Project Report: Plant Disease Detection Using CNN Model

1. Introduction

Plant diseases significantly affect agricultural productivity and food security. Traditional methods of disease identification involve manual inspection, which is time-consuming and prone to human error. This project leverages Convolutional Neural Networks (CNN) to build an intelligent system that can automatically detect plant diseases from leaf images.

2. Objective

- To develop a deep learning model using CNN that can accurately classify images of plant leaves into healthy or diseased categories.
- To deploy the model for practical use in assisting farmers or agriculturalists in disease identification.

3. Dataset

Dataset Used: PlantVillage Dataset

- Total Classes: Multiple plant species with various disease categories.
- Image Format: RGB images of size 256x256.
- Data Split: Training set, Validation set, Test set
- Image Augmentation Techniques Used:
  - Rescaling
  - Rotation
  - Shear
  - Zoom
  - Horizontal Flip

4. Model Architecture (CNN)

The model is built using the Keras deep learning library with the TensorFlow backend.

Layers Used:

- Input Layer: Accepts 256x256 RGB images
- Conv2D: Filters (32, 64, 128), Kernel size 3x3
- Activation: ReLU
- MaxPooling2D: Pool size 2x2
- Dropout: Regularization to avoid overfitting
- Flatten: Convert 2D feature maps to 1D

- Dense Layer: 128 neurons
- Output Layer: Softmax activation for multi-class classification

5. Model Training

- Optimizer: Adam
- Loss Function: Categorical Crossentropy
- Metrics: Accuracy
- Epochs: 10 (can be increased for better performance)
- Batch Size: 32

6. Results & Evaluation

- Accuracy Achieved: (e.g., 95% on validation data — update based on actual training logs)
- Loss: Gradually decreasing trend during training
- Model Evaluation: Performed on unseen test data

Visualization: Accuracy and loss graphs plotted using Matplotlib.

7. Model Saving

The trained model is saved in `.h5` format for future use:
model.save("plant_disease_model.h5")

8. Future Scope

- Deploy the model on a web or mobile application using Flask or TensorFlow Lite.
- Add more datasets for better generalization.
- Use Transfer Learning (e.g., with ResNet, VGG16) to improve accuracy and reduce training time.

9. Conclusion

This project demonstrates that a CNN-based approach is highly effective in detecting plant diseases from leaf images. By automating disease detection, this model can play a crucial role in enhancing modern agriculture practices, enabling early intervention and improving crop yield.