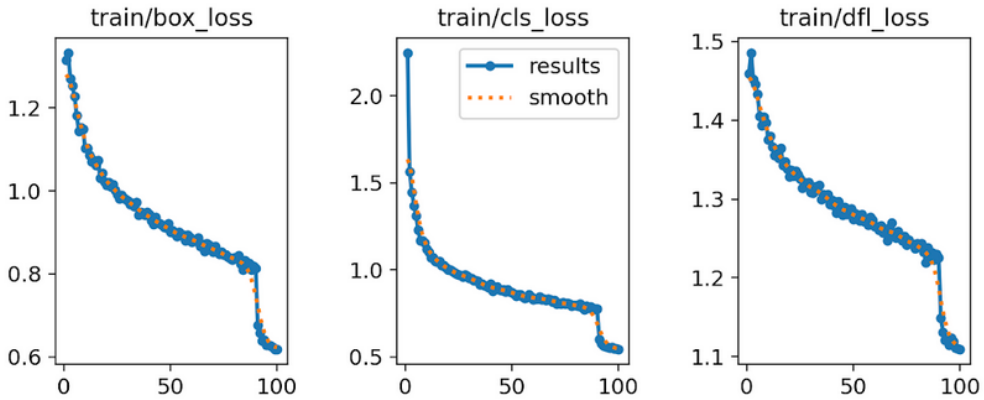


Figure 5. YOLOv12n 100 epochs training test loss values (A) Training box loss, (B) Training classification loss, (C) Training distribution focal loss (dfl)

From Box loss is a matrix used to measure the reliability of the predicted bounding box overlap with the actual bounding box, thus illustrating the model's ability to determine object location. Therefore, the smaller the box loss value, the more accurate the detected object's position. Classification loss measures the difference between the predicted class probability and the actual class label. Distribution Focal Loss (dfl) is used to address class imbalance when detecting objects across different classes and to refine bounding box predictions, especially for objects that are difficult to distinguish [20][21]. The training results for the YOLOv12n model with 100 epochs, as seen in Figure 5, show an average box loss of 0,95014; a classification loss of 0,94543; and a Distribution Focal Loss (dfl) of 1,29476.

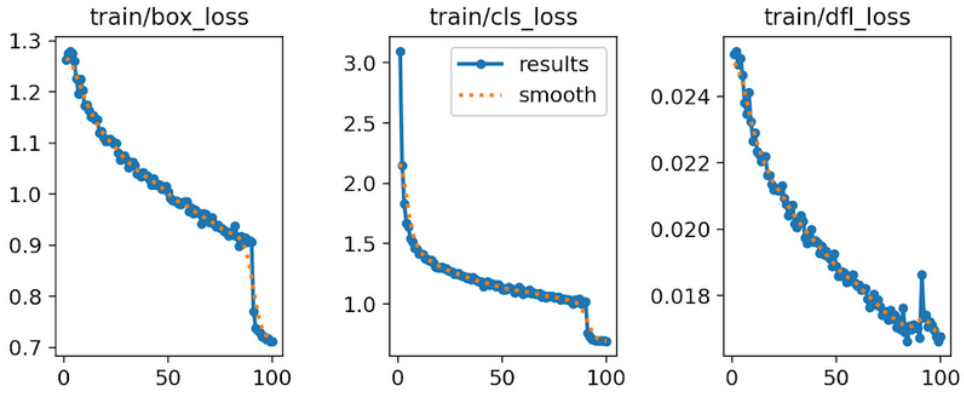


Figure 6. YOLOv26 100 epochs training test loss values (A) Training box loss, (B) Training classification loss, (C) Training distribution focal loss (dfl)

Meanwhile, the training performance results of the YOLOv26n dataset with epoch 100 can be seen in Figure 6; the average box loss value is 1,04624; the classification loss is 1,19796; and the distribution focal loss is 0,01951. Based on the general training results of both YOLO models in this case, YOLOv12n tends to have better results in adjusting the training data in terms of bounding box placement and object classification because it has lower box loss and classification loss values compared to YOLOv26n. However, when viewed from the distribution of focal loss (dfl), YOLOv26n is better when modeling the distribution of bounding box coordinates with precision, especially in capturing complex object details. This is because YOLOv26 has a smaller distribution focal loss (dfl) value than YOLOv12n.

The difference in loss values indicates that each model has different characteristics during the training stage. In general, box loss and classification loss reflect the model's ability to adapt to the training data, while (dfl) contributes to the quality of the bounding box's spatial representation. This difference may also indicate variations in learning mechanisms influenced by the architecture of each model. Furthermore, the difference in loss values can provide an initial insight into the stability of the training process. However, further evaluation analysis is needed to assess the overall performance of the model.

B. Validation Performance

Then, in the model validation test process, in this research, 20% of the dataset was used for each YOLOv12n and YOLOv26n model with a total of 100 epochs. The results of this validation process are in the form of recall, precision, mean average precision (mAP), box loss, confusion matrix, classification loss, and distribution focal loss (dfl).

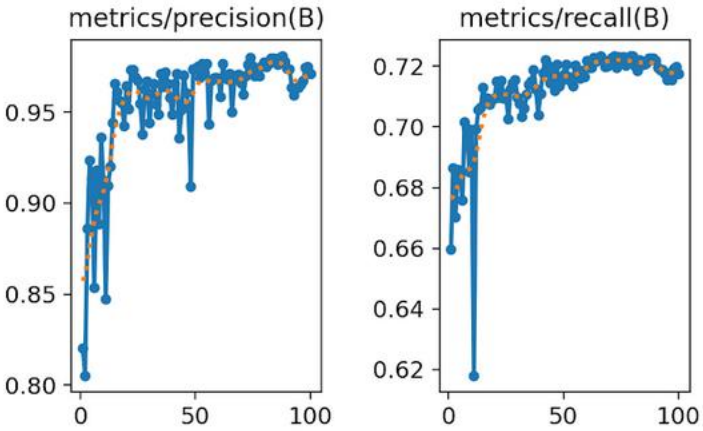


Figure 7. Precision values (left) and recall values (right) of validation test results on YOLOv12n 100 epochs

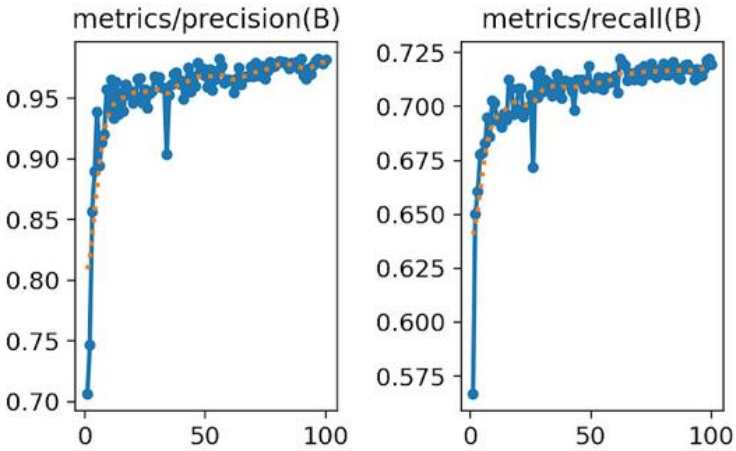


Figure 8. Precision values (left) and recall values (right) of validation test results on YOLOv26n 100 epochs

Figures 7 and 8 show the precision and recall values for each model. The YOLOv12n model with 100 epochs had a precision of 0,95661; while the YOLOv26n model had a precision of 0,95734. Precision results, values close to 1 indicate a low false positive rate [22]. This means that most positive predictions are consistent with the actual situation. Therefore, a higher precision value indicates a more selective model when detecting objects. The ratio of the system's success in finding the input dataset is called recall [23]. The recall value for the YOLOv12n model was 0,71396 and for YOLOv26n, the result was 0,70462. By definition, a low recall value indicates a high number of missed objects (false negatives).

Based on the results obtained from the model validation process, YOLOv26n has a higher precision, indicating that the model is more selective when detecting objects, thus minimizing false positives. However, a lower recall value indicates that the model tends to miss some objects (false negatives) when detecting objects. However, conversely, YOLOv12n has a higher recall value. This indicates that the model is able to detect more objects, but allows a greater number of false positives. Thus, in this case, YOLOv26n is better at reducing false positives, while YOLOv12n is better at reducing false negatives.

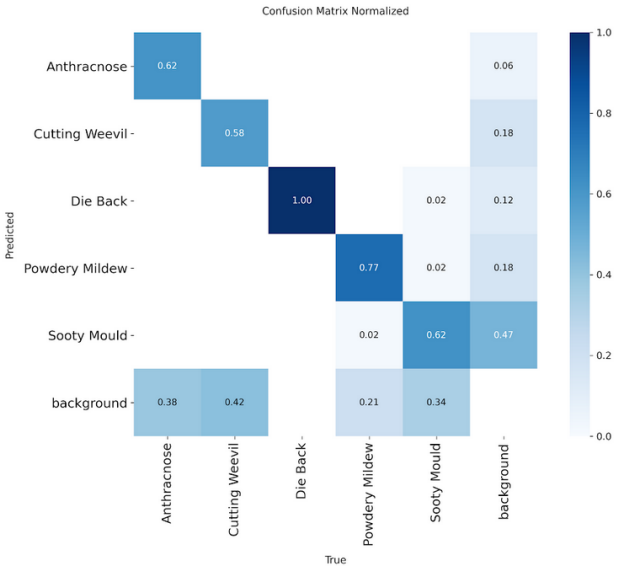


Figure 9. Confusion matrix validation test on YOLOv12n 100 epochs

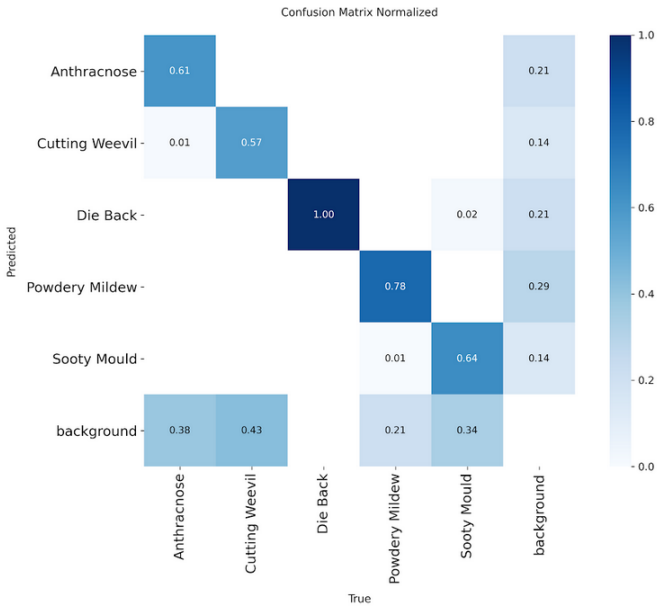


Figure 10. Confusion matrix validation test on YOLOv26n 100 epochs

Figures 9 and 10 show the performance results of each model for each class in the dataset. In the YOLOv12n model (Figure 9), the anthrachnose class reached 62%, the cutting weevil reached 58%, the dieback reached 100%, powdery mildew reached 77%, and sooty mold reached 62%. Meanwhile, for the YOLOv26n model (Figure 10), the anthrachnose class reached 61%, the weevil reached 57%, the dieback reached 100%, powdery mildew reached 78%, and sooty mold reached 64%. Based on the confusion matrix results, the average YOLOv12n reached 71.8%, while YOLOv26n reached 72%. This indicates an improvement in YOLOv26n.

These results indicate that the cutworm beetle is the most difficult to detect due to its lowest performance score. This is due to its visual characteristics, which lack contrast. Furthermore, anthracnose and sooty mold also performed quite poorly. This could be due to the visual similarity between anthracnose and sooty mold, which both display dark patterns on the leaf surface, albeit with different shapes. Furthermore, dieback had the highest performance score due to its distinctive visual characteristics. Powdery mildew also scored quite high due to its distinctive appearance, which resembled a white powder on the leaf surface.

Detection performance tends to vary depending on brightness, size, and disease intensity. In this case, based on the confusion matrix results, YOLOv26n demonstrated improved detection performance compared to YOLOv12n.

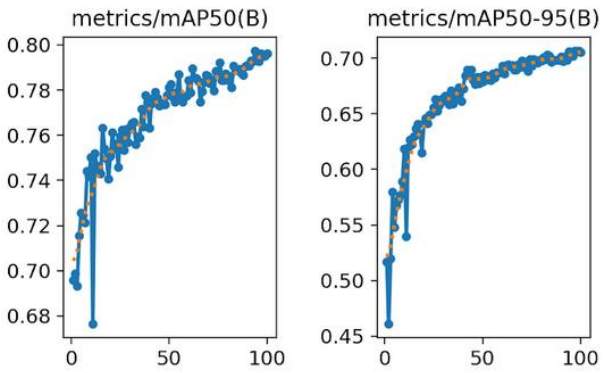


Figure 11. YOLOv12n mAP values of 100 epochs at a threshold of 0,5 (left) and at a threshold of 0,5 - 0,95 (right)

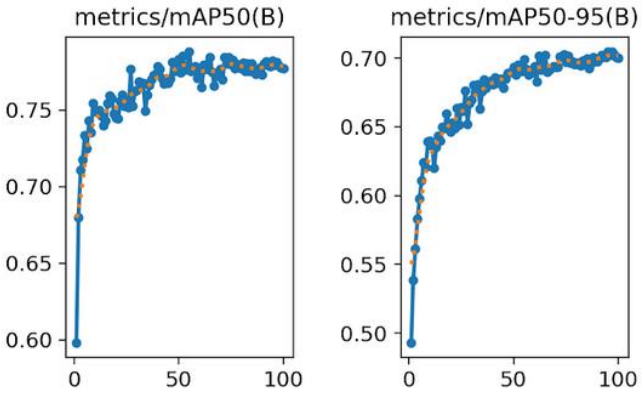


Figure 12. YOLOv26n mAP values of 100 epochs at threshold 0,5 (left) and at threshold 0,5 - 0,95 (right)

There are several ways to measure system performance, one of which is by using the mAP value. Mean precision (mAP) is one of the most widely used evaluation metrics for object detection. Mean average precision (mAP) takes the average precision across all classes and calculates it at a predetermined Intersection over Union (IoU) threshold [24]. Based on Figures 11 and 12, the YOLOv12n model (Figure 11) has a Mean average precision (mAP) value at a threshold of 0.5 of 0.76987 and at a threshold of 0.5–0.95 of 0.66792. Meanwhile, in the YOLOv26n model (Figure 12), the mean average precision (mAP) value at a threshold of 0.5 reaches 0.76492, and at a threshold of 0.5–0.95 reaches 0.67597.

These results indicate that YOLOv12n is slightly superior to YOLOv26n at an IoU threshold of 0.5, indicating better performance when detecting common objects. However, YOLOv26n's mAP value is slightly higher in the IoU range of 0.5–0.95. This reflects YOLOv26n's slightly better bounding box localization accuracy across various IoU thresholds.

This aligns with the previous precision and recall validation analysis, where YOLOv26n has a higher precision value, reflecting more selective detection with a lower number of false positives. Meanwhile, YOLOv12n has a higher recall value, reflecting its ability to detect a wider range of objects. Although YOLOv26n is superior in localization accuracy, the performance difference between the two models is relatively small. This suggests a trade-off between selectivity and detection coverage. Therefore, mAP needs to be further analyzed in conjunction with loss metrics to gain a more comprehensive understanding of model performance.

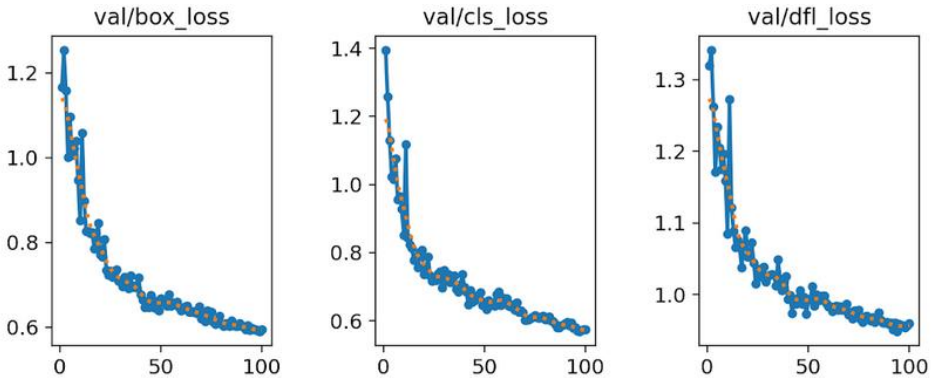


Figure 13. Validation test loss value on YOLOv12n 100 epochs (A) Box loss validation, (B) Classification loss validation, (C) Distribution focal loss (dfl) validation

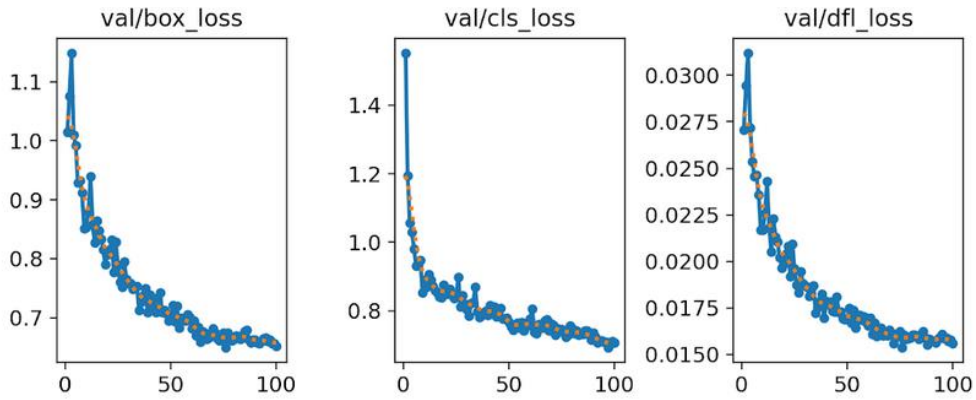


Figure 14. Validation test values on YOLOv26n 100 epochs (A) Box loss validation, (B) Classification loss validation, (C) Distribution focal loss (dfl) validation

In the validation process, the classification loss, box loss, and distribution focal loss (dfl) values were also obtained. For each model, these are shown in Figures 13 and 14. In Figure 13, for the YOLOv12n model, the average box loss value reached 0,70331, the classification loss reached 0,71931, and the distribution focal loss (dfl) reached 1,02436. Meanwhile, in Figure 14, for the YOLOv26n model, the average box loss value reached 0,74857, the classification loss reached 0,80466, and the distribution focal loss (dfl) reached 0,01859.

In general, the lower the loss value, it can be interpreted that the model has a good classification [25]. However, YOLOv26n showed lower DFL values with higher box bounding loss and classification values compared to YOLOv12n, indicating better box bounding distribution but weaker localization and classification performance.

Other validation results confirmed that YOLOv26n had higher precision but lower recall, while YOLOv12n showed the opposite. In terms of mAP, YOLOv12n performed slightly better at an IoU of 0.5, while YOLOv26n excelled at an IoU of 0.5–0.95. This indicates more consistent localization at tighter thresholds. These results suggest that loss value alone is not a strong enough indicator of overall model performance and should be evaluated in conjunction with evaluation metrics such as precision, recall, confusion matrix, and mAP for a more comprehensive assessment.

Based on the overall evaluation results, YOLOv26n can be considered the superior model in this case due to its superior performance at mAP IoU 0.5-0.95 and high precision values. This indicates more accurate and consistent detection.

C. testing and evaluation

After the validation process, the next step is to conduct trials and evaluations. This trial process used 10% of the total dataset used. The test results were taken from one image in each class for samples, which can be seen in Table 2 for YOLOv12n and Table 3 for YOLOv26n. This was done to determine how the system performs for each model in detecting objects, namely leaf diseases (*Mangifera indica*), compared to the annotation that has been done. Below is a table of the results of YOLOv12n and YOLOv26n object detection.

Table 2. Test results for object detection on YOLOv12n 100 epochs

No	Object detection results	information	System trust (%)
1.		Correctly detected (*Powdery Mildew*)	90
2.		Correctly detected (*Antrachnose*)	96
3.		Correctly detected (*Cutting Weevil*)	91
4.		Correctly detected (*Sooty Mould*)	91

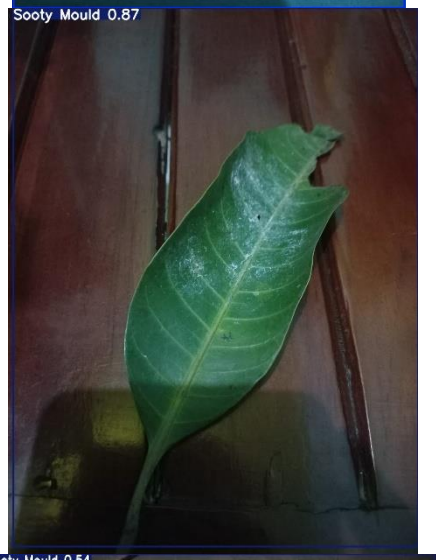
5.		Correctly detected (*Dieback*)	86
6.		Misclassified (Cutting Weevil as Antrachnose)	94
7.		Misclassified (Sooty Mould as Anthracnose)	66
8.		Misclassified (Sooty Mould as Anthracnose)	56

9.		Misclassified (Antrachnose as Powdery Mildew)	83
10.		Missed detection (Powdery Mildew as Sooty Mould)	60

Table 3. Test results for object detection on YOLOv26n 100 epochs

No	Object detection results	Information	System trust (%)
1.		Correctly detected (*Powdery Mildew*)	91

2.		Correctly detected (*Antrachnose*)	94
3.		Correctly detected (*Cutting Weevil*)	94
4.		Correctly detected (*Sooty Mould*)	95
5.		Correctly detected (*Dieback*)	90

6.		Misclassified (Cutting Weevil as Antrachnose)	82
7.		Misclassified (Powdery Mildew as Sooty Mould)	87
8.		Low confidence detection (Sooty Mould)	54

9.		Low confidence detection (Powdery Mildew)	51
10.		Misclassified (Sooty Mould as Anthracnose)	68

Then, the test results were calculated using equations 1, 2, 3, and 4 in the method section, which are then displayed in Table 4 for YOLOv12n and Table 5 for YOLOv26n. Both tables show the system's performance in detecting leaf diseases (*Mangifera indica*).

Table 4. Evaluation results of testing on YOLOv12n epoch 100

Class	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Powdery Mildew	95,94	96,43	99,47	97,92
Antrachnose	93,91	95,36	98,40	96,86
Cutting Weevil	89,85	92,19	97,25	94,63
Sooty Mould	96,45	96,45	100	98,19
Die Back	96,45	96,94	99,48	98,18
Average	**94,52**	**95,47**	**98,92**	**97,16**

Table 5. Evaluation results of testing on YOLOv26n epoch 100

Class	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Powdery Mildew	97,46	97,96	99,48	98,71
Antrachnose	96,45	98,45	97,94	98,20

Cutting Weevil	95,43	97,92	97,41	97,66
Sooty Mould	97,46	98,46	98,97	98,70
Die Back	98,48	98,48	100	99,23
Average	**97,06**	**98,25**	**98,76**	**98,50**

Based on the F1-score values in Tables 3 and 4, among the classes evaluated in the YOLOv12n and YOLOv26n models, cutting weevil consistently demonstrated the lowest F1-score across both models; dieback, sooty mold, and powdery mildew performed better and more consistently. This is also supported by the confusion matrix results from the validation process. Dieback class achieved the highest detection accuracy, indicating that its visual characteristics are easily recognized by the model. Similarly, powdery mildew and sooty mold also demonstrated strong performance with low classification errors. In contrast, anthracnose performed slightly lower. This was due to the model's confusion when detecting visually similar classes, particularly sooty mold.

For the cutting weevil class, error analysis indicates that the lower performance is associated with frequent misclassifications. Several detection examples in Tables 2 and 3 indicate that the model incorrectly classified this class as other diseases. This suggests that the visual characteristics of the Cutworm Beetle are less distinctive and more difficult to detect.

In addition to misclassification errors, borderline cases were also observed, particularly in ambiguous visual conditions. These cases were characterized by lower confidence scores and less accurate localization, indicating the model's challenges in distinguishing similar patterns. Overall, YOLOv26n outperformed YOLOv12n, as demonstrated by its higher F1 score and more stable performance across classes. This indicates that the architectural improvements in YOLOv26n contributed to its improved detection capabilities.

IV. Conclusion

This study evaluated the YOLOv12n and YOLOv26n models for automatically detecting mango leaf diseases. Based on the validation results, both models demonstrated a stable learning process, characterized by a decrease in loss values (box loss, classification loss, and distribution focus loss). However, the YOLOv26n model displayed better convergence and lower loss values. This indicates a more optimal model generalization capability compared to YOLOv12n.

Furthermore, during testing, both models were able to detect objects in real time with a good level of accuracy. However, the YOLOv26n model consistently demonstrated superior performance, reaching an accuracy of 97.06% compared to YOLOv12n's 94.52%. This indicates that the architectural improvements in YOLOv26n significantly contributed to improved detection performance. This is also supported by the higher F1 score for YOLOv26n than for YOLOv12n.

Overall, the primary scientific contribution of this study is providing a comprehensive analysis of the performance of two state-of-the-art lightweight YOLO models on a multi-class mango leaf disease dataset, encompassing the training, validation, and testing stages. This research also confirms the trade-off between computational efficiency and accuracy and indicates that newer models can provide significant performance improvements without sacrificing efficiency. This research supports the application of efficient and accurate deep learning models to agriculture, particularly for the early detection of plant diseases such as leaf spot disease (Mangifera indica) with limited device resources.

Recommendations for future research include using larger and more diverse datasets to enhance model learning, further refining the image annotation process to achieve optimal results, and utilizing other advanced deep learning methods.

V. Acknowledgements

The author would like to express his gratitude to all parties who have provided support and contributions to the implementation of this research. He also expresses his gratitude to the institution and his supervisor, Prof. Dr. Lilik Anifah, S.T., M.T., for their invaluable guidance, direction, and input throughout the research process and the preparation of this manuscript.

VI. Bibliography

[1] E. T. Oviana, R. Alpaizon, D. P. Khalifah, and P. Dwirotama, "Sistem Deteksi Hama dan Penyakit Tanaman Mangga (Mangifera indica L .) Berbasis Deep Learning Menggunakan Model Pra Latih YOLOv5," vol. 35, no. 1, pp. 151–163, 2024, doi: https://doi.org/10.24198/agrikultura.v35i1.53834.

[2] T. R. Siagian, I. Taufik, Z. Indra, and S. Susiana, "KLASIFIKASI PENYAKIT DAN HAMA PADA DAUN MANGGA MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 9, no. 6, pp. 10370–10376, 2025, doi: https://doi.org/10.36040/jati.v9i6.16477.

[3] M. Iqbal *et al.*, "Automated Identification of Mango Leaf Diseases Using Deep Convolutional Neural Networks," vol. XX, no. X, pp. 1–14, 2026, doi: 10.15244/pjoes/215218.

[4] D. E. Arissandi, "Sistem Monitoring Gambar Daun Mangga Untuk Deteksi Awal Penyakit," no. I, 2025, doi: 10.20473/KNJ.X.X.

[5] K. U. Shetty, R. Javed Kutty, K. Donthi, A. Patil, and N. Subramanyam, "Plant Disease Detection for Guava and Mango using YOLO and Faster R-CNN," in *2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)*, 2024, pp. 1–6. doi: 10.1109/IATMSI60426.2024.10503209.

[6] K. Dhawan, R. S. Perumal, and R. K. Nadesh, "Image-based Mangifera Indica Leaf Disease Detection using Transfer Learning for Deep Learning Methods," vol. 22, no. 2, pp. 27–40, 2023.

[7] V. K. Pratap and N. S. Kumar, "Deep Learning based Mango Leaf Disease Detection for Classifying and Evaluating Mango Leaf Diseases," vol. 15, no. 02, pp. 261–277, 2024, doi: 10.54216/FPA.150222.

[8] Q. Wang and X. Duan, "Detection and Identification of Mango Disease Leaves Based on Improved YOLOv7," in *2024 5th International Conference on Computer Vision, Image and Deep Learning (CVIDL)*, 2024, pp. 1263–1267. doi: 10.1109/CVIDL62147.2024.10603942.

[9] G. O. S *et al.*, "Deep Learning-based Mango Leaf Disease Detection using Yolov8," in *2025 3rd International Conference on Advancement in Computation & Computer Technologies (InCACCT)*, 2025, pp. 881–885. doi: 10.1109/InCACCT65424.2025.11011320.

[10] Y. Tian, Q. Ye, and D. Doermann, "Yolov12: Attention-centric real-time object detectors," *arXiv Prepr. arXiv2502.12524*, 2025, doi: 10.48550/arXiv.2502.12524.

[11] R. Sapkota, R. H. Cheppally, A. Sharda, and M. Karkee, "YOLO26: key architectural enhancements and performance benchmarking for real-time object detection," *arXiv Prepr. arXiv2509.25164*, 2025, doi: 10.48550/arXiv.2509.25164.

[12] P. Hidayatullah and R. Tubagus, "YOLO26: A Comprehensive Architecture Overview and Key Improvements," *arXiv Prepr. arXiv2602.14582*, 2026, doi: 10.48550/arXiv.2602.14582.

[13] A. Zargar, "Mango Leaf Diseases Dataset," 2023. [Online]. Available: https://www.kaggle.com/datasets/ahmadzargar/mango-leaf-diseases-dataset?select=val

[14] K. Vaishnavi and R. Malinidevi, "Biological and Optimal Control Model for Powdery Mildew Disease in Mango Plants and Fruits," *Appl. Model. Simul.*, vol. 9, pp. 174–188, 2025, [Online]. Available: http://arqiipubl.com/ojs/index.php/AMS_Journal/article/view/869

[15] G. V. Benatar, Y. Nurhayati, and N. Febryani, "Identifikasi Colletotrichum asianum penyebab antraknosa mangga kultivar Golek di Indramayu," *Media Pertan.*, vol. 8, no. 1, pp. 1–13, 2023, doi: 10.37058/mp.v8i1.6900.

[16] A. Hossain, S. Sakib, H. Muhammad, and S. E. Arman, "Heliyon Deep learning for mango leaf disease identification : A vision transformer perspective," *Heliyon*, vol. 10, no. 17, p. e36361, 2024, doi: 10.1016/j.heliyon.2024.e36361.

[17] S. D. Anggraeni, E. Y. Puspaningrum, and A. L. Nurlaili, "Klasifikasi Penyakit Daun Mangga Menggunakan ANFIS," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 9, no. 3, pp. 5467–5473, 2025, doi: 10.36040/jati.v9i3.14311.

[18] A. Niscioli, "Impact of Twig-Tip Dieback on Leaf Nutrient Status and Resorption Efficiency of Mango (Mangifera indica L .) Trees," 2024, doi: 10.3390/horticulturae10070678.

[19] N. F. Y. Putra, "Evaluasi Pengaruh Variasi Tingkat Anotasi Terhadap Performa dan Efisiensi Model YOLOv11 pada Deteksi Multi-objek Buah," 2025, *Universitas Islam Indonesia*. [Online]. Available: https://dspace.uii.ac.id/handle/123456789/58823

[20] I. Ahmad *et al.*, "Deep learning based detector YOLOv5 for identifying insect pests," *Appl. Sci.*, vol. 12, no. 19, p. 10167, 2022, doi: 10.3390/app121910167.
[21] A. Lahmdani, A. Elboushaki, and S. Saoud, "Targeted detection of plant disease infection levels using YOLOv8 and YOLOv9 for agricultural monitoring and public healthcare support," *Discov. Artif. Intell.*, vol. 6, no. 1, p. 258, 2026, doi: 10.1007/s44163-026-00841-z.
[22] T. Diwan, G. Anirudh, and J. V Tembhurne, "Object detection using YOLO: challenges, architectural successors, datasets and applications," *Multimed. Tools Appl.*, vol. 82, no. 6, pp. 9243–9275, 2023, doi: 10.1007/s11042-022-13644-y.
[23] P. Krishna, B. Chintan, and M. P. Luigi, "Improved Ship Detection Algorithm from Satellite Images Using YOLOv7 and Graph Neural Network [J]," *Algorithms*, vol. 15, no. 12, p. 473, 2022, doi: 10.3390/a15120473.
[24] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-time flying object detection with YOLOv8," *arXiv Prepr. arXiv2305.09972*, 2023, doi: 10.48550/arXiv.2305.09972.
[25] G. Thiodorus, A. Prasetia, L. A. Ardhani, and N. Yudistira, "Klasifikasi citra makanan/non makanan menggunakan metode Transfer Learning dengan model Residual Network," *Teknol. J. Ilm. Sist. Inf.*, vol. 11, no. 2, pp. 74–83, 2021, doi: 10.26594/teknologi.v11i2.2402.