PLANTGUARD AI

Ms. J. Aiswarya, M.Sc. (Associate Professor),

Department of Computer Applications,

Sri Krishna Arts and Science College, Coimbatore-641 008

Sanjay Karthi S, Department of Computer Applications,

Sri Krishna Arts and Science College, Coimbatore-641 008

ABSTRACT

Agriculture remains one of the most critical sectors globally, contributing significantly to economic stability and food security. However, plant diseases continue to pose major threats to crop yield, farmer income, and sustainable agricultural development. Early and accurate detection of plant diseases is essential to minimize losses and optimize treatment strategies. This paper presents **PlantGuard AI**, a deep learning-based full-stack web application designed for automated plant disease detection using transfer learning techniques. The system employs the MobileNetV2 convolutional neural network architecture trained on a dataset of 87,867 plant leaf images categorized into 38 disease classes. The model utilizes image preprocessing techniques including resizing to 224×224 pixels and data augmentation strategies such as rotation, zooming, and horizontal flipping to improve generalization performance. The training process uses the Adam optimizer with a learning rate of 0.001 and categorical cross-entropy as the loss function.

The system integrates a React.js 18 and Vite-based frontend for responsive and interactive user experience, styled using TailwindCSS and enhanced with Framer Motion animations. The backend

is developed using FastAPI (Python 3.11+) and exposes RESTful APIs for prediction, authentication, and data management. MongoDB 8.0 serves as the NoSQL database, accessed through PyMongo for persistent storage of user credentials, detection history, and metadata. Authentication is implemented using JWT tokens with bcrypt-based password hashing to ensure secure access control. The application allows users to upload leaf images, obtain real-time disease predictions with confidence scores, receive treatment recommendations, and generate downloadable PDF reports. The proposed system demonstrates how deep learning, modern web technologies, and secure database systems can be integrated into a scalable agricultural decision-support platform.

Keywords:

Plant Disease Detection, Deep Learning, Transfer Learning, MobileNetV2, Convolutional Neural Networks (CNN), Computer Vision, FastAPI, React.js, MongoDB, Full -Stack Web Application, Image Classification, Agricultural Decision Support System, JWT Authentication, Data Augmentation.

1. INTRODUCTION

Agriculture plays a vital role in sustaining global populations and supporting economic growth. However, crop diseases are one of the primary causes of reduced agricultural productivity. According to global agricultural studies, plant diseases are responsible for significant annual yield losses, impacting both large-scale agricultural industries and small-scale farmers.

Traditional plant disease detection methods rely heavily on manual inspection by agricultural experts. These approaches are time-consuming, subjective, and often inaccessible in rural or underdeveloped regions. Farmers frequently lack access to expert guidance, leading to delayed treatment and increased crop damage.

Recent advancements in Artificial Intelligence (AI), particularly Deep Learning (DL) and Computer Vision (CV), have revolutionized image-based classification tasks. Convolutional Neural Networks (CNNs) have demonstrated high accuracy in visual recognition applications, including medical diagnosis, object detection, and agricultural disease classification.

PlantGuard AI is designed to bridge the gap between advanced AI research and practical agricultural use. By deploying a deep learning model within a scalable full-stack web

architecture, the system enables real-time disease detection accessible through a standard web browser. This research focuses on the design, development, and implementation of an AI-powered web application capable of classifying plant diseases accurately and efficiently.

2. LITERATURE REVIEW

Numerous studies have explored the use of deep learning for plant disease detection. Early approaches relied on traditional image processing techniques such as color histogram analysis, texture features, and edge detection combined with classical machine learning classifiers like Support Vector Machines (SVM) and Random Forests. However, these methods were limited in scalability and accuracy.

With the emergence of deep convolutional neural networks, researchers began utilizing architectures such as AlexNet, VGG16, ResNet, and Inception for plant disease classification tasks. Transfer learning has become particularly popular due to its ability to leverage pretrained weights from large datasets such as ImageNet.

MobileNetV2, introduced by Sandler et al., is a lightweight CNN architecture optimized for mobile and embedded vision applications. It uses depthwise separable convolutions and inverted residual blocks to reduce computational complexity while maintaining high accuracy.

Existing research prototypes often lack full-stack deployment, user authentication, secure database storage, and scalable API integration. Many implementations remain limited to Jupyter notebooks or offline demonstrations.

PlantGuard AI addresses these limitations by integrating:

- A trained MobileNetV2 model
- FastAPI backend for inference
- React-based frontend
- MongoDB database storage
- Secure JWT authentication

This makes it not just a research prototype but a deployable real-world application.

3. PROBLEM STATEMENT

Plant diseases cause significant agricultural losses due to delayed detection and lack of

expert accessibility. Farmers in remote areas often rely on visual estimation, which may lead to incorrect diagnosis and improper pesticide usage.

Key challenges include:

- Lack of real-time disease identification systems
- Limited accessibility to agricultural experts
- Absence of centralized detection history
- Inconsistent disease classification accuracy
- Poor integration of AI models into usable platforms

The problem addressed in this research is:

How can deep learning-based image classification be integrated into a scalable, secure, and user-friendly web application to provide real-time plant disease detection and management?

4. SYSTEM ARCHITECTURE

PlantGuard AI follows a three-tier architecture:

4.1 Presentation Layer (Frontend)

- Built with React.js 18 and Vite
- Styled using TailwindCSS
- Animated using Framer Motion
- Navigation handled by React Router
- API communication via Axios

4.2 Application Layer (Backend)

- Developed using FastAPI
- RESTful API architecture
- Image preprocessing pipeline
- Model inference execution
- JWT authentication system
- CORS configuration

4.3 Data Layer (Database)

- MongoDB 8.0
- PyMongo integration
- Stores user information
- Stores detection history
- Stores metadata

5. METHODOLOGY

5.1 Dataset

The dataset contains 87,867 plant leaf images categorized into 38 disease classes. The dataset includes healthy and infected leaves across multiple crop types.

5.2 Data Preprocessing

- Image resizing to 224×224
- Normalization
- Data augmentation:
  - Rotation
  - Zoom
  - Horizontal flipping

5.3 Model Selection

MobileNetV2 was selected due to:

- Lightweight architecture
- Efficient computation
- High classification accuracy
- Suitable for real-time inference

5.4 Transfer Learning

Pretrained ImageNet weights were used. The top classification layer was replaced with a dense layer corresponding to 38 output classes.

5.5 Training Configuration

- Optimizer: Adam
- Learning Rate: 0.001
- Loss Function: Categorical Cross-Entropy
- Evaluation Metrics: Accuracy
- Batch Size: Configurable
- Epochs: Based on convergence

6. IMPLEMENTATION DETAILS

6.1 Frontend Implementation

The frontend provides:

- User registration and login
- Image upload interface
- Dashboard with detection history
- PDF report generation
- Notification system using React Hot Toast

6.2 Backend Implementation

FastAPI provides:

- /register endpoint
- /login endpoint
- /predict endpoint
- /history endpoint

Image files are processed and passed into the trained model for inference.

6.3 Security Implementation

- Password hashing using bcrypt
- JWT token-based authentication
- Protected API routes
- Environment variable configuration using .env files

7. DATABASE DESIGN

Two primary collections:

Users Collection

- user_id
- name
- email
- password_hash
- created_at

Detections Collection

- detection_id
- user_id
- image_path
- disease_name
- confidence_score
- timestamp

Indexes improve query efficiency.

8. SYSTEM TESTING

Testing was conducted at multiple levels:

Unit Testing

Individual API endpoints tested.

Integration Testing

Frontend-backend communication validated.

System Testing

Complete workflow tested from image upload to PDF generation.

Black Box Testing

Input validation and output verification.

White Box Testing

Code-level logic testing and route protection.

Validation Testing

Ensured accurate disease classification.

9. RESULTS AND PERFORMANCE ANALYSIS

The trained MobileNetV2 model achieved high classification accuracy on the validation dataset. The system demonstrated:

- Real-time inference capability
- Stable backend performance
- Secure authentication workflow

- Efficient database storage and retrieval

Latency was minimal due to optimized architecture.

10. DISCUSSION

PlantGuard AI demonstrates the feasibility of integrating deep learning models into production-ready web applications. Unlike research-only implementations, this system includes authentication, persistent storage, and scalable architecture.

Challenges encountered:

- Model size optimization
- Managing image upload performance
- Ensuring secure token management
- Database indexing for faster queries

11. FUTURE ENHANCEMENTS

- Cloud deployment (AWS/Azure)
- Mobile application version
- Real-time camera streaming detection
- Multi-language support
- Integration with IoT sensors
- Weather-based disease prediction models
- Advanced analytics dashboard

12. CONCLUSION

PlantGuard AI successfully integrates deep learning, full-stack development, and secure database management into a comprehensive plant disease detection system. By leveraging MobileNetV2 transfer learning, FastAPI backend services, and a responsive React frontend, the system provides accurate, scalable, and user-friendly disease classification. The research demonstrates how AI can be effectively deployed in agriculture to improve productivity and reduce crop losses.

13. REFERENCES

1. Sandler, M., et al. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks.
2. TensorFlow Documentation.
3. FastAPI Documentation.
4. React Official Documentation.