Project Report: Plant Disease Classification Using CNN

1. Title

Plant Disease Detection Using Convolutional Neural Networks

---

2. Abstract

This project focuses on building a deep learning model to classify plant diseases using images. Using the PlantVillage dataset, a Convolutional Neural Network (CNN) was developed and trained to identify healthy and diseased plant leaves across multiple crops. The model achieved high accuracy on the validation set and can be used for real-time disease detection in agriculture.

---

3. Introduction

Agriculture is a critical sector, and early detection of plant diseases can prevent significant crop losses. Traditional manual inspection is time-consuming and requires expertise. Machine learning and computer vision provide an automated solution for detecting and classifying plant diseases from leaf images. This project aims to implement a CNN-based model to classify images of plant leaves into multiple disease categories or healthy class.

---

4. Dataset Description

Dataset: PlantVillage (Available on Kaggle)
Dataset Size: Over 50,000 images spanning 38 classes.
Classes Example: Apple scab, Black rot, Cedar apple rust, Grape healthy, Tomato mosaic virus, etc.
Data Split: 80% training, 20% validation using Keras ImageDataGenerator.

The dataset contains three types of images: color, grayscale, and segmented. For this project, **color images** were used.

---

5. Methodology

5.1 Environment Setup

Python environment with TensorFlow and Keras was used. Random seeds were set for reproducibility:

```
import random, numpy as np, tensorflow as tf

random.seed(0)

np.random.seed(0)

tf.random.set_seed(0)
```

5.2 Data Preprocessing

- Images resized to 224x224 pixels.
- Pixel values normalized to the range [0,1].
- Training and validation generators were created with an 80-20 split using ImageDataGenerator.

5.3 Model Architecture

A **Convolutional Neural Network (CNN)** was designed:

Layer	Output Shape	Parameters
Conv2D (32 filters, 3x3)	(222, 222, 32)	896
MaxPooling2D	(111, 111, 32)	0
Conv2D (64 filters, 3x3)	(109, 109, 64)	18,496
MaxPooling2D	(54, 54, 64)	0
Conv2D (128 filters, 3x3)	(52, 52, 128)	73,856
MaxPooling2D	(26, 26, 128)	0
Flatten	(86528,)	0
Dense (256)	(256)	22,151,424
Dense (38, softmax)	(38)	9,766

Total Parameters: 22,254,438 (all trainable)

5.4 Training

- Optimizer: Adam
- Loss: Categorical Crossentropy
- Metrics: Accuracy
- EarlyStopping was used to prevent overfitting (monitoring val_loss, patience=3).
- Epochs: Up to 50

```
history = model.fit(
  train_generator,
  validation_data=validate_generator,
  epochs=50,
  callbacks=[early_stop]
)
```

5.5 Model Evaluation

The model was evaluated on the validation set:

```
test_loss, test_acc = model.evaluate(validate_generator)
print(f"Test accuracy: {test_acc}")
```

5.6 Prediction Function

A utility function was implemented to preprocess any leaf image and predict its class:

```
predicted_class_name = predict_class(model, image_path, class_indices)
print("Predicted Class Label:", predicted_class_name)
```

5.7 Saving the Model

The trained model was saved as .h5 for future use:

```
model.save('plant_disease_model.h5')
```

---

6. Results

- **Training and Validation Accuracy & Loss Curves:**

Insert plots here of accuracy and loss over epochs.

- **Validation Accuracy:** Example output: ~92% (can include your actual number).
- **Sample Predictions:**

Image	True Class	Predicted Class
Apple leaf image	Apple___Black_rot	Apple___Black_rot
Tomato leaf image	Tomato___Early_blight	Tomato___Early_blight

- The model successfully distinguishes healthy vs diseased leaves across multiple crops.

---

7. Discussion

- The CNN effectively extracted features from leaf images.
- EarlyStopping prevented overfitting.
- The model can be deployed for real-time plant disease detection using a mobile application.
- Limitations: Requires sufficient lighting and clear leaf images; performance may decrease with occluded or low-resolution images.

---

8. Conclusion

This project demonstrates a robust deep learning approach to detect and classify plant diseases. The CNN achieved high validation accuracy and can assist farmers in early disease detection, reducing crop losses and improving agricultural productivity.

---

9. Future Work

- Augment dataset with more leaf images from diverse environments.
- Use transfer learning with pre-trained models like ResNet or EfficientNet for better performance.
- Deploy as a mobile or web app for practical use.
- Implement segmentation to focus on diseased regions.

10. References

1. PlantVillage Dataset – Kaggle: https://www.kaggle.com/abdallahalidev/plantvillage-dataset
2. Chollet, F. (2017). Deep Learning with Python. Manning Publications.
3. TensorFlow/Keras Documentation: https://www.tensorflow.org/