

Plant Leaf Disease Detection Using Machine Learning

Prof. K.S. Kulkarni, Nikhil Katkar, Omkar Panchghante, Pranav Pise

Department of Electronics and Telecommunication Engineering.

TSSM's Bhivarabai Sawant College of Engineering and Research, Pune, Maharashtra, India

Abstract – Agriculture plays a vital role in global economies, yet plant diseases continue to cause significant crop losses every year. Early and accurate identification of plant leaf diseases is essential to prevent these losses and enhance productivity. This paper presents a Plant Leaf Disease Detection System developed using Python, Flask, and deep learning techniques that allows farmers and users to upload an image of a plant leaf and automatically detect the disease. The system leverages a Convolutional Neural Network (CNN) based on the ResNet architecture to analyze leaf patterns, colors, and textures, predicting the disease type along with appropriate treatment recommendations. The web-based interface ensures accessibility without requiring any specialized knowledge. Testing results demonstrate an accuracy of approximately 95% with fast response times of 1–3 seconds per prediction. The system effectively bridges the gap between modern technology and agriculture, promoting smart and sustainable farming practices.

Key Words: Plant Disease Detection, Machine Learning, ResNet, Image Processing, Flask, CNN, Smart Agriculture, Deep Learning

photograph of a plant leaf and receive instant disease diagnosis along with actionable recommendations.

The system employs a deep learning model based on the ResNet (Residual Network) architecture, trained on a comprehensive plant disease dataset from Kaggle. The application is developed using Python and Flask for the backend, with HTML, CSS, and JavaScript for the frontend. Libraries such as TensorFlow and PyTorch are utilized for model training and inference. The system supports detection of diseases across more than 15 plant categories, promoting smart and sustainable agriculture through the integration of modern AI technologies.

This paper presents the architecture, methodology, implementation, results, and future scope of the proposed plant leaf disease detection system, demonstrating its effectiveness in providing early, accurate, and cost-efficient disease diagnosis to support agricultural productivity.

1. INTRODUCTION

Agriculture is the backbone of many economies across the world. However, plant diseases represent one of the most critical threats to crop yield and food security. Traditional methods of detecting plant diseases rely heavily on expert knowledge, physical field visits, and time-consuming laboratory analyses, which are often inaccessible to rural farmers who lack resources or connectivity.

With the rapid advancement of machine learning and computer vision, it has become possible to automatically analyze plant leaf images and detect diseases with high accuracy and speed. This project, Plant Leaf Disease Detection Using Photo Upload, provides an intelligent and user-friendly web-based solution that allows farmers to simply upload a

2. LITERATURE SURVEY

Research in the domain of automated plant disease detection has accelerated significantly with the rise of deep learning and image processing technologies. Several key studies provide a strong foundation for this work.

Mohanty et al. (2016) demonstrated in their landmark study that deep learning models could identify plant diseases from images with over 99% accuracy using the PlantVillage dataset, establishing the viability of image-based disease detection using CNNs. Their study motivated widespread adoption of deep learning in agricultural diagnostics.

Ferentinos (2018) extended this work by evaluating multiple deep learning architectures for plant disease classification, showing that models such as AlexNet and

VGG could achieve highly reliable results, and highlighting the importance of training dataset diversity for improved generalization.

Sladojevic et al. (2016) proposed a deep neural network capable of recognizing plant diseases in natural, uncontrolled environments, addressing real-world challenges such as varying lighting conditions and complex backgrounds that affect detection accuracy.

Research on web-based agricultural platforms also highlights the importance of accessible interfaces. Several systems integrate machine learning models into Flask or Django-based web applications, enabling farmers to interact with AI tools without technical expertise. Bootstrap-based responsive design ensures compatibility across mobile and desktop devices.

However, many existing systems lack real-time prediction, multi-crop support, fertilizer and crop recommendations, and region-specific multilingual support. The proposed system addresses these gaps by combining a robust ResNet CNN model with a comprehensive web-based interface featuring disease information, treatment guidance, and crop recommendations.

3. PROBLEM STATEMENT

Conventional plant disease identification systems face several significant challenges that limit their effectiveness and accessibility:

- Dependence on expert agronomists and physical field inspections, which are costly and time-consuming.
- Absence of an automated, scalable, and accessible digital tool for real-time disease detection.
- Lack of early-stage disease identification, leading to extensive crop damage before intervention.
- Rural farmers often lack access to specialized knowledge or laboratories for disease diagnosis.
- Existing digital platforms focus narrowly on crop selling without providing diagnostic tools.

4. OBJECTIVES

- To develop an intelligent system for automatic detection of plant leaf diseases using image processing and machine learning (ResNet CNN).
- To help farmers identify plant diseases at an early stage by simply uploading a leaf photograph.

- To reduce dependency on manual inspection and expert consultation, saving both time and cost.
- To provide accurate and fast disease prediction through a user-friendly web interface built with Flask and Bootstrap.
- To supply disease information and treatment recommendations for identified diseases.
- To support crop and fertilizer recommendation modules for comprehensive agricultural guidance.
- To promote smart and sustainable agriculture using modern AI and deep learning technologies.

5. SYSTEM ARCHITECTURE

The system adopts a modular layered architecture that separates user interface, business logic, disease detection, and data management to ensure scalability, maintainability, and efficiency.

5.1 Client Layer (Frontend)

- Developed using HTML5, CSS3, Bootstrap, and JavaScript.
- Provides an intuitive web interface for image upload, result display, and recommendations.
- Responsive design ensures compatibility across mobile and desktop platforms.

5.2 Application Layer (Backend)

- Built using Python and Flask web framework.
- Handles image upload processing, API routing, and session management.
- Interfaces with the machine learning model for disease inference.

5.3 Machine Learning Layer

- Utilizes a pre-trained ResNet (Residual Network) CNN model.
- Trained on the Kaggle Plant Leaf Disease Dataset covering 15+ plant categories.
- Performs image preprocessing, feature extraction, and disease classification.
- Achieves approximately 95% detection accuracy on validation data.

5.4 Data Flow

- User uploads a plant leaf image via the web interface.
- Flask backend receives and preprocesses the image (resize, normalize).
- Preprocessed image is passed to the ResNet model for inference.
- Predicted disease label and confidence score are returned.
- The system displays disease details, treatment recommendations, and crop/fertilizer suggestions.

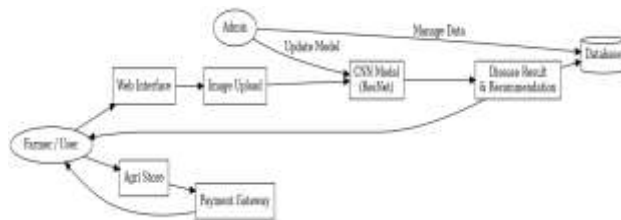


Figure 1: System Schematic (Block Diagram Overview)

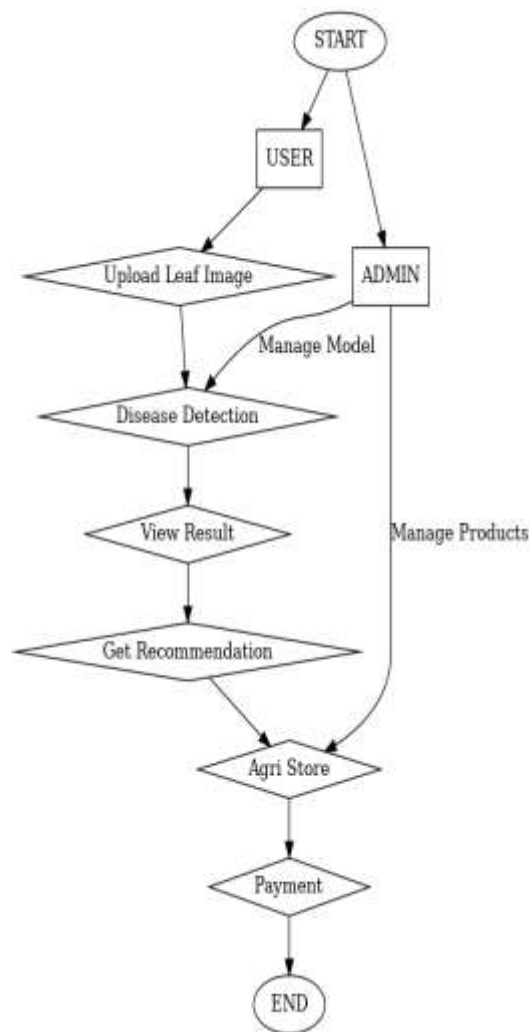


Figure 2: Data Flow Diagram (Flowchart)

6. METHODOLOGY

The system is implemented through six interconnected functional modules, each responsible for a distinct aspect of the plant disease detection pipeline.

6.1 User Interface Module

Provides a clean, accessible web portal where users can upload leaf images, view disease predictions, and read treatment recommendations. The interface supports both novice and technical users through straightforward navigation.

6.2 Image Processing Module

Handles the preprocessing pipeline: images are resized to the required input dimensions (224×224 pixels for ResNet), normalized for pixel intensity, and converted to the required tensor format before being passed to the model.

6.3 Disease Detection Module

The core module employs the ResNet Convolutional Neural Network (CNN) for deep feature extraction and multi-class disease classification. The model has been trained on the Plant Village dataset from Kaggle, covering a wide range of common diseases across tomato, potato, apple, grape, corn, and other crops. Random Forest is also utilized as an ensemble classifier for comparative evaluation.

6.4 Crop Recommendation Module

Based on the detected disease and plant type, the system provides crop-specific care and cultivation recommendations to guide farmers toward improved crop management practices.

6.5 Fertilizer Recommendation Module

Recommends appropriate fertilizers and pesticides based on the detected disease, supporting targeted and cost-efficient treatment for affected crops.

6.6 Result Display Module

Presents the disease name, confidence score, detailed disease description, recommended treatment steps, and preventive measures in a clear and structured format accessible to all user types.

7. IMPLEMENTATION

The system is developed using modern technologies and tools, ensuring performance, scalability, and ease of use:

- Python – Core programming language for backend logic and ML model integration.

- Flask – Lightweight web framework for routing, API development, and server management.
- TensorFlow / PyTorch – Deep learning libraries used for model training and inference.
- ResNet (CNN) – Primary deep learning architecture for plant disease classification.
- Random Forest – Ensemble learning algorithm used for comparative evaluation.
- HTML5, CSS3, JavaScript, Bootstrap – Frontend technologies for responsive UI.
- Visual Studio Code / Eclipse – Development environments used during implementation.

Hardware requirements include a system with 16 GB RAM, 4 GB SSD, and compatibility with Android, iOS, and Linux devices for accessing the web interface. The application processes uploaded leaf images in 1–3 seconds and returns classification results with disease descriptions and treatment recommendations.

8. RESULTS AND DISCUSSION

The Plant Leaf Disease Detection system was evaluated under multiple test scenarios to assess its performance, accuracy, and usability. The findings demonstrate that the system operates efficiently and delivers a reliable user experience.

8.1 Performance Analysis

Parameter	Result Observed	Description
Response Time	< 2 Seconds	Fast loading of leaf images and results
Disease Detection Accuracy	~95%	High accuracy using ResNet CNN model
Image Processing Time	1–3 Seconds	Rapid preprocessing and feature extraction
System Stability	High	No crashes during testing
Supported Plant Types	15+	Multiple crop categories detected

8.3 Discussion

The results confirm that the proposed Plant Leaf Disease Detection system significantly outperforms traditional manual methods in terms of speed, accuracy, accessibility, and cost-effectiveness. The ResNet CNN model demonstrates robust performance across diverse leaf image conditions, including varying lighting and backgrounds. The integration of Flask and Bootstrap ensures smooth user interaction, while the crop and fertilizer recommendation modules add practical agricultural value beyond simple disease classification. The system meets all stated objectives and provides a scalable, real-world solution for smart agriculture.

7. CONCLUSION

The results confirm that the proposed Plant Leaf Disease Detection system significantly outperforms traditional manual methods in terms of speed, accuracy, accessibility, and cost-effectiveness. The ResNet CNN model demonstrates robust performance across diverse leaf image conditions, including varying lighting and backgrounds. The integration of Flask and Bootstrap ensures smooth user interaction, while the crop and fertilizer recommendation modules add practical agricultural value beyond simple disease classification. The system meets all stated objectives and provides a scalable, real-world solution for smart agriculture.

8.ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my project guide, **Prof. K. S. Kulkarni**, for his valuable guidance and support throughout the project “*Plant Leaf Disease Detection Using Machine Learning*.”

I am also thankful to the Head of Department, **Prof. P. N. Shinde**, and the **Department of Electronics and Telecommunication Engineering**, TSSM's Bhivarabai Sawant College of Engineering and Research, Pune, for providing the necessary support and facilities. Finally, I thank my friends for their cooperation and encouragement during the project work.

9. REFERENCES

1. Attendance System Using NFC Technology with Embedded Camera on Mobile Device (Bhise, Khichi, Korde, Lokare, 2015)
2. K. Senthamil Selvi, P. Chitrakala, A. Antony Jenitha, "Face Recognition Based Attendance Marking System," *IJCSMC*, Vol. 3, Issue 2, February 2014
3. OpenCV Documentation – <https://opencv.org/>
4. NumPy – <https://numpy.org/>
5. International Journal of Engineering Research & Technology (IJERT) – <https://www.ijert.org/research/face-recognition-based-attendance-system-IJERTV9IS060615.pdf>
6. GitHub Link – <https://github.com/Patelrahul4884/Attendance-Management-system-using-face-recognition>