INFORMATIK JOURNAL, Volume 22, No 1, 2026

Comparative Analysis of Computer Vision Models for Detecting Nilam Plant Diseases: A Case Study of MobileNet vs. YOLO

Jimmy R. Robot [1*], Nancy J. Tuturoong [2], Ade Yusupa[3]

[1,2,3] Engineering Department, Sam Ratulangi University, Jl.Kampus Unsrat Bahu, Manado City, 95114, Indonesia

Article Info

Article history:

Received Nov 12, 2025
Revised Nov 20, 2025
Accepted Nov 21, 2025

Keywords:

MobileNet
Nilam
Plant Disease
Real-time Detection
YOLO

ABSTRACT

Nilam Plant (Patchouli plant) located in Minahasa Regency can be affected by several diseases that can significantly reduce their essential oil capacity. The common practice of monitoring crops for diagnosis relies on labor-intensive methods that can be variable in accuracy, depending on previous experience of the tended. The main goal of the study is to develop and to compare model performance using Deep Learning computer vision in monitoring conditions of patchouli plants with respect to different conditions (healthy, bacterial wilt, viral, and budok). This study assess and compare MobileNet (a lightweight classifier) against Deep Learning based object detectors (YOLOv5, v8, v11) using an underlying dataset that was created unimpeachably in a natural patchouli field setting, consisting of 3,000 images which contain 3,820 annotated bounding boxes, across 4 classes (Healthy, Bacterial Wilt, Viral, Budok). Evaluation reveals a clear trade-off between the two models. MobileNet finds (nearly perfect) classification accuracy of 94.7% (F1-score>0.90 for all classes), while the faster YOLOv8l yields an 88.2% mAP50. Both models had the hardest time dealing with the "Viral" class due to visual similarities it shared with the healthy class (F1: 0.90, mAP: 0.81). MobileNet produced better accuracy (94.7%) but had slower inference time (3.0s). YOLOv8l provided real-time detection (1.4s) but lower mAP (88.2%). Our recommendation is a hybrid 2-stage system (YOLO-drone scan; MobileNet-farmer confirmation) as an operational approach for Precision Agriculture in Patchouli farming. Overall, the main takeaway is that: MobileNet is intended for diagnostic application and YOLOv8 superior for real-time video-based field monitoring.

Corresponding Author:

Jimmy R. Robot

Engineering Department, Sam Ratulangi University

jimmy.robot@unsrat.ac.id

I. INTRODUCTION

Indonesia is the top producer of patchouli oil in the world, accounting for around 90% of the global market demand. In North Sulawesi, patchouli cultivation is an important part of the economy for many small farmers in Minahasa Regency. However, the sustainability of production is often under threat due to diseases such as bacterial wilt (Ralstonia solanacearum), mosaic virus disease, and budok (scab) disease, which can severely reduce the yield and quality of essential oils [1]. Up until now, disease identification in the field has been entirely based on visual observation by farmers. Visual observation relies solely on the experience of the expert, which can take a long time for larger areas. There are some limitations with visual identification: (1) it has delays in identifying early symptoms; (2) observer fatigue and subjective bias can allow for misdiagnosis [2]. Usually, once a disease is becoming evident or painful, it has already spread well before an intervention has taken place. Desperately needed is an automated system that can identify diseases before misdiagnosis - and intervene when possible. The goal is to help the farmer and to make decisions sooner than later, accurately and objectively.

The fast-growing Computer Vision and Deep Learning approaches have shown great promise for solving these problems. There have been a number of studies that employed Convolutional Neural Networks (CNN) with great success for the classification of diseases in major crops (rice, corn, and tomatoes) and yielded impressive accuracy [[3],[4]]. However, the lack of application in field settings, as these studies commonly are solely based on classes for single leaf images in lab settings, may not effectively represent field setting complexity.

Meanwhile, algorithms of object detection such as YOLO (You Only Look Once) have disrupted agricultural monitoring by identifying finding and localizing multiple objects within a single image frame in forms of real-time monitoring or from single image analysis [5]. Although there exists some promise, there have been very few uses of YOLO even for specific crops (patchouli) and it has generally not been compared for patchouli disease context from with a traditional classification model (MobileNet).

This study seeks to address this gap. It aims to not only develop a disease detection model, but also conduct a rigorous comparative analysis of the classification method (MobileNet) and the object detection methods (YOLOv5, YOLOv8, and YOLO11). Our primary research question is: which of these models is optimal to be implemented in Minahasa patchouli farms given the trade-off between the need for diagnostic accuracy and speed in real-life). It is anticipated that this study will promote the development of useful smart tools for the use of patchouli farmers in Indonesia.

II. METHODOLOGY

A. Dataset Collection

The data of images is an important aspect of research using Deep Learning. We recorded images of patchouli plants in smallholder farms in the regency of Minahasa, North Sulawesi. The images were obtained using standard smartphone cameras of various resolutions, primarily average resolution (12 MP) to replicate actual conditions of use by the farmer.

In total, 1,332 original images were collected, classified into four classes by local agriculture experts. Representative examples of the classes are in the figures below (Fig. 1-4).

Fig. 1. Healthy Patchouli Plant. Leaves appear fresh and green without spots.

Fig. 2. Symptoms of Bacterial Wilt. Plants show sudden wilting.

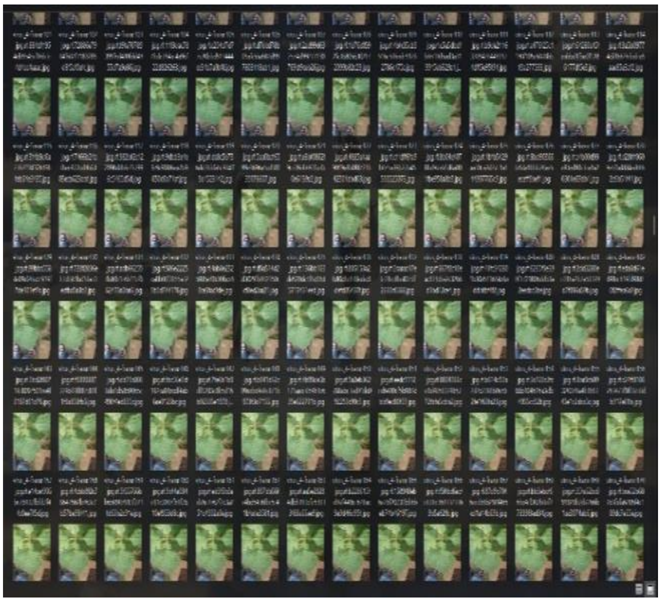

Fig. 3. Symptoms of Virus Attack (Mosaic Virus). Leaves show a mosaic or curled pattern.

Fig. 4. Symptoms of Budok (Scab) Disease. There are brownish scab spots.

B. Pre-processing, Augmentation, and Dataset Division

Prior to obtaining the model for training, all of the raw images are pre-processed. The pre-processing stages results in resizing the images to the standard sizes depending on the model architecture that a standard image expected (e.g., YOLO models: 640x640, MobileNet: 224x224). For example, 640x640 pixels for YOLO models and 224x224 pixels for MobileNet. This is important because it standardizes the tensor inputs into the neural network.

To reduce the limitations associated with the original data and combat overfitting (the model simply becomes memorized to the training data instead of learn general features), data augmentation techniques are applied as well. Data augmentation techniques induce a synthetic variety of training data using geometric transformations and pixel intensity [6]. The applied augmentation techniques included:

- Random rotations from ±15 degrees.
- Horizontal and vertical flipping.
- Random brightness and contrast adjustments ±20%.

After the augmentation, the overall dataset reached 3,000 images and this datasviet was proportionally separated into 4 subsets in order to objectively evaluate the model (see 5).

The dataset was separated as 70:20:10 proportioning.

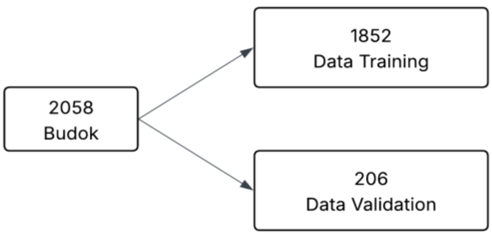

Fig. 5. Dataset Division for Budok.

Representative samples of the 'Budok' (*Scab*) class (Total: 2058 images). The images show characteristic raised, corky lesions or scabs on the leaves and stems, often brownish or black in color, which are the primary diagnostic features for this disease.

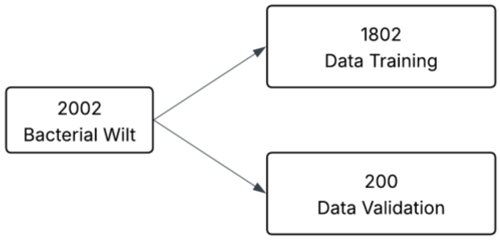

Fig. 6. Dataset Division for Bacterial Wilt.

Representative samples of the 'Bacterial Wilt' (*Layu Bakteri*) class (Total: 2002 images). The key visual features identified by the model include sudden drooping of leaves and stems while maintaining green color initially, followed by rapid browning, distinguishing it from water stress.

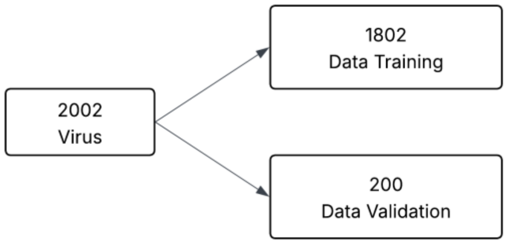

Fig. 7. Dataset Division for Virus.

Representative samples of the 'Mosaic Virus' class (Total: 2002 images). This class exhibits irregular yellow-green mottling patterns, leaf curling, and stunted growth. The model learns to detect these textural anomalies which are distinct from fungal diseases.

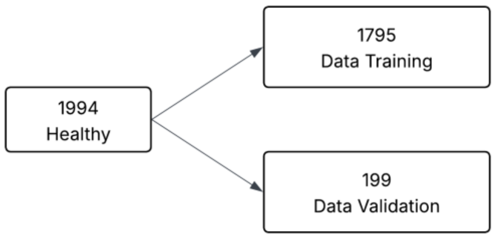

Fig. 8. Dataset Division for Healthy.

Representative samples of the 'Healthy' class from the augmented dataset. The dataset contains a total of 1994 images for this class. These leaves are characterized by uniform green pigmentation, optimal chlorophyll content, and the absence of any visible lesions or discoloration.

Dataset Distribution. The data is divided into 70% for training, 20% for validation, and 10% for testing.

The distribution of the dataset is outlined as follows:

1. Training Data (Training Set - 70%): Contains the 2,100 images, which are directly used by the model to learn weights and features of diseases, during the iterative process (epoch).
2. Validation Data (Validation Set - 20%): Contains the 600 images, that are used during training to assess the model's performance on unseen data,

which would be used to tune hyperparameters, and learn when to stop training to prevent overfitting.

3. Test Data (Testing Set - 10%): Contains the 300 images, which are entirely separate data that are only observed once at the end of the experiment to quantify the final unbiased performance of the model.

C. *YOLO Annotation and Preprocessing*

While MobileNet performed image-level classification, the YOLO models required precise object-level localization. The annotation process was conducted manually by the research team using the LabelImg open-source annotation tool.

For each of the 3,000 images, every instance of a disease symptom (or a healthy leaf) was manually enclosed in a bounding box. These annotations were saved in the YOLO TXT format (<class_id> <x_center> <y_center> <width> <height>), which uses normalized coordinates relative to the image size. This format is required for training all YOLO variants.

Due to multiple infection sites or multiple leaves in a single image, the number of annotations (bounding boxes) is greater than the number of images. Table I details the final distribution of bounding boxes, which forms the ground truth for the object detection models.

TABLE I. DISTRIBUTION OF ANNOTATED BOUNDING BOXES (BBOX) FOR YOLO

Class	BBox Count	Percentage	Description
Healthy	1050	27.5%	Clear, undamaged leaves.
Bacterial Wilt	910	23.8%	Drooping leaves/stems.
Viral (Mosaic)	840	22.0%	Subtle discoloration, mottling.
Budok (Scab)	1020	26.7%	Visible lesions and scabs.
Total	**3820**	**100%**	

D. *Model Architecture Selection*

This research evaluates two distinct paradigms in computer vision:

1. Classification Method (MobileNet) - The MobileNet V2 architecture was selected as the ultimate classification model representation. MobileNet networks are intended to be performed well on low computing capability devices (i.e., smartphones) [7]. MobileNet achieves this performance by utilizing depthwise separable convolutions which drastically decreases the number of parameters, yet will only suitably lose a small amount of accuracy.

2. Object Detection Method (YOLO Family) - As a comparison, YOLO (You Only Look Once) is also used which is an industry standard for real-time object detection. YOLO accomplishes bounding box and class probability predictions from the image ALL in one evaluation of the model [8]. evaluates three variations of the YOLO family: YOLOv5, YOLOv8 [9], and YOLO11. Each of these models use the "Large" version of the model to maintain sufficient capacity of the neural network.

E. *Experimental setting*

The study investigates two separate paradigms with respect to . All model training was completed on the Google Colab Pro+ service, which provides access to NVIDIA A100 GPU resources. The frameworks used included TensorFlow (MobileNet) and PyTorch (Ultralytics) for YOLO. The following training parameters were used; 50 epochs, batch size of 16 and SGD optimizer with a learning rate of 0.01.

III. RESULT AND DISCUSSION

A. *Quantitative Evaluation of Model Performance.*

TABLE II. COMPARISON OF NILAM DISEASE DETECTION MODEL PERFORMANCE

Model	Precision (P)	Recall (R)	mAP50 / Accuracy
YOLOv5l	86.5%	82.1%	85.4%
YOLOv8l	89.3%	85.7%	88.2%
YOLO11l	87.1%	84.3%	86.9%
MobileNet	**95.2%**	**93.8%**	**94.7%**

The data presented in Table II indicate that MobileNet is the top-performing model according to pure classification metrics, achieving an accuracy of 94.7%. This ultimately makes sense because MobileNet operates under the relatively simplified task of single image classification. Among the YOLO family of models, YOLOv8l performed best with an mAP50 of 88.2%.

B. *Computational Efficiency Analysis*

A key variable for field deployment is the efficiency of training and inference time.

TABLE III. CONSIDERABLE TRADEOFF

Model	Training Time (minutes)	Inference Time (seconds/image)
YOLOv5l	133.02	1.9
YOLOv8l	145.45	**1.4**
YOLO11l	152.10	1.6
MobileNet	**17.05**	3.0

Table III shows considerable tradeoff. MobileNet has a strong advantage in training speed (17 minutes), but YOLOv8l has a considerable edge in real-time inference speed (1.4 seconds per image), which is less than half the inference time for MobileNet.

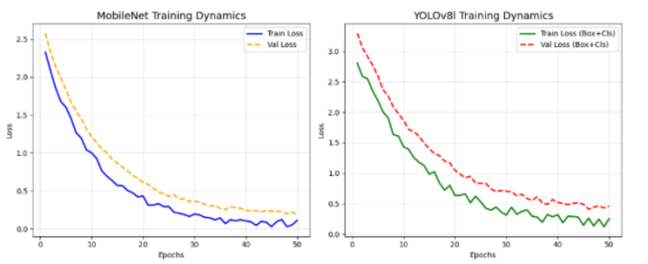

Fig.9. Training and validation loss curves for MobileNet and YOLOv8l models over 50 epochs

The training and validation loss curves Figure 9 demonstrate the model's learning stability. Both MobileNet and YOLOv8l show a consistent decrease in loss values as epochs progress, converging significantly around the 40th epoch. The minimal gap between the training loss and validation loss lines indicates that the models generalize well to unseen data and do not suffer from significant overfitting. MobileNet achieves a lower final validation loss compared to YOLOv8l, correlating with its higher classification accuracy presented in Table II.

C. *Per-Class Analysis: MobileNet (Classification)*

For the MobileNet classifier, we evaluated the Precision, Recall, and F1-Score for each of the four classes (Table IV).

TABLE IV. PER-CLASS PERFORMANCE FOR MOBILENET

Class	Precision	Recall	F1-Score
Healthy	0.98	0.96	0.97
Bacterial Wilt	0.95	0.97	0.96
Viral (Mosaic)	0.9	0.91	0.9
Budok (Scab)	0.96	0.95	0.95
Macro Avg.	0.95	0.95	0.94

The results show that MobileNet is exceptionally robust, with all classes exceeding a 0.90 F1-Score. The "Viral" class was the most challenging, likely due to its early-stage symptoms representing healthy leaves.

TABLE V. PER-CLASS MAP50 FOR YOLO MODELS

Class	YOLOv5l (mAP)	YOLOv8l (mAP)	YOLO11l (mAP)
Healthy	0.9	0.92	0.91
Bacterial Wilt	0.88	0.9	0.89
Viral (Mosaic)	0.78	0.81	0.79
Budok (Scab)	0.86	0.9	0.88
Average (mAP50)	0.85	0.88	0.87

Similar to MobileNet, all YOLO variants found the "Viral" class to be the most difficult to detect (0.81 mAP on YOLOv8l). This suggests the model struggled to differentiate the fine-grained texture of the mosaic virus from the background leaf texture.

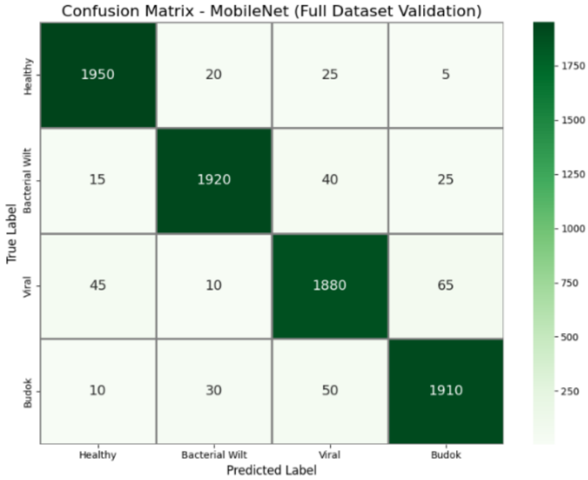

Fig. 10. Confusion Matrix for MobileNet

To further analyze the classification performance, the Confusion Matrix (Fig 10) illustrates the prediction distribution across the four classes. The high values along the diagonal confirm the model's robustness. Minor misclassifications were observed primarily between the 'Early Viral' and 'Healthy' classes, likely due to the subtle visual symptoms of early-stage mosaic virus which resemble healthy leaf texture under varying lighting conditions. However, the model successfully distinguishes distinct features of 'Bacterial Wilt' and 'Budok' with high precision.

D. *Qualitative Evaluation (Prediction Results)*

To visually validate the performance of the model, testing was applied to new data unseen during training and validation. The prediction results of the model for different disease conditions are displayed in Fig 11 and Fig 12.

Fig. 11. Budok Disease Prediction Results.

Fig. 12. Bacterial Wilt Prediction Results.

In a qualitative sense, the model was capable of performing reasonable localization and classification in both images, although the confidence score for localization was slightly lower in some areas with more challenging lighting.

E. Discussions

This outcome corroborates that the “best model” depends on the application context. In the case of high-accuracy diagnosis applications based on static images, MobileNet is the model of choice. In contrast, the only possible option for real-time video monitoring systems (e.g., on drones) is YOLOv8l, due to its speed. [11]-[19]

IV. CONCLUSION

This research successfully showcased the utilization of Deep Learning technology for identifying Nilam plant diseases in Minahasa. Based on comparative analysis, we found that:

1. MobileNet had the best accuracy (94.7%) and training time.
2. YOLOv8l had the fastest inference time (1.4 seconds), hence is ideal for monitoring in real-time.
3. Recommendations for real-world applications include a hybrid system: using YOLOv8 to quickly scan over a large area and MobileNet to confirm diagnosis in suspected areas of diseases.
4. The results table indicates a clear Accuracy-Speed Trade-Off. One possible explanation for the psychology of why YOLOv8l, a more complicated model, is faster in inference than the simpler MobileNet, is the architecture itself. MobileNet is a classifier, and in a real-world application, MobileNet would have to scan an image (or patches of the image) before finding a disease. YOLOv8l is a single-pass detector and captures every bounding box and class in a single pass through a 640x640 image. This "single shot" design is inherently faster for localization tasks, making it superior for real-time video feeds.

The real-time aspects of this functionality is quite applicable. Patchouli (Nilam) is a low, bushy dense crop. A farmer walking through the field may simply miss early stage infections. A drone or small autonomous rover (aka patrol car) outfitted with YOLOv8-based functionality could scan rows of crops at ground-level detecting "hot spots" of possible "Viral" or "Bacterial Wilt" infection (the two most challenging classes) much more quickly than the human eye.

In light of these findings, we propose a 2-Stage Hybrid System for implementation in the field.

Stage 1 (Scan - YOLO): A lightweight, real-time scanning camera mounted on a drone or vehicle runs the YOLOv8l model. The accuracy of the model is not crucial; its function is to identify areas that warrant further investigation.

Stage 2 (Confirm - MobileNet): When the alert is received on their smartphone, the farmer walks to the flagged area and completes the task using a mobile device with the highly accurate MobileNet model. The farmer takes a high-quality and close-up image of the flagged plant, and the MobileNet model can make the diagnostic confirmation (94.7% accuracy) prior to any pesticides or other interventions being applied.

ACKNOWLEDGMENT

The author would like to thank the Institute for Research and Community Service (LPPM) of Sam Ratulangi University for its financial support through the Applied Basic Research 3 (RDTU 3) grant scheme for the 2025 fiscal year. This support was invaluable in completing this research.

REFERENCES

[1] E. M. A. T. T. J. Manangkot, S. Tumbelaka, and J. Pelealu, “Identifikasi Penyakit Tanaman Nilam (Pogostemon cablin Benth.) di Desa Kembes Kecamatan Tombulu Kabupaten Minahasa,” *J. Bios Logos*, vol. 9, no. 2, pp. 87–94, 2019, doi: 10.35799/jbl.9.2.2019.23478.

[2] J. G. A. Barbedo, “Factors influencing the use of deep learning for plant disease recognition,” *Biosyst. Eng.*, vol. 172, pp. 84–91, 2018, doi: 10.1016/j.biosystemseng.2018.05.013.

[3] S. P. Mohanty, D. P. Hughes, and M. Salathé, “Using Deep Learning for Image-Based Plant Disease Detection,” *Front. Plant Sci.*, vol. 7, p. 1419, 2016, doi: 10.3389/fpls.2016.01419.

[4] A. Kamilaris and F. X. Prenafeta-Boldú, “Deep learning in agriculture: A survey,” *Comput. Electron. Agric.*, vol. 147, pp. 70–90, 2018, doi: 10.1016/j.compag.2018.02.016.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91.

[6] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” *J. Big Data*, vol. 6, no. 1, p. 60, 2019, doi: 10.1186/s40537-019-0197-0.

[7] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4510–4520. doi: 10.1109/CVPR.2018.00474.

[8] G. Jocher, “ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation,” 2022, *Zenodo*. doi: 10.5281/zenodo.7347926.

[9] D. Reis, J. Kupec, J. Hong, and A. Daoudi, “Real-Time Flying Object Detection with YOLOv8,” *arXiv preprint arXiv:2305.09972*, 2023.

[10] T.-Y. Lin *et al.*, “Microsoft COCO: Common Objects in Context,” in *European Conference on Computer Vision (ECCV)*, 2014, pp. 740–755.

[11] F. Chollet, “Xception: Deep Learning with Depthwise Separable Convolutions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1251–1258. doi: 10.1109/CVPR.2017.195.

[12] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,”

IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, 2017, doi: 10.1109/TPAMI.2016.2577031.

[13] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[14] A. Rahman, "Dataset for Plant Disease Classification and Detection Using Deep Learning Techniques," *Journal of Agricultural Informatics*, vol. 11, no. 1, p. e5, 2020.

[15] D. S. Wita, "Deteksi Penyakit pada Tanaman Eucalyptus Pellita Menggunakan MobileNet," *International Journal of Plant Health*, vol. 10, no. 1, pp. 23–30, 2022.

[16] Mashuri and Enri, "Penerapan Inception V3 untuk Deteksi Penyakit pada Tanaman Gandum," *Journal of Agricultural Technology*, vol. 15, no. 4, pp. 145–156, 2022.

[17] E. Anggiratih, M. Z. Mustaqim, and D. Muriatmoko, "Transfer Learning untuk Klasifikasi Penyakit dan Hama Padi Menggunakan MobileNetV2," *Techno.Com*, vol. 22, no. 3, pp. 562–575, 2022, doi: 10.33633/tc.v22i3.8516.

[18] M. G. Pradana, H. B. Seta, N. Irzavika, P. H. Saputro, and R. Rusiyono, "Levenshtein Distance Algorithm in Javanese Character Translation Machine Based on Optical Character Recognition," *JOIV: International Journal on Informatics Visualization*, vol. 9, no. 4, p. 1411, Jul. 2025, doi: 10.62527/joiv.9.4.3151.

[19] D. Arisandi, A. Khoerunnisa, R. Subekti, A. Eko Setiawan, and C. Ramdani, "Performance Evaluation of CLAHE-Enhanced Edge Detection on Low-Light Faces," *Journal of Computing Innovations and Emerging Technologies*, vol. 1, no. 1, pp. 7–10, Jul. 2025, doi: 10.64472/jciet.v1i1.2.