Plant Disease Detection Project Report

Abstract

This project is about detecting diseases in tomato leaves using deep learning. I used **MobileNetV2** as a pre-trained model and trained it on **PlantVillage dataset** images. The model classifies leaves into **five categories**:

- Tomato Late Blight
- Tomato Early Blight
- Tomato Leaf Mold
- Tomato Bacterial Spot
- Healthy Leaf

I applied **data preprocessing, augmentation, and class weighting** to improve model performance. Finally, I created a **Streamlit web app** to allow users to upload leaf images and get predictions instantly.

Introduction

Plant diseases can reduce crop yield and affect farmers' income. Detecting these diseases manually takes time and knowledge. This project uses deep learning to automatically identify diseases from leaf images. It provides a fast and easy way to check the health of tomato plants.

Tools Used

- **Python** – programming language
- **TensorFlow & Keras** – for building and training the model
- **OpenCV & NumPy** – for image processing
- **Matplotlib** – to visualize training results
- **Scikit-learn** – to compute class weights for imbalance handling
- **Streamlit** – to create a simple web application

Steps Involved in Building the Project

1. **Dataset Preparation**
    - Used the **PlantVillage dataset** with tomato leaf images.

   - Organized images into **five classes**.

2. **Data Preprocessing and Augmentation**
   - Resized images to **128×128 pixels**.
   - Applied **rotation, width/height shift, zoom, flips, brightness changes** using `ImageDataGenerator`.
   - Split dataset into **training (80%)** and **validation (20%)** sets.
3. **Handling Class Imbalance**
   - Calculated **class weights** using `sklearn.utils.class_weight` to give more importance to smaller classes.
4. **Model Building**
   - Used **MobileNetV2** as the base model (pre-trained on ImageNet) with **frozen weights**.
   - Added **GlobalAveragePooling2D**, **Dense layers**, and **Dropout** for classification.
   - Used **softmax activation** for 5-class output.
5. **Training the Model**
   - Optimizer: **Adam** with learning rate 0.0001
   - Loss function: **categorical crossentropy**
   - Callbacks: **EarlyStopping** and **ModelCheckpoint**
   - Trained for **20 epochs** using augmented data and class weights
6. **Testing the Model**
   - Created a script to test single leaf images.
   - Model outputs **predicted class** and **confidence score**.
7. **Deploying the Model**
   - Developed a **Streamlit app** (`app.py`).
   - Users can upload an image and see the predicted disease along with confidence.

Conclusion

The project successfully demonstrates a system for detecting tomato leaf diseases using deep learning. The trained **MobileNetV2 model** can classify leaf images into five categories accurately.

The **Streamlit app** provides a user-friendly interface for real-time predictions. This system can help farmers detect diseases early and take action to protect crops.
