Plant Disease Detection with Machine Learning

Mansi Dupte, Khushi Moolya, Sakshi Maurya, Shweta Phanse, Prof. Sarita Bopalkar
Students, Department of Computer Science and Engineering (IOT and Cybersecurity Including Blockchain Technology), Smt. Indira Gandhi College of Engineering, Navi Mumbai, India
Guide, Department of Computer Science and Engineering (IOT and Cybersecurity Including Blockchain Technology), Smt. Indira Gandhi College of Engineering, Navi Mumbai, India

Abstract: Plant diseases significantly impact agricultural productivity, necessitating early and accurate detection methods. Traditional manual inspection techniques are time-consuming and prone to inaccuracies. In this study, we propose a convolutional neural network (CNN)-based model for plant disease detection, with a specific fopcus on tomato leaf diseases. The dataset used comprises 11,000 images sourced from the PlantVillage database, encompassing multiple disease classes and healthy samples. The proposed CNN model consists of convo2D, batch normalization, max pooling, global average pooling, dense and dropout convolutional layers, and 3 fully connected layers, and employs a learning rate of 0.0002, trained over 50 epochs. Compared with traditional machine learning approaches such support vector machines (SVMs) and random forests (RFs), our model has a superior training accuracy of 98.15% and a validation accuracy of 96.2%, significantly reducing the number of false positives. The model incorporates advanced techniques such as batch normalization and dropout regularization, ensuring robustness and generalizability. The experimental results demonstrate the model's efficacy in accurately diagnosing tomato diseases, providing a reliable tool for precision agriculture. The proposed system not only automates the disease identification process but also offers treatment recommendations, thus contributing to enhanced crop health management.

Keywords: **Plant Disease Detection, Convolutional Neural Network (CNN), Tomato Leaf Diseases, PlantVillage Dataset, Batch Normalization, Dropout Regularization, Precision Agriculture, Machine Learning (ML), Learning Rate, Accuracy Rate**

I. INTRODUCTION

The global agricultural sector faces significant challenges, including increasing demand for food due to population growth, climate change, and crop diseases. Plant diseases are a major threat to agricultural productivity, causing considerable losses in yield and quality. According to the Food and Agriculture Organization (FAO), plant diseases are responsible for reducing crop yields by up to 40%, which not only affects food security but also poses economic threats to farmers. The timely and accurate identification of plant diseases is, therefore, a critical task in modern agriculture.

Manual plant disease detection relies on manual observation, where experienced agronomists or farmers inspect the plants visually to identify symptoms. However, this method is labor intensive, time-consuming, and prone to human error. In many cases, by the time symptoms are visible, the disease may have already spread, leading to severe damage. Moreover, the lack of access to skilled experts, especially in remote regions, further complicates the process of early disease detection. These limitations highlight the need for automated, efficient, and scalable solutions.

A. Background and Motivation

Agriculture is a crucial sector that supports global food security and economic stability. However, plant diseases pose a significant threat to crop yields, leading to substantial economic losses and reduced productivity. Traditional disease detection methods, such as manual inspection by agricultural experts, are often time-consuming, subjective, and error-prone. With the rapid advancements in artificial intelligence (AI) and machine learning (ML), automated approaches have emerged as effective alternatives. Among these, convolutional neural networks (CNNs) have shown remarkable success in image-based classification tasks, making them an ideal choice for plant disease detection. CNNs can analyze plant leaf images to identify specific disease patterns with high accuracy, reducing reliance on human expertise.

The increasing demand for efficient, scalable, and precise disease detection solutions has motivated the integration of deep learning techniques in agriculture. Machine learning models can process large-scale datasets, enabling early detection of plant diseases, which is crucial for timely intervention and disease control. This project is driven by the need to:

1. By providing an automated, cost-effective, and scalable solution, economic losses due to crop infections can be reduced.
2. Compared with traditional methods, these methods increase disease detection accuracy, thereby minimizing human error.
3. Real time diagnosis can be enabled, allowing farmers to take preventive actions and improve overall crop health.
4. Advance smart agriculture practice and promote the adoption of AI-based technologies in precision farming.

To address these challenges, this research implements a CNN-based plant disease detection model that leverages image processing and deep learning to classify tomato leaf diseases accurately. The results demonstrate the potential of AI-driven solutions in revolutionizing modern agriculture and ensuring food security.

B. Objectives

The primary objective of this project is to develop a convolutional neural networks (CNN)-based model for the automated detection of tomato plant diseases via leaf images. The specific goals include the following:

1. **Accurate Classification**: To classify tomato leaf diseases with high precision via deep learning.
2. **Automated detection:** This eliminates the need for manual inspection, reducing human effort and error.
3. **Early Diagnosis**: To enable timely detection, farmers should take preventive measures.
4. **Optimized Performance**: To fine-tune the CNN model for high accuracy while maintaining computational efficiency.
5. **User-Friendly Deployment**: To design a system that can be integrated into real-world agricultural applications.

This project focuses exclusively on CNN-based image classification for plant disease detection. The scope includes:

1. **Dataset Preparation**: Preprocessing and augmentation of tomato leaf disease images.
2. **Model training and evaluation**: Implementing and training a CNN model to classify diseases.
3. **Performance Metrics**: Evaluating accuracy, and precision, to measure model effectiveness.
4. **Limited Disease Categories**: The system is trained on a predefined dataset covering nine tomato plant diseases.
5. **No other ML models**: The project does not use other machine learning techniques such as SVM, random forest, or decision trees, and uses only a CNN.
6. **Future enhancements**: Potential improvements include expanding the dataset, optimizing model performance, and integrating real-time mobile or IoT-based applications.

II. LITERATURE REVIEW

Over the past decade, significant efforts have been made to automate plant disease identification to address the limitations of traditional methods such as manual observation and expert diagnosis. These conventional approaches are often slow, expensive, and subjective, making them insufficient for large-scale farming operations, especially in developing countries. To overcome these challenges, researchers have turned to machine learning (ML) and deep learning (DL) techniques, which have proven effective in analyzing vast amounts of image data to identify disease patterns. Among these techniques, convolutional neural networks (CNNs) have emerged as the most widely adopted model because of their ability to automatically extract features from images and classify them with high accuracy. [1]

A. Challenges & Limitations of Existing Models

Traditional plant disease detection models that do not use convolutional neural networks (CNNs) face several challenges and limitations, including the following:

1. **Feature Extraction Dependency**: Traditional ML models (e.g., SVM and decision trees) rely on handcrafted features and require domain expertise and extensive preprocessing.
2. **Limited accuracy**: Feature-based methods struggle with complex patterns, leading to lower classification accuracy.
3. **Poor generalization**: Variability in lighting, leaf angles, and environmental conditions reduces model effectiveness.
4. **High False Positives/Negatives**: Traditional models may misclassify diseases, leading to ineffective treatments.
5. **Scalability Issues**: Many methods fail to scale across diverse datasets or plant species.
6. **Manual feature selection:** This requires careful feature engineering, unlike CNNs, which automatically learn patterns.
7. **Inability to handle Large Datasets**: Traditional models struggle with big data, whereas CNNs efficiently process large image datasets.
8. **Limited Adaptability**: Poor performance when dealing with new diseases or plant conditions.
9. **Less Robust to Noise**: Background variations and image distortions significantly affect model performance.

B. Emergence of CNN Models in Plant Disease Detection

The rise of convolutional neural networks **(CNNs)** has significantly transformed plant disease detection, addressing many limitations of traditional models. CNNs excel in image-based classification by automating feature extraction**,** reducing dependency on manual feature selection. Unlike conventional models, CNNs learn hierarchical patterns, making them highly accurate in detecting complex leaf diseases.

1. **High Accuracy & Robustness**: CNNs outperform traditional models by distinguishing fine-grained disease features, even under varying environmental conditions.
2. **Scalability & Adaptability**: CNNs generalize well across different plant species and disease types, improving detection efficiency.
3. **Automate feature learning**: Unlike SVM or decision trees, CNNs learn patterns directly from images without manual intervention.
4. **Handling Large Datasets**: CNNs effectively process vast datasets via parallel computation and GPU acceleration.
5. **Noise resistance**: CNN architectures can differentiate disease-affected regions from background noise, enhancing real-world applicability.

CNNs have revolutionized agricultural disease management, enabling **real-time detection, precision agriculture, and reduced crop losses**.

C. Research Findings

On the basis of a comparison of the literature and our proposed model, the following research findings are identified:

1. **Dataset Quality and Diversity:** Existing models rely on datasets that primarily consist of images taken under controlled environments, often capturing only single leaves in optimal lighting conditions. These datasets fail to incorporate real-world challenges such as varying lighting conditions, multiple leaves per image, and natural background noise, limiting model generalizability. Our approach overcomes this by utilizing the PlantVillage dataset with 11,000 images**,** ensuring diverse represntations of real-world scenarios. [1]

2. **Model architecture and performance:** Many previous studies have focused on general deep learning architectures such as VGG16, MobileNet, and ResNet for plant disease detection. While these architectures are effective, they may not be optimized for plant disease classification, often requiring many parameters and extensive computational resources. Our proposed CNN model integrates convo2D, batch normalization, max pooling, global average pooling, and dense and dropout layers with 13 fully connected layers, achieving 98.15% training accuracy and 96.2% validation accuracy, surpassing traditional models in performance and efficiency. [2]

3. **Automation and Practical Deployment:** Previous research has focused primarily on improving classification accuracy but has not emphasized real-time application and farmer accessibility. Many models lack a direct implementation framework, making them impractical for widespread use in agricultural settings. Our model addresses this by incorporating a real-time deployment system via a web-based application, allowing farmers to upload images and receive instant disease diagnosis and treatment recommendations. [3]

By addressing these research gaps, our model significantly improves the accuracy, efficiency, and real-world applicability of plant disease detection via CNNs. [1],[2],[3]

III. PROPOSED SYSTEM ARCHITECTURE

A. Overview

The proposed system for plant disease detection leverages convolutional neural networks (CNNs) to overcome the limitations of traditional approaches. CNNs are highly efficient in image classification tasks because they automatically extract relevant features from raw image data, eliminating the need for manual feature engineering. This deep learning-based solution enhances the accuracy, scalability, and robustness of plant disease detection by incorporating a diverse dataset, advanced preprocessing techniques, and optimized classification models.

B. System Components

1. **Image Acquisition and Data Collection:** The system captures high-resolution images of plant leaves, incorporating diverse datasets such as PlantVillage to ensure generalizability across different plant species, lighting conditions, and angles. This comprehensive data collection process enhances the model's ability to perform accurately in real-world agricultural environments.
2. **Image Preprocessing**: Advanced data augmentation techniques, including rotation, flipping, and zooming, are applied to artificially expand the dataset. This ensures that the model can learn to handle variations in real-world conditions, improving its robustness and adaptability to unseen data.
3. **Automated feature extraction via CNNs:** The CNN model learns to detect edges and contours in the initial layers. Texture and shape patterns in intermediate layers. Complex disease-specific features in deeper layers. This hierarchical feature learning enables the model to capture subtle variations in plant diseases that may not be detected by conventional feature extraction methods.
4. **Classification Layer**: The final layer of the CNN assigns disease labels to plant leaves, which provides accuracy scores for each disease category. The model classifies images into multiple disease types or as healthy, enabling confident and precise predictions.
5. **Model training**: The system is trained on a large and diverse dataset covering 11,000 tomato plant species, 9 disease types and 1 healthy type. Furthermore, cross-validation and early stopping are implemented to prevent overfitting and ensure that the model generalizes well to unseen data.
6. **User Interface and Deployment**: To facilitate real-time disease detection, a web-based interface is integrated, enabling users to upload tomato leaf images and receive instant disease diagnoses. The system also provides treatment recommendations, making it an effective tool for precision agriculture.

The proposed deployment ensures accessibility even in low-resource environments, improving its practicality for widespread agricultural applications.

Fig 3.1 Different Disease Types of Tomato Leaves

The above figure shows different types of diseased tomato leaves, totaling 9 disease types, which we have used to train our model.

C. Advantages of the proposed system:

1. **Deep architecture:** Multiple convolutional layers enable the model to extract complex patterns and features.
2. **Regularization Techniques:** Batch normalization and dropout reduce overfitting and improve generalizability.
3. **Data Augmentation:** Improves model robustness by introducing variations such as flipping, rotation, and zooming.
4. **Adaptive Learning:** The Adam optimizer and learning rate scheduling help the model converge faster and avoid local minima.
5. **Early Stopping:** Stop training at the optimal point to prevent overfitting.
6. **Efficient Data Pipeline:** Caching and prefetching improve the data loading speed and training efficiency.
7. **Best Model Saving:** Ensures that the best-performing model (on the validation data) is saved automatically.

Combining these techniques creates a well-balanced, efficient, and highly generalizable model.

IV. IMPLEMENTATION DETAILS

In this section, we delve into the technical aspects of developing plant disease detection with machine learning, including front- and back-end development, as well as database management and integration.

A. Front-End Development

The front-end of the plant disease detection system is designed to provide a user-friendly and interactive interface, allowing users to upload images of tomato leaves for disease classification. Developed via modern web technologies, the interface ensures seamless user experience, efficient data handling, and real-time interaction with the deep learning model. The front-end is built via the following technologies:

1. **HTML5**: To structure the web application and ensure compatibility across devices.
2. **CSS3**: For styling the interface and making it visually appealing.
3. **JavaScript (JS)**: To add interactivity and handle client-side operations.

Fig 3.2 Front-End Web Application of Plant Disease Detection

The above figure shows the web application features in a minimalistic yet functional design to maximize usability. The main components include the following:

1. **Image Upload Section**: Users can upload an image of a tomato leaf by clicking the “Choose an Image” button.
2. **Prediction button**: Once an image is selected, clicking the "predict" button triggers the disease classification model.
3. **Prediction Results Display**: The system presents the following:

- **Identified diseases** (e.g., late blight, early blight, etc.).
- **The confidence score indicates** the reliability of the prediction.
- **The description** includes information on tomato plant leaf disease.
- **Treatment suggestions**, including fungicide recommendations and preventive measures.

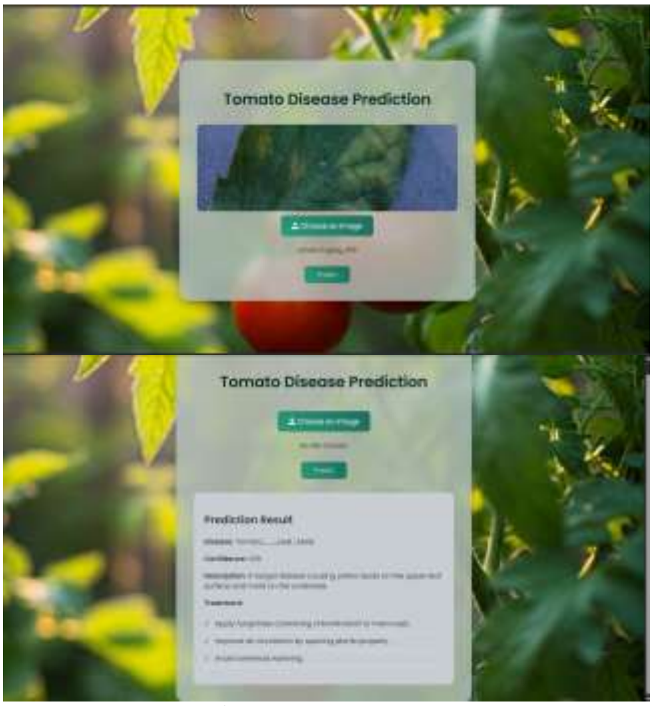

Fig 3.3 Result of the input image in Plant Disease Detection

The figure above shows the results of how the model analyzes the disease of tomato leaves and correctly gives a description, confidence, and three treatments required after.

B. Back-End Development

The back-end of the plant disease detection system is responsible for handling image processing, model inference, and API communication with the front-end interface. It is designed to use FastAPI to ensure high-speed responses, enabling real-time disease prediction with minimal latency.
The back-end system is developed via the following technologies:

1. **FastAPI**: A modern, asynchronous web framework for handling HTTP requests.

2. **TensorFlow/Keras**: Used for loading and running the trained convolutional neural network (CNN) model.
3. **NumPy**: Utilized for efficient numerical operations on image data.
4. **Logging & Exception Handling**: Implementing Python’s built-in logging library to capture errors and debug issues efficiently.

This is how the system workflow happens here:

1. **Image Uploading and Validation**: The system receives images from the front-end and checks their format (JPEG, PNG). File size validation ensures that the uploaded image does not exceed 5 MB. A heuristic function determines if the uploaded image is a tomato leaf before processing.
2. **Preprocessing and Model Inference**: Images are preprocessed via OpenCV and NumPy, including resizing (128×128 pixels) and normalization. The preprocessed image is passed through the trained CNN model to classify plant diseases.
3. **Prediction Generation**: The model outputs the predicted disease category and confidence score. The system retrieves treatment recommendations from a predefined knowledge base.
4. **API response handling**: The results, including disease type, confidence percentage, and treatment recommendations, are returned as a JSON response to the front-end.

The back-end is structured as a FastAPI with the following key endpoints:

1. **POST/predict:** Accepts image input and returns disease classification.
2. **GET/history:** Retrieves past predictions for a user.
3. **POST/feedback:** Stores user feedback for model improvement.

To ensure smooth deployment and scalability, the model is optimized as follows:

1. **Model quantization**: Reducing the model size without sacrificing accuracy for faster inference.
2. **Early Stopping & Learning Rate Scheduling**: Implemented to prevent overfitting and improve generalizability.

Layer (type)	Output Shape	Param #
conv2d_32 (Conv2D)	(None, 128, 128, 32)	896
batch_normalization_26 (BatchNormalization)	(None, 128, 128, 32)	128
max_pooling2d_32 (MaxPooling2D)	(None, 64, 64, 32)	0
conv2d_33 (Conv2D)	(None, 64, 64, 64)	18,496
batch_normalization_27 (BatchNormalization)	(None, 64, 64, 64)	256
max_pooling2d_33 (MaxPooling2D)	(None, 32, 32, 64)	0
conv2d_34 (Conv2D)	(None, 32, 32, 128)	73,856
batch_normalization_28 (BatchNormalization)	(None, 32, 32, 128)	512
max_pooling2d_34 (MaxPooling2D)	(None, 16, 16, 128)	0
global_average_pooling2d_9 (GlobalAveragePooling2D)	(None, 128)	0
dense_22 (Dense)	(None, 128)	16,512
dropout_11 (Dropout)	(None, 128)	0
dense_23 (Dense)	(None, 10)	1,290

Fig 3.4 CNN layers used to train the model

This image shows the architecture of a CNN model used for tomato leaf disease detection. It includes three convolutional layers with batch normalization and max pooling, followed by global average pooling, dense layers with dropout, and a final softmax output layer for classifying 10 disease types.

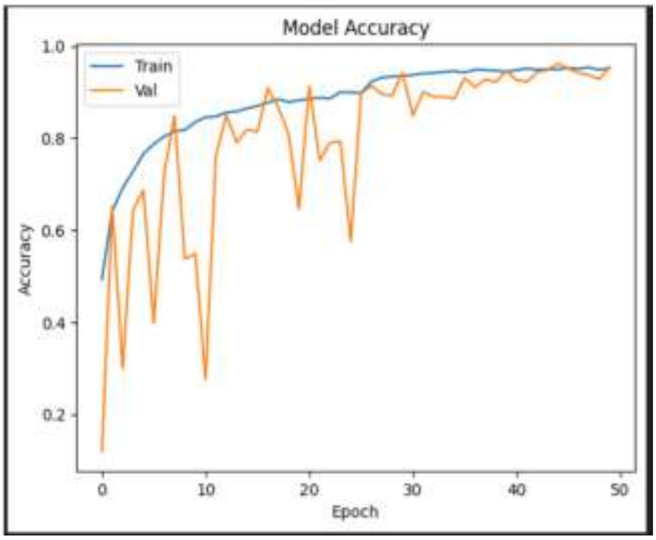

Fig 3.5 Graph of Model Accuracy

This graph shows the training and validation accuracy of the CNN model over 50 epochs. The model demonstrates a steady improvement in accuracy, with the training accuracy reaching nearly 100% and the validation accuracy stabilizing approximately 96%, indicating effective learning and generalization.

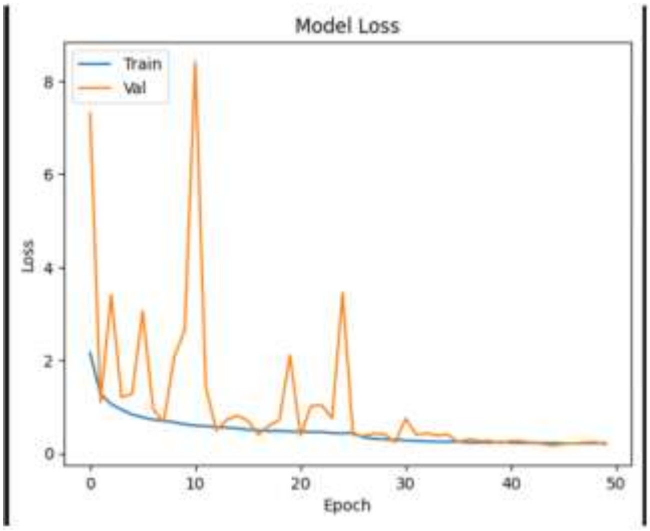

Fig 3.6 Graph of Model Loss

This graph displays the training and validation loss over 50 epochs. The training loss consistently decreases, whereas the validation loss initially fluctuates but eventually stabilizes, indicating effective training with minimal overfitting.

C. Results

Dataset Preparation	Image size: **(256, 256)** Mini-batch size: **32**
Data Visualization	Randomly selects **12 images** from the dataset Displays in a **3×4 grid** with class names
Data Preprocessing	**Normalization:** Scales pixel values from **[0, 255] to [0, 1]** **Caching & Optimization:** [.cache()] keeps data in memory [.shuffle(1000)] shuffles training data [.prefetch(AUTOTUNE)] loads batches in advance
Data Augmentation	Prevents overfitting with **random transformations**: RandomFlip (horizontal) RandomRotation (10%) RandomZoom (10%) RandomBrightness (20%)
CNN Model Architecture	**Sequential CNN Model**: Input: **128×128 RGB images** **Three convolutional layers**: Filters: **32 → 64 → 128**, Activation: **ReLU** **Batch Normalization** for stable training **MaxPooling** for downsampling **Global Average Pooling →** Reduces parameters **Fully Connected Layer**: Dense 128, L2 Regularization, **Dropout (60-70%)** **Output Layer:** Softmax activation, **10 classes**
Model Compilation	**Optimizer:** Adam (adaptive learning) **Loss Function:** sparse_categorical_crossentropy **Evaluation Metric:** Accuracy
Data Augmentation (ImageDataGenerator)	Additional transformations applied: **Rotation, shifting, shearing, zooming, flipping, brightness adjustment**
Training Configuration	**Early Stopping:** Stops training when val_loss doesn't improve for **10 epochs** **ReduceLROnPlateau:** Reduces learning rate if val_loss doesn't improve for **3 epochs** **Training duration: 50 epochs**
Key Features & Optimizations	Image Preprocessing & Normalization for better convergence Data Augmentation to avoid overfitting Batch Normalization for stable training Dropout Regularization to prevent overfitting Global Average Pooling to reduce parameters Early Stopping & Learning Rate Reduction for efficiency

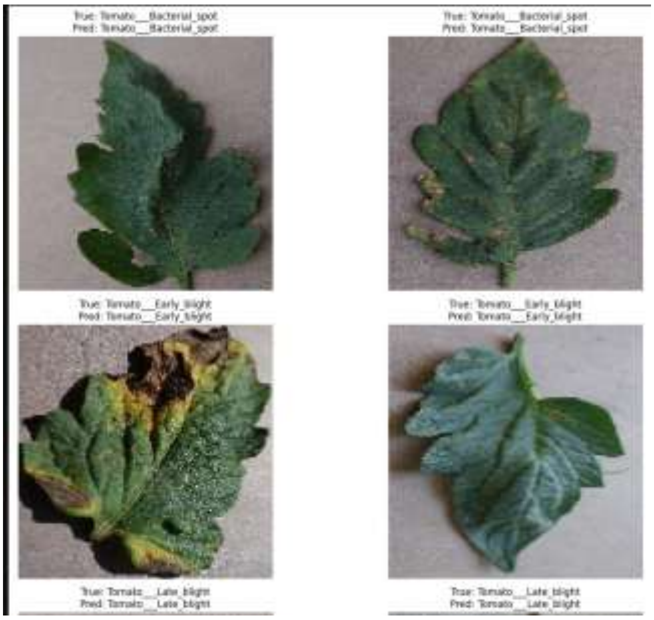

Fig 3.7 Results for the final tomato plant leaves

This image shows the prediction results of the CNN model on tomato leaf samples. The model accurately identifies different diseases such as bacterial spot, early blight, and late blight with the predicted labels matching the true labels in all four cases.

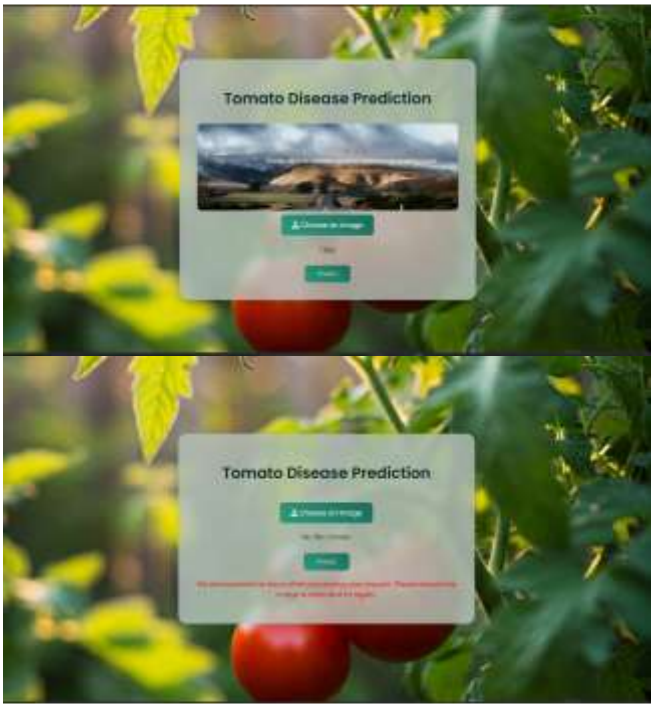

Fig 3.8 Random image results

The figure given above shows that the model detects the random image, and asks to input the right image to identify and provide the desired results.

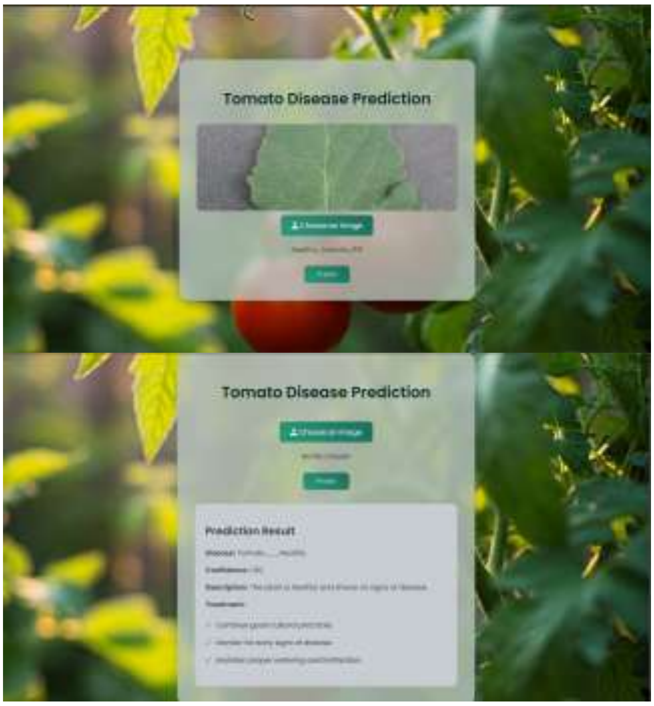

Fig 3.9 Healthy image results

The above figure shows how the model detects the healthy image and shows the correct description with confidence and treatment.

V. IMPLICATIONS AND FUTURE WORK

The Plant Disease Detection System using CNNs has significant implications in the field of precision agriculture, contributing to early disease detection, reduced crop losses, and improved yield quality. The key implications include the following:

1. **Enhanced Disease Diagnosis**: This system enables accurate identification of tomato leaf diseases, allowing farmers to take prompt corrective actions.
2. **Reduction in Chemical Usage**: By providing specific treatment recommendations, the model helps minimize the overuse of pesticides and fungicides, leading to sustainable farming practices.
3. **Scalability for Large-Scale Implementation**: The system can be integrated into agricultural advisory services and used by farmers, researchers, and agronomists to monitor crop health efficiently.
4. **Automation in Smart Farming**: This technology can be extended to autonomous drones and IoT-enabled smart farms, allowing real-time disease detection and automatic treatment applications.
5. **Economic and Environmental Benefits**: By reducing crop damage and unnecessary chemical use, the system promotes cost-effective and eco-friendly farming solutions.

Although the system demonstrates promising results, several areas for improvement and expansion remain. The following enhancements are planned for future research:

1. **Expansion of the dataset**: Increasing the dataset size with diverse plant species, environmental conditions, and disease variations can improve the model's generalizability and accuracy.
2. **Integration of multimodal data:** Future versions may incorporate sensor-based data (temperature, humidity, and soil conditions) along with image-based analysis for more robust disease prediction.
3. **Mobile application development**: Developing a mobile-friendly application with offline capabilities will increase accessibility for farmers in remote areas with limited internet connectivity.
4. **Edge and Fog Computing Implementation**: Deploying the model on edge devices (e.g., Raspberry Pi, NVIDIA Jetson) for real-time, low-latency inference in the field without dependency on the cloud infrastructure.
5. **Explainable AI (XAI) Integration**: Implementing explainability techniques to provide transparent and interpretable predictions, helping farmers understand the model's reasoning for each diagnosis.
6. **Real-Time Monitoring with Drones**: Future work may involve automated drone-based imaging and disease detection to monitor large farmlands efficiently and continuously.
7. **Blockchain for Data Security**: Using blockchain-based decentralized databases to securely store and share disease data ensures traceability and reliability of agricultural insights.
8. **Cross-Platform Deployment**: Making the system compatible with different operating systems and hardware to ensure wider adoption and usability.

The proposed plant disease detection system lays a strong foundation for AI-driven agricultural solutions, significantly impacting crop health monitoring and precision farming. By addressing the limitations and incorporating advanced technologies such as the IoT, blockchain, and edge computing, future iterations of the system will further improve the efficiency, accessibility, and sustainability of smart agriculture.

VI. CONCLUSION

This study presents a deep learning-based approach for automated plant disease detection, addressing the limitations of traditional methods such as manual inspection and feature-based machine learning models. By leveraging a CNN of 13 layers, which consists of convo2D, batch normalization, max pooling, global average pooling, dense and dropout convolutional layers and optimized hyperparameters, our model achieves a superior training accuracy of 98.15% a validation accuracy of 96.2%, and a learning rate of 0.0002. The model effectively differentiates between multiple disease categories, ensuring rapid and precise identification. Additionally, integrating treatment suggestions enhances its practical applicability for farmers and agricultural experts.

Future work will focus on expanding the dataset to include real-world images from diverse environmental conditions, optimizing the model for mobile and edge computing applications, and improving interpretability through explainable AI techniques. The proposed approach has significant potential for revolutionizing agricultural disease management, minimizing crop losses, and promoting sustainable farming practices.

1. **Corresponding author:** Khushi Moolya, Sakshi Maurya, Shweta Phanse, Prof. Sarita Bopalkar **Email:** 2021ci38@sigce.edu.in, 2021ci39@sigce.edu.in, 2021ci50@sigced.edu.in, sarita.bopalkar@sigce.edu.in
2. **Funding:** This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.
3. **Ethics Declaration:** Not applicable.
4. **Data Availability:** The dataset used in this study is publicly available and can be accessed from the PlantVillage dataset repository: https://plantvillage.psu.edu. The processed dataset and trained model will be made available upon reasonable request from the corresponding author.
5. **Consent to Publish:** Not applicable.
6. **Author Contributions:** Shweta Phanse and Mansi Dupte conceptualized the research idea and led the model development. Khushi Moolya performed data collection, preprocessing, and experimentation. Sakshi Maurya contributed to results analysis, visualizations, and front-end integration. And Prof. Sarita Bopalkar helped us throughout this process to make sure the model gets trained well and gives accurate results. All authors participated in writing and revising the manuscript. All authors have read and approved the final version of the paper.
7. **Competing Interests:** The authors declare that they have no competing interests.

REFERENCES

[1] Joseph, D. S., Pawar, P. M., & Chakradeo, K. (2024). *Real-time plant disease dataset development and detection of plant disease via deep learning*. IEEE Access.
[2] Aria Dolatabadian, Ting Xiang Neik, Monica F. Danilevicz, Shriprabha R. Upadhyaya, Jacqueline Batley, David Edwards (2024) *Image Based Crop Disease Detection Using Machine Learning*
[3] Sahil Verma, Prabhat Kumar, Jyoti Prakash Singh (2023) *A Unified Lightweight CNN-based Model for Disease Detection and Identification in Corn, Rice, and Wheat*
[4] Nain, S., Mittal, N., & Hanmandlu, M. (2024). *CNN-based plant disease recognition via color space models*. Taylor & Francis.
[5] Verma, S., Kumar, P., & Singh, J. P. (2024). *A unified lightweight CNN-based model for disease detection and identification in corn, rice, and wheat*. Taylor & Francis.
[6] Sherly, K. K., & Sonia, A. (2024). *Hybrid CNN Models for Plant Species Recognition and Disease Detection*. Springer.
[7] PlantVillage Dataset. A publicly available dataset for plant disease detection.
[8] TensorFlow Documentation. (2023). *TensorFlow for Deep Learning*. [Online] Available at: https://www.tensorflow.org/
[9] PyTorch Documentation. (2023). *PyTorch for Machine Learning*. [Online] Available at: https://pytorch.org/\
[10] OpenCV Documentation. (2023). *Open Source Computer Vision Library*. [Online] Available at: https://opencv.org/