Advance Sustainable Science, Engineering and Technology (ASSET)
Vol. 7, No.1, January 2025, pp. 02501019-01 ~ 02501019-015
ISSN: 2715-4211 DOI: https://doi.org/10.26877/asset.v7i1.1152

Development of a Robotic System for Agricultural Pest Detection: A Case Study on Chili Plants

Nur Sultan Salahuddin, Fathi Muthia Tarie*, Trini Saptariani

Faculty of Computer Science and Information Technology, Universitas Gunadarma, Jl. Margonda Raya No.100, West Java, Indonesia

*fathimtari55@gmail.com

Abstract. Chili peppers, a key agricultural commodity in Indonesia, are highly susceptible to pest infestations and diseases, leading to significant economic losses and challenges in sustainable farming. This study presents the design and implementation of a real-time pest detection system that integrates robotics, computer vision, and deep learning to enhance agricultural productivity. The system is built on a Raspberry Pi 5 and Arduino Mega Pro Mini, utilizing a camera for image capture and ultrasonic sensors for navigation. A ResNet-based model was trained on a dataset of 2,703 chili leaf images, categorized into healthy and diseased classes, achieving a detection accuracy of 91%. The system provides early warnings to farmers through a web-based interface, allowing timely intervention and reducing reliance on chemical pesticides. While promising, the system faced challenges such as environmental variability, which influenced image recognition accuracy. By automating pest detection and promoting precision farming, this innovation addresses the need for sustainable agricultural practices, contributing to global food security and reducing environmental impact.

Keywords: ResNet, Pest Detection, Chili Plants, Computer Vision, Raspberry Pi, Machine Learning, Sustainable Agriculture, AI in Agriculture,

(Received 2024-10-25, Accepted 2024-12-31, Available Online by 2024-12-31)

1. Introduction

Chili plants (Capsicum annuum L.) are a crucial agricultural commodity in Indonesia, cultivated widely for both domestic consumption and export [7]. However, chili farming faces significant challenges from pests and diseases, including leaf curl, powdery mildew, and pest infestations such as whitefly, which directly impact crop yield and quality [14][17]. These challenges lead to economic setbacks, particularly when early signs of infestation are not detected in time. Traditionally, pest and disease detection relies on manual inspection, a labor-intensive and error-prone process, especially over large agricultural areas [1][11].

Recent advancements in artificial intelligence (AI) and computer vision have revolutionized pest and disease detection, offering more efficient and accurate methods. Machine learning (ML), particularly deep learning, has shown great potential in automating image classification tasks, enabling the recognition of diseases and pests through plant leaf images [5][24]. Among the various deep learning models, Residual Networks (ResNet) have emerged as a leading architecture due to their ability to handle complex image data and deliver high classification accuracy [5]. The use of ResNet for pest and disease detection in agriculture is a promising application of AI, addressing critical issues of sustainability in farming by minimizing the need for manual labor and reducing pesticide use.

This research focuses on developing a mobile, real-time pest detection system for chili plants, leveraging ResNet for leaf image classification. The system, built on a Raspberry Pi 5 and integrated with an Arduino Mega Pro Mini for motor control and sensor management, captures images of plant leaves using a camera and employs ultrasonic sensors for navigation, ensuring minimal crop disturbance. Real-time analysis of the images allows for early detection of pest infestations, providing farmers with timely feedback through a web-based interface [18].

In line with recent trends in AI-powered agricultural tools, this system not only improves pest detection for chili plants but is also adaptable to other crops such as tomatoes, eggplants, and cucumbers. By incorporating a database of pest and disease information, the robot can be expanded to identify infestations in various crops, thereby enhancing the system's versatility. This capability aligns with sustainable agricultural practices by promoting early intervention, reducing the need for excessive pesticide use, and supporting farmers with data-driven solutions for crop protection [26].

Through this approach, the research contributes to the broader goal of sustainable agriculture, demonstrating how AI and robotics can be harnessed to improve crop health, reduce environmental impact, and increase productivity.

2. Methods

In this research, there are research stages which are divided into several stages, namely identifying problems, literature study, observation and determination of case studies, data collection, system design, tool assembly, testing, and evaluation and conclusions

2.1. Hardware Design

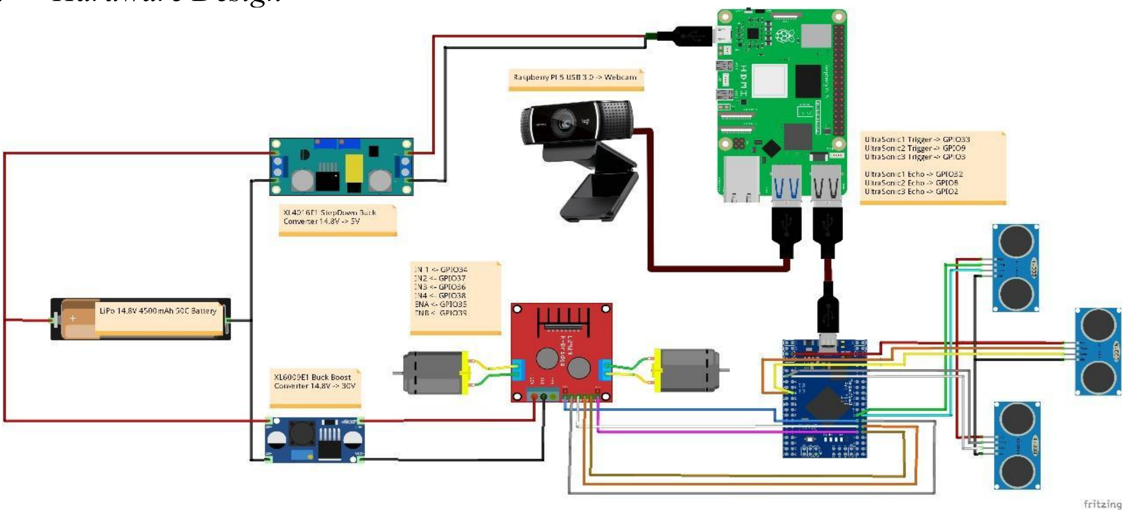


Figure 1. Hardware Design

The system uses a 14.8V 4500mAh 30C LiPo battery as the main power source. To meet the different voltage requirements of the components, an XL401611 StepDown Buck converter reduces the battery voltage from 14.8V to 5V for the Raspberry Pi, while an XL6009E1 Buck Boost converter increases the battery voltage from 14.8V to 24V to power the motors. The Raspberry Pi 5 acts as the primary microcontroller, connected to a webcam via a USB port for image capture and processing. It is also linked to the motor module through its GPIO pins.

The system employs three ultrasonic sensors to detect objects in front of the robot. Ultrasonic Sensor 1 is connected to GPIO33 (trigger) and GPIO32 (echo) on the Raspberry Pi, Sensor 2 is connected to GPIO9 (trigger) and GPIO8 (echo), and Sensor 3 is connected to GPIO5 (trigger) and GPIO2 (echo). For actuation, an L298N motor module controls two DC motors. Motor 1 is connected to the Raspberry Pi via IN1 (GPIO34), IN2 (GPIO37), ENA (GPIO35), and INB (GPIO35), while Motor 2 is connected via IN3 (GPIO36), IN4 (GPIO38), ENA (GPIO35), and INB (GPIO35).

The wiring layout is color-coded for clarity: red wires indicate positive connections, black wires represent negative connections, yellow wires are assigned to Motor 1, green wires to Motor 2, blue wires to Ultrasonic Sensor 1, purple wires to Ultrasonic Sensor 2, and brown wires to Ultrasonic Sensor 3. This configuration ensures efficient operation and control of the robot's movement and object detection capabilities.

2.2. System Block Diagram

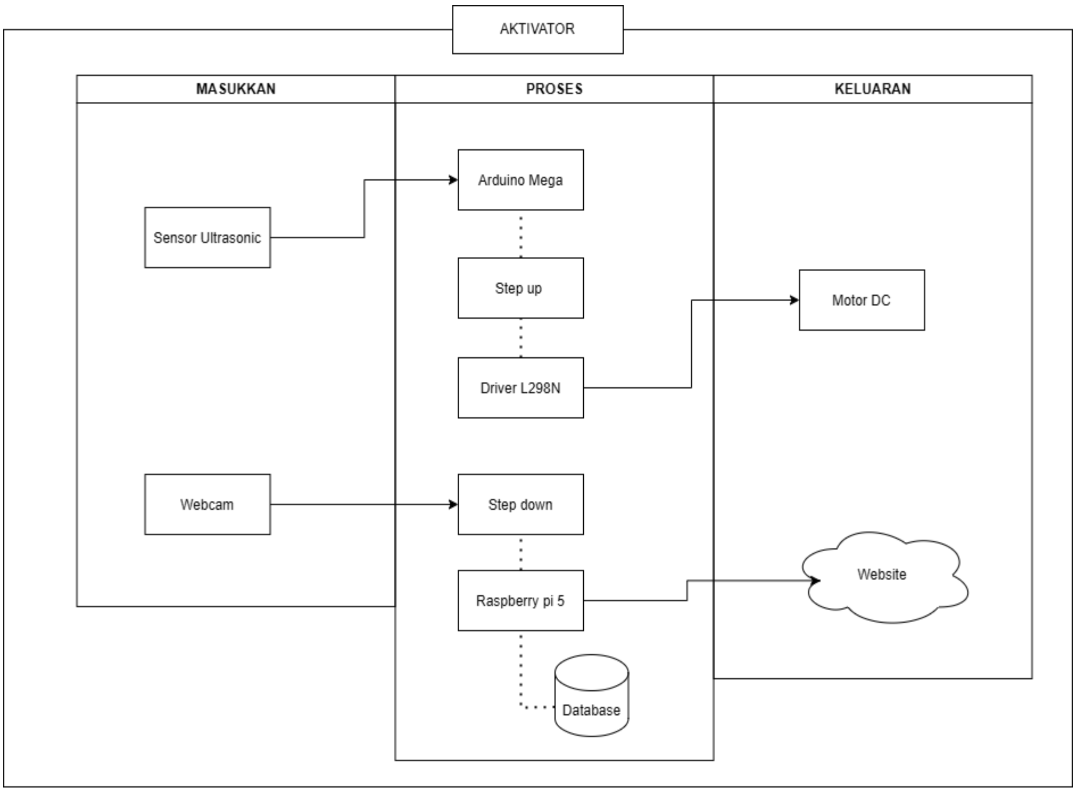


Figure 2. Block Diagram

In figure 1, the system architecture for the pest detection robot consists of three main sections: input, process, and output. In the input section, the ultrasonic sensor is used to detect obstacles in the robot's environment. It sends distance measurements to the Arduino Mega, allowing the robot to navigate the field without colliding with objects. Additionally, the system includes a webcam that captures images of chili plant leaves. These images are processed in real-time for pest and disease detection using a Raspberry Pi 5.

In the process section, the Arduino Mega acts as the main controller for the robot's motor and sensor operations. It receives data from the ultrasonic sensor and controls the movement of the motors via a motor driver (L298N). It also manages the power distribution using step-up and step-down circuits, which regulate voltage according to the system's needs. The step-up circuit increases the voltage for components like motors, while the step-down circuit reduces voltage for more sensitive components, such as the Raspberry Pi 5. The L298N motor driver is responsible for controlling the DC motors that enable the robot to move and adjust its direction based on input from the sensors. The Raspberry Pi 5 handles more complex tasks, such as processing the images captured by the webcam. By utilizing machine learning algorithms like ResNet, the Raspberry Pi performs image recognition to detect pests or diseases. The results are stored in a database and sent to a web-based platform.

In the output section, the movement of the robot is controlled by the DC motors, which are operated through the motor driver. The system also provides real-time feedback to the user via a website interface. Farmers can remotely access this website to monitor the health of their crops and receive information on pest infestations. Additionally, a database is used to store image data and pest detection results for further analysis and future reference. This system offers an efficient and technology-driven solution for automating pest detection and improving agricultural productivity

2.3. System workflow

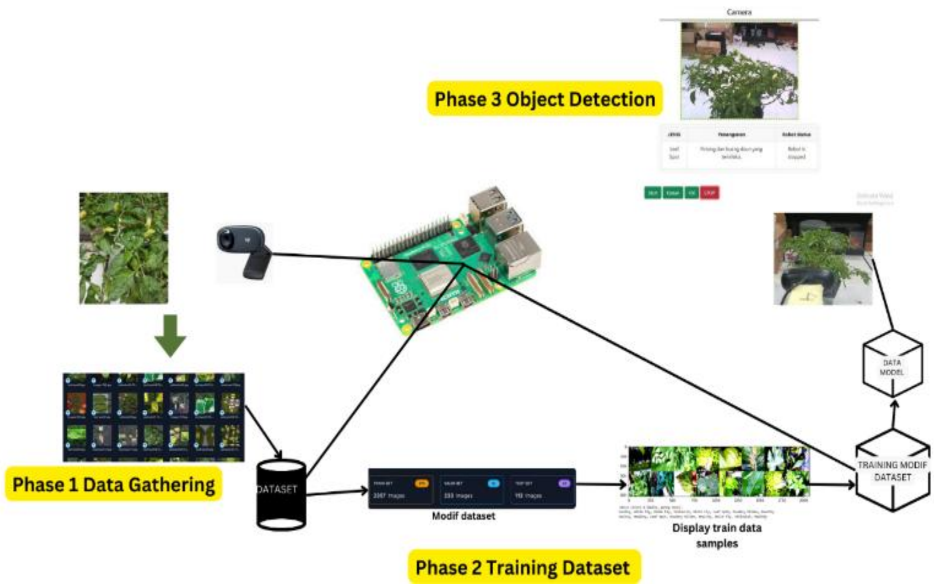


Figure 3. Workflow System

The image illustrates a three-phase workflow for developing an object detection system using a Raspberry Pi and camera. It begins with Phase 1: Data Gathering, where images are collected using the camera connected to the Raspberry Pi, capturing various objects of interest, such as leaves, and storing them in a dataset. In Phase 2: Training Dataset, the collected data is processed to prepare it for training a machine learning model, which may involve labeling the images, augmenting the dataset, or splitting it into training and validation sets. The model is then trained to recognize patterns and objects based on the sample images. Finally, in Phase 3: Object Detection, the trained model is deployed for real-time object detection. The Raspberry Pi, equipped with the camera, detects and identifies objects in the environment, displaying the results on a screen with the detected objects and their labels. This workflow demonstrates an iterative process of gathering data, training a model, and deploying the system for practical object detection tasks.

2.4. Dataset and Modelling

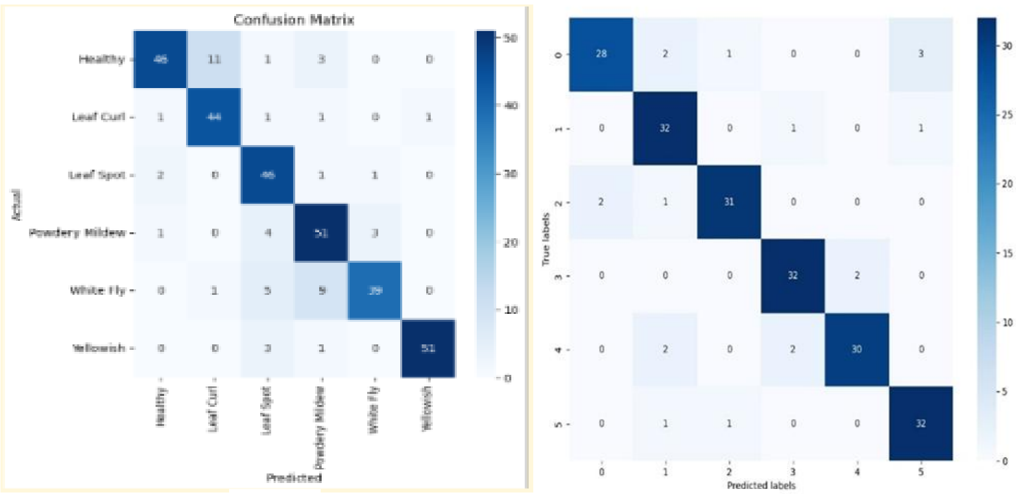
The dataset contains images of plant leaves classified into two main categories: "Healthy" and "Diseased." The "Healthy" category includes 340 images, while the "Diseased" category is divided into five classes: Leaf Curl, Leaf Spot, Powdery Mildew, White Fly, and Yellowish, each with 340 images. This results in a total of 2,703 images, with a nearly uniform distribution across all classes.

The dataset of 2,703 images was divided into three sets:

- Training Set : 1,305 images for model training.
- Validation Set : 327 images to evaluate model performance during training.
- Test Set : 204 images to assess the model's ability to generalize to new data.

Data cleaning enhanced classification accuracy, particularly for the "Healthy" class, by adding 90 images to increase data variability. This process reduced classification errors and improved correct predictions across all classes, as shown in the confusion matrices before and after data cleaning.

Data transformation included two steps: preprocessing and augmentation. Images were resized to 256x256 pixels, normalized, and converted into tensors for model compatibility. Augmentation involved random vertical and horizontal flips, and Gaussian Blur to reduce overfitting and increase data variability.



a b

Figure 4. Data Cleaning Results (a) Before training (b) After training

Figure 4 have two confusion matrices from the classification results before and after data cleaning. The confusion matrix on the left represents the state before data cleaning, where several classes depict plant conditions such as Healthy, Leaf Curl, Leaf Spot, Powdery Mildew, and White Fly. Each cell in the matrix shows the number of correct and incorrect predictions for each class. The main diagonal, stretching from the top left to the bottom right, indicates the number of correct predictions that match the actual labels. In contrast, the cells outside the main diagonal represent misclassifications, where the model incorrectly classified one class as another. Before data cleaning, some classes had a higher number of incorrect predictions, particularly Leaf Spot and Powdery Mildew, which were often predicted as other classes.

On the right, the confusion matrix after data cleaning shows an improvement in classification performance. More values appear on the main diagonal, indicating an increase in the number of correct predictions for almost all classes. Additionally, the number of misclassifications outside the main diagonal significantly decreased, suggesting that the model became better at distinguishing between classes after the data cleaning process. For example, the Leaf Curl and White Fly classes experienced an increase in the number of correct predictions after data cleaning. Overall, the model appears more accurate and precise in predicting each class, thanks to a cleaner and more balanced data distribution.

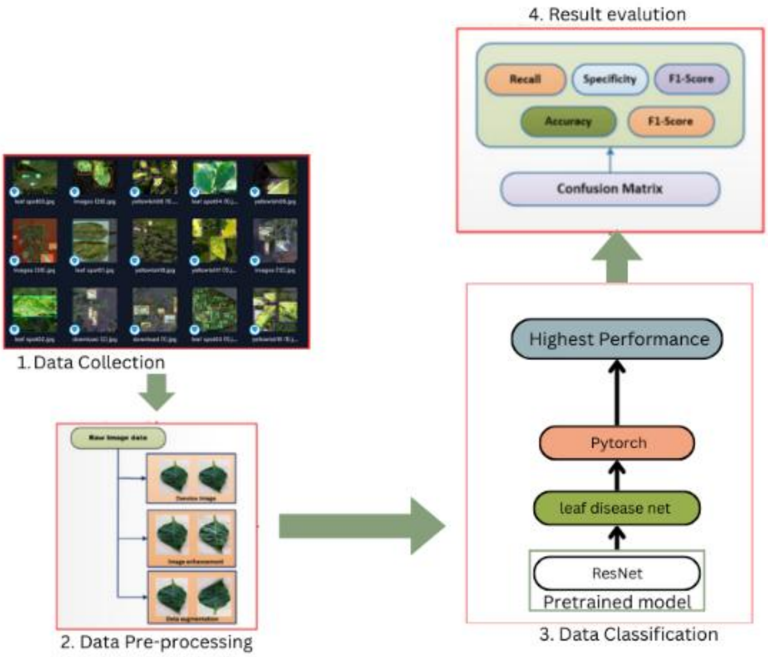


Figure 5. Modelling

Figure 5, process of developing a leaf disease classification system using deep learning, comprising four main stages: data collection, data preprocessing, data classification, and result evaluation. In the data collection stage, raw images of leaves showing various disease symptoms, such as spots, discoloration, or abnormal textures, are gathered to serve as the foundation for training the classification model. During data preprocessing, the collected images undergo enhancement steps, such as contrast adjustment, resolution improvement, or noise reduction, to improve image quality and make critical features more recognizable for the model. The processed images are then used in the data classification stage to train a deep learning model based on the ResNet architecture, a pretrained model recognized for capturing deep patterns in images. The model, implemented using PyTorch, is fine-tuned to identify leaf diseases by learning visual patterns from the images. Finally, in the result evaluation stage, the model's performance is assessed using metrics such as accuracy, recall, specificity, and F1-score to evaluate its reliability in classification, with a confusion matrix providing a visual representation of correct and incorrect predictions across different disease categories. This end-to-end process outlines the steps from data collection to performance evaluation to develop an accurate leaf disease classification system.

2.5. Flowchart

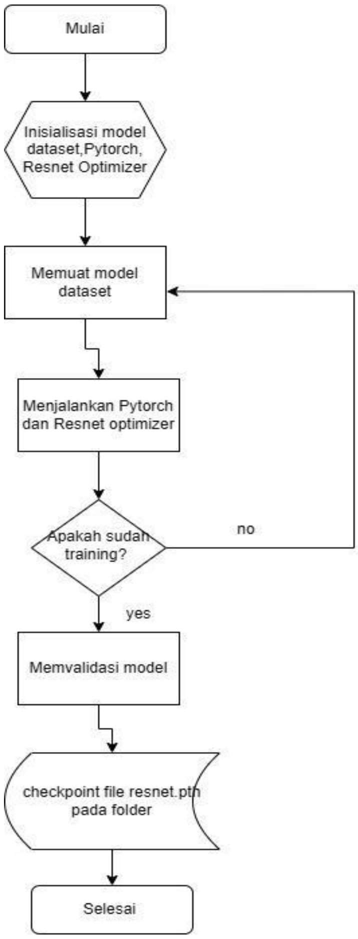


Figure 6. Dataset Flowchart

The flowchart outlines the process of training a ResNet model using PyTorch. The process begins with the initialization of the model, dataset, PyTorch framework, and the ResNet optimizer. Once these components are set up, the system proceeds to load the dataset into the model. After the dataset is loaded, PyTorch and the ResNet optimizer are run to begin the training process. The system then checks if the training is complete. If it is not, the process loops back to continue running PyTorch and the optimizer. Once the training is finished, the model undergoes validation to ensure its accuracy and performance. After validation, a checkpoint file, typically named resnet.pth, is saved in a designated folder to store the trained model's parameters. The process concludes after saving the checkpoint, marking the model as ready for use. This sequence is a common approach in machine learning workflows, where models are trained, validated, and saved for future applications, such as image classification or object detection.

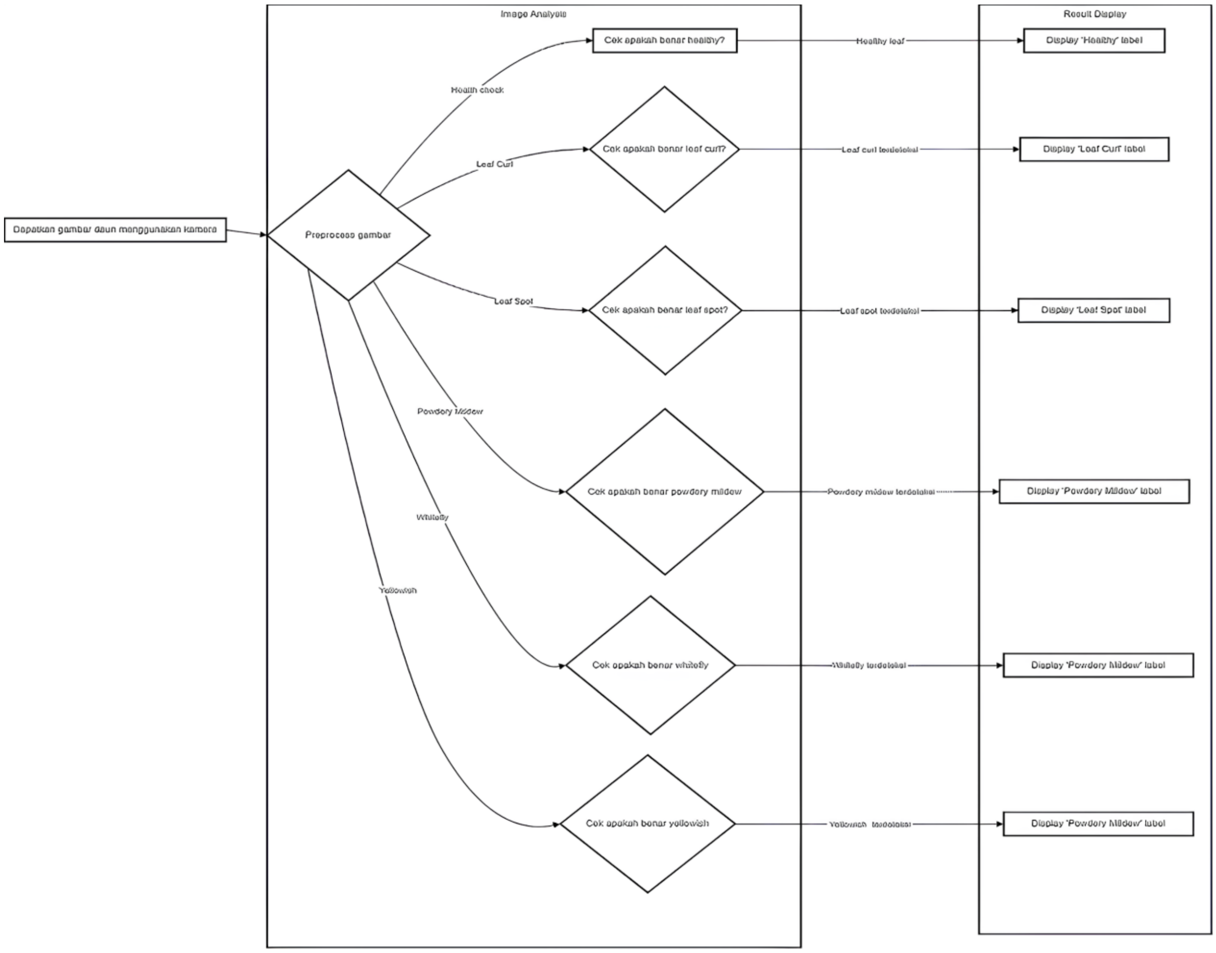


Figure 7. Flowchart Image Recognition

In the flow diagram of leaf image recognition, the disease identification process begins by capturing an image of the leaf using a camera. The image is then processed to prepare it for analysis. After processing, the system checks whether the leaf is healthy or not. If it is not healthy, the application will examine whether the leaf is affected by leaf curl disease, leaf spot, or powdery mildew. Once the disease is identified, the application will display a label indicating the type of disease detected.

3. Results and Discussion

In this research, the results are obtained in the form of a series of tools and website displays described in the results section.

Figure 8. Implementation on Robot

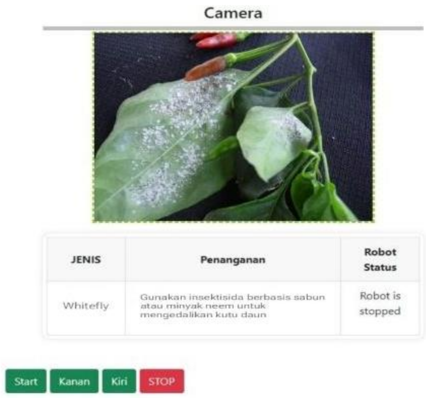


Figure 9. *Implementation on Website*

Figure 8 the physical implementation of the pest detection robot, built on a wheeled platform equipped with various sensors and a camera mounted on a vertical pole. The robot is outfitted with ultrasonic sensors, a motor driver, and other components necessary for mobility and image capture. The camera is positioned at the top to scan and capture images of plant leaves for pest detection. All wires and electronic components are connected to the control system, enabling the robot to operate autonomously or through remote control.In Figure 9, the web interface, designed for controlling the robot and presenting real-time image analysis results. The camera feed shows close-up images of plant leaves, identifying pests such as Whitefly. The interface provides details on the pest type, suggested treatments (e.g., neem oil-based insecticides), and the robot's current status, such as "Robot is stopped." Users can control the robot's movements via buttons for starting, stopping, or navigating in different directions (right, left).

3.1 Tool Set Results

The pest detection robot was designed with a robust rectangular base frame measuring 27.5 cm in width, supported by four wheels for mobility. At the center of the frame, a vertical pole with a height of 31 cm supports a camera or primary sensor module for leaf image acquisition. The electronic components, including sensor modules, wires, and circuit boards, are strategically housed on the frame, ensuring compactness and functionality. The integration of ultrasonic sensors, a motor driver, and other essential components enables the robot to move autonomously or be controlled remotely.

3.2 Modelling Results

Classification Report:

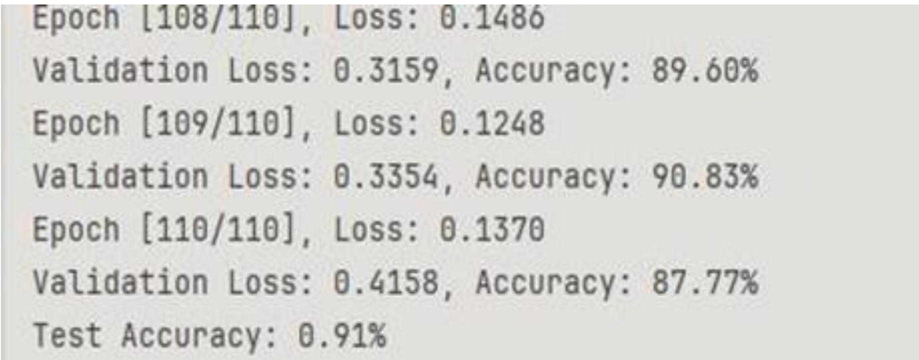
	precision	recall	f1-score	support
0	0.93	0.82	0.88	34
1	0.84	0.94	0.89	34
2	0.94	0.91	0.93	34
3	0.91	0.94	0.93	34
4	0.94	0.88	0.91	34
5	0.89	0.94	0.91	34
accuracy			0.91	204
macro avg	0.91	0.91	0.91	204
weighted avg	0.91	0.91	0.91	204

Figure 10. Model Modification Results

Figure 10, each class performed quite well, with precision ranging between 0.84 and 0.94, recall between 0.82 and 0.94, and an F1-score between 0.88 and 0.93. The support, indicating the number of samples in each class, is 34 for each class. The macro average and weighted average values are all at 0.91, indicating that the model consistently performs well across individual classes and overall.

3.3 Data Accuracy

This model was trained using a predefined formula, with a learning rate configuration of 1×10^{-3}, a batch size of 16, and trained for 110 epochs. During the training process, the model showed performance improvements and ultimately achieved an accuracy of 91% on the test data. This result reflects the model's ability to recognize patterns in the data effectively, despite some variations in the loss and accuracy values over the last few epochs. This accuracy indicates strong potential for the model, though further adjustments might be needed to enhance performance on larger or different datasets.



```
Epoch [108/110], Loss: 0.1486
Validation Loss: 0.3159, Accuracy: 89.60%
Epoch [109/110], Loss: 0.1248
Validation Loss: 0.3354, Accuracy: 90.83%
Epoch [110/110], Loss: 0.1370
Validation Loss: 0.4158, Accuracy: 87.77%
Test Accuracy: 0.91%
```

Figure 11. Accuracy Data Result

3.4 Leaf Image Recognition Results

The system successfully identified a leaf with Leaf Curl classification at an 81% confidence level and detected Powdery Mildew on another leaf with 67% confidence, along with other classes. This testing aims to assess the system's reliability and accuracy in real conditions, where leaves vary in shape, color, and disease severity.

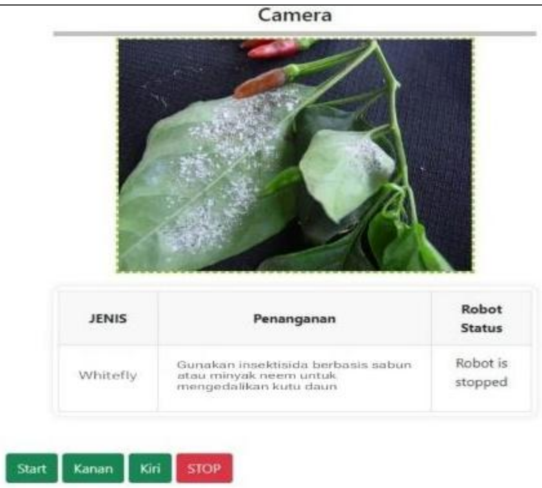
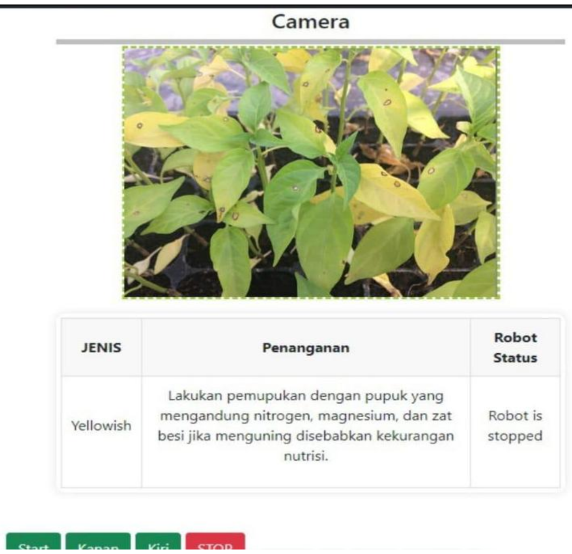
Table 1. Result Leaf Recognition

Condition	Confidence Level
Healthy	89%
Yellowish	78%
Whitefly	92%
Leaf Spot	75%
Leaf Curl	84%
Powdery Mildew	67%

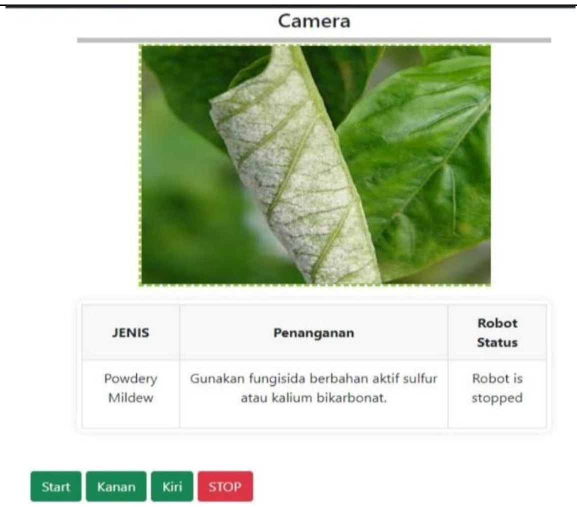
3.5 *Website Result*

Tabel 2 , leaf images representing each label are processed by the camera mounted on the robot, and the analysis results are displayed on the website connected to the system. The test results show that the tool is capable of detecting pests and diseases on chili leaves with good accuracy, providing the necessary information to the user through the web interface.

Table 2. Website Results

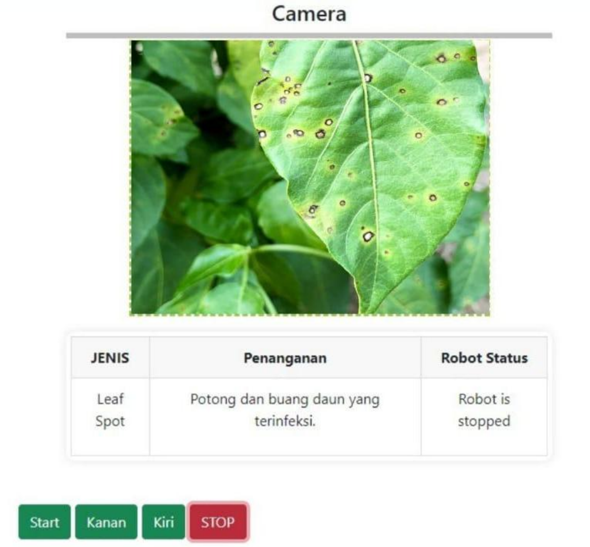
No.	Tampilan Website	Hasil Prediksi
1		Whitefly
2		Yellowish

3



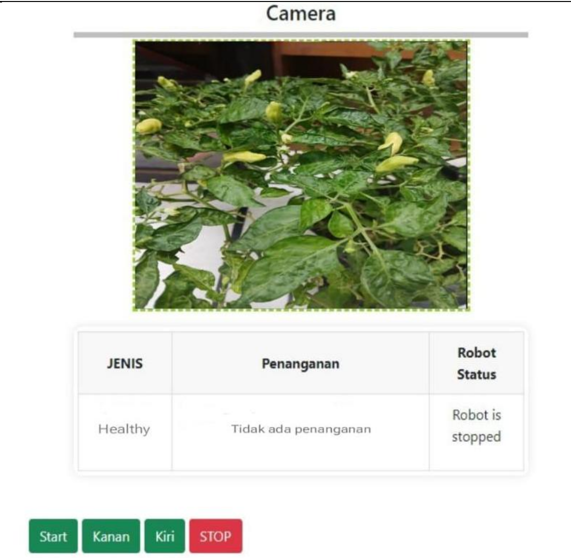
Powdery Mildew

4

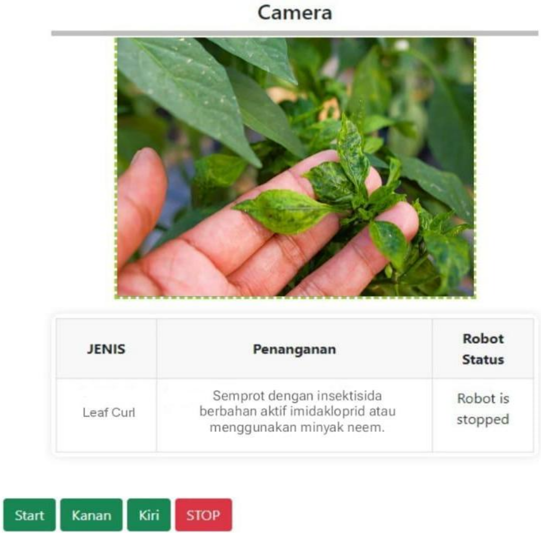


Leaf Spot

5



Healthy

6		Leaf Curl

3.6 *Implications, Limitations, and Future Enhancements*

This pest detection system contributes significantly to sustainable farming practices by enabling early detection of plant diseases and pests, reducing reliance on manual monitoring. Compared to existing technologies, this system integrates advanced machine learning with real-time robotic control, offering precise and efficient pest management.

Limitations

- The current model's accuracy may vary under extreme environmental conditions such as low light or heavy foliage density.
- Limited testing on diverse datasets may impact generalizability to other crops or pest types.

Proposed Enhancements

- Incorporating additional sensors for environmental data collection, such as humidity and temperature, to improve detection accuracy.
- Expanding the dataset to include a broader range of pests and plant diseases.
- Developing a mobile app for remote monitoring and control, providing farmers with greater accessibility.

By addressing these limitations, the system could be further optimized for broader applicability, supporting sustainable agricultural practices across different regions and crops.

4. **Conclusion**

The pest detection robot for chili plants represents a significant milestone in agricultural innovation, offering a reliable, efficient, and technology-driven approach to pest management. By achieving 91% accuracy in real-time pest detection, the robot provides farmers with actionable insights, fostering confidence in the adoption of smart farming technologies. The integration of advanced hardware, such as the Raspberry Pi 5 and Arduino MegaPro Mini 2560, along with ultrasonic sensors for safe navigation, ensures effective operation while protecting crops. Furthermore, this innovation promotes sustainable practices by reducing the reliance on manual inspections and enabling targeted pest control measures.

To maximize the robot's potential, several directions should be considered. For practitioners, the modular design and web-based interface present opportunities to seamlessly integrate the robot into existing farming systems, allowing for real-time monitoring, precise interventions, and enhanced decision-making. Its adaptability supports scalability to larger agricultural fields, particularly when combined with complementary smart farming technologies like automated irrigation, fertilization systems, and environmental monitoring tools.

For researchers, future studies should prioritize improving the robot's environmental adaptability to ensure consistent performance under diverse weather and field conditions. Expanding the system’s compatibility to detect pests across various crop types will enhance its versatility and market reach.

Acknowledgements
The authors would like to thank Gunadarma University and Laboratory of Multimedia and Robotics for supporting the implementation of this project.

References

[1] Anwar, M. T., Nugrohadi, S., Tantriyati, V., & Windarni, V. A. (2020). Rain Prediction Using Rule-Based Machine Learning Approach. Vol 2, No 1 (2020): November-April.

[2] Cahyani, A. N., Zeniarja, J., Winarno, S., Putri, R. T. E., & Maulani, A. A. (2024). Heart Disease Classification Using Deep Neural Network with SMOTE Technique for Balancing Data. Vol 6, No 1 (2024): November-January..

[3] Crisostomo, J. L. B., & Dizon, J. R. C. (2021). 3D Printing Applications in Agriculture, Food Processing, and Environmental Protection and Monitoring. Vol 3, No 2 (2021): May-October.

[4] El Hakim, E. B., & Aryanto, J. (2024). Automated Maintenance System For Freshwater Aquascape Based On The Internet Of Things (IoT). Vol 6, No 1 (2024): November-January.

[5] Firmansyah, M. R., & Astuti, Y. P. (2024). Stroke Classification Comparison with KNN through Standardization and Normalization Techniques. Vol 6, No 1 (2024): November-January.

[6] Hanggara, P., Alfianto, A. S., Purnama, L. P., & Tanra, I. (2024). Portable Low-Cost Home Sleep Monitor using Wemos D1 Mini. Vol 6, No 2 (2024): February-April.

[7] Kamila, I. P., Sari, C. A., Rachmawanto, E. H., & Cahyo, N. R. D. (2024). A Good Evaluation Based on Confusion Matrix for Lung Diseases Classification using Convolutional Neural Networks. Vol 6, No 1 (2024): November-January

[8] Nabila, E. A., Sari, C. A., Rachmawanto, E. H., & Doheir, M. (2023). A Good Performance of Convolutional Neural Network Based on AlexNet in Domestic Indonesian Car Types Classification. Vol 5, No 3 (2023): August-October

[9] Pradisthi, A. S., & Aryanto, J. (2023). Monitoring and Automation System for Bird Feeding and Drinking Based on Internet of Things Using ESP32. Vol 5, No 3 (2023): August-October..

[10] Pramudita, P., & Aryanto, J. (2024). Overcoming The Buildup of Queues By Carrying Out the Concept of Self-Service Using Responsive Web-Based Applications. Vol 6, No 1 (2024): November-January.

[11] Permatasari, P. W., & Aryanto, J. (2024). Implementation Open Artificial Intelligence ChatGPT Integrated With WhatsApp. Vol 6, No 1 (2024): November-January.

[12] Prastiti, F. F., Sari, C. A., Rachmawanto, E. H., & Cahyo, N. R. D. (2023). Mangrove Tree Species Classification Based on Leaf, Stem, and Seed Characteristics Using Convolutional Neural Networks with K-Folds Cross Validation Optimization. Vol 5, No 3 (2023): August-October.

[13] Salsabilla, A., Handayani, A. S., Fadhilah, A., Ziad, I., Husni, N. L., & Huda, M. H. (2023). Design of Android and iOS Applications for Mobile Health Monitoring Devices. Vol 5, No 2 (2023): May-July.

[14] Sugiarto, D., & Farikhah, F. (2024). Sustainability Strategies of Traditional Vannamei Shrimp Cultivation in East Java: A Case Study in Kudu Hamlet, Lamongan District. Vol 6, No 2 (2024): February-April.

[15] Suryadi, G. R., Maslebu, G., & Trihandaru, S. (2021). Artificial Neural Network for Classifying Injected Materials under Ultrasonography. Vol 3, No 1 (2021): November-April.

[16] Utari, G. R., Maslebu, G., & Trihandaru, S. (2021). Automatic Complaints Categorization Using Random Forest and Gradient Boosting. Vol 3, No 1 (2021): November-April.

[17] Utami, E. B., & Aryanto, J. (2024). Automated Maintenance System for Freshwater Aquascape Based on IoT. Vol 6, No 1 (2024): November-January

[18] Wahyu, R. A., & Sari, C. A. (2023). Utilization of Heat from Geothermal Well Pipes as Electricity for Road Lighting Based on IoT. Vol 5, No 1 (2023): November-April.

[19] Wijayanto, A. W., & Rachmawanto, E. H. (2023). Sustainability Strategies of Traditional Vannamei Shrimp Cultivation in East Java: A Case Study. Vol 6, No 2 (2024): February-April

[20] Windarni, V. A., Nugrohadi, S., Tantriyati, V., & Anwar, M. T. (2020). Rule-Based Machine Learning Approach for Rain Prediction. Vol 2, No 1 (2020): November-April.

[21] Wijayanto, A. W., et al. (2020). *Klasifikasi Penyakit Tanaman Cabai menggunakan Algoritma K-Nearest Neighbor (K-NN)*.

[22] Y. L. and Y. Z. (2020). Residual Network for Image Classification of Plant Diseases. *Journal of Intelligent Information Systems*.

[23] Y. Z. and Y. L. (2020). Image-Based Plant Disease Detection using Deep Learning. *Sensors*.

[24] Zhao, Z., Zheng, P., Xu, S., & Wu, X. (2019). Object Detection with Deep Learning: A Review. IEEE Transactions on Neural Networks and Learning Systems, 30(11), 3212-3232

[25] Zhu, X., & Goldberg, A. B. (2009). Introduction to Semi-Supervised Learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, 3(1), 1-130.

[26] Zhang, W., et al. (2020). Plant Disease Recognition Based on ResNet Deep Learning Model. *IEEE Access*, 8, 96413-9646