

Plant Disease Detection using AI

Dr. M C Padma¹, Ananya M J², Chandana S³, Dhruthi G N⁴

¹Professor

^{1,2,3,4}Department of Computer Science and Engineering
PES College of Engineering, Mandya 571401, Karnataka, India

Abstract:

Early and accurate detection of tomato leaf diseases is essential for improving agricultural productivity and reducing crop losses, yet many existing farming applications focus mainly on environmental monitoring and yield estimation while ignoring automated disease diagnosis and treatment guidance. This paper presents an AI-Based Tomato Leaf Disease Detection and Treatment Recommendation System that utilizes computer vision and deep learning techniques to monitor plant health using only a smartphone or standard digital camera, without requiring specialized agricultural equipment. The system employs image-based disease analysis to identify infected tomato leaves in real time. A MobileNetV2 deep learning model trained on a tomato-specific PlantVillage dataset extracts visual disease patterns such as discoloration, lesions, and texture abnormalities from uploaded leaf images. The generated feature representations are processed to classify diseases including Early Blight, Late Blight, Bacterial Spot, Septoria Leaf Spot, Mosaic Virus, and Yellow Leaf Curl Virus with high accuracy. The integrated pipeline automatically predicts disease categories, estimates severity levels, and generates AI-powered treatment recommendations. Upon detecting a disease, the system provides instant corrective guidance including fungicide suggestions, organic treatment methods, preventive measures, and crop management advice through an interactive web and mobile interface. The backend architecture is implemented using Node.js and Express, while the frontend is developed using Flutter for cross-platform accessibility. Operating with fast cloud-based inference and real-time response capability, the system ensures efficient disease monitoring for farmers. Experimental evaluation demonstrates high classification accuracy, reliable disease prediction, and effective treatment recommendation performance, making intelligent agricultural assistance accessible and affordable for farmers regardless of geographical limitations or resource availability.

Keywords: Tomato leaf disease detection, Deep learning, MobileNetV2, PlantVillage dataset, Computer vision, Artificial intelligence, Image classification, Disease severity estimation, Smart agriculture, Real-time disease prediction, Treatment recommendation system, Precision farming.

1. INTRODUCTION

Agriculture is one of the most important sectors supporting food production and economic development across the world. Among various crops, tomato is widely cultivated due to its nutritional value, commercial demand, and extensive usage in food industries. However, tomato plants are highly susceptible to several bacterial, fungal, and viral diseases that significantly reduce crop quality and yield. Diseases such as Early Blight, Late Blight, Bacterial Spot, Septoria Leaf Spot, Mosaic Virus, and Yellow Leaf Curl Virus can spread rapidly if not identified during the early stages, leading to severe economic losses for farmers. Traditional disease diagnosis methods mainly depend on manual inspection by agricultural experts, which is often time-consuming, expensive, and inaccessible in rural farming regions.

In recent years, advancements in Artificial Intelligence, computer vision, and deep learning have transformed agricultural disease detection by enabling automated image-based diagnosis systems. Deep learning models such as Convolutional Neural Networks (CNNs) and MobileNetV2 are capable of

extracting complex visual features from plant leaf images, including discoloration, lesions, spots, and texture abnormalities. These technologies make it possible to detect diseases accurately using only smartphone cameras or standard digital imaging devices without requiring specialized laboratory equipment.

This paper presents an AI-Based Tomato Leaf Disease Detection and Treatment Recommendation System designed to provide real-time disease diagnosis and intelligent treatment guidance. The proposed system allows users to upload or capture tomato leaf images through a mobile or web application. The uploaded images are processed using a MobileNetV2 deep learning model trained on a tomato-specific PlantVillage dataset. The model predicts disease categories with high accuracy and estimates severity levels based on confidence scores and visible symptoms.

In addition to disease detection, the system integrates an AI-powered advisory module that generates treatment recommendations including fungicide usage, organic remedies, preventive measures, irrigation guidance, and crop management practices. The application is developed using Flutter for cross-platform accessibility and Node.js with Express for backend processing and cloud-based inference. By combining deep learning, cloud computing, and intelligent recommendation systems, the proposed solution bridges the gap between traditional farming practices and modern AI-driven agricultural support systems, helping farmers improve crop health, reduce losses, and increase productivity.

2. RELATED WORK

2.1 Deep Learning in Plant Disease Detection

Deep learning techniques have been widely explored for automated plant disease identification using leaf image analysis. Early research focused on Convolutional Neural Networks (CNNs) trained on agricultural datasets such as PlantVillage. Studies by Mohanty et al. demonstrated that deep learning models could achieve high accuracy in detecting plant diseases from leaf images under controlled conditions. However, many existing systems were computationally expensive and difficult to deploy on lightweight mobile platforms. MobileNetV2 later emerged as an efficient alternative due to its lightweight architecture, reduced computational complexity, and suitability for real-time agricultural applications on low-resource devices.

2.2 Artificial Intelligence in Smart Agriculture

The application of Artificial Intelligence in agriculture has grown significantly with advancements in computer vision, cloud computing, and machine learning. AI-based agricultural systems are increasingly being used for crop monitoring, disease prediction, irrigation management, and yield estimation. Recent intelligent farming solutions provide automated diagnosis and decision support using image processing and deep learning techniques. However, most existing agricultural applications focus primarily on environmental monitoring or generalized crop analysis, while disease-specific systems with integrated treatment recommendation modules remain limited, particularly for tomato crops.

2.3 Image-Based Disease Classification Systems

Image-based disease classification systems have become an important research area due to the availability of large annotated agricultural datasets. The PlantVillage dataset enabled researchers to train robust machine learning and deep learning models for identifying diseases in crops such as tomato, potato, and grape. Architectures including AlexNet, VGGNet, ResNet, and MobileNet have been applied for disease classification with high accuracy. Recent cloud-hosted inference platforms further improved accessibility by enabling real-time disease prediction through web and mobile applications. Despite these advancements, many systems lack real-time treatment guidance, severity estimation, and farmer-friendly interfaces for practical field usage.

3. SYSTEM ARCHITECTURE

The proposed AI-Based Tomato Leaf Disease Detection and Treatment Recommendation System follows a multi-layer architecture consisting of frontend, backend, AI inference, and database

components. The architecture is designed to provide real-time disease detection, treatment recommendation, and cross-platform accessibility through mobile and web applications. The system integrates deep learning, cloud computing, and REST API communication to ensure accurate and efficient disease analysis.

The architecture mainly consists of the following layers:

3.1 Presentation Layer

The presentation layer is developed using Flutter for both Android and Web platforms. It provides an interactive user interface through which farmers can capture tomato leaf images using a smartphone camera or upload images from the gallery. The frontend also displays prediction results, severity levels, treatment recommendations, and scan history in a user-friendly format.

The presentation layer includes:

- Image capture and upload module
- Disease prediction dashboard
- AI treatment recommendation interface
- Scan history and feedback system
- Real-time result visualization

3.2 Application Layer

The application layer is implemented using Node.js and Express.js. It acts as the communication bridge between the frontend application, AI inference services, and database storage. The backend handles image processing, API request management, disease prediction responses, and treatment recommendation generation.

The backend functionalities include:

- REST API management
- Base64 image processing
- Request validation and preprocessing
- Communication with AI inference APIs
- Treatment recommendation generation
- Database interaction and storage

3.3 AI Inference Layer

The AI inference layer performs tomato disease classification using a MobileNetV2 deep learning model trained on a tomato-specific PlantVillage dataset. The uploaded tomato leaf image is analyzed to identify visual disease symptoms such as lesions, discoloration, spots, and texture abnormalities.

The model predicts diseases including:

- Tomato Early Blight
- Tomato Late Blight
- Tomato Bacterial Spot
- Tomato Leaf Mold
- Tomato Mosaic Virus
- Tomato Yellow Leaf Curl Virus
- Tomato Septoria Leaf Spot
- Tomato Target Spot
- Tomato Spider Mites
- Healthy Tomato Leaf

The AI layer also calculates confidence scores and severity estimation levels for the detected disease.

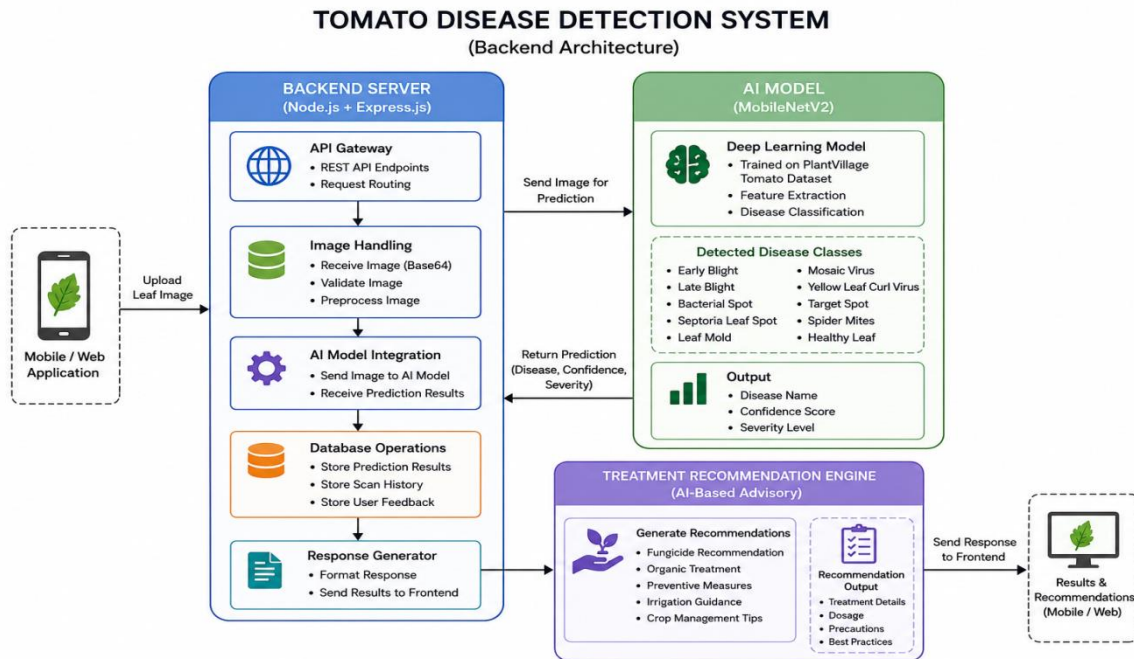
3.4 Treatment Recommendation Layer

The treatment recommendation layer uses an AI-powered advisory system to generate intelligent crop treatment suggestions based on the predicted disease and severity level. The system provides:

- Fungicide recommendations
- Organic treatment methods

- Preventive measures
- Irrigation guidance
- Crop management practices
- Recovery suggestions

This layer improves farmer decision-making and helps reduce crop damage through timely corrective actions.



4. DEVELOPMENT JOURNEY

The development of the Tomato Disease Detection and Treatment Recommendation System involved systematic evaluation of multiple technological approaches before finalizing the current architecture. This section explains the complete development process, including experimental approaches, implementation strategies, encountered challenges, and final optimization techniques used in the system.

4.1 Phase 1: Initial Design and System Analysis

The project began with detailed system analysis focused on tomato disease identification and farmer accessibility. A Use Case Diagram was designed identifying the primary actors: Users (farmers/agricultural workers) and Administrators. Core use cases identified included: Upload Tomato Leaf Image, Capture Leaf Image, Detect Disease, View Severity Level, Get Treatment Recommendation, View Scan History, and Submit Feedback.

Simultaneously, the overall system architecture was planned with the following components:

- Flutter Mobile/Web Interface
- Backend Server
- AI Disease Detection Module
- Treatment Recommendation Engine
- User Feedback System

The initial design focused on creating a lightweight and user-friendly application suitable for real-time disease detection using smartphones.

4.2 Phase 2: CNN-Based Disease Classification Approach

The first disease detection approach implemented was a custom Convolutional Neural Network (CNN) model trained using tomato leaf images from the PlantVillage dataset. The purpose of this approach was to build a customized deep learning model specifically for tomato disease classification.

Implementation involved:

- Image preprocessing and normalization
- CNN model development using TensorFlow/Keras
- Data augmentation techniques
- Multi-class disease classification training

Challenges encountered:

- High computational requirements during training
- Slower prediction speed during testing
- Increased model size affecting application performance
- Overfitting due to limited dataset diversity
- Difficulty achieving smooth real-time prediction

Although the CNN model achieved good classification results, it was not efficient enough for lightweight real-time deployment.

Decision: The custom CNN architecture was replaced with a more optimized lightweight model suitable for fast disease prediction.

4.3 Phase 3: MobileNetV2-Based Disease Detection

The second and final disease detection approach used MobileNetV2, a lightweight deep learning architecture optimized for mobile and web applications. The main reason for selecting MobileNetV2 was its high accuracy combined with low computational complexity.

Implementation included:

- Fine-tuning MobileNetV2 using tomato leaf disease images
- Transfer learning using pretrained weights
- Real-time image classification integration
- Confidence score generation for severity estimation

The model was trained to classify diseases such as:

- Early Blight
- Late Blight
- Bacterial Spot
- Leaf Mold
- Septoria Leaf Spot
- Mosaic Virus
- Yellow Leaf Curl Virus
- Spider Mites
- Target Spot
- Healthy Tomato Leaf

Advantages observed:

- Faster prediction speed
- Reduced memory usage
- Better real-time performance
- Lightweight architecture suitable for mobile devices
- Improved classification accuracy

Decision: MobileNetV2 was finalized as the primary disease detection model due to its efficient performance and optimized prediction capability.

4.4 Phase 4: Backend Integration and System Development

The next phase focused on integrating the frontend application with backend services and AI prediction modules. The backend was developed using Node.js and Express.js to manage image processing, API communication, and disease prediction workflows.















Implementation involved:

- REST API development for disease prediction
- Image upload and processing
- Prediction result handling


- User feedback integration
 - Severity estimation generation
- and reduced unnecessary API calls to improve response speed.

5. IMPLEMENTATION DETAILS


5.1 Technology Stack

SYSTEM ARCHITECTURE AND TECHNOLOGIES USED		
Component	Technology Used	Purpose
 Frontend	   HTML, CSS, JavaScript	Web Application (User Interface)
 Backend	 Node.js + Express.js <code>express</code>	API Handling and Processing
 AI Model	 MobileNetV2	Tomato Disease Classification
 Dataset	PlantVillage Tomato Dataset	Model Training
 Image Processing	OpenCV	Image Preprocessing
 Programming Language	 Python, JavaScript	Application Development
 API Communication	REST API	Frontend and Backend Communication
 IDE	Visual Studio Code	Development Environment


FRONTEND TECHNOLOGIES (HTML, CSS, JAVASCRIPT)



HTML
Structure of the web pages



CSS
Styling and layout of the web pages



JavaScript
Interactivity and dynamic functionality

5.2 Key Design Decisions

5.2.1 MobileNetV2 for Disease Detection

Initially, a traditional CNN model was considered for tomato disease classification. However, the model required higher processing power and produced slower predictions. MobileNetV2 was selected because it provides high accuracy with lower computational complexity, making it suitable for real-time disease detection in mobile and web applications.

5.2.2 Flutter for Cross-Platform Development

Separate Android and Web development would increase development complexity and maintenance effort. Flutter was selected because it supports both mobile and web platforms using a single codebase, reducing development time and improving application consistency.

5.2.3 Base64 Image Processing







Direct file handling caused compatibility issues during image upload, especially for web applications. To solve this problem, the system uses Base64 image conversion before sending images to the backend server. This ensured smooth image transfer and cross-platform compatibility.

5.2.4 AI-Based Recommendation System

The project initially focused only on disease prediction. Later, an AI-powered treatment recommendation module was integrated to provide fungicide suggestions, preventive measures, and crop management guidance. This improved the practical usefulness of the system for farmers.

5.3 Disease Coverage

5.3 DISEASE COVERAGE

DISEASE CATEGORY		DISEASES SUPPORTED
 FUNGAL DISEASES		Early Blight, Late Blight, Leaf Mold, Septoria Leaf Spot
 BACTERIAL DISEASES		Bacterial Spot
 VIRAL DISEASES		Mosaic Virus, Yellow Leaf Curl Virus
 PEST DISEASES		Spider Mites
 OTHER DISEASES		Target Spot
 HEALTHY CLASS		Healthy Tomato Leaf

Total Classes Supported: 10

6. RESULTS AND DISCUSSION

6.1 Disease Detection Results

The MobileNetV2 model successfully identified tomato leaf diseases from uploaded images with high accuracy. Diseases such as Early Blight, Late Blight, and Yellow Leaf Curl Virus produced highly accurate predictions due to their visually distinctive symptoms.

The system effectively analyzed:

- Leaf discoloration
- Lesions and spots
- Texture abnormalities
- Leaf curling symptoms

The application also correctly classified healthy tomato leaves and reduced incorrect disease predictions.











6.2 Treatment Recommendation Results

The AI-powered recommendation module generated treatment suggestions based on the detected disease and severity level. The generated recommendations included:

- Fungicide recommendations
- Organic treatment methods
- Preventive measures
- Irrigation guidance
- Crop management suggestions

This feature improved the usefulness of the application by providing immediate agricultural guidance to farmers.

6.3 System Performance

SYSTEM PERFORMANCE	
FEATURE	AVERAGE RESPONSE TIME
 Image Upload	 < 2 seconds
 Disease Prediction	 2–4 seconds
 Severity Estimation	 < 1 second
 Treatment Recommendation	 2–3 seconds
 Result Display	 < 1 second

7. CHALLENGES AND LIMITATIONS

7.1 Model Prediction Accuracy Challenges

One of the major technical challenges encountered during development was achieving consistent disease prediction accuracy for visually similar tomato leaf diseases. Diseases such as Early Blight and Septoria Leaf Spot contain overlapping visual symptoms including brown lesions and circular spots, causing occasional classification confusion during initial testing. Variations in lighting conditions, image quality, and leaf orientation also affected prediction consistency. To improve model performance, image preprocessing, normalization, and data augmentation techniques such as rotation, zooming, and flipping were implemented to enhance dataset diversity and improve generalization capability.

7.2 Dataset Limitations

The system primarily uses the PlantVillage Tomato Dataset, which contains images captured under controlled environmental conditions with clean backgrounds. Real-world agricultural environments often include varying lighting conditions, shadows, soil background noise, and partially damaged leaves, which may affect prediction accuracy. Expanding the dataset with field-level images and real farm conditions would improve model robustness and practical deployment performance.

7.3 Cross-Platform Image Upload Issues

During development, Flutter Web compatibility introduced several image upload challenges because direct dart:io file operations are not supported in browser environments. Initial implementations caused image transfer failures and inconsistent frontend behavior across Android and Web platforms. This issue was resolved by implementing Base64 image encoding before transmitting images to the backend server, ensuring smooth cross-platform compatibility and stable image processing.

7.4 Real-Time Prediction Performance

Maintaining fast prediction speed while preserving high classification accuracy was another significant challenge. Larger deep learning architectures initially produced slower inference times, making real-time disease detection impractical for low-resource devices. Traditional CNN models increased memory consumption and response delays during testing. This issue was mitigated by replacing the custom CNN model with MobileNetV2, a lightweight architecture optimized for faster inference and reduced computational overhead.

8. FUTURE WORK

Several extensions are planned for future development of the proposed Tomato Disease Detection and Treatment Recommendation System:

- Multi-crop disease detection: Extend the system to support additional crops such as potato, chili, rice, and maize along with their respective plant diseases
- Offline prediction support: Integrate TensorFlow Lite for offline disease detection on mobile devices without requiring continuous internet connectivity
- Regional language support: Provide treatment recommendations and application interfaces in regional languages such as Kannada, Hindi, Tamil, and Telugu for better farmer accessibility
- Real-time field monitoring: Integrate IoT sensors and smart farming devices for continuous crop health monitoring and environmental analysis
- Advanced severity analysis: Improve disease severity estimation using enhanced image segmentation and symptom area calculation techniques

9. CONCLUSION

This paper presented an AI-Based Tomato Leaf Disease Detection and Treatment Recommendation System designed to assist farmers in identifying tomato plant diseases using deep learning and computer vision techniques. The system successfully integrates a MobileNetV2-based disease classification model, real-time image processing, AI-powered treatment recommendation generation, and a cross-platform Flutter application for mobile and web accessibility.

The development process demonstrated the importance of iterative model evaluation and system optimization. Initial traditional CNN architectures provided good classification accuracy but resulted in higher computational complexity and slower inference speed. MobileNetV2 provided the best balance between prediction accuracy, lightweight performance, and real-time disease detection capability suitable for practical agricultural applications. Key technical contributions of the proposed system include: (1) implementation of a tomato-specific disease detection model trained using the PlantVillage dataset; (2) real-time classification of multiple tomato diseases including Early Blight, Late Blight, Mosaic Virus, and Yellow Leaf Curl Virus; (3) integration of an AI-powered treatment recommendation module capable of generating fungicide suggestions, organic remedies, and preventive measures; (4) development of a cross-platform Flutter-based user interface supporting image upload and disease analysis; and (5) implementation of efficient backend communication and image preprocessing techniques for smooth real-time prediction. The proposed system demonstrates that modern Artificial Intelligence and deep learning technologies can significantly improve smart agricultural practices by enabling accurate, fast, and accessible disease diagnosis. By reducing dependency on manual inspection and expert consultation, the system helps farmers identify diseases at early stages, minimize crop damage, improve productivity, and support sustainable farming practices through intelligent AI-driven agricultural assistance.

REFERENCES:

- [1] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using Deep Learning for Image-Based Plant Disease Detection," *Frontiers in Plant Science*, vol. 7, pp. 1–10, 2016.
- [2] K. P. Ferentinos, "Deep Learning Models for Plant Disease Detection and Diagnosis," *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, 2018.
- [3] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A Comparative Study of Fine-Tuning Deep Learning Models for Plant Disease Identification," *Computers and Electronics in Agriculture*, vol. 161, pp. 272–279, 2019.
- [4] 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.

- [5] D. P. Hughes and M. Salathé, “An Open Access Repository of Images on Plant Health to Enable the Development of Mobile Disease Diagnostics,” *arXiv preprint arXiv:1511.08060*, 2015.
- [6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- [7] TensorFlow Documentation, “TensorFlow and Keras Deep Learning Framework,” 2024.
- [8] Flutter Documentation, “Flutter SDK for Cross-Platform Application Development,” 2024.
- [9] Node.js Documentation, “Express.js Web Application Framework,” 2024.
- [10] OpenCV Documentation, “Open Source Computer Vision Library,” 2024.
- [11] PlantVillage Dataset Documentation, “Tomato Leaf Disease Dataset for Deep Learning Applications,” 2024.
- [12] Google AI Documentation, “Generative AI and Gemini API Documentation,” 2024.
- [13] J. G. A. Barbedo, “Impact of Dataset Size and Variety on the Effectiveness of Deep Learning and Transfer Learning for Plant Disease Classification,” *Computers and Electronics in Agriculture*, vol. 153, pp. 46–53, 2018.
- [14] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, “Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification,” *Computational Intelligence and Neuroscience*, vol. 2016, pp. 1–11, 2016.