International Journal of Scientific Research and Engineering Development-– Volume 9 Issue 2, Mar-Apr 2026
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Powdery Mildew Disease Detection of Mango Trees Using CNN

Dr.V. Suganthi, Sai Barth Krishna P M
[1]Assistant Professor, PG & Reasearch Department of Computer Science, Sri Ramakrishna College of Arts & Science
[2]Student of Computer Science, PG & Reasearch Department of Computer Science, Sri Ramakrishna College of Arts & Science

Abstract:
This project presents a web-based **Mango Tree Disease Detection and Treatment Recommendation System developed using Python and Flask**. The system analyzes mango leaf images to identify diseases such as Powdery Mildew, Anthracnose, and Bacterial Canker. Image preprocessing and feature extraction techniques are used to detect visual symptoms and classify the disease. The application also estimates severity levels and provides structured treatment recommendations with dosage details. A step-by-step recovery plan is generated to guide farmers in effective disease management. The system aims to support precision agriculture by combining intelligent detection with practical treatment guidance.

I. Introduction

Artificial intelligence and image processing technologies have significantly transformed agricultural monitoring systems by enabling automated disease detection and decision support mechanisms. With the advancement of machine learning and web-based platforms, scalable agricultural solutions can now be deployed with minimal infrastructure overhead.

Despite these technological developments, plant disease identification in many farming regions remains largely manual and experience-based. Farmers often rely on visual inspection or expert consultation, which leads to:
• Delayed disease detection
• Inaccurate diagnosis
• Lack of severity estimation
• Absence of structured treatment planning
• Increased crop loss and economic damage

To address these limitations, this paper proposes a web-based Mango Tree Disease Detection and Treatment Recommendation System that integrates image preprocessing, rule-based classification, severity estimation, and structured recovery scheduling within a scalable architecture.

Major Contributions

The primary contributions of this work are:
1. Design of a structured image-based disease detection framework
2. Implementation of a modular web-based client–server architecture
3. Development of a rule-based classification and severity estimation model
4. Integration of dosage-based treatment recommendation system
5. Structured recovery lifecycle planning mechanism
6. Scalable architecture adaptable for CNN-based future enhancement

Image processing techniques have been widely adopted in agricultural applications due to their ability to analyze leaf symptoms such as discoloration, lesion patterns, and fungal growth. Machine learning-based disease detection models, particularly Convolutional Neural Networks (CNN), have demonstrated strong performance in visual classification tasks.

Structured decision-support systems further enhance practical usability by combining disease prediction with treatment guidance. However, many existing systems focus primarily on classification accuracy and lack integrated treatment planning or severity-based recommendations.

To overcome these limitations, the proposed system integrates image analysis, classification logic, severity estimation, and recovery planning into a unified web-based platform. This integration improves accessibility, transparency, and practical applicability in agricultural environments.

II. System Architecture

The proposed system follows a structured three-tier architecture to ensure modularity, scalability, and efficient processing of disease detection tasks. The architecture separates user interaction, application logic, and data management to improve maintainability, reliability, and performance. This separation of concerns allows independent development and future enhancement of each layer without affecting overall system stability.

The three-tier design also supports extensibility, enabling integration of advanced machine learning models, cloud storage systems, and mobile-based interfaces in future versions.

A. Presentation Layer

The Presentation Layer represents the user interface of the system. It is developed using HTML and CSS to provide a simple, responsive, and accessible web-based environment for users. This layer is responsible for handling user interaction, including image upload and visualization of diagnostic results.

It enables:
• Leaf image submission
• Display of predicted disease
• Visualization of severity level
• Presentation of treatment and recovery plan
• Validation of uploaded image formats
• Clear navigation between result and treatment pages

The interface is designed to be intuitive and farmer-friendly, ensuring that users without technical expertise can easily operate the system. By providing structured output visualization, the presentation layer enhances transparency and user trust.

B. Application Layer

The Application Layer functions as the core processing unit of the system. It is implemented using Python and the Flask framework. This layer handles server-side logic, executes image preprocessing operations, performs feature extraction, and applies classification algorithms.

Its responsibilities include:
• Receiving and validating user-uploaded images
• Performing image resizing, normalization, and noise reduction
• Extracting disease-related visual features
• Comparing features with stored disease profiles
• Estimating severity levels based on infection spread
• Generating structured treatment recommendations
• Rendering dynamic response pages

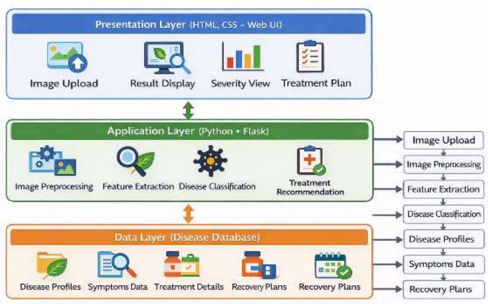
This layer ensures accurate analysis and structured decision-making. It also manages communication between the presentation layer and data layer, acting as the central control unit of the system. The modular design allows future integration of Convolutional Neural Network (CNN)-based deep learning models for automated and higher-accuracy classification.

C. Data Layer

The Data Layer manages storage and retrieval of disease-related information. It is implemented using a structured database in Python dictionary or JSON format. The database contains predefined disease profiles, symptoms, environmental conditions, treatment solutions, dosage details, and recovery plans.

This layer ensures:
• Organized data management
• Fast retrieval of disease information
• Consistent treatment and dosage mapping
• Secure storage of predefined agricultural guidelines
• Easy update and modification of disease profiles
• Scalability for future cloud database integration

The structured data model supports efficient rule-based classification and enables seamless expansion to relational or cloud-based database systems.



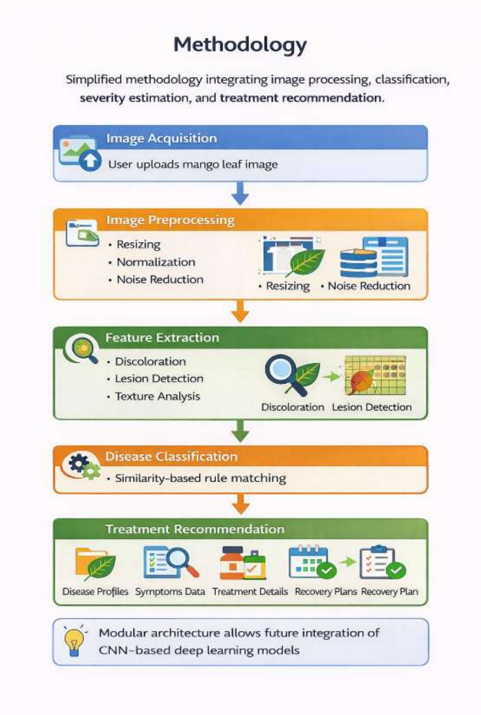
IV. Methodology

The proposed Mango Tree Disease Detection and Treatment Recommendation System follows a structured and modular methodology integrating image processing, classification, severity estimation, and decision-support mechanisms within a web-based framework.

The process begins with **image acquisition**, where the user uploads a mango leaf image through the application interface. The image undergoes **preprocessing**, including resizing, normalization, and noise reduction to ensure consistency and improve feature clarity.

Next, the **feature extraction module** analyzes visual characteristics such as discoloration, lesion patterns, texture irregularities, and infection spread. These features are structured into a representative feature vector for classification.

The **disease classification module** compares extracted features with predefined disease profiles stored in the database. A similarity-based rule mapping approach is used to determine the most probable disease category. Following classification, the **severity estimation module** evaluates the infection level (Low, Moderate, High) based on the proportion of affected regions. Finally, the **treatment recommendation framework** generates structured guidance including chemical dosage, application frequency, preventive measures, and a time-based recovery plan. The modular architecture ensures scalability and allows future integration of CNN-based deep learning models for enhanced prediction accuracy.



V. System Design

The proposed Mango Tree Disease Detection and Treatment Recommendation System follows a modular client–server architecture to ensure scalability, efficiency, and maintainability. The system is designed to integrate image analysis, disease classification, and treatment planning within a unified web-based platform.

The architecture consists of three primary components: the **Presentation Layer**, **Application Layer**, and **Data Layer**.

A. Presentation Layer

The presentation layer provides the user interface through which farmers upload leaf images and view diagnostic results. It is developed using HTML and CSS to ensure simplicity and accessibility. The interface supports image upload, result visualization, severity display, and structured recovery plan presentation.

B. Application Layer

The application layer is implemented using Python and the Flask framework. It handles request processing, image preprocessing, feature extraction, classification logic, and treatment recommendation. The server processes user inputs, executes disease detection algorithms, retrieves structured disease data, and dynamically generates output responses.

C. Data Layer

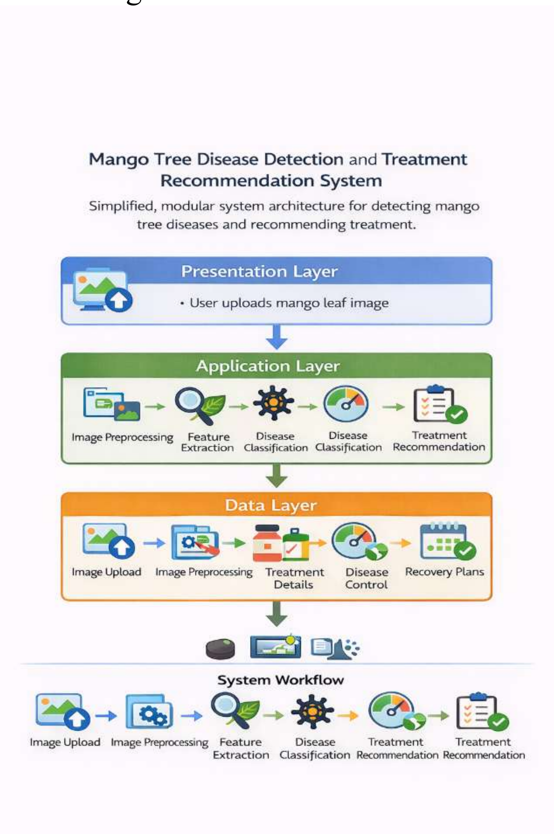
The data layer consists of a structured disease database implemented using Python dictionaries or JSON format. It stores disease-related information including symptoms, environmental conditions, severity levels, treatment solutions, dosage specifications, and recovery schedules. The modular database design allows future extension to relational or cloud-based databases.

D. System Workflow

The overall system workflow follows:

Image Upload → Preprocessing → Feature Extraction → Disease Classification → Severity Estimation → Treatment Recommendation → Recovery Plan Display

The modular system design ensures separation of concerns, easy maintenance, and future scalability. The architecture supports integration of Convolutional Neural Network (CNN) models and cloud-based deployment for real-time agricultural monitoring.



VII Security Model

Security is critical in cloud-based systems. According to Chandramouli et al. cryptographic controls and authentication are essential in cloud environments.

Security measures implemented:

- Role-Based Access Control
- Firebase Authentication
- Secure Transaction Verification
- Firestore Security Rules

Future enhancement may integrate multi-factor authentication.

VII. Conclusion

This paper presented a web-based Mango Tree Disease Detection and Treatment Recommendation System designed to support intelligent agricultural decision-making. The system integrates image preprocessing, feature extraction, rule-based classification, and severity estimation to accurately identify common mango leaf diseases. In addition to disease detection, the framework provides structured treatment recommendations, dosage guidance, and a time-based recovery plan. Unlike conventional classification-focused models, the proposed system emphasizes practical applicability by combining diagnostic accuracy with actionable treatment support. The modular architecture ensures scalability and enables future integration of Convolutional Neural Network (CNN) models for enhanced performance. The proposed solution demonstrates how intelligent web-based systems can contribute to precision agriculture, improve crop health management, and reduce economic losses caused by plant diseases.

VII References

[1] S. Mohanty, D. 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] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Computers and Electronics in Agriculture*, vol. 147, pp. 70–90, 2018.

[4] J. G. A. Barbedo, "A review on the main challenges in automatic plant disease identification based on visible range images," *Biosystems Engineering*, vol. 144, pp. 52–60, 2016.

[5] S. Sladojevic et al., "Deep neural networks based recognition of plant diseases by leaf image classification," *Computational Intelligence and Neuroscience*, 2016.

[6] H. Al-Hiary et al., "Fast and accurate detection and classification of plant diseases," *International Journal of Computer Applications*, vol. 17, no. 1, pp. 31–38, 2011.

[7] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.

[8] R. Szeliski, *Computer Vision: Algorithms and Applications*, Springer, 2010.

[9] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[10] W. Stallings, *Cryptography and Network Security*, 7th ed., Pearson, 2017.

[11] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," Google Research, 2016.

