Jurnal Teknologi Informasi dan Pendidikan
Volume 19, No. 1, March 2026
https://doi.org/10.24036/jtip.v19i1.1073

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

A Web Application for Classification and Detection of Tomato Leaf Diseases Using CNN and Yolo Models

Dony Novaliendry[1*✉], Fachri Rizal[1], Muhammad Anwar[1], Dedy Irfan[1]
[1]Faculty of Engineering, Universitas Negeri Padang, Padang, Indonesia
*Corresponding Author: fachrir680@gmail.com

Article Information

Article history:

No. 1073
Rec. October 30, 2025
Rev. February 14, 2026
Acc. February 14, 2026
Pub. February 18, 2026
Page. 1179 – 1192

Keywords:

- *tomato leaf disease*
- *CNN*
- *YOLO*
- *classification*
- *web application*

ABSTRACT

This study developed a web-based application for the classification and detection of tomato leaf diseases using Convolutional Neural Network (CNN) and You Only Look Once (YOLO) models. The research followed a Research and Development approach that consisted of requirement analysis, system design, implementation, model training, and testing. The CNN model was trained to classify tomato leaf images into specific disease categories, while the YOLO model was designed to detect and localize diseased areas in real time. Both models were integrated into a Flask-based web system to provide accessible and interactive functionality through standard web browsers. Testing results showed that the CNN model achieved an accuracy of 96.1%, effectively identifying disease types such as Early Blight and Bacterial Spot. The YOLO model reached a mean Average Precision (mAP) of 87.3% for real-time detection, successfully locating and labeling infected regions on tomato leaves. The integration of CNN and YOLO models demonstrated strong classification and detection performance, offering an efficient and scalable solution to support early disease diagnosis and digital transformation in precision agriculture.

How to Cite:

Novaliendry, D., & et al. (2026). A Web Application for Classification and Detection of Tomato Leaf Diseases Using CNN and Yolo Models. Jurnal Teknologi Informasi Dan Pendidikan, 19(1), 1179-1192. https://doi.org/10.24036/jtip.v19i1.1073

1. INTRODUCTION

Agriculture plays a crucial role in sustaining economic growth and food security in many developing countries, including Indonesia. As one of the primary sectors absorbing 28.61% of the national workforce in 2022, agriculture contributes significantly to national stability and resilience [1], [2]. However, crop productivity remains vulnerable to numerous biotic stresses, particularly plant diseases that cause substantial yield losses and economic impacts. Studies indicate that global crop losses due to pests and pathogens can reach 20–40% annually, posing a persistent challenge to food sustainability [3], [4].

Early and accurate disease detection is essential to prevent extensive crop damage and enable targeted treatment [5]. Nevertheless, traditional detection methods that rely on manual visual inspection remain dominant among farmers. These methods are often subjective, time-consuming, and prone to misdiagnosis due to variations in environmental conditions, lighting, and the similarity of symptoms among different diseases [6], [7]. Consequently, they are inefficient for large-scale agricultural applications and delay the implementation of effective control measures [8], [9], [10].

The selection of tomato plants as the primary focus of this research is driven by their immense economic value and high consumption rate both locally in Indonesia and globally. Tomato crops are particularly susceptible to a wide range of pathogenic infections, such as Early Blight and Bacterial Spot, which can rapidly diminish fruit quality and yield if not detected early. Unlike some other staple crops, the visual symptoms of tomato leaf diseases are often subtle and highly similar in their initial stages, making manual diagnosis exceptionally difficult for farmers. Therefore, focusing on tomato leaf diseases provides a critical testbed for deep learning models, as accurate classification and real-time detection in this specific crop can significantly mitigate substantial financial losses for smallholder farmers and support the broader goals of precision agriculture.

In response to these limitations, the agricultural sector is undergoing a digital transformation toward Agriculture 4.0, characterized by the adoption of advanced technologies such as Artificial Intelligence (AI), the Internet of Things (IoT), and Computer Vision [11], [12]. These innovations enable precision agriculture practices that improve efficiency and sustainability. Computer Vision, supported by deep learning models like Convolutional Neural Network (CNN) and You Only Look Once (YOLO), has demonstrated remarkable performance in classifying and detecting plant diseases based on visual features such as texture, color, and shape [13], [14], [15].

The CNN architecture excels at extracting spatial features from digital images, making it highly effective for disease classification tasks [16], [17], [18]. Meanwhile, YOLO offers real-time object detection capabilities, enabling the localization of infected regions on plant leaves with high accuracy [19], [20]. Integrating both models in a unified system provides a robust approach to disease analysis, combining classification accuracy with spatial awareness.

This research aims to develop a web-based system for the classification and detection of tomato leaf diseases using CNN and YOLO models. The system is designed to assist farmers and agricultural practitioners in identifying plant diseases more efficiently and objectively. Through this study, it is expected that AI-based Computer Vision technology

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

can contribute to enhancing agricultural productivity and supporting Indonesia's national agenda for digital transformation and food security.

2. RESEARCH METHOD

This research develops a web-based tomato leaf disease classification and detection system using Convolutional Neural Network (CNN) and You Only Look Once (YOLO) models [20], [21]. The development process follows the Research and Development (R&D) approach with stages of requirement analysis, system design, implementation, model training, and testing.

2.1. Requirement Analysis

The primary objective of this research is to design a web application that can classify and detect tomato leaf diseases automatically based on image input. The application provides two main functions:
Classification mode, which uses the CNN model to classify tomato leaf images into disease categories such as bacterial spot, early blight, septoria leaf spot, and leaf healthy [5].

Detection mode, which employs the YOLO model to identify and localize the diseased areas in real time [19].

The system is designed to assist farmers and agricultural practitioners in identifying tomato leaf diseases efficiently. Data used for model training and validation are derived from open-access public datasets, including the PlantVillage and PlantDoc repositories, to ensure both controlled and field-condition variability [9].

2.2. Dataset and Data Preprocessing

The primary dataset for this research was curated from two reputable open-access repositories, namely the PlantVillage and PlantDoc datasets, which are publicly available for academic and research purposes. A total of 3,251 tomato leaf images were utilized in this study, consisting of 1,200 images for healthy leaves and 2,051 images representing various disease categories, including Bacterial Spot, Early Blight, and Septoria Leaf Spot. These images were obtained by downloading the raw data directly from the official Kaggle and GitHub repositories associated with the original studies. The inclusion of both PlantVillage and PlantDoc ensures a diverse collection of data, encompassing both high-quality images from controlled laboratory environments and challenging real-world field conditions with complex backgrounds.

The dataset contains tomato leaf images categorized as healthy and diseased. To improve model generalization, preprocessing includes resizing images to 224×224 pixels for the CNN model and 640×640 for YOLO, normalizing pixel values to the [0,1] range, and applying augmentation such as rotation, flipping, zooming, and brightness adjustments [7].

Bounding boxes are annotated in Roboflow for YOLO training, and the dataset is split into 80% training, 10% validation, and 10% testing to ensure balanced performance evaluation [12].

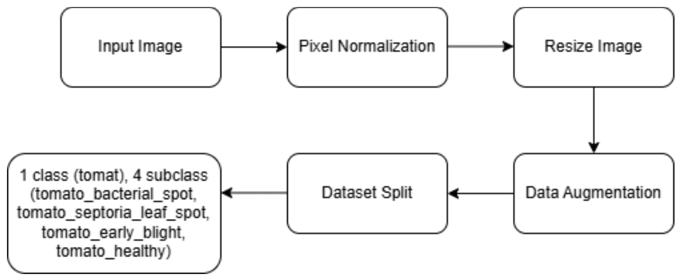
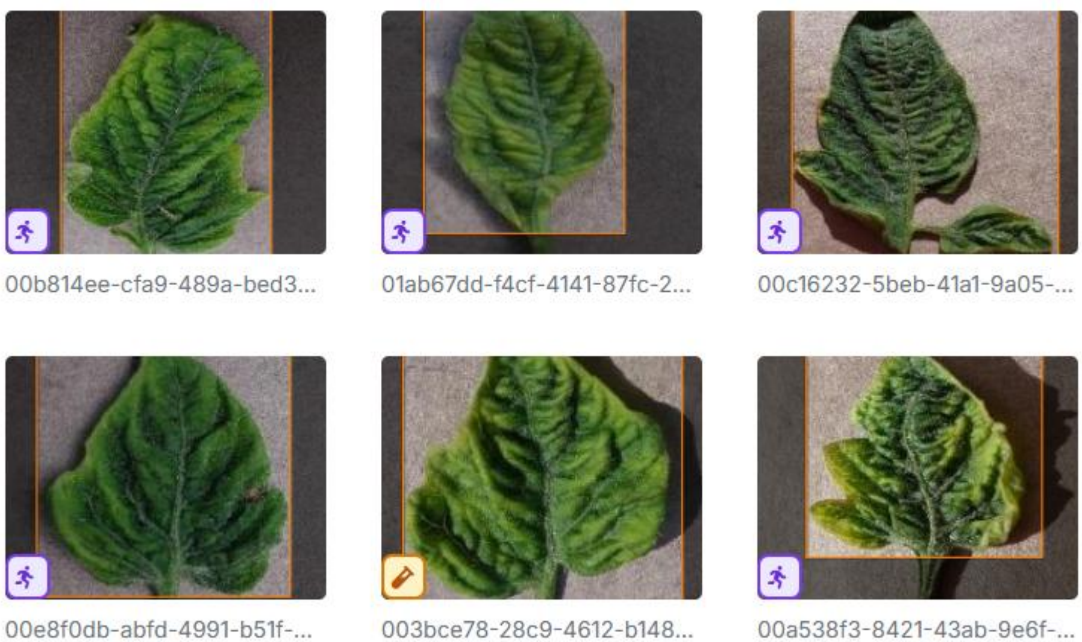


Figure 1. Dataset Preprocessing Workflow for Tomato Leaf Images

The annotation process for the YOLO model was performed manually using the Roboflow platform, where bounding boxes were meticulously drawn around diseased areas to ensure precise localization. To address class imbalances within the combined dataset, we employed specific data augmentation techniques for underrepresented classes, such as localized brightness adjustments and synthetic oversampling. These measures ensured that the model received sufficient exposure to rare disease variants, preventing a bias toward more prevalent healthy samples.



https://doi.org/10.24036/jtip.v19i1.1073

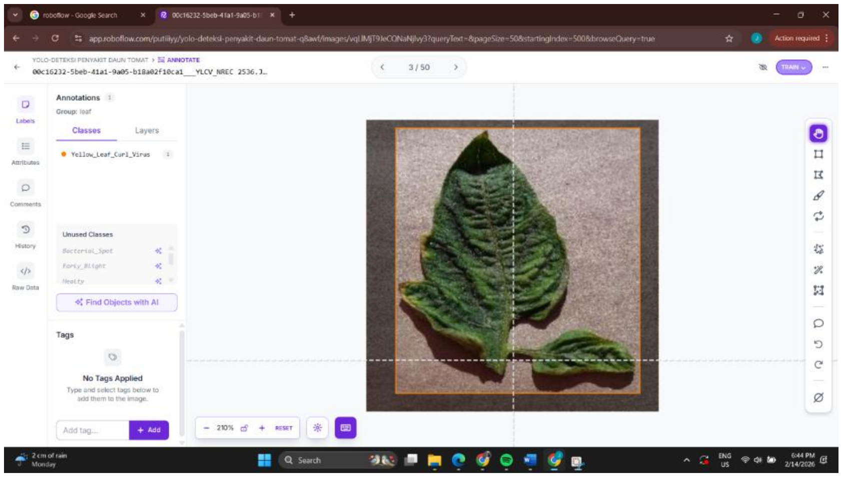


Figure 2. The manual annotation workflow: (a) Grid overview of multiple annotated samples representing dataset diversity, and (b) Close-up view of the annotation process for specific disease identification, such as Yellow Leaf Curl Virus, ensuring high-quality ground truth for model training.

2.3. System Design

2.3.1. Model Training Architecture

In this phase, CNN and YOLO models are trained separately using preprocessed images. CNN is responsible for feature extraction and disease classification, while YOLO handles object detection and localization [15].

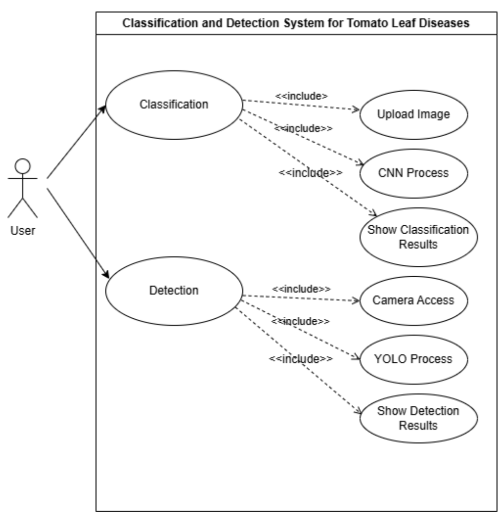


Figure 3. Use Case Diagram for Plant Leaf Disease Detection System

2.3.2. Web Application

The trained models are integrated into a Flask-based web application. The system workflow begins when a user uploads or captures a tomato leaf image. The image is processed and analyzed by the corresponding deep learning model, and the results are displayed as either a disease classification label or bounding boxes marking infected areas.

The overall system is designed to be lightweight, responsive, and accessible via standard web browsers on desktop and mobile platforms.

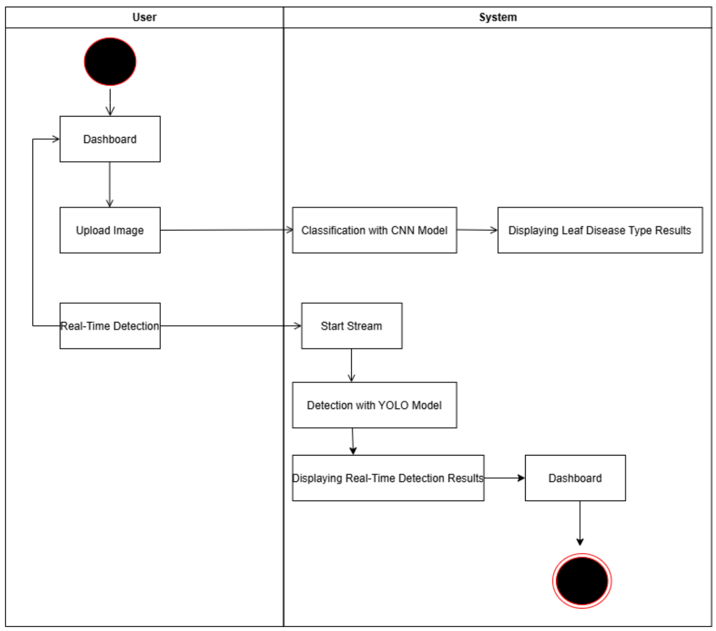


Figure 4. Web Activity Diagram of Plant Leaf Disease Detection System

2.4. Model Development

The CNN model for classification is developed using the MobileNetV2 architecture, which employs depthwise separable convolutions to reduce computational complexity while maintaining high feature extraction efficiency. This process involves several key stages, including convolution layers for spatial feature detection, pooling layers for dimensionality reduction, and fully connected layers that utilize a Softmax activation function to categorize images into specific disease classes. Simultaneously, the YOLO (You Only Look Once) algorithm is implemented for real-time detection by treating object detection as a single regression problem. The YOLO framework operates by dividing the input image into an $S\ X\ S\ grid$, where each grid cell simultaneously predicts multiple bounding boxes and confidence scores for the detected diseased regions. By processing the entire image in a single forward pass of the network, YOLO ensures minimal latency and

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

high inference speed, which is essential for dynamic field monitoring through the web application's camera interface.

2.4.1. CNN Model for Classification

The CNN model is implemented using TensorFlow and Keras frameworks. A pre-trained architecture such as MobileNetV2 is fine-tuned using transfer learning [22]. The model uses Adam optimizer with a categorical cross-entropy loss function and an early stopping callback to prevent overfitting [17].

Evaluation metrics include accuracy, precision, recall, and F1-score, calculated from the confusion matrix during testing.

2.4.2. YOLO Model for Detection

The YOLO model is trained using the annotated dataset to detect diseased regions on tomato leaves. The model divides the image into grids and predicts bounding boxes and class probabilities simultaneously, enabling real-time detection [19], [20].
Hyperparameters such as batch size, learning rate, and IoU threshold are adjusted to optimize performance. Evaluation is based on mean Average Precision (mAP) and inference speed (frames per second) [15].

2.5. Implementation

The web application is developed using the Flask framework as the backend [23]. The trained CNN and YOLO models are stored as serialized files (.h5 and .pt) and loaded into the server for inference. The frontend is built using HTML, CSS, and JavaScript, providing interactive pages for classification and detection.

The system supports image uploads and real-time detection using a device camera. All outputs, including prediction labels and bounding boxes, are rendered dynamically through Flask endpoints.

2.6. Testing and Evaluation

The testing phase is conducted to verify that all primary functions of the system operate according to the predefined design specifications. This study employs the Black Box Testing method to evaluate two core features: Classification Testing, which assesses the CNN model's accuracy in identifying disease types, and Detection Testing, which evaluates the YOLO model's precision in localizing diseased regions. To rigorously measure the performance of these features, several standard evaluation metrics are utilized, including Accuracy, Precision, Recall, and F1-score. Accuracy measures the proportion of correct predictions among the total cases, calculated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision indicates the model's ability to correctly identify positive instances, defined as:

$$Precision = \frac{TP}{TP + FP}$$

Meanwhile, Recall reflects the ability to detect all actual positive cases, expressed as:

$$Recall = \frac{TP}{TP + FN}$$

The F1-score provides a balanced evaluation as the harmonic mean of precision and recall:

$$F1 - score = 2\ x\ \frac{Precision\ x\ Recall}{Precision + Recall}$$

For the YOLO detection model, the Mean Average Precision (mAP) is specifically employed to measure accuracy in localizing diseased regions by considering the Intersection over Union (IoU) threshold across different classes. This comprehensive testing approach ensures the reliability and effectiveness of the integrated diagnostic system for practical agricultural use.

2.7. Deployment

After successful implementation and testing, the web application is deployed on a local server for demonstration. The architecture allows further cloud deployment to enable online access without additional installation.

3. RESULTS AND DISCUSSION

3.1. System Integration and Web Deployment

The final implementation successfully integrated both the CNN and YOLO models into a single web-based system using the Flask framework as the backend. The integration process ensured that both classification and detection modules operated cohesively under a unified user interface. The system was designed with an emphasis on usability, accessibility, and responsiveness, allowing users to interact with the application seamlessly through standard web browsers on both desktop and mobile platforms.

The web interface consists of three primary components: a dashboard, a classification page, and a detection page. The dashboard provides introductory information about the system, including a concise guide on how to operate the web application. Although this component does not directly influence the model's performance, it plays a crucial role in user experience by introducing non-technical users—such as farmers and agricultural technicians—to the system's core functions.

The classification page integrates the trained CNN model, enabling users to upload tomato leaf images in standard formats such as JPG, JPEG, and PNG, with a maximum image size of 5 MB. Once an image is uploaded, it is automatically preprocessed and classified into one of the predefined categories: *Bacterial Spot*, *Early Blight*, *Septoria Leaf Spot*, or *Healthy Leaf*. The detection page, on the other hand, incorporates the YOLO model to facilitate real-time analysis via a device's camera. When the camera is activated, the model continuously detects and localizes diseased regions on the captured frames, marking them with bounding boxes and confidence scores.

Through this integration, the system effectively combines deep learning-based computer vision with web technology, providing a scalable and interactive solution for agricultural diagnostics.

3.2. Model Testing and Performance Analysis

The testing phase aimed to assess both the accuracy and efficiency of the CNN and YOLO models after deployment. Black Box Testing was used to verify that the system's functionalities operated as intended, including classification accuracy, detection performance, and real-time response.

Table 1. CNN Model Performance

Metric	Score
Accuracy	0.96
Precision	0.96
Recall	0.96
F1-score	0.96

The CNN model achieved an overall accuracy of 96.1%, indicating strong reliability in classifying tomato leaf diseases. Precision, recall, and F1-score all reached 0.96, demonstrating that the model was able to consistently identify the correct disease class with minimal false positives or false negatives. This performance can be attributed to the use of transfer learning from the MobileNetV2 architecture, which allowed efficient feature extraction while avoiding overfitting through early stopping and data augmentation.

Table 2. YOLO Model Performance

Metric	Score
mAP50	0.995
mAP50-95	0.873
Precision	0.997
Recall	0.998

Jurnal Teknologi Informasi dan Pendidikan
Volume 19, No. 1, March 2026
https://doi.org/10.24036/jtip.v19i1.1073

Meanwhile, the YOLO detection model achieved a mean Average Precision (mAP50) of 99.5% and a mAP50–95 of 87.3%, with precision and recall values close to 1.0. These results confirm that the detection module performs exceptionally well in identifying and localizing disease regions even under varying lighting and image quality conditions. The model's lightweight architecture also ensures real-time inference, with minimal latency between frame capture and detection output.

Furthermore, the robustness of both models was evaluated under diverse environmental factors, including varying lighting conditions and different leaf orientations. By integrating the PlantDoc dataset, which contains images captured in actual field conditions, the evaluation moved beyond controlled laboratory setups. The results indicate that the YOLO model maintains a high mAP even when faced with complex backgrounds and shadows, confirming its readiness for real-world agricultural monitoring.

3.3. Result of Tomato Leaf Disease Classification Testing

The classification testing was conducted to validate the CNN model's predictive capability within the deployed web environment. Two representative trials were performed using different tomato leaf samples.

In the first trial, the model correctly classified the uploaded image as Early Blight with a confidence score of 99.88%, while in the second trial, the model identified the leaf as Bacterial Spot with a confidence of 97.59%. These outcomes demonstrate that the CNN model can effectively capture and interpret visual cues such as color patterns, texture variations, and lesion morphology. The strong confidence levels further confirm the robustness of the model against intra-class variability, meaning it can distinguish between diseases with similar visual symptoms.

The classification outputs were displayed dynamically on the website, allowing users to receive immediate diagnostic feedback along with the confidence percentage. This capability is particularly valuable for field practitioners who require quick and reliable assessments without relying on laboratory analysis.

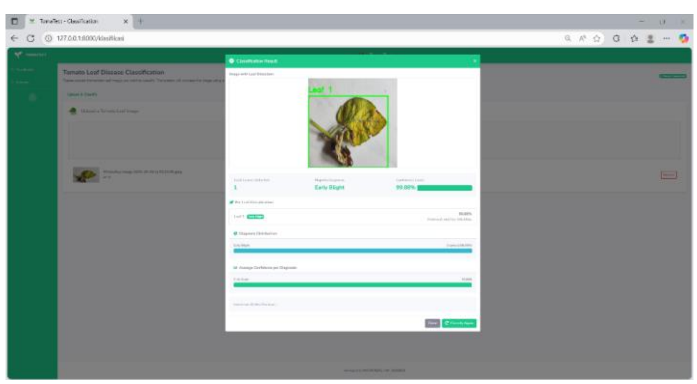


Figure 4. Result of Tomato Leaf Disease Classification Test 1 (Early Blight, 99.88%)

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

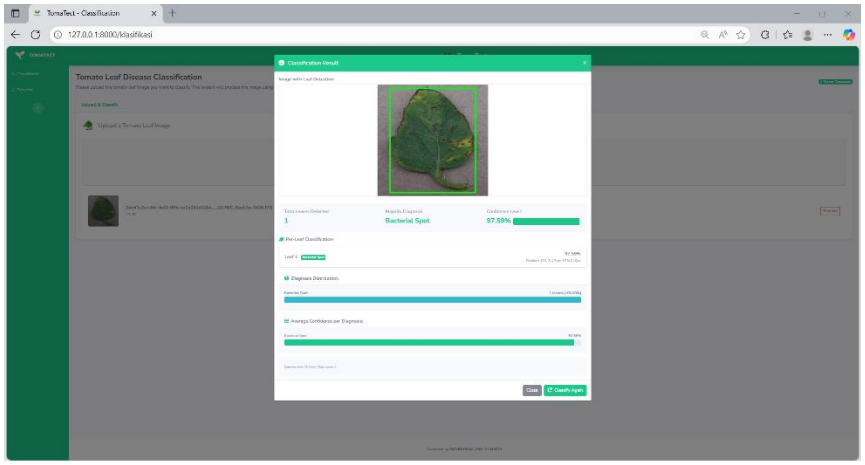


Figure 5. Result of Tomato Leaf Disease Classification Test 2 (Bacterial Spot, 97.59%)

3.4. Real-Time Tomato Leaf Disease Detection Test Results

The YOLO model was further evaluated through real-time detection experiments using a live camera feed. This test aimed to assess the system's ability to detect diseases dynamically as the camera captured tomato leaves in different orientations and lighting conditions.

The first experiment successfully detected Septoria Leaf Spot with a confidence score of 84.47%, while the second identified Bacterial Spot with 79.45% confidence. Bounding boxes were automatically rendered around the infected areas, accompanied by the corresponding disease name and confidence percentage. These detections occurred with minimal delay, confirming that the YOLO model can perform real-time inference effectively in practical conditions.

The combination of CNN and YOLO provides a synergistic diagnostic framework: CNN ensures categorical precision in disease classification, whereas YOLO delivers spatial localization of infected regions. This dual approach enhances both interpretability and practicality, offering a holistic understanding of disease conditions rather than a simple categorical label.

Volume 19, No. 1, March 2026
https://doi.org/10.24036/jtip.v19i1.1073

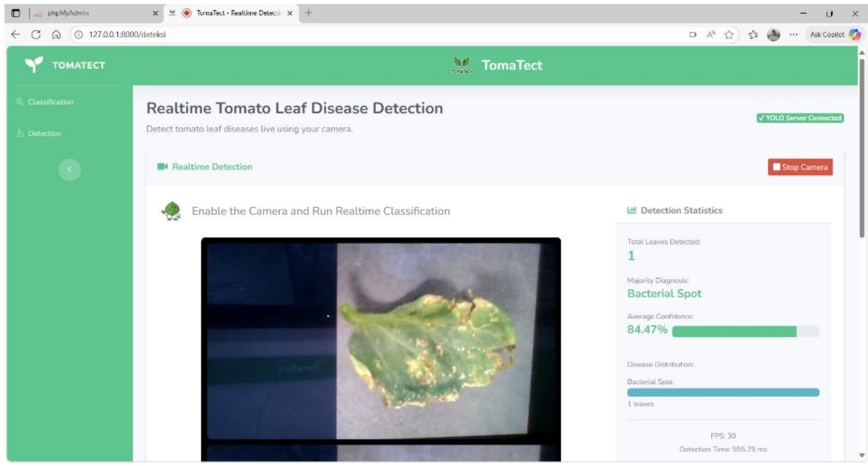


Figure 6. Result of Real-Time Detection Test 1 (Bacterial Spot, 84.47%)

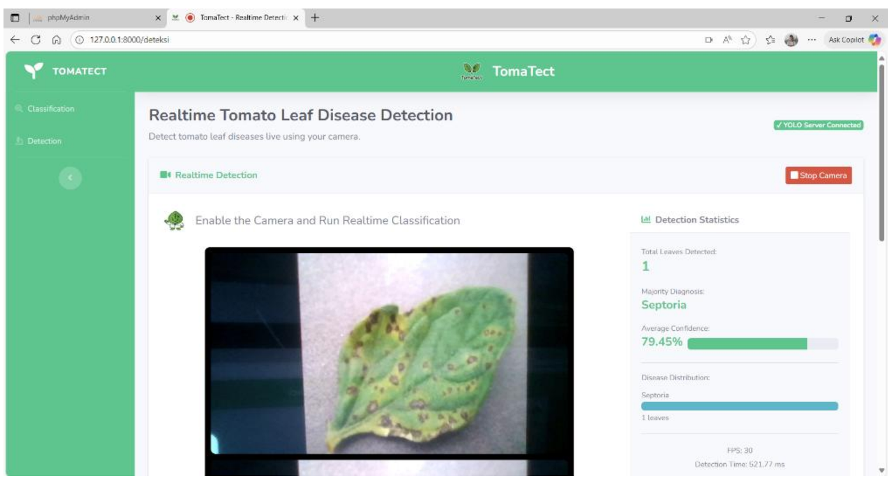


Figure 7. Result of Real-Time Detection Test 2 (Septoria, 79.45%)

4. CONCLUSION

The results confirm that integrating CNN and YOLO in a unified web environment creates an efficient and scalable system for plant disease diagnostics. The CNN's high classification accuracy demonstrates its ability to generalize across varying dataset conditions, including both controlled laboratory images and field data. Meanwhile, the YOLO model's real-time detection capability highlights its suitability for on-site monitoring, enabling early and proactive disease management.

Compared to traditional manual inspection, this AI-based approach significantly reduces diagnosis time and subjectivity, providing quantitative, repeatable, and interpretable results. The web-based design also removes the need for local installations, making the system accessible to a wider user base, including farmers in remote areas.

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

To further enhance the system's robustness, future research will focus on incorporating more diverse datasets from varying geographical locations to account for regional environmental variations. Plans are also in place to explore the integration of transformer-based architectures and self-supervised learning to improve the model's ability to generalize across unpredictable real-world field conditions without requiring extensive manual annotations.

REFERENCES

[1] A. P. Siregar *et al.*, "The Trend of Agricultural Sector Resilience in Indonesia During 2008-2020," *J. Agric. Sci. - Sri Lanka*, vol. 19, no. 2, pp. 336–357, 2024, doi: 10.4038/jas.v19i2.10154.

[2] D. Johan, M. S. Maarif, N. Zulbainarni, and B. Yulianto, "Agricultural Digitalization In Indonesia: Challenges and Opportunities for Sustainable Development," *Educ. Adm. Theory Pract.*, vol. 30, no. 7, pp. 640–648, 2024, doi: 10.53555/kuey.v30i7.6599.

[3] "FAO's Plant Production and Protection Division," *FAO's Plant Prod. Prot. Div.*, 2022, doi: 10.4060/cc2447en.

[4] J. B. Ristaino *et al.*, "The persistent threat of emerging plant disease pandemics to global food security," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 118, no. 23, pp. 1–9, 2021, doi: 10.1073/pnas.2022239118.

[5] C. Nguyen, V. Sagan, M. Maimaitiyiming, M. Maimaitijiang, S. Bhadra, and M. T. Kwasniewski, "Early detection of plant viral disease using hyperspectral imaging and deep learning," *Sensors (Switzerland)*, vol. 21, no. 3, pp. 1–23, 2021, doi: 10.3390/s21030742.

[6] S. U. Khan, A. Alsuhaibani, A. Alabduljabbar, F. Almarshad, Y. N. Altherwy, and T. Akram, *A review on automated plant disease detection: motivation, limitations, challenges, and recent advancements for future research*, vol. 37, no. 3. Springer International Publishing, 2025. doi: 10.1007/s44443-025-00040-3.

[7] M. U. I. Tamim, S. A. Hamim, S. Malik, M. F. Mridha, and S. Mahmood, "InsightNet: A Deep Learning Framework for Enhanced Plant Disease Detection and Explainable Insights," *Plant Direct*, vol. 9, no. 5, 2025, doi: 10.1002/pld3.70076.

[8] F. O. Isinkaye, M. O. Olusanya, and P. K. Singh, "Deep learning and content-based filtering techniques for improving plant disease identification and treatment recommendations: A comprehensive review," *Heliyon*, vol. 10, no. 9, p. e29583, 2024, doi: 10.1016/j.heliyon.2024.e29583.

[9] M. Xu, J. E. Park, J. Lee, J. Yang, and S. Yoon, "Plant disease recognition datasets in the age of deep learning: challenges and opportunities," *Front. Plant Sci.*, vol. 15, no. September, pp. 1–13, 2024, doi: 10.3389/fpls.2024.1452551.

[10] A. Y. Ashurov *et al.*, "Enhancing plant disease detection through deep learning: a Depthwise CNN with squeeze and excitation integration and residual skip connections," *Front. Plant Sci.*, vol. 15, no. January, pp. 1–16, 2024, doi: 10.3389/fpls.2024.1505857.

[11] D. E. Dewi, P. N. A. Cahyani, and L. R. Megawati, *Increasing Adoption of the Internet of Things in Indonesian Agriculture Based on a Review of Everett Rogers' Diffusion Theory of Innovation*, vol. 1, no. Un 2019. Atlantis Press International BV, 2023. doi: 10.2991/978-94-6463-144-9_29.

[12] W. Wang and Y. Kang, "A Review of Computer Vision Technologies in Precision Agriculture,"

vol. 0, pp. 35–40, 2025, doi: 10.54254/2753-8818/2025.CH22224.
[13] A. Jafar, N. Bibi, R. A. Naqvi, A. Sadeghi-Niaraki, and D. Jeong, "Revolutionizing agriculture with artificial intelligence: plant disease detection methods, applications, and their limitations," *Front. Plant Sci.*, vol. 15, no. March, pp. 1–20, 2024, doi: 10.3389/fpls.2024.1356260.
[14] I. A. M. Dioses, P. N. Santiago, R. P. Pascual, R. Dellosa, J. L. Dioses, and J. B. A. Tababa, "Detection of Philippine Rice Plant Diseases: A ResNet50 CNN Approach," *2024 IEEE 15th Control Syst. Grad. Res. Colloquium, ICSGRC 2024 - Conf. Proceeding*, no. August, pp. 256–260, 2024, doi: 10.1109/ICSGRC62081.2024.10690946.
[15] A. Tripathi, V. Gohokar, and R. Kute, "Comparative Analysis of YOLOv8 and YOLOv9 Models for Real-Time Plant Disease Detection in Hydroponics," *Eng. Technol. Appl. Sci. Res.*, vol. 14, no. 5, pp. 17269–17275, 2024, doi: 10.48084/etasr.8301.
[16] S. V. Sinha and B. M. Patil, "INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING Comparative Analysis of CNN, EFFICIENTNET and RESNET for Grape Disease Prediction : A Deep Learning Approach," 2024.
[17] F. Zubair, M. Saleh, Y. Akbari, and S. Al Maadeed, "A Robust Ensemble Model for Plant Disease Detection Using Deep Learning Architectures," *AgriEngineering*, vol. 7, no. 5, 2025, doi: 10.3390/agriengineering7050159.
[18] M. A. A. Khandaker, Z. S. Raha, S. Islam, and T. Muhammad, "Explainable AI-Enhanced Deep Learning for Pumpkin Leaf Disease Detection: A Comparative Analysis of CNN Architectures," *2024 27th Int. Conf. Comput. Inf. Technol. ICCIT 2024 - Proc.*, no. December, pp. 2428–2433, 2024, doi: 10.1109/ICCIT64611.2024.11021957.
[19] A. Raj *et al.*, "YOLO-ODD: an improved YOLOv8s model for onion foliar disease detection," *Front. Plant Sci.*, vol. 16, no. May, pp. 1–18, 2025, doi: 10.3389/fpls.2025.1551794.
[20] L. T. Ramos and A. D. Sappa, "A Decade of You Only Look Once (YOLO) for Object Detection," pp. 1–39, 2025, [Online]. Available: http://arxiv.org/abs/2504.18586
[21] D. Sasmoko and E. Siswanto, "Systematic Literature Review on CNN and YOLO Algorithms for Detecting Plant Diseases in Precision Agriculture," 2025.
[22] Q. Yan, B. Yang, W. Wang, B. Wang, P. Chen, and J. Zhang, "Apple leaf diseases recognition based on an improved convolutional neural network," *Sensors (Switzerland)*, vol. 20, no. 12, pp. 1–14, 2020, doi: 10.3390/s20123535.
[23] F. Rahman, A. Hadi, D. Novaliendry, and A. Dwinggo Samala, "A Web-Based SIBI Sign Language Translator Application with Speech-to-Text Feature Using CNN and MediaPipe," *J. Teknol. Inf. dan Pendidik.*, vol. 18, no. 2, pp. 1033–1042, 2025, doi: 10.24036/jtip.v18i2.971.