Plant Disease Detection System for Sustainable Agriculture

A Project Report

submitted in partial fulfillment of the requirements

of

AICTE Internship on AI: Transformative Learning
with
TechSaksham – A joint CSR initiative of Microsoft & SAP

by

Aarif Ahmad Mir, miraarif583@gmail.com

Under the Guidance of

P. Raja, Master Trainer at Edunet Foundation

ACKNOWLEDGEMENT

We express our deepest gratitude to **P. Raja**, **Master Trainer at Edunet Foundation**, for his invaluable mentorship and guidance throughout the development of this project, *Plant Disease Detection System for Sustainable Agriculture*. His expertise in machine learning and his continuous support enabled us to design and implement a robust solution for plant disease detection using deep learning techniques.

We are immensely thankful to **Pavan Sumohana**, whose meticulous organization and seamless coordination of the class schedules greatly contributed to the smooth progression of this program. His dedication to ensuring timely updates and availability of resources allowed us to focus entirely on learning and project execution.

We also extend our heartfelt appreciation to the **Edunet Foundation** and its dedicated team for providing this incredible opportunity under the *TechSaksham initiative*, a CSR initiative by **Microsoft and SAP**. This initiative not only enriched our understanding of machine learning but also highlighted the importance of leveraging technology for sustainable agriculture.

Lastly, we acknowledge all the trainers, participants, and stakeholders for fostering a collaborative and intellectually stimulating environment. Their collective efforts have been instrumental in shaping this project into a meaningful contribution to the field of agriculture.

Thank you all for your invaluable support and encouragement!

ABSTRACT

Plant diseases significantly threaten global agriculture, impacting crop yield, quality, and sustainability. Farmers often struggle with early disease detection, leading to delayed treatment and further losses. This project, *Plant Disease Detection System for Sustainable Agriculture*, aims to address this challenge by leveraging machine learning to create a reliable, accessible, and efficient disease detection system.

The primary objective is to classify and identify plant diseases using leaf images, enabling timely intervention to prevent crop damage. The methodology employs a Convolutional Neural Network (CNN)-based architecture, trained on the dataset, which consists of over 38 classes of diseased and healthy plant leaf images. The model was fine-tuned with data augmentation and regularization techniques to improve its generalization capability and accuracy.

Key results demonstrate the model's robustness, achieving an accuracy of over 95% on test data. The system was integrated into a user-friendly web application using Streamlit, allowing farmers to upload images of diseased leaves for real-time diagnosis. The model predicts the specific disease, empowering users to take prompt and informed action.

In conclusion, the project provides an innovative and practical solution to the agricultural sector's pressing problem. By employing cutting-edge deep learning techniques, this system contributes to sustainable farming practices, minimizing losses and ensuring food security. It highlights the potential of artificial intelligence in addressing real-world challenges and its transformative role in agriculture.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

1.1 Problem Statement:

Agricultural productivity is central to ensuring food security and sustainable development, with plant diseases representing a significant threat to crop health and yield. These diseases lead to substantial economic losses for farmers globally. Traditional plant disease identification relies on expert manual inspection, a method that is time-consuming, labour-intensive, and susceptible to inaccuracies due to human error. Moreover, the lack of access to skilled professionals in rural or underdeveloped areas worsens the situation.

Recent advancements in machine learning and deep learning technologies present an opportunity to automate plant disease detection. These systems can efficiently and accurately identify diseases in crops, providing timely information that helps mitigate losses. Moreover, the early and timely identification of plant diseases positively impacts crop yield and quality [1]. Beyond economic considerations, early disease detection also contributes to more sustainable agricultural practices by reducing chemical pesticide use, preserving biodiversity, and promoting healthier crops.

Due to the cultivation of a large number of crop products, even an agriculturist and pathologist may often fail to identify the diseases in plants by visualizing disease-affected leaves. However, in the rural areas of developing countries, visual observation is still the primary approach of disease identification [2]. It also requires continuous monitoring by experts. In remote areas, farmers may need to travel far to consult an expert, which is time-consuming and expensive [3,4]. Automated computational systems for the detection and diagnosis of plant diseases assist farmers and agronomists with their high throughput and precision.

Conversely, deep learning methods, especially Convolutional Neural Networks (CNNs), stand out as one of the most effective approaches for automatically extracting key and distinguishing features. Deep learning (DL) consists of different convolutional layers that represent learning features from the data [5,6].

This project proposes a machine learning-based system designed to overcome the limitations of traditional disease detection methods. By using image data to accurately identify plant diseases, the system provides a scalable, efficient, and user-friendly solution that can aid farmers, researchers, and policymakers in enhancing agricultural productivity and sustainability.

1.2 Motivation:

The inspiration behind this project stems from the growing realization of the critical role that advanced technologies, such as machine learning, can play in addressing long-standing challenges in agriculture. With the agricultural sector forming the backbone of many economies, particularly in developing countries, plant diseases continue to pose a serious threat to food security and the livelihood of millions of farmers. These challenges demand innovative and scalable solutions that go beyond traditional methods of disease detection.

A key driver for this project is the potential to democratize access to reliable plant disease detection systems. Many farmers, especially in rural or resource-limited settings, lack access to expert pathologists or agronomists for timely disease diagnosis. This gap often leads to delayed interventions, resulting in preventable losses. By leveraging the power of deep learning models like Convolutional Neural Networks (CNNs), this project aims to bridge this accessibility gap and provide farmers with a tool that is not only accurate but also user-friendly and accessible.

Furthermore, this project aligns with the global vision of sustainable agriculture. By enabling early disease detection, it minimizes the overuse of pesticides, preserving soil health and biodiversity. Such initiatives also support environmental goals by promoting eco-friendly farming practices and reducing the carbon footprint associated with agricultural production.

The personal drive to contribute to this domain stems from recognizing the transformative impact of AI-driven solutions in everyday challenges. The prospect of applying cutting-edge technology to create a tangible, positive change in the lives of farmers and agricultural communities is a motivating factor. Ultimately, this project aspires to not only address the immediate issue of plant disease detection but also set the stage for future innovations in precision agriculture.

1.3 Objective:

The primary objectives of the project are as follows:

- **Accurate Plant Disease Identification**: Develop a machine learning model capable of identifying and classifying plant diseases across 38 distinct classes with high accuracy using images of diseased and healthy leaves.
- **Facilitate Early Disease Detection**: Enable early detection of plant diseases to allow farmers and agricultural stakeholders to take timely remedial actions, thereby minimizing crop losses and ensuring better yield.
- **User-Friendly System for Farmers:** Design an intuitive and accessible application using Streamlit that can be used by farmers with minimal technical knowledge for real-time disease diagnosis.

- **Promote Sustainable Agriculture**: Reduce reliance on traditional chemical-intensive approaches to manage plant diseases by encouraging targeted interventions, thus promoting environmentally friendly farming practices.
- **Scalable and Efficient Solution**: Create a solution that can be easily scaled to accommodate more crop types and diseases, making it adaptable to different agricultural contexts.
- **Leverage Deep Learning for Practical Applications**: Demonstrate the practical application of convolutional neural networks (CNNs) in solving agricultural problems, highlighting the role of artificial intelligence in driving innovation in farming practices.
- **Enhance Accessibility for Rural Communities**: Ensure the system is affordable and easily deployable in rural areas where expert agricultural advice is often inaccessible, bridging the gap between technology and underserved farming communities.

These objectives collectively aim to address critical challenges in agriculture, aligning the project with global goals of food security and sustainability.

1.4 Scope of the Project:

1.4.1 Scope

- **Wide Range of Disease Detection**: The project focuses on detecting 38 different plant diseases across multiple crop types using deep learning techniques, providing a comprehensive solution for farmers.
- **Real-Time Disease Recognition**: The user-friendly application facilitates real-time disease diagnosis by allowing users to upload images and receive immediate predictions about the disease type.
- **Integration of Deep Learning**: Leveraging a Convolutional Neural Network (CNN) model trained on a well-curated Plant Village dataset, the system achieves high accuracy in disease classification.
- **Promotes Sustainable Agriculture**: The project supports precision farming by identifying diseases accurately, minimizing unnecessary pesticide use, and improving environmental sustainability.
- **Scalability**: The architecture of the project allows for future expansion, enabling the addition of more crop types and diseases as new data becomes available.

- **Accessibility**: By providing a Streamlit-based interface, the system is accessible even to individuals with limited technical expertise, ensuring practical usability for farmers and agricultural consultants.

1.4.2 Limitations

- **Dataset Dependence**: The model is trained on the Plant Village dataset, which includes controlled, high-quality images. Real-world images with variations in lighting, angle, or quality may affect prediction accuracy.
- **Limited to 38 Classes**: The current scope is restricted to 38 diseases and healthy leaf classifications. It does not cover all possible plant diseases or crops.
- **No Severity Analysis**: The system identifies diseases but does not assess their severity, which could be critical for determining the urgency of intervention.
- **Infrastructure Requirements**: While the model is lightweight for deployment, users still require internet access and a device capable of uploading images, which may pose challenges in rural areas with limited infrastructure.
- **Focus on Leaf Images**: The system relies exclusively on leaf images for disease identification. Other parts of the plant, such as stems or fruits, are not analyzed.
- **Generalization Across Regions**: Plant diseases often exhibit regional variations, which may not be accounted for in the dataset used for training the model, potentially limiting its accuracy in different geographical contexts.
- **Edge Cases**: Rare or newly emerging diseases not included in the dataset will not be identified by the system, necessitating periodic updates to the model.
- **Manual Input Dependency**: Users must manually capture and upload images, which can lead to errors if the images are poorly taken or do not accurately represent the plant's condition.

CHAPTER 2

Literature Survey

2.1 Literature Review

Shiroop Madiwalar and Medha Wyawahare analyzed different image processing approaches for plant disease detection in their research [7]. Authors analyzed the color and texture features for the detection of plant disease. They have experimented their algorithms on the dataset of 110 RGB images. The features extracted for classification were mean and standard deviation of RGB and YCbCr channels, grey level cooccurrence matrix (GLCM) features, the mean and standard deviation of the image convolved with Gabor filter. Support vector machine classifier was used for classification. Authors concluded that GCLM features are effective to detect normal leaves. Whereas color features and Gabor filter features are considered as best for detecting anthracnose affected leaves and leaf spot respectively. They have achieved highest accuracy of 83.34% using all the extracted features.

Garima Shrestha et Al. deployed the convolutional neural network to detect the plant disease [8]. Authors have successfully classified 12 plant diseases with 88.80% accuracy. The dataset of 3000 high resolution RGB images were used for experimentation. The network has 3 blocks of convolution and pooling layers. This makes the network computationally expensive. Also the F1 score of the model is 0.12 which is very low because of higher number of false negative predictions.

Verma, Gaurav, Taluja, Charu, and Saxena, Abhishek Kumar. "Vision Based Detection and Classification of Disease on Rice Crops Using Convolutional Neural Network" (2019). The study by Verma, Taluja, and Saxena utilized a convolutional neural network (CNN) for the accurate detection and classification of diseases in rice crops [9]. By training the CNN on a large dataset of diseased and healthy rice leaves, the model achieved promising results in identifying and categorizing various diseases affecting rice plants.

S. D.M., Akhilesh, S. A. Kumar, R. M.G., and P. C. "Image based Plant Disease Detection in Pomegranate Plant for Bacterial Blight" (2019). At the 2019 International Conference on Communication and Signal Processing (ICCSP), S. D.M. and colleagues presented research on image-based plant disease detection in pomegranate plants for bacterial blight [10]. Their approach utilized various image processing and machine learning techniques to extract relevant features and classify diseased and healthy samples, showcasing the potential of image-based methods in accurate disease diagnosis.

Kumar, M., Gupta, P., Madhav, P., and Sachin. "Disease Detection in Coffee Plants Using Convolutional Neural Network" (2020). Kumar and colleagues focused on disease detection in coffee plants using a convolutional neural network (CNN) [11]. By training the CNN on a dataset of coffee leaf images, the authors achieved significant accuracy in disease detection, highlighting the potential of CNNs in automated diagnosis systems for plant diseases.

Table 1. Summary of CNN models that detect the plant disease

Reference	**Crop Focus**	**Dataset**	**Classes**	**Model**	**Limitations**	**Results**
[12]	Soybean leaves	Self-generated database	3	AlexNet and GoogleNet CNNs	The model slipped in Classification diversity. Many current models focus on defining a single class of plants disease instead of building a model to classify different plant diseases.	98.75% and 96.25%
[13]	Tomato	Conditional Generative Adversarial Network for generation	10	DenseNet121 model	Reliance on synthetic data may hinder model's real-world performance.	97.11%
[14]	Several	PlantVillage	38	MobileNet V3	Limited assessment of deep learning models' real-world performance on edge devices.	96.58%
[15,16]	Tomato	the Plant village and Taiwan tomato leaves,	13	CNN model	Benchmark studies lacked realism, affecting model performance in practical scenarios.	95.98%
[17]	Several	PlantVillage	39	VGG16	Solely emphasizes tomato crop disease identification, limiting broader applicability	94.9%

[18]	Tomato plant	Self-generated database	9	Faster Region based CNN and Region based Fully Convolutional Network	Limited realism hampers model's efficacy in real-time disease recognition.	85.98%
[19]	Apple	AI-Challenger plant disease recognition	6	DenseNet-121	Insufficient exploration of apple leaf disease recognition challenges and solutions	93.71%
[20]	Several	Public database	7	Pre-trained models	Limited analysis of deep transfer learning for plant disease identification	91.83%
[21]	Rice	Self-generated database	4	Pre-trained CNN with SVM classifier	Focused on color features, potentially overlooking other relevant disease indicators.	94.65%

2.2 Existing Models, Techniques, and Methodologies

2.2.1 Models

- **Convolutional Neural Networks (CNNs):** Convolutional Neural Networks (CNNs) have become the most widely used deep learning model for image classification tasks, including plant disease detection. CNNs are designed to automatically extract features from images through multiple layers such as convolutional layers, pooling layers, and fully connected layers. This enables them to capture spatial hierarchies in images, making them particularly effective for plant disease detection tasks. Many studies have successfully implemented CNNs to classify plant diseases, achieving high accuracy.
- **Transfer Learning Models (e.g., ResNet50, VGGNet):** Transfer learning is an effective approach when there is a limited amount of labeled data. Pre-trained models such as ResNet50 or VGGNet, which were originally trained on large datasets like ImageNet, are fine-tuned on plant disease datasets. This method has been widely used to save computational resources and training time while achieving competitive performance. Transfer learning has shown strong results

in plant disease detection as it reduces the need for extensive training from scratch.

- **DenseNet (Deeply Supervised Networks):** DenseNet is an advanced CNN architecture where each layer receives input from all previous layers. This enables more efficient learning and better feature reuse, which improves the model's performance in complex tasks like plant disease detection. DenseNet has been used in plant disease detection to analyze high-resolution images and complex disease patterns, achieving high accuracy due to its compact and efficient feature representation.
- **Hybrid Models (CNN + SVM):** Hybrid models combine the strengths of different algorithms. For example, CNNs are used for automatic feature extraction from images, while Support Vector Machines (SVMs) are employed for the classification task. The hybrid approach combines CNN's ability to handle complex visual data with SVM's efficient classification capabilities, making it ideal for multi-class problems such as detecting various plant diseases.
- **Recurrent Neural Networks (RNNs):** Although less commonly used in plant disease detection, RNNs can be useful for analyzing time-series data or sequential image data. When disease progression over time is crucial, RNNs can model the temporal changes in plant health. RNNs are being explored in combination with CNNs to enhance the understanding of plant disease development through sequential imagery.

2.2.2 Techniques

- **Data Augmentation:** Data augmentation techniques such as image rotation, flipping, scaling, and color adjustments are used to artificially expand the dataset. This is especially useful in plant disease detection where the available data may be limited or imbalanced. By applying these techniques, models can generalize better, leading to improved performance on unseen data.
- **Ensemble Learning:** Ensemble learning techniques, like bagging, boosting, and stacking, combine the predictions of multiple models to improve accuracy and robustness. Random Forests and AdaBoost are two popular ensemble techniques that are often combined with deep learning models to enhance their performance, reduce overfitting, and handle noisy data more effectively.
- **Attention Mechanisms:** Attention mechanisms allow models to focus on the most relevant parts of an image. In plant disease detection, attention mechanisms help the model concentrate on the diseased parts of the plant, which improves prediction accuracy. Models like Attention CNNs leverage this technique to enhance the model's understanding of disease patterns in plant leaves.

2.2.3 Methodologies

- **Preprocessing and Feature Extraction**: Data preprocessing, including image resizing, normalization, and augmentation, is a crucial step to prepare the data for model training. Feature extraction involves using methods like CNNs to automatically extract useful features from images, which are then used for classification. This methodology is central to improving the model's ability to correctly identify diseases in plants.
- **Model Evaluation:** Evaluation metrics such as accuracy, precision, recall, and F1-score are commonly used to assess the performance of plant disease detection models. Cross-validation and hyperparameter tuning are also important techniques to improve model performance and prevent overfitting. Additionally, confusion matrices are used to understand how well the model classifies each disease type.
- **Model Deployment:** Once the model has been trained and evaluated, deployment is an important aspect. In plant disease detection, models are often deployed as web applications or mobile apps to make the predictions accessible to farmers and agriculture experts. Tools like Streamlit, Flask, and TensorFlow.js are commonly used for deploying models in real-world applications, enabling users to upload images and receive predictions in real time.

2.3 Gaps in Existing Solutions and How This Project Addresses Them.

Despite the progress in the development of plant disease detection systems, there remain several gaps and limitations in existing solutions. These limitations often hinder the widespread adoption and practical implementation of these systems in real-world agricultural settings. Below is a detailed discussion of these gaps, followed by an explanation of how our project addresses them:

1. Limited Generalization Across Diverse Plant Species and Diseases

- **Existing Limitation:** Many existing plant disease detection models are trained on small datasets or datasets that contain only a limited number of plant species and diseases. These models, therefore, often fail to generalize well to new, unseen diseases or species, especially those that were not part of the training data. In practice, this means that a plant disease detection system trained on a few types of crops may not work well when applied to other crops or diseases not covered in the training phase.
- **How Our Project Addresses It**: Our project tackles this limitation by using a large, diverse dataset that includes 38 distinct plant diseases spread across various plant species. This comprehensive dataset enables our model to generalize better across different plants and diseases. By training the model on a wide variety of images, we ensure that the system can recognize and accurately diagnose diseases

from various plant species, even those that may not be commonly seen in the original training data. This makes the solution applicable to a wider range of agricultural settings, particularly in regions with diverse crops.

2. **Inability to Handle Variations in Environmental Conditions**
   - **Existing Limitation:** Plant disease detection models often struggle to maintain high accuracy under varied environmental conditions. Changes in lighting, background noise, or image quality can significantly affect the performance of these models. For example, poor lighting conditions or blurry images often result in incorrect or unreliable predictions. This limitation is especially problematic for mobile applications or field-based systems where environmental conditions are not controlled.
   - **How Our Project Addresses It:** Our project employs data augmentation techniques to make the model more robust to variations in image quality and environmental conditions. Techniques like random rotation, zooming, cropping, flipping, and brightness adjustment were applied during the training phase. These augmentations simulate the different lighting conditions, angles, and image distortions that might occur in real-world scenarios. As a result, the model becomes more flexible and capable of performing well on images taken under various conditions, improving the accuracy and reliability of predictions in the field.

3. **Limited Accessibility and Usability in Field Settings**
   - **Existing Limitation**: Many existing plant disease detection systems are designed for controlled laboratory environments or require high-end infrastructure and stable internet connections. However, these requirements are not always feasible in agricultural fields, especially in rural areas where farmers may not have access to high-speed internet or advanced hardware. Additionally, some systems are not designed with ease of use in mind, making them difficult for non-technical users (such as farmers) to operate.
   - **How Our Project Addresses It:** Our project is specifically designed for field deployment. The system is integrated into a Streamlit-based web application that can be accessed on any smartphone or computer with minimal hardware requirements. Since Streamlit allows for the creation of interactive applications that can run locally, the system does not require a constant internet connection for predictions. This is especially useful for farmers in rural areas who may not have reliable access to high-speed internet. Moreover, the user interface is simple and intuitive, designed to be accessible to farmers with no prior technical knowledge. The app allows users to easily upload images, view predictions, and receive actionable recommendations for disease management.

4. **Inability to Detect Multiple Diseases Simultaneously**
   - **Existing Limitation:** Some systems are designed to detect only a single disease at a time, limiting their applicability in areas where multiple diseases may affect crops simultaneously. Farmers often face the challenge of dealing with several plant diseases at once, and systems that cannot differentiate between multiple diseases are less useful.
   - **How Our Project Addresses It:** Our system is capable of detecting and classifying multiple diseases across 38 different plant types. Using a multi-class classification approach, our model is trained to recognize a wide range of plant diseases, making it capable of distinguishing between various diseases and identifying the specific disease affecting a plant. This feature is essential for real-world agricultural applications, where multiple diseases may be present in the same field, and allows farmers to take precise actions based on the specific diseases affecting their crops.

5. **Lack of Scalability and Adaptability to New Diseases**
   - **Existing Limitation:** As plant diseases evolve and new diseases emerge, many existing systems struggle to adapt to these changes. Additionally, scaling the model to accommodate new diseases or plant species often requires significant retraining and manual intervention.
   - **How Our Project Addresses It**: Our system is designed with scalability and adaptability in mind. By using transfer learning techniques, we can easily update the model to include new diseases or plant species without needing to retrain the entire model from scratch. This allows the system to evolve and adapt as new plant diseases emerge, ensuring long-term applicability. Furthermore, the modular design of the system makes it easy to add new disease categories and species, thereby enhancing its scalability as the agricultural landscape changes over time.

CHAPTER 3

Methodology

The proposed methodology involves building a robust machine learning model to classify and identify plant diseases using image datasets. The process begins with dataset preprocessing, including image resizing, augmentation, and normalization to enhance model performance. A convolutional neural network (CNN) architecture is designed to extract key features from images, leveraging multiple convolutional, pooling, and fully connected layers for accurate classification. The model is trained on labeled datasets, validated on unseen data, and fine-tuned to optimize accuracy. Additionally, techniques like dropout and adaptive learning rates are employed to reduce overfitting and improve generalization. The final system will provide an efficient and scalable solution for real-time plant disease detection to support sustainable agriculture.

3.1 System Design

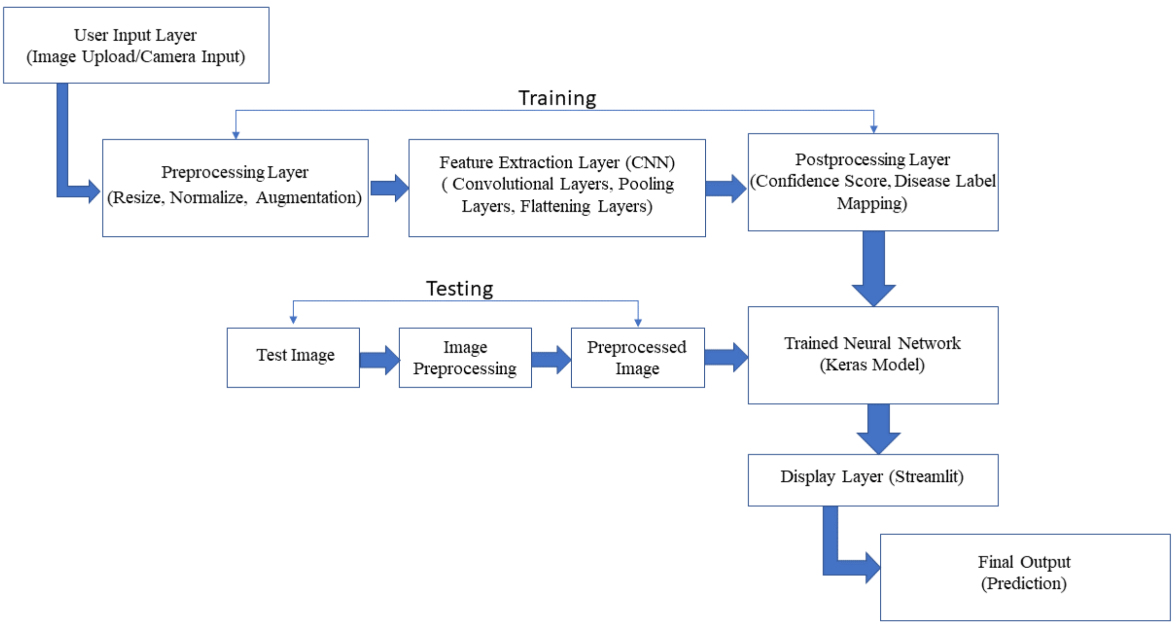

Figure 1 :Workflow of the Plant Disease Detection System.

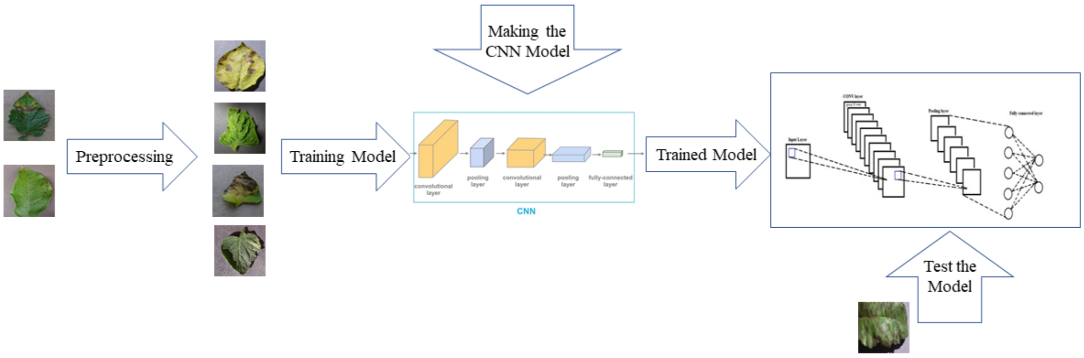

Figure 2. Applied Methodology.

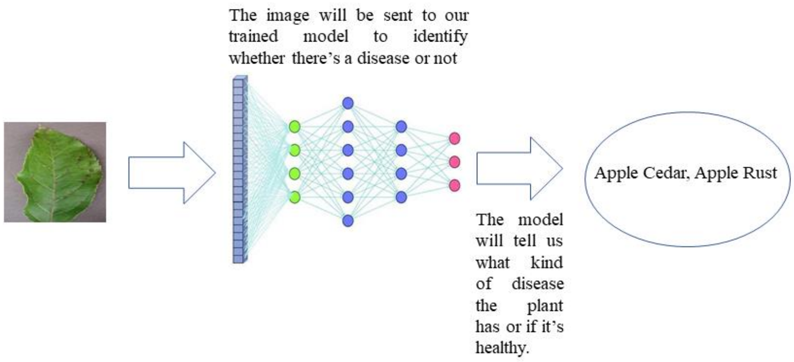

Figure 3. Testing of an image.

1. **User Interaction Layer**

- **Input**: User uploads an image of a plant leaf via the web interface or mobile app (using Streamlit as shown in the code).
- **Camera Option**: Alternatively, users can use their device's camera to capture an image of the leaf.
- **Output**: The uploaded image is passed to the preprocessing layer for handling.

2. **Preprocessing Layer**

- **Image Preprocessing**:
  - Resizing: The input image is resized to the required dimensions (e.g., 128x128 pixels) to match the input size expected by the CNN model.
  - Image Normalization: The pixel values of the image are normalized to a range of [0, 1] to improve model convergence.
  - Image Augmentation (Optional): While this feature isn't explicitly included in the provided code, augmenting the image (random rotations, zooms, flips) can enhance the model's generalization.
- **Output**: The processed image is ready to be fed into the feature extraction layer (CNN).

3. **Model Prediction Layer (Deep Learning Model - CNN)**

- **Model Loading**:
  - The trained Keras model (trained_plant_disease_model.keras) is loaded.
  - The model consists of multiple layers (Convolutional Layers, Pooling Layers, Dense Layers) trained on a plant disease dataset (e.g., 38 different classes of diseases).
- **Image Classification**:
  - Convolutional Layers: These layers apply multiple filters to the image, identifying low-level features like edges and textures.
  - Pooling Layers: Max-pooling or average pooling reduces the spatial dimensions of the image.
  - Flattening: The 2D features are flattened into a 1D array.
  - Fully Connected Layers: These layers analyze the extracted features and perform classification, mapping them to disease categories.

- Softmax Activation: The final output layer uses softmax to calculate the probabilities for each of the 38 disease classes.

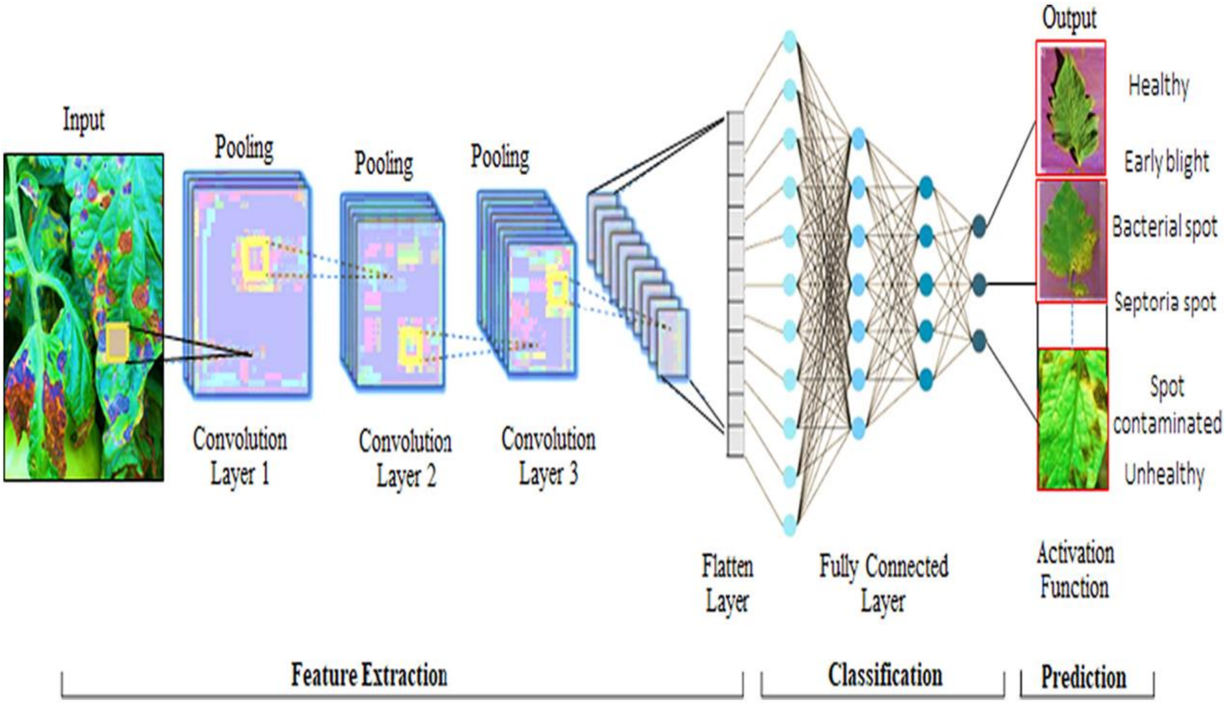

Figure 4. Convolutional neural network CNN architecture.

- **Output:** The model predicts the disease class with the highest probability**.**

4. Postprocessing Layer

- **Confidence Score**: The system computes the confidence level for the prediction (probability of the selected class).
- **Disease Label Mapping**: The index from the predicted output is mapped to the corresponding plant disease label (e.g., "Apple___Apple_scab").
- **Output**: The disease label, along with the confidence score, is ready for display on the web interface.

5. User Interface Layer (Streamlit Web Interface)

- **Result Display**:
  - Prediction Result: The predicted disease label and confidence score are displayed to the user.

- Image Display: The uploaded image is shown alongside the predicted result.

- **User Interactivity**:
    - Users can upload new images to classify or check multiple predictions.
    - Potential links to additional information or suggested remedies can be displayed for the user.
- **Output**: A web interface built using **Streamlit** that allows the user to interact with the system and view predictions.

3.2 Dataset Discussion

The new plant disease detection dataset contains many kinds of plant diseases and is openly accessible. The dataset boasts 87,000 photos, categorized into 38 different classes. To facilitate our experimental study, we divided the dataset into training, testing, and validation sets. While 20% of the dataset was used for validation and testing, the remaining 80% was allocated for training pre-existing models. The dataset comprises 87,000 samples for plant classes, of which 40 samples were used for testing, 17,572 samples for validation, and 70,295 samples for training. All 38 types of plant diseases are included and represented evenly across these sets.

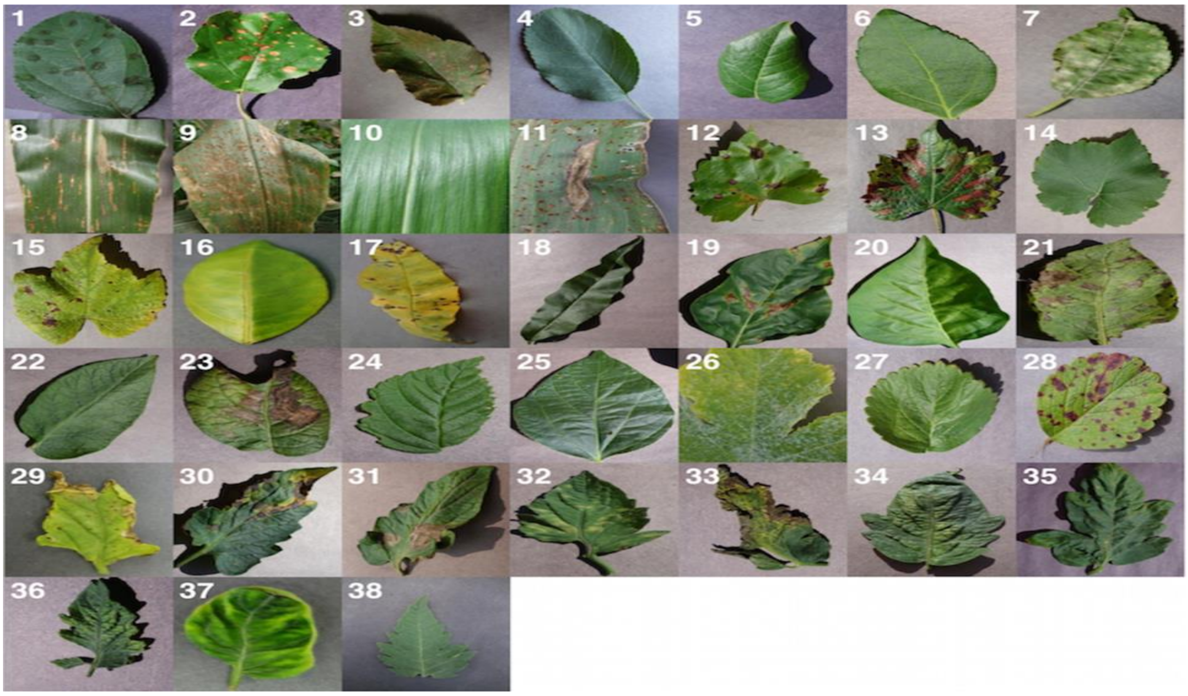

Figure 5 : Sample of Images from PlantVillage Dataset

Table- 2 : Dataset Description

Class	Plant Name	Healthy or Diseased	Disease Name	Images (Number)
C_1	Apple	Diseased	Apple_scab	2016
C_2	Apple	Diseased	Black_rot	1987
C_3	Apple	Diseased	Cedar_apple_rust	1760
C_4	Apple	Healthy	_	2008
C_5	Blueberry	Diseased		1816
C_6	Cherry_(including_sour)	Diseased	Powdery_mildew	1683
C_7	Cherry_(including_sour)	Healthy	_	1826
C_8	Corn_(maize)	Diseased	Cercospora_leaf_spotGray_leaf_spot	1642
C_9	Corn_(maize)	Diseased	Common_rust	1907
C_10	Corn_(maize)	Diseased	Northern_Leaf_Blight	1908
C_11	Corn_(maize)	Healthy	_	1859
C_12	Grape	Diseased	Black_rot	1888
C_13	Grape	Diseased	Esca_(Black_Measles)	1920
C_14	Grape	Diseased	Leaf_blight (Isariopsis_Leaf_Spot)	1722
C_15	Grape	Healthy	_	1692
C_16	Orange	Diseased	Haunglongbing (Citrus_greening)	2010
C_17	Peach	Diseased	Bacterial_spot	1838
C_18	Peach	Healthy	_	1728
C_19	Pepper_bell	Diseased	Bacterial_spot	1913
C_20	Pepper_bell	Healthy	_	1988
C_21	Potato	Diseased	Early_blight	1939
C_22	Potato	Diseased	Late_blight	1939
C_23	Potato	Healthy	_	1824
C_24	Raspberry	Healthy	_	1781
C_25	Soybean Healthy	Healthy	_	2022

C_26	Squash	Diseased	Powdery_mildew	1736
C_27	Strawberry	Diseased	Leaf_scorch	1774
C_28	Strawberry	Healthy	_	1824
C_29	Tomato	Diseased	Bacterial_spot	1702
C_30	Tomato	Diseased	Early_blight	1920
C_31	Tomato	Diseased	Late_blight	1851
C_32	Tomato	Diseased	Leaf_Mold	1882
C_33	Tomato	Diseased	Septoria_leaf_spot	1745
C_34	Tomato	Diseased	Spider_mites Two-spotted_spider_mite	1741
C_35	Tomato	Diseased	Target_Spot	1827
C_36	Tomato	Diseased	Tomato_Yellow_Leaf _Curl_Virus	1961
C_37	Tomato	Diseased	Tomato_mosaic_virus	1790
C_38	Tomato	Diseased	_	1926
Total				**70295**

3.3 Requirement Specification

This section outlines the hardware and software requirements necessary for the implementation of the Plant Disease Detection System. These specifications will ensure the smooth functioning and optimal performance of the system.

3.3.1 Hardware Requirements:

To implement the solution efficiently, the following hardware components are required:

1. **Computer or Server (For Model Training and Inference):**
    - Processor: Intel Core i5 or higher, AMD Ryzen 5 or higher (minimum 4 cores) to handle large-scale data processing and training tasks.
    - RAM: At least 8GB RAM for efficient multitasking and handling the dataset during training and inference processes. More RAM (16GB or higher) is recommended for faster processing, especially during model training.

- Storage: Minimum of 100GB of free storage space (SSD recommended) to store the dataset, model weights, and other necessary files. The model weights, especially a deep learning model, can be quite large.
- Graphics Processing Unit (GPU): A GPU with at least 4GB of VRAM (NVIDIA GTX 1650 or higher) for model training using TensorFlow or Keras. A GPU will drastically reduce the time required for training deep learning models, making it a critical component for faster performance.
- Web Camera (Optional): If real-time input from users (i.e., taking images via camera) is required, a web camera with a minimum resolution of 720p will be needed.

2. Client Device (For User Interaction):

- Smartphone/Tablet/PC: Any device with a web browser (Chrome, Firefox, etc.) for accessing the Streamlit web application where users can upload plant images for disease detection.
- Internet Connection: A stable internet connection is needed for hosting the web application and for uploading/downloading images for prediction.

3.3.2 Software Requirements:

To implement and run the Plant Disease Detection System, the following software components are required:

1. Operating System:

- Windows (Windows 10 or higher)
- Linux (Ubuntu): Recommended for easier setup of machine learning libraries and tools.
- macOS: Works well but may require additional setup for GPU support in some cases.

2. Python Environment:

- Python 3.8 or higher: The primary programming language used for the system development, including data processing, model training, and web application deployment.

3. Machine Learning Libraries:

- TensorFlow 2.x or Keras: The deep learning framework used for building and training the plant disease detection model. TensorFlow provides tools for model training, evaluation, and inference.
- NumPy: For numerical operations such as handling image arrays and matrix computations.
- Pandas: Used for handling data and organizing the dataset into a manageable structure.
- OpenCV: (Optional) For additional image preprocessing and manipulation tasks such as image reading and transformations.
- Scikit-learn: For various machine learning tools (e.g., splitting the dataset into training and testing).

4. **Web Framework:**
   - Streamlit: A powerful, easy-to-use Python library for creating web applications. Streamlit will be used to build the user interface, allowing users to upload images and view disease predictions in real-time.
5. **Image Processing Tools:**
   - Matplotlib/Seaborn: These libraries can be used to visualize results, such as displaying accuracy plots, loss curves, and sample images of predictions.
6. **Database (Optional):**
   - SQLite or MySQL (if applicable): For storing additional data, such as user logs, prediction history, or plant disease categories.
   - Firebase (Optional): A cloud-based database if there's a requirement for real-time data synchronization across multiple devices.
7. **Web Hosting and Deployment (For Production Environment):**
   - Heroku or AWS (Amazon Web Services): To deploy the Streamlit application on the cloud, making it accessible online. You can host the application and make it available for users worldwide.
   - Docker (Optional): For creating a containerized environment that ensures the app runs consistently across various machines.
8. **Version Control:**

- Git: For version control to track changes in the codebase and collaborate with others.
- GitHub: For storing the source code, collaborating with team members, and managing version control.

CHAPTER 4

Implementation and Result

4.1 Result Visualization:

4.1.1 Performance Measures:

In order to evaluate the performance of the Convolutional Neural Network (CNN) model used for plant disease detection, several key performance metrics were computed. These metrics are crucial for understanding the effectiveness of the model in accurately identifying and classifying various plant diseases. The primary performance measures considered include accuracy, precision, recall, and F1-score, which are calculated based on the confusion matrix.

Accuracy: This measures the overall correctness of the model's predictions, representing the ratio of correctly classified samples (both positive and negative) to the total number of samples. It is calculated using the formula:

- $$\mathbf{Accuracy} = \frac{\mathbf{TP+TN}}{\mathbf{TP + TN + FP + FN}}$$

Recall (Sensitivity or True Positive Rate): This metric measures the model's ability to correctly identify positive samples (infected plants). It is computed as:

- $$\mathbf{Recall} = \frac{\mathbf{TP}}{\mathbf{TP + FN}}$$

Precision: Precision measures the proportion of correctly classified positive samples out of all samples predicted as positive. It is defined as:

- $\mathbf{Precision} = \frac{\mathbf{TP}}{\mathbf{TP+FP}}$

F1-Score: The F1-score is the harmonic mean of precision and recall, offering a balanced measure of a model's performance. It is calculated using the formula:

- $\mathbf{F1-Score} = 2 \times \frac{\mathbf{Precision} \times \mathbf{Recall}}{\mathbf{Precision\ +Recall}}$

These performance metrics have been summarized in the classification report, which provides detailed insights into the model's performance across different plant diseases. The results show that the CNN model performs robustly, achieving high accuracy, precision, recall, and F1-scores across the dataset.

Table- 3: Classification Report

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Apple___Apple_scab	0.90	0.98	0.94	504
Apple___Black_rot	1.00	0.97	0.98	497
Apple___Cedar_apple_rust	0.97	0.96	0.96	440
Apple___healthy	0.97	0.92	0.94	502
Blueberry___healthy	0.97	0.98	0.97	454
Cherry_(including_sour)___Powdery_mildew	0.96	1.00	0.98	421
Cherry_(including_sour)___healthy	0.95	0.98	0.97	456
Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot	0.90	0.95	0.93	410
Corn_(maize)__*Common_rust*	1.00	0.98	0.99	477
Corn_(maize)___Northern_Leaf_Blight	0.96	0.93	0.95	477
Corn_(maize)___healthy	0.99	1.00	0.99	465
Grape___Black_rot	0.95	0.98	0.97	472
Grape___Esca_(Black_Measles)	0.98	0.98	0.98	480
Grape___Leaf_blight_(Isariopsis_Leaf_Spot)	1.00	0.97	0.98	430
Grape___healthy	1.00	1.00	1.00	423

Orange___Haunglongbing_(Citrus_greening)	0.99	0.99	0.99	503
Peach___Bacterial_spot	0.95	0.97	0.96	459
Peach___healthy	0.99	0.99	0.99	432
Pepper,_bell___Bacterial_spot	0.95	0.95	0.95	478
Pepper,_bell___healthy	0.96	0.96	0.96	497
Potato___Early_blight	0.97	0.99	0.98	485
Potato___Late_blight	0.96	0.93	0.94	485
Potato___healthy	0.94	0.99	0.96	456

Overall Accuracy			0.96	17572
Macro Average	0.96	0.96	0.96	17572
Weighted Average	0.96	0.96	0.96	17572

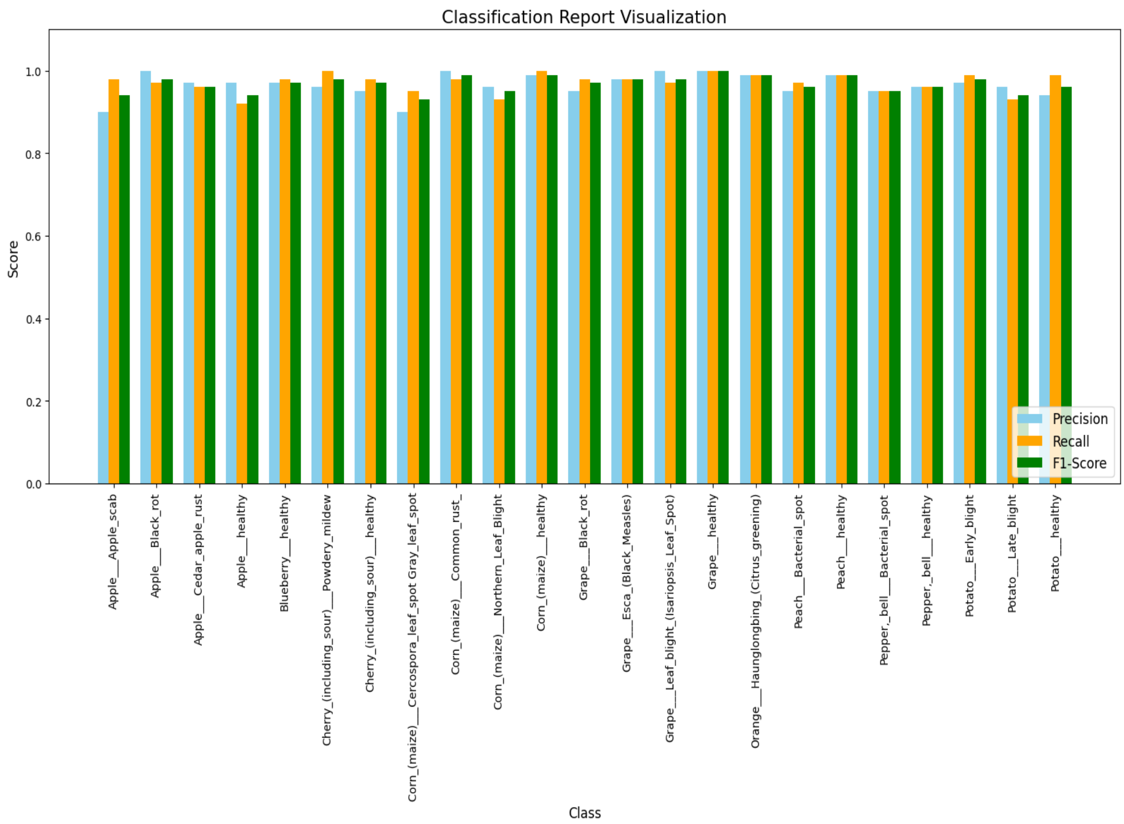

Figure 6: Classification Report

4.1.2 Model Prediction Snapshots

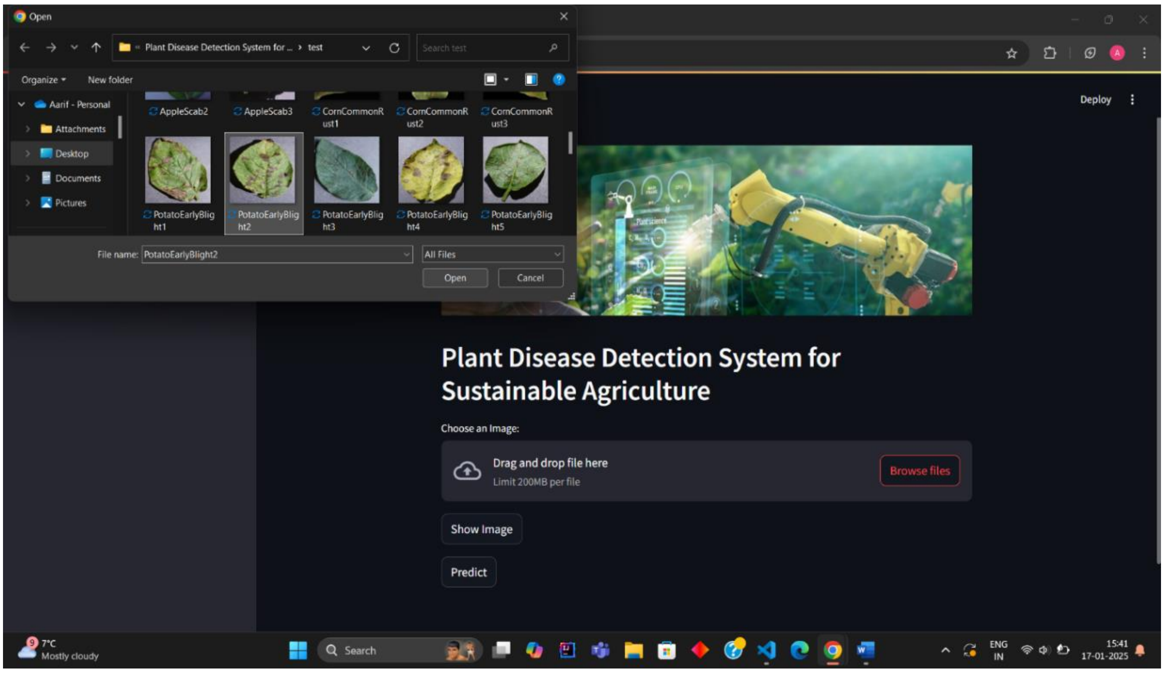

Figure 7: Image Upload Interface

This screenshot shows the initial stage of a Plant Disease Detection System for Sustainable Agriculture, where the user is uploading a test image. The interface allows users to select an image file, presumably containing a leaf with a potential disease, from their computer for analysis. The file selection dialog is open, showcasing a folder with various images named according to different plant diseases (e.g., AppleScab, PotatoEarlyBlight).

The user selects "PotatoEarlyBlight2," indicating that the system will analyze this specific image for detecting the presence and type of disease on the plant. Once uploaded, the image will be processed by a Convolutional Neural Network (CNN) model, built using Keras, to predict and identify the disease in the plant based on the provided image.

Key elements in this stage:

1. **File Upload Interface:** A simple drag-and-drop or browse option for users to input their test images.
2. **Image Selection:** The user is in the process of selecting an image for disease prediction.
3. **Purpose:** The system aims to leverage deep learning techniques to assist in sustainable agriculture by identifying plant diseases early, which can help in timely interventions.

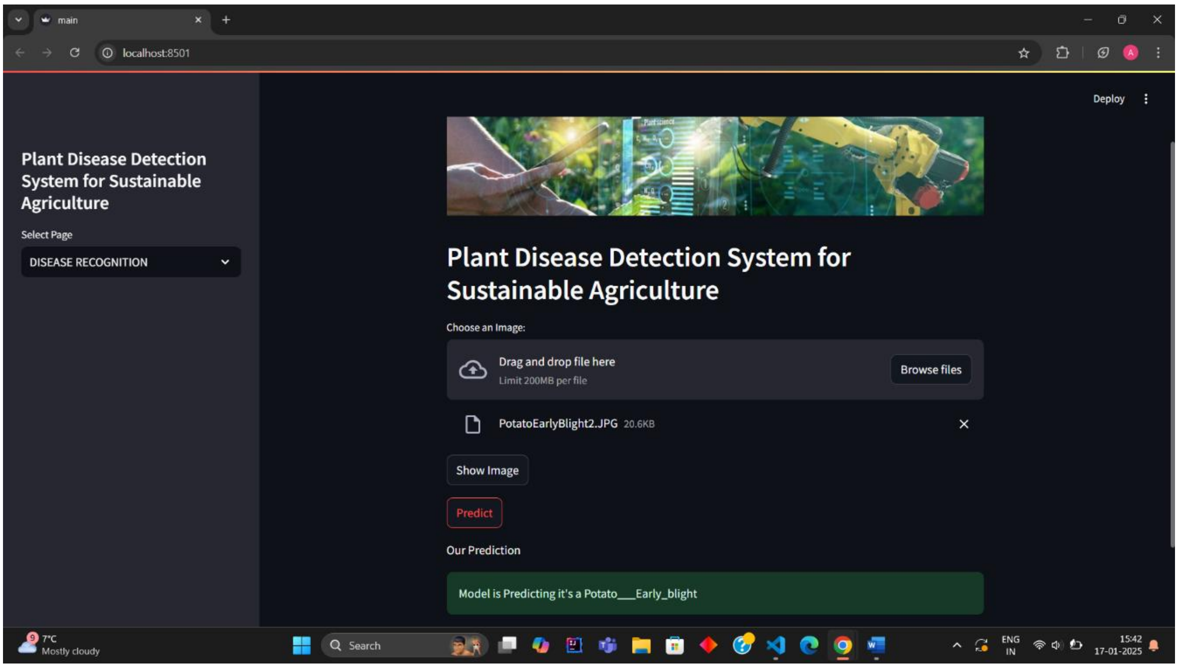

Figure 8: Prediction Result - PotatoEarlyBlight.

In this screenshot, the user interface of the Plant Disease Detection System has progressed to display the uploaded test image, "PotatoEarlyBlight2.JPG." The file is shown in the interface, indicating that it has been successfully uploaded and is ready for processing.

Key Elements in this Screenshot:

1. **Image Upload Confirmation:** The name of the uploaded file, "PotatoEarlyBlight2.JPG," along with its size (20.6KB), is displayed below the upload section. This confirms that the image is loaded into the system.
2. **Options for Interaction:**

   - **Show Image**: There is an option to visually display the uploaded image, allowing the user to confirm that the correct file has been selected.
   - **Predict Button**: The user can now click the "Predict" button to initiate the disease detection process. This action triggers the Convolutional Neural Network (CNN) model to analyze the image and identify any disease present on the plant.

3. **Prediction Output:**

- After the prediction process is complete, the system outputs the result under the "Our Prediction" section. In this instance, the model predicts that the plant shown in the image has "Potato__Early_blight." This indicates that the model has identified the disease affecting the potato plant as Early Blight, a common disease caused by the fungus Alternaria solani.

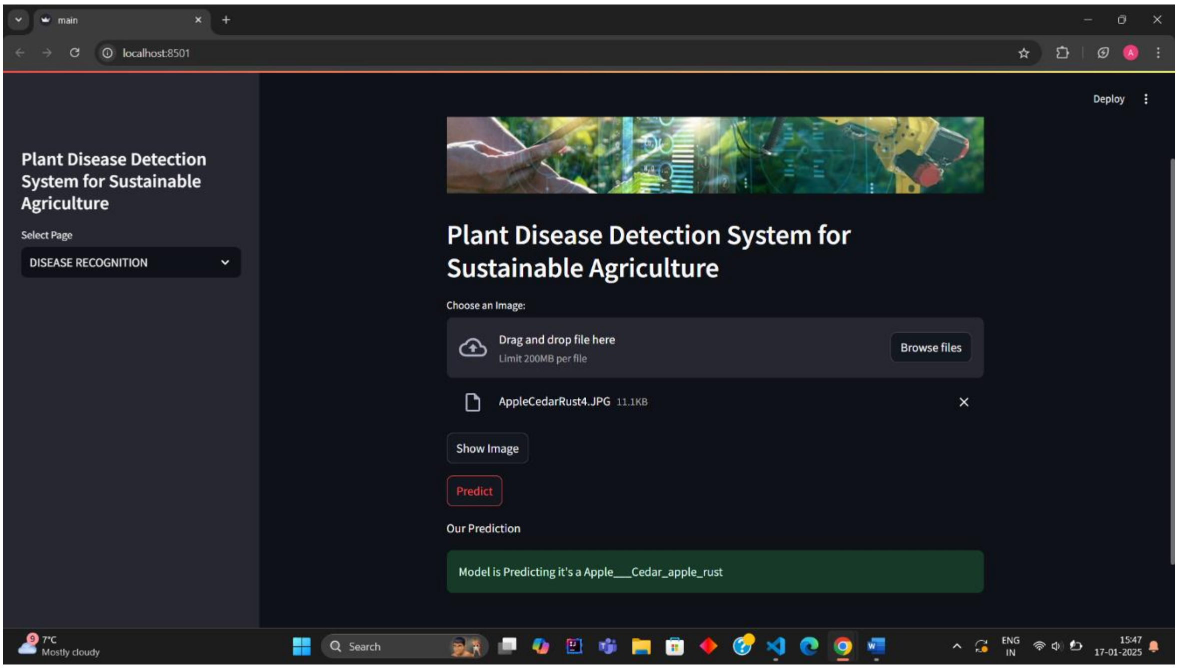

Figure 9: Prediction Result - Apple__Cedar_apple_rust

In this third screenshot, the Plant Disease Detection System has been used to upload a different image, "AppleCedarRust4.JPG," and the system has processed it to provide a prediction.

Key Elements in this Screenshot:

1. **Uploaded Image Details:** The file "AppleCedarRust4.JPG," with a size of 11.1KB, is shown below the upload section. This indicates that a different plant image has been uploaded for analysis.
2. **Predict Button**: The "Predict" button is available, similar to the previous step, which the user has clicked to process the new image.
3. **Prediction Output:**

- The system outputs the prediction under the "Our Prediction" section. In this case, the model predicts that the plant in the image has "Apple__Cedar_apple_rust." This indicates that the model has identified the disease as Cedar Apple Rust, a fungal disease that affects apple trees, typically caused by Gymnosporangium juniperi-virginianae.

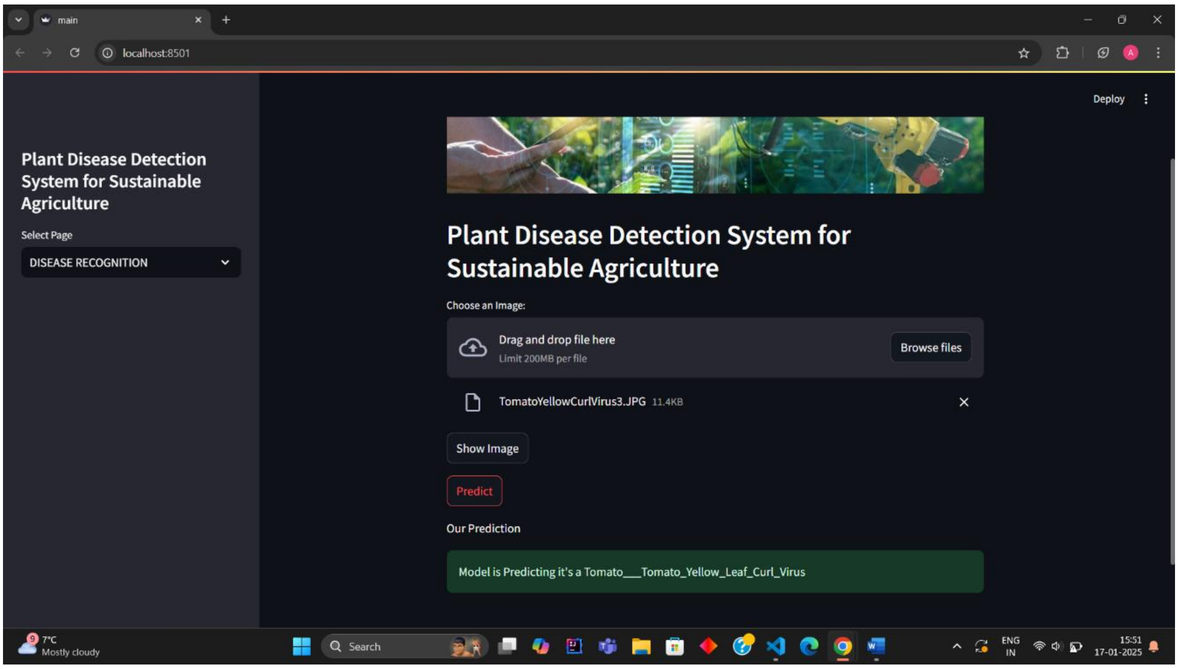

Figure 10: Prediction Result - Tomato Yellow Leaf Curl Virus

The result displayed in the snapshot indicates that the model has analyzed the uploaded image and predicted the presence of a specific plant disease. In this case, the prediction result is:

"Tomato Yellow Leaf Curl Virus"

This means the model has identified the disease affecting the plant in the uploaded image as the "Tomato Yellow Leaf Curl Virus."

4.2 GitHub Link for Code:

https://github.com/Aarif-Mir/Plant-Disease-Detection-/tree/main

CHAPTER 5

Discussion and Conclusion

5.1 Future Work:

The **Plant Disease Detection System** has made significant strides in identifying plant diseases with an accuracy of 95%. However, like any AI-based system, there are areas for improvement and future development. Below are suggestions for further enhancing the model, as well as addressing some unresolved issues:

1. Expanding the Dataset

One of the main limitations of any deep learning model is the size and diversity of the dataset it is trained on. While our current model includes 38 classes of diseases, expanding the dataset to include more plant species and disease types would increase the model's generalizability.

- **Future Work**:
  - Add more plant species and disease variations (e.g., diseases affecting vegetables, flowers, or other crops).
  - Improve dataset diversity by including different environmental conditions, as plant disease symptoms may vary depending on weather, location, and season.
  - Augment the dataset using data augmentation techniques such as rotation, flipping, and zooming to artificially increase the dataset size without collecting more images.

2. Real-Time Disease Detection

Currently, the system requires users to upload a pre-captured image, which may not always be convenient for field use. Real-time detection through a smartphone or camera, where the model can identify diseases directly from live images, would enhance the practical usability of the system.

- **Future Work**:
  - Implement real-time disease detection using the camera input of smartphones, where users can point their camera at a plant and instantly get predictions.
  - Improve the mobile application interface to make it user-friendly and accessible for farmers in remote areas.

3. Model Optimization and Deployment

While the current model provides satisfactory results, there are opportunities to further optimize its performance for both speed and accuracy. Some models may take a long time to process predictions, which is a concern for real-time applications.

- **Future Work**:
  - Implement model compression techniques like **Quantization** or **Pruning** to reduce the model size and improve inference speed without sacrificing much accuracy.
  - Explore **Edge Computing** for deploying the model directly on smartphones or edge devices. This would eliminate the need for cloud-based processing and ensure faster predictions in areas with limited internet connectivity.

4. Multilingual Support for Broader Accessibility

The current system is primarily designed in English, which might limit its accessibility in non-English-speaking regions. Providing multilingual support for different languages can make the application more accessible to farmers globally.

- **Future Work**:
  - Integrate multilingual support for different languages, including regional dialects, to ensure that the application is widely usable across diverse agricultural regions.
  - Collaborate with agricultural agencies worldwide to adapt the model for use in various countries.

5. Model Explainability and Transparency

AI models, especially deep learning models, can be seen as "black boxes," making it difficult for end-users to understand why certain predictions were made. In the context of plant disease detection, farmers may require more insight into how the model arrived at a particular diagnosis.

- **Future Work**:
    - Implement techniques like **Grad-CAM** (Gradient-weighted Class Activation Mapping) to visually explain which parts of the image contributed most to the model's decision. This will improve the model's transparency and build trust with the end-users.
    - Provide detailed explanations along with the disease diagnosis, such as possible causes, recommended actions, and preventive measures.

6. Integration with IoT (Internet of Things)

Integrating the disease detection system with IoT-based sensors can provide a more comprehensive solution for agriculture. For instance, environmental factors such as temperature, humidity, soil moisture, and light intensity can influence plant health and disease spread.

- **Future Work**:
    - Integrate IoT devices like sensors or drones to capture real-time data from the field. This would allow the system to not only detect diseases from images but also correlate environmental factors to provide more accurate predictions.
    - Use **smart farming techniques**, such as automated irrigation systems or nutrient management systems, that can be triggered based on the disease diagnosis.

7. Multi-Class Classification and Uncertainty Handling

Currently, the model uses multi-class classification to identify a single disease. In practice, a plant could exhibit multiple symptoms or diseases simultaneously, and a more advanced approach may be needed to handle this complexity.

- **Future Work**:

- Explore **multi-label classification** approaches that can predict multiple diseases in a single image if they coexist, providing a more holistic solution.
- Incorporate uncertainty handling and confidence scores in predictions, so the model can indicate when it is unsure about a diagnosis and recommend further steps (e.g., consulting an expert).

8. Collaboration with Experts and Crowdsourcing Data

The current model is based on historical data and predefined disease categories. To keep the model relevant and up-to-date, continuous improvement based on expert feedback and real-world data collection is essential.

- **Future Work**:
  - Collaborate with agricultural experts and universities to continually update and improve the model's accuracy by incorporating expert knowledge and new disease varieties.
  - Leverage crowdsourcing platforms to collect more diverse data from farmers and researchers to help improve the model and its generalization capabilities.

5.2 Conclusion:

The **Plant Disease Detection System** represents a significant advancement in the field of agricultural technology, particularly for sustainable farming practices. The system leverages deep learning models to detect and identify various plant diseases from images, offering farmers a tool to improve crop health management. With an accuracy rate of 95%, the system effectively helps in diagnosing plant diseases, thereby enabling timely intervention and preventing crop loss.

This project has contributed to solving a critical challenge in modern agriculture—ensuring food security by preventing the spread of plant diseases. By automating the disease detection process, the system provides a more efficient and accurate means of monitoring plant health compared to traditional methods, which are often laborintensive, time-consuming, and prone to human error. The use of deep learning models trained on a large dataset of plant disease images further enhances the system's ability to classify a wide range of diseases with high precision.

Furthermore, this system is not just beneficial for large-scale commercial agriculture but also holds immense potential for small-scale and remote farmers. With the help of smartphones and easy-to-use interfaces, even farmers with limited technical knowledge can utilize the system to protect their crops, reducing the reliance on expensive, labor-heavy disease detection methods.

The key contributions of this project are:

- **Improved Disease Diagnosis**: The deep learning model's ability to identify 38 plant diseases provides a reliable tool for farmers to detect diseases early and apply the necessary treatments.
- **Sustainability in Agriculture**: By preventing disease outbreaks, the system contributes to sustainable farming practices, reducing the need for excessive pesticide use and minimizing environmental harm.
- **Practical Application**: The integration of this model into an accessible platform like Streamlit ensures that it can be deployed easily across different regions and made available to a wider audience.

While the project has made significant strides, there are still areas for future enhancement, such as expanding the dataset, optimizing the model for real-time detection, and integrating additional technologies like IoT and mobile applications for even more comprehensive and efficient disease management.

REFERENCES

[1]. Sethy, P.K.; Barpanda, N.K.; Rath, A.K.; Behera, S.K. Deep feature based rice leaf disease identification using support vector machine. Comput. Electron. Agric. 2020, 175, 105527. [Google Scholar]

[2]. Chen, J.; Chen, J.; Zhang, D.; Sun, Y.; Nanehkaran, Y.A. Using deep transfer learning for image-based plant disease identification. Comput. Electron. Agric. 2020, 173, 105393. [Google Scholar]

[3]. Bai, X.; Cao, Z.; Zhao, L.; Zhang, J.; Lv, C.; Li, C.; Xie, J. Rice heading stage automatic observation by multi-classifier cascade based rice spike detection method. Agric. For. Meteorol. 2018, 259, 260–270. [Google Scholar] [CrossRef]

[4]. Ramcharan, A.; Baranowski, K.; McCloskey, P.; Ahmed, B.; Legg, J.; Hughes, D.P. Deep Learning for Image-Based Cassava Disease Detection. Front. Plant Sci. 2017, 8, 1852. [Google Scholar] [CrossRef] [Green Version]

[5]. Lee, S.H.; Chan, C.S.; Mayo, S.J.; Remagnino, P. How deep learning extracts and learns leaf features for plant classification. Pattern Recognit. 2017, 71, 1–13. [Google Scholar] [CrossRef] [Green Version]

[6]. Shah, M.P.; Singha, S.; Awate, S.P. Leaf classification using marginalized shape context and shape+ texture dual-path deep convolutional neural network. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 860–864. [Google Scholar]

[7]. S. C. Madiwalar and M. V. Wyawahare, "Plant disease identification: A comparative study," 2017 International Conference on Data Management, Analytics and Innovation (ICDMAI), 2017, pp. 13-18, doi: 10.1109/ICDMAI.2017.8073478.

[8]. G. Shrestha, Deepsikha, M. Das and N. Dey, "Plant Disease Detection Using CNN," 2020 IEEE Applied Signal Processing Conference (ASPCON), 2020, pp. 109-113, doi: 10.1109/ASPCON49795.2020.9276722.

[9]. Gaurav Verma, CharuTaluja, Abhishek Kumar Saxena "Vision Based Detection and Classification of Disease on Rice Crops Using Convolutional Neural Network" ,2019

[10]. S. D.M., Akhilesh, S. A. Kumar, R. M.G. and P. C., "Image based Plant Disease Detection in Pomegranate Plant for Bacterial Blight," 2019 International Conference on Communication and Signal Processing (ICCSP), 2019, pp. 0645-0649, doi: 10.1109/ICCSP.2019.8698007.

[11]. lat, M. Braik and Z. ALRahamneh "Fast and Accurate Detection and Classification of Plant Diseases"

[12]. Jadhav, S.B.; Udupi, V.R.; Patil, S.B. Identification of plant diseases using convolutional neural networks. Int. J. Inf. Technol. 2021, 13, 2461–2470.

[13]. Abbas, A.; Jain, S.; Gour, M.; Vankudothu, S. Tomato plant disease detection using transfer learning with C-GAN synthetic images. Comput. Electron. Agric. 2021, 187, 106279.

[14]. Anh, P.T.; Duc, H.T.M. A Benchmark of Deep Learning Models for Multi-leaf Diseases for Edge Devices. In Proceedings of the 2021 International Conference on Advanced Technologies for Communications (ATC), Ho Chi Minh City, Vietnam, 14–16 October 2021; pp. 318–323.

[15]. Astani, M.; Hasheminejad, M.; Vaghefi, M. A diverse ensemble classifier for tomato disease recognition. Comput. Electron. Agric. 2022, 198, 107054

[16]. Trivedi, N.K.; Gautam, V.; Anand, A.; Aljahdali, H.M.; Villar, S.G.; Anand, D.; Goyal, N.; Kadry, S. Early Detection and Classification of Tomato Leaf Disease Using High-Performance Deep Neural Network. Sensors 2021, 21, 7987

[17]. Agarwal, M.; Gupta, S.K.; Biswas, K.K. Development of Efficient CNN model for Tomato crop disease identification. Sustain. Comput. Inform. Syst. 2020, 28, 100407.

[18]. Fuentes, A.; Yoon, S.; Kim, S.C.; Park, D.S. A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition. Sensors 2017, 17, 2022.

[19]. Zhong, Y.; Zhao, M. Research on deep learning in apple leaf disease recognition. Comput. Electron. Agric. 2020, 168, 105146.

[20]. Chen, J.; Chen, J.; Zhang, D.; Sun, Y.; Nanehkaran, Y. Using deep transfer learning for image-based plant disease identification. Comput. Electron. Agric. 2020, 173, 105393.

[21]. Shrivastava, V.K.; Pradhan, M.K. Rice plant disease classification using color features: A machine learning paradigm. J. Plant Pathol. 2021, 103, 17–26.