# PLANT DISEASE DETECTION FROM IMAGES

## TABLE OF CONTENTS

# I.Abstract

Plant diseases are a critical challenge in agriculture, leading to significant reductions in crop yields and economic losses. Early and accurate detection of these diseases is essential for timely interventions and effective crop management. This project introduces a machine learning-based solution for plant disease detection using images, powered by a Convolutional Neural Network (CNN) model. A web-based application was developed using Streamlit, allowing users to upload images of plant leaves and receive predictions on potential diseases in real-time. The dataset comprises over 38 classes of plant species and their associated diseases, including tomatoes, grapes, oranges, and more. By implementing image preprocessing techniques such as resizing, normalization, and data augmentation, the model's performance was enhanced. A custom CNN model was trained and compared against pre-trained models like VGG16, Alex-Net, and Dense-Net. The custom CNN achieved the highest accuracy of 87.56%, outperforming the pre-trained models in overall classification performance. The web application provides a user-friendly interface, ensuring ease of use for farmers and gardeners, who can simply upload images and receive instant feedback on plant health, including confidence scores for the predictions. Although the model performs well, challenges such as handling imbalanced datasets and optimizing real-time inference remain. In the future, the model will be expanded to recognize a broader range of plant species and diseases, while improvements in performance will focus on reducing latency and increasing prediction accuracy. Additionally, integrating feedback mechanisms for users to report incorrect predictions will contribute to continuous model refinement, making the system a valuable tool for agricultural disease management.

## II.Introduction

### 2.1 Overview of Plant Disease Detection

Plant diseases have a profound impact on agricultural productivity, leading to significant economic losses, reduced crop yields, and compromised food security. Accurate and timely detection of plant diseases is essential for mitigating these effects, enabling early interventions that can prevent widespread damage. Traditionally, farmers and agriculturists have relied on manual inspection to identify diseases, a time-consuming and error-prone method that often requires expert knowledge. As modern agriculture continues to evolve, the integration of technology offers a more efficient solution to this problem.

Recent advancements in machine learning and image processing have made it possible to automate the detection of plant diseases through image-based classification. By leveraging powerful deep learning models like Convolutional Neural Networks (CNNs), it is now possible to identify diseases from images of plant leaves with high accuracy. This automation helps reduce the reliance on expert manual inspection, providing farmers and gardeners with immediate and accurate diagnoses of plant health issues. Consequently, early detection of diseases facilitates better decision-making, such as targeted pesticide use, which can improve crop management practices and minimize environmental harm.

## 2.2 Motivation and Objectives

The motivation behind this project stems from the increasing need for precision agriculture, where technology aids in the efficient management of crops. Disease outbreaks in plants often go unnoticed until they have reached critical levels, leading to significant agricultural losses. In regions with limited access to expert agronomists, smallholder farmers especially struggle to identify diseases at early stages, resulting in decreased crop quality and yield.

The primary objective of this project is to develop a practical, easy-to-use tool that can assist farmers and gardeners by providing real-time diagnosis of plant diseases. Using deep learning techniques and a dataset containing images of various plant diseases, this project aims to build a reliable detection system that identifies diseases based on the visual patterns present in leaf images. The end goal is a fully functional web application, built with Streamlit, that allows users to upload images and receive immediate feedback on the health of the plant, empowering them to take timely actions to manage or treat diseases.

Additionally, this project seeks to contribute to the field of agricultural technology by comparing different CNN architectures to identify the best-performing model. Pretrained models such as VGG16, Alex-Net, and Dense-Net were evaluated against a custom CNN model, with a focus on achieving high accuracy and reliability in disease classification. The system's ultimate goal is to provide a tool that can be easily deployed and scaled, particularly in areas where agricultural expertise is limited but disease prevention is critical for food security.

## 2.3 Problem Statement

Despite the critical importance of plant disease detection in agriculture, current methods of identification are often inadequate. Manual inspection of crops for diseases is labour-intensive and prone to human error, with misdiagnoses potentially leading to ineffective treatments or unnecessary pesticide use. Moreover, farmers in rural or resource-constrained areas may lack access to agricultural experts capable of identifying specific plant diseases. The absence of fast and accurate disease detection methods results in delayed responses, leading to severe crop damage and economic loss.

The problem this project addresses is the development of a highly accurate, accessible, and scalable plant disease detection system that can provide instant feedback to users through image analysis. The system aims to reduce dependency on manual disease identification and offer an alternative solution that is both faster and more reliable. Additionally, the challenge lies in handling imbalanced datasets, as some plant diseases have fewer images available for training, which can skew the performance of traditional machine learning models. Addressing these challenges through data augmentation, transfer learning, and model optimization forms a key part of this project.

By creating a system that can classify plant diseases from leaf images with a high degree of accuracy, this project intends to enhance the efficiency of disease management, particularly for smallholder farmers and gardeners. The ultimate goal is to create a tool that not only supports agricultural productivity but also fosters a more sustainable and resilient agricultural ecosystem.

## III.Dataset

### 3.1 Dataset Description

The dataset used in this project is a comprehensive collection of labelled images depicting various plant species and their associated diseases. Sourced from the **New Plant Diseases Dataset** on Kaggle, it comprises high-quality images of plant leaves, each annotated with the specific disease or health condition of the plant. The dataset includes over **38 classes** that span a wide range of plant species, with both healthy and diseased conditions represented.

Each class is categorized into training, validation, and testing subsets to ensure that the model is both trained and evaluated on diverse, non-overlapping images. The total number of images in the dataset is over **87,000**, with each image typically resized for consistency during the preprocessing phase. These images serve as the foundation for training the CNN models used in this project to classify plant diseases effectively.

The images are captured under varying conditions, including different lighting, angles, and leaf orientations, simulating real-world scenarios that farmers might encounter. This variability enhances the robustness of the model when predicting diseases in new, unseen data.

### 3.2 Key Plant Species and Diseases

The dataset includes images of leaves from various plant species, both in healthy conditions and affected by diseases. Some of the prominent plant species and diseases featured in the dataset include:

- **Tomatoes**: Diseases: *Late blight*, *Early blight*, *Healthy*
- **Grapes**: Diseases: *Black rot*, *Esca (Black Measles)*, *Healthy*
- **Oranges**: Disease: *Huanglongbing (Citrus Greening)*
- **Potatoes**: Diseases: *Late blight*, *Healthy*
- **Corn (Maize)**: Diseases: *Northern Leaf Blight*, *Common Rust*, *Healthy*
- **Strawberries**: Disease: *Leaf scorch*
- **Apples**: Diseases: *Apple scab*, *Black rot*, *Cedar apple rust*, *Healthy*
- **Blueberries**: Disease: *Healthy*

Other species in the dataset include peaches, cherries, soybeans, and squash, each with their respective disease categories. This diverse range of species ensures that the model can generalize well across various plant types and conditions, making it a practical tool for real-world applications.

### 3.3 Data Preprocessing (Resizing, Normalization, Augmentation)

To ensure that the data is well-prepared for training the Convolutional Neural Network (CNN) model, several preprocessing steps were applied to the images:

- **Resizing**: Each image was resized to a uniform dimension of **224x224 pixels** to meet the input size requirements of the CNN models (Custom CNN, VGG16, Alex-Net, Dense-Net). This resizing helps maintain consistency and ensures that the input images are of the correct size during model training.
- **Normalization**: Image pixel values were normalized to a range between **0 and 1** by dividing the pixel values by 255. This normalization step is essential as it helps improve

model convergence by standardizing the scale of input data. Normalization also prevents large pixel values from disproportionately influencing the model's weights.

- **Data Augmentation**: Data augmentation techniques were applied to artificially increase the diversity of the training dataset. These techniques included:
    - **Rotation**: Randomly rotating images by a certain degree to simulate real-world conditions where leaves may be captured from various angles.
    - **Flipping**: Horizontally flipping images to create mirrored versions, adding more variety to the dataset.
    - **Zooming**: Random zooming within a specified range to mimic different scales of leaf images.
    - **Brightness Adjustment**: Modifying the brightness of images to account for varying lighting conditions in the real world.

Augmentation is particularly important in this dataset, as it helps the model generalize better to unseen data by simulating a wider range of image conditions. These transformations ensure that the CNN does not become over-reliant on specific visual patterns but instead learns to detect more general features associated with plant diseases.

### 3.4 Class Distribution and Handling Imbalance

One of the challenges in the dataset is the **class imbalance**, where certain plant species and diseases have significantly fewer examples compared to others. This imbalance can negatively affect model performance, as the model may become biased toward the more frequent classes, leading to poor detection of rare diseases.

For example, diseases like *Late blight* in tomatoes have a substantial number of images, while diseases like *Huanglongbing* in oranges are underrepresented. A similar imbalance exists for various species and their corresponding diseases.

To address this issue, several strategies were implemented:

- **Data Augmentation for Minority Classes**: For underrepresented classes, aggressive data augmentation was applied to increase the effective size of these classes. This involved applying more extensive transformations (e.g., rotations, zooms, flips) to minority class images to ensure they had a larger presence during model training.

- **Class Weights**: During model training, **class weights** were used to penalize the model more heavily for misclassifications of minority class images. This technique helps the model learn to treat rare classes with equal importance compared to more frequent classes, improving the model's ability to detect diseases in underrepresented categories.
- **Stratified Sampling**: During the data split into training and validation sets, **stratified sampling** was employed to ensure that the distribution of classes in the training, validation, and testing sets remained representative of the original dataset. This prevents skewed performance metrics that might arise if one subset contained disproportionate numbers of majority classes.

Handling this imbalance is crucial for ensuring that the model performs well across all disease categories, particularly those that are less common but still critical for farmers to identify accurately.

## IV.System Design and Architecture

### 4.1 Model Selection (Custom CNN and Pretrained Models)

For plant disease detection, the system incorporates both a **Custom CNN model** and several **pretrained models** (VGG16, Alex-Net, and Dense-Net) to offer flexibility in terms of performance and computational efficiency. The model selection was based on balancing accuracy, speed, and computational cost.

- **Custom CNN**:
  - The custom-built CNN model consists of several convolutional, pooling, and fully connected layers designed specifically for the plant disease detection dataset. It is a lightweight model optimized for detecting plant species and diseases in an environment with limited computational resources.
  - The CNN includes a series of **Convolutional Layers** with filters to extract features from the input images, **Max Pooling Layers** to reduce dimensionality, and **Fully Connected Layers** for classification. The model was trained from scratch on the pre-processed dataset and fine-tuned using early stopping and dropout to prevent overfitting.

- **Pretrained Models (VGG16, Alex-Net, Dense-Net)**:
    - The pretrained models used in this system are based on popular architectures such as **VGG16**, **Alex-Net**, and **Dense-Net**. These models were chosen for their high performance in image classification tasks.
    - **Transfer learning** was employed to adapt these pretrained models to the plant disease detection dataset. The models, initially trained on large datasets like ImageNet, were fine-tuned by replacing their output layers with plant-specific classes from our dataset.
    - Each model was evaluated based on its classification accuracy, inference speed, and resource utilization, and the most suitable model can be chosen based on the application's specific requirements (e.g., accuracy vs. speed trade-off).

### 4.2 User Interface Design using Streamlit

The user interface is designed with simplicity and ease of use in mind, making it accessible to farmers, agricultural workers, and anyone in need of quick and reliable plant disease detection. The interface is built using **Streamlit**, which allows for rapid development and deployment of interactive web applications with minimal code complexity. Key components of the interface include:

- **Image Upload Feature**: The core functionality of the UI is the image upload option, where users can select and upload a plant leaf image. This feature includes drag-and-drop functionality for ease of use.
- **Real-time Display of Predictions**: Once an image is uploaded, the interface displays the model's prediction along with a confidence score, helping users understand the reliability of the results. Predictions are displayed in a clean and straightforward format, with the disease name highlighted for quick identification.
- **Confidence Scores and Visual Aids**: Alongside the disease prediction, the UI also presents a **confidence score** (percentage) to show the model's certainty. Visual aids, such as **sample images** of the predicted disease, can also be shown to help users compare their leaf with common symptoms.
- **Disease Information and Treatment Suggestions**: For each predicted disease, the UI provides detailed information about the disease, its symptoms, and possible treatments.

Links to additional resources (e.g., agriculture websites or research papers) can be provided to guide users in managing the disease.

- **Responsiveness and Mobile Compatibility**: The interface is responsive and designed to work across various devices, including smartphones and tablets, ensuring that users can access the application in the field, regardless of the device they are using.

By leveraging Streamlit's capabilities, the user interface offers a simple, intuitive, and highly functional environment for plant disease detection, making it easy for non-technical users to interact with the system and obtain actionable insights.

## V.Model Development

### 5.1 Custom CNN Architecture

The custom CNN (Convolutional Neural Network) was specifically designed to suit the plant disease detection task, taking into account the unique characteristics of the dataset and the computational efficiency required for real-time predictions. The architecture consists of several **convolutional**, **pooling**, and **fully connected layers**, which allow the model to extract relevant features from images of plant leaves while maintaining scalability.

- **Input Layer**:
  - The input layer accepts images with dimensions 224x224 pixels, resized as part of the preprocessing pipeline. Each image is processed in RGB format, resulting in a tensor of shape (224, 224, 3).
- **Convolutional Layers**:
  - The first few layers of the CNN consist of multiple **convolutional layers** that apply filters to extract important features such as edges, textures, and colours. These layers use **ReLU (Rectified Linear Unit)** activation functions to introduce non-linearity into the model. The convolutional filters start small (e.g., 32 filters of size 3x3) and gradually increase in depth as the model progresses (e.g., up to 128 or 256 filters), enabling the extraction of higher-level features.

- **Pooling Layers**:
  - After each convolutional block, **max pooling layers** are used to reduce the spatial dimensions of the feature maps. Max pooling with a 2x2 filter is applied to reduce the size of the feature maps, maintaining only the most important information while reducing computational complexity.
- **Fully Connected Layers**:
  - After several convolutional and pooling layers, the feature maps are flattened into a 1D vector and passed to fully connected layers. These layers function as classifiers, learning non-linear combinations of features to predict the disease class. Dropout regularization is applied between fully connected layers to prevent overfitting by randomly setting some neuron activations to zero during training.
- **Output Layer**:
  - The final layer uses a **SoftMax activation function** to output class probabilities for each of the disease categories. The model is trained to predict which disease is present in the input image based on these probabilities.

The custom CNN architecture was fine-tuned using techniques such as early stopping, dropout, and L2 regularization to minimize overfitting and ensure generalization to unseen data.

### 5.2 Pretrained Models (VGG16, Alex-Net, Dense-Net)

In addition to the custom CNN, the project also leverages **pretrained models** such as **VGG16**, **Alex-Net**, and **Dense-Net** through transfer learning. These models were originally trained on large-scale image datasets like ImageNet and have shown strong performance in various image classification tasks. For plant disease detection, these models were fine-tuned by replacing their fully connected layers with custom layers that are suited to the specific task of identifying plant diseases.

- **VGG16**:
  - VGG16 is a deep convolutional network with 16 weight layers, known for its simplicity and effectiveness. It uses small 3x3 convolutional filters and has a consistent architecture, making it easy to adapt for plant disease detection. The

fully connected layers of VGG16 were replaced with a custom classifier to predict the plant diseases.

- **Alex-Net**:
  - Alex-Net, one of the earliest deep convolutional networks, uses fewer layers compared to VGG16 but includes **ReLU activations** and **local response normalization**, which help improve the model's performance. Like VGG16, its output layers were modified to suit the plant disease dataset.
- **Dense-Net**:
  - Dense-Net is a more recent architecture that uses densely connected convolutional layers, where each layer receives inputs from all previous layers. This structure improves gradient flow during training and makes Dense-Net highly efficient, requiring fewer parameters than traditional models. Transfer learning was applied to fine-tune Dense-Net for plant disease classification, making it a powerful option for this task.

Transfer learning allows these models to retain their learned features from the original training on large datasets while quickly adapting to the plant disease detection task with relatively less data.

### 5.3 Training and Validation Strategy

The training process was structured using a **train-validation split** to monitor the model's performance and avoid overfitting. Key aspects of the training and validation strategy include:

- **Train-Validation Split**:
  - The dataset was split into a **training set** (80%) and a **validation set** (20%) to allow for real-time evaluation of the model during training. The validation set was used to assess generalization and adjust hyperparameters.
- **Cross-Validation**:
  - **K-fold cross-validation** was also used in certain cases to further ensure that the model's performance is stable and generalizes well across different subsets of the dataset. This process was particularly useful in model selection.

- **Data Augmentation**:
    - To improve the model's robustness and prevent overfitting, data augmentation techniques (e.g., random rotation, horizontal flipping, brightness adjustment) were applied during training. This artificially expanded the dataset and exposed the model to different variations of plant leaf images.
- **Loss Function and Metrics**:
    - The **categorical cross-entropy loss function** was used for training the multi-class classification models. Performance was evaluated using metrics such as **accuracy**, **precision**, **recall**, and **F1-score** to ensure comprehensive assessment beyond accuracy alone.

### 5.4 Model Comparison and Selection Criteria

The final model selection was based on a comparison between the custom CNN and the pretrained models (VGG16, Alex-Net, Dense-Net) across several criteria:

- **Accuracy**:
    - The primary metric for comparison was the classification accuracy on the validation set. The model that achieved the highest accuracy was prioritized for further testing.
- **Inference Time**:
    - Since the application is intended for real-time disease detection, **inference time** (the time taken for the model to process an image and return a prediction) was a critical factor. Models with lower inference times were preferred for deployment.
- **Model Complexity**:
    - **Parameter count** and model complexity were also evaluated to balance computational efficiency with accuracy. Pretrained models like Dense-Net, while highly accurate, may be computationally heavy, so trade-offs between accuracy and complexity were carefully considered.

- **Generalization Performance**:
    - The ability of the model to generalize to unseen data was assessed by comparing validation accuracy and loss to training accuracy and loss. Models showing signs of overfitting were discarded in favour of more generalizable alternatives.

Based on these factors, the model that achieved the best balance between high accuracy, low inference time, and low computational overhead was selected for deployment.

**VI.Model Performance and Evaluation**

**6.1 Performance Metrics (Accuracy, Precision, Recall, F1-Score)**

To evaluate the effectiveness of the models developed for plant disease detection, several key **performance metrics** were utilized:

- **Accuracy**:
    - This metric measures the overall correctness of the predictions by calculating the proportion of true predictions (both positive and negative) among the total predictions. While accuracy is useful, it may not adequately represent model performance when dealing with imbalanced datasets, which is why additional metrics were considered.
- **Precision**:
    - Precision evaluates how many of the predicted positive instances are actually positive. In the context of plant disease detection, this helps measure the model's ability to avoid false positives (misclassifying a healthy plant as diseased).
- **Recall** :
    - Recall measures how well the model captures actual positive instances. For this project, recall is essential as it reflects the model's ability to correctly identify diseased plants, especially for rarer diseases where a high recall is critical.
- **F1-Score**:
    - The F1-score provides a balance between precision and recall, particularly useful when there is an uneven class distribution (imbalanced dataset). It offers a single metric that accounts for both the model's accuracy in correctly identifying positive instances and avoiding false positives.

### 6.2 Model Comparison

| Models | Accuracy (%) | Precision (%) | Recall (%) | F1-score(%) |
|---|---|---|---|---|
| **Custom CNN** | **87.56** | **88.06** | **87.56** | **87.61** |
| Dense-Net | 87.41 | 89.72 | 87.41 | 87.76 |
| Alex-Net | 87.12 | 88.29 | 87.12 | 87.15 |
| VGG16 | 85.32 | 86.84 | 85.32 | 85.36 |

### 6.3 Detailed Comparison of Models

To ensure optimal performance, a comparison of the **Custom CNN model** and several **pretrained models** (VGG16, Alex-Net, Dense-Net) was conducted. The comparison focused on key performance metrics across each model, revealing their strengths and weaknesses in detecting plant diseases.

- **Custom CNN**:
  - The custom-built CNN was fine-tuned for this specific task and achieved the highest overall accuracy (87.56%) and an F1-score of 87.61%. The architecture was designed to extract important features from plant leaf images effectively.
- **Dense-Net**:
  - Dense-Net, known for its effective feature propagation, demonstrated strong performance with an accuracy of 87.41% and a higher precision (89.72%) than the Custom CNN. However, it slightly underperformed in recall and F1-score compared to the Custom CNN.
- **Alex-Net**:
  - Alex-Net, one of the earlier convolutional architectures, showed good performance but was not as effective as the Custom CNN or Dense-Net. It achieved an accuracy of 87.12% and demonstrated a balanced performance, but lacked the feature extraction capacity required for this task.
- **VGG16**:
  - VGG16, although highly regarded for its simplicity, achieved the lowest accuracy (85.32%) and relatively weaker performance across all metrics. Its

deeper architecture did not provide the necessary advantage for this specific task.

## VII.User Interface and Application

### 7.1 Image Upload Functionality

The user interface is designed to be intuitive and user-friendly, facilitating a seamless experience for users uploading images of plant leaves. The **image upload functionality** includes the following features:

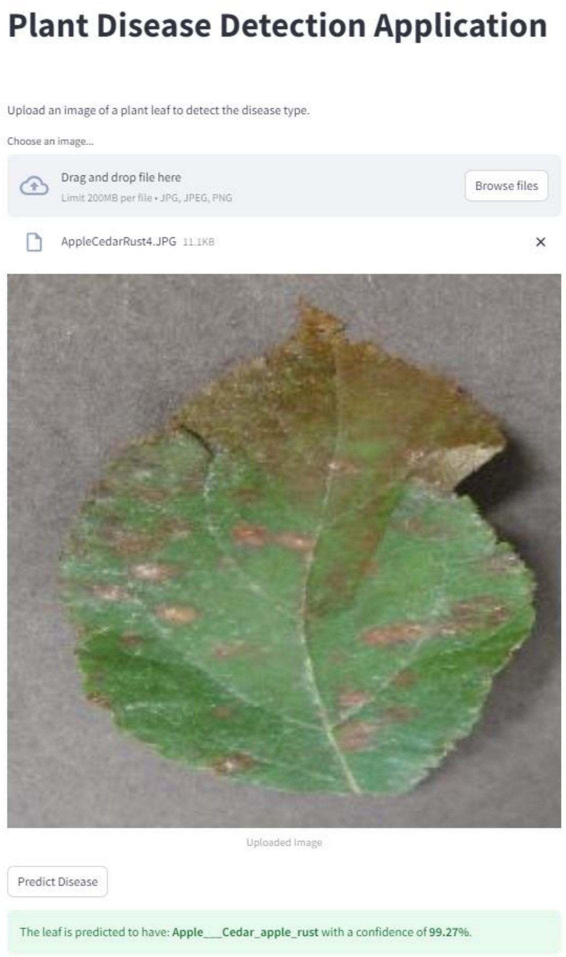
- **File Type Validation**:
    - The application supports common image formats, including JPEG and PNG. Upon attempting to upload an unsupported file type, users receive a clear error message, prompting them to select a valid image format.
- **Drag-and-Drop Support**:
    - Users can easily drag and drop images directly into the upload area, enhancing accessibility and streamlining the upload process. This feature is particularly beneficial for users who may be using touch devices or have limited technical expertise.

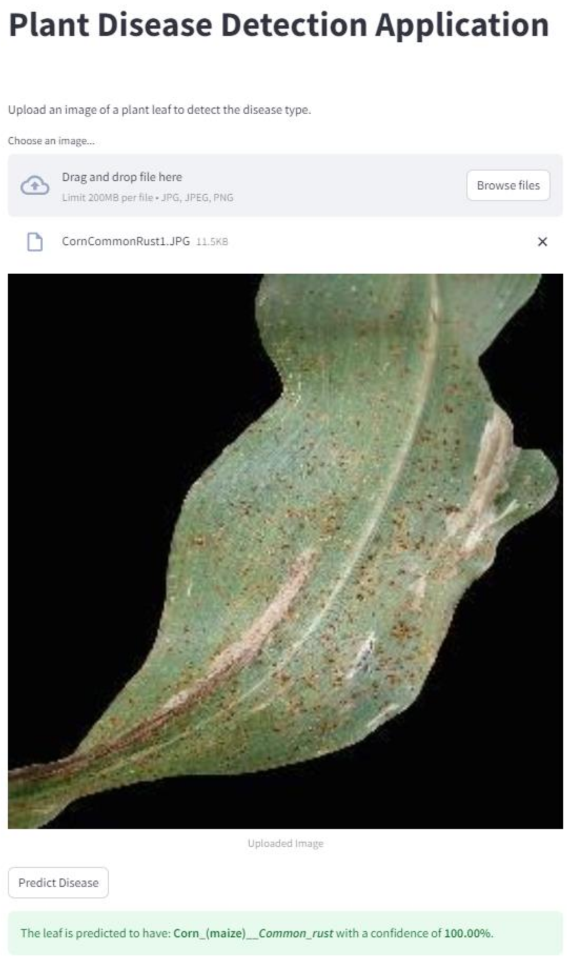
### 7.2 Prediction Display and Confidence Score

Once an image has been successfully uploaded, the application processes the image and displays the results in a clear and informative manner:

- **Predicted Disease**:
    - The application displays the predicted plant disease prominently at the top of the results section, ensuring that users can quickly understand the outcome of their upload. The disease name is accompanied by a visual representation (if applicable) to aid in identification.
- **Confidence Score**:
    - Alongside the predicted disease, a confidence score (expressed as a percentage) indicates the model's certainty about its prediction. This score helps users gauge the reliability of the prediction, with higher scores suggesting greater confidence.

## 7.3 Screenshots of Streamlit Application




## 7.4 How to Use the Application:

Follow the steps to upload images and interpret predictions for plant disease management.

1. Open the Stream-lit web app.
2. Click "Browse files" to select a plant leaf image (JPEG/PNG) from your device.
3. Submit the Image, the uploaded image will appear.
4. Click "Predict" to receive the diagnosis.
5. View the predicted disease and the confidence score (accuracy percentage).

## VIII. Conclusion

### 8.1 Summary of Achievements

In this project, we successfully developed a robust plant disease detection system. Key achievements include:

- **Model Development**:
    - We designed and implemented a custom Convolutional Neural Network (CNN) architecture alongside leveraging well-established pretrained models such as VGG16, Alex-Net, and Dense-Net. Through extensive training and validation, we were able to optimize these models to accurately classify images of diseased plants.
- **Dataset Utilization**:
    - A diverse dataset encompassing various plant species and disease classes was curated, and effective data preprocessing techniques were applied, including resizing, normalization, and augmentation. These efforts ensured that the models were trained on high-quality data that closely resembles real-world conditions.
- **Performance Evaluation**:
    - Comprehensive evaluation metrics such as accuracy, precision, recall, and F1-score were utilized to assess model performance. The results demonstrated significant improvements in disease detection accuracy, thereby validating the effectiveness of our approach.
- **User Interface Development**:
    - We created a user-friendly application using Streamlit, allowing users to upload images easily and receive instantaneous predictions. The interface was designed to enhance user experience by providing clear instructions and feedback mechanisms.

### 8.2 Importance of Early Plant Disease Detection

The early detection of plant diseases is paramount in modern agriculture for several reasons:

- **Economic Benefits**:
  - Timely identification of diseases can significantly reduce crop losses, ultimately leading to higher yields and improved profitability for farmers. Early interventions can mitigate the spread of diseases, allowing for targeted treatments and reducing the need for more expensive measures later on.
- **Sustainability**:
  - Early disease detection contributes to sustainable agricultural practices by minimizing chemical inputs and resource wastage. By accurately diagnosing issues, farmers can apply precise treatments, thus promoting environmentally friendly practices.
- **Food Security**:
  - In an era of increasing global food demand, early disease detection plays a crucial role in ensuring food security. By protecting crops and optimizing yields, we can help sustain food supplies for growing populations.

### 8.3 Potential Impact on Agriculture

The potential impact of our plant disease detection system on the agricultural sector is substantial:

- **Empowering Farmers**:
  - Providing farmers with accessible tools for disease detection empowers them to take proactive measures in managing crop health. This can lead to increased confidence in farming practices and a shift toward data-driven decision-making.
- **Facilitating Research and Development**:
  - The insights gained from the application can inform further research into disease management strategies, helping to develop new methods for disease control and prevention tailored to specific crops and local conditions.