Project Title: Plant Disease Detection using Deep Learning

Introduction / Overview

This is a deep learning–based web application for detecting plant leaf diseases. Built using Streamlit and TensorFlow, the app classifies uploaded leaf images into one of three categories: Early Blight, Late Blight, or Healthy.

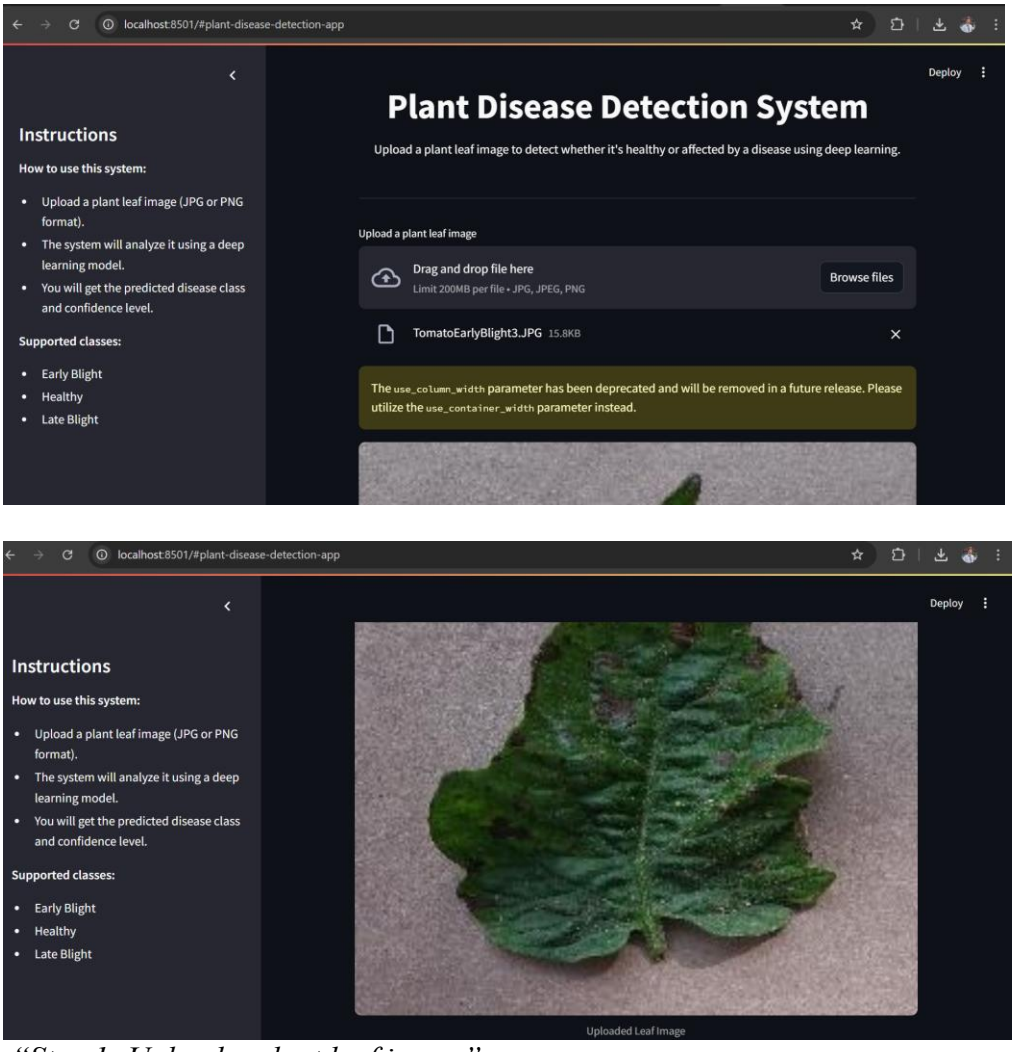
Instructions to Use the App

Follow these steps to use the application:

1. **Launch** the app (either from deployment or by running app.py via Streamlit).
2. **Upload** a plant leaf image (JPEG/PNG format).
3. **View results**, including disease prediction and a confidence bar chart

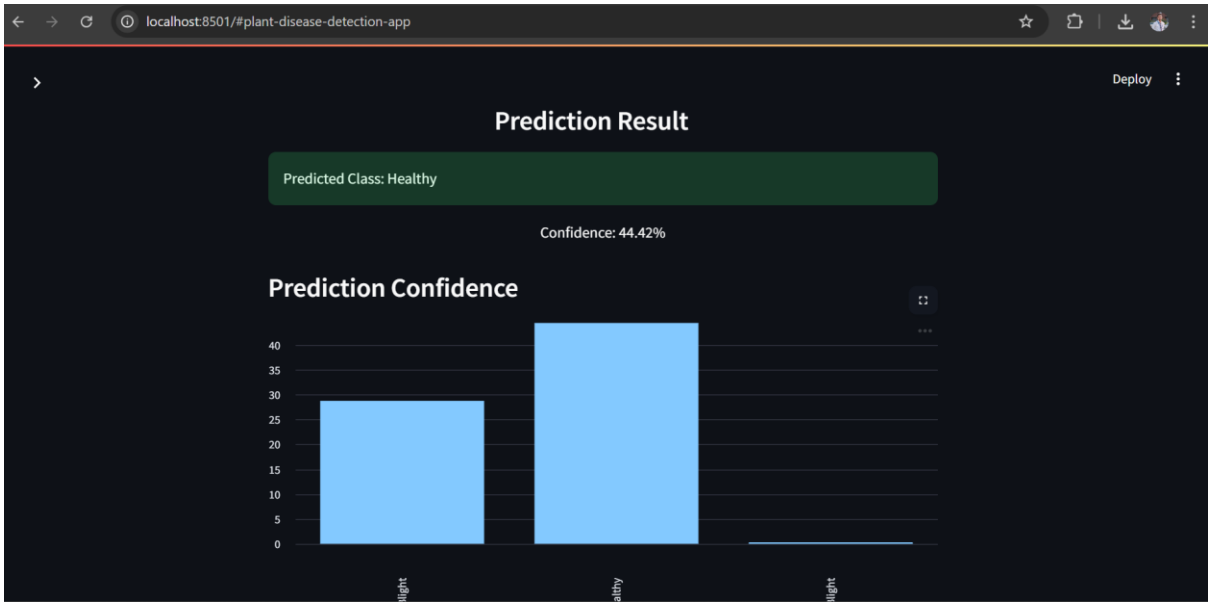
Screenshots

- Upload section



"Step 1: Upload a plant leaf image"

- Prediction result



"Step 2: Model predicts the disease class and shows confidence level"

- Confidence bar chart

"Step 3: Visualization of confidence scores"

Tools & Technologies

- Python, TensorFlow, Keras
- Streamlit for web interface
- Pillow, NumPy for image processing
- Trained model: plant_disease_model.keras

GitHub Repository

🔗 GitHub: github.com/DeveshNathJha/Plant-Disease-Detection

Notes on Training Dataset

The model was trained using a subset of the publicly available **PlantVillage dataset**, which contains thousands of labeled images of healthy and diseased plant leaves. Specifically, for this project:

- **Classes used**:
  - Early Blight
  - Late Blight
  - Healthy
- **Image size**:
  All images were resized to 128x128 pixels to reduce computation and standardize input.
- **Preprocessing steps**:
  - Normalized pixel values (divided by 255)
  - Converted images to NumPy arrays
  - Expanded dimensions to fit model input shape
- **Model performance**:
  The trained model achieved high accuracy during testing on a validation split, indicating good generalization for the selected disease types.

Challenges faced:

- **Model Size**:
  The trained Keras model (.keras) file was too large for direct upload to GitHub, so it required Git LFS or ZIP handling.
- **Dataset Balance**:
  Originally, some classes had more samples than others, which required data balancing and augmentation to avoid biased predictions.
- **Web App Styling**:
  Streamlit's theme customization has limitations. Attempts to use custom themes faced issues where changes did not reflect dynamically.
- **Performance on Similar Classes**:
  The model sometimes confused Early and Late Blight due to similar visual patterns in leaves.

Future Improvements (Planned Work)

- **We will expand the number of disease classes and plant types** to make the model more versatile and applicable to a wider range of crops.
- **We will optimize the model** using techniques like **quantization** and **pruning** to reduce file size and improve inference speed, especially for edge deployment.