Project Document: Plant Disease Detection using CNN in Google Colab

# Abstract

This project focuses on using Convolutional Neural Networks (CNNs) in Google Colab to detect plant diseases from leaf images. The model is trained on the PlantVillage dataset and implemented in a beginner-friendly Colab environment. This document explains the project, methodology, and provides an in-depth explanation of the code.

# 1. Introduction

Plant diseases significantly impact crop yield and food security. Early and accurate detection of plant diseases is crucial. This project leverages deep learning, particularly CNNs, to automate the identification process using leaf image classification.

# 2. Dataset Description

The dataset used is the 'PlantVillage' dataset from Kaggle, containing over 50,000 images of healthy and diseased plant leaves. The images are categorized by plant type and disease label, which makes it suitable for multi-class classification using CNNs.

# 3. Project Setup in Google Colab

The model is implemented in Google Colab for accessibility and ease of GPU usage. Necessary libraries are installed using pip commands and the dataset is loaded from Kaggle using API access.

# 4. Code Explanation

\*\*Step 1: Import Required Libraries\*\*

The first step involves importing necessary libraries such as TensorFlow, Keras, NumPy, Matplotlib, and libraries for image preprocessing and plotting.

\*\*Step 2: Load and Preprocess Dataset\*\*

Images are resized uniformly (e.g., 128x128 pixels) and normalized to a 0-1 scale. They are then split into training, validation, and test sets. ImageDataGenerator is used to apply data augmentation techniques like rotation, flipping, and zooming.

\*\*Step 3: Define the CNN Model\*\*

A sequential model is built with layers:  
- Convolutional layers (Conv2D) with ReLU activation  
- MaxPooling layers to reduce spatial dimensions  
- Dropout layers to prevent overfitting  
- Dense layers for classification, ending with a softmax layer for multi-class output

\*\*Step 4: Compile the Model\*\*

The model is compiled using 'categorical\_crossentropy' loss and the 'adam' optimizer, with accuracy as the evaluation metric.

\*\*Step 5: Train the Model\*\*

Model is trained using the fit() function with training and validation data. Epochs and batch size are set depending on GPU availability and dataset size.

\*\*Step 6: Evaluate the Model\*\*

The trained model is evaluated on test data. Accuracy, loss, and confusion matrix are plotted for performance analysis.

\*\*Step 7: Predict and Visualize Results\*\*

The model predicts plant disease for new images. Predictions are visualized using matplotlib with predicted and actual labels.

# 5. Results

The model achieved high accuracy on the PlantVillage dataset. Results include accuracy plots, confusion matrix, and correctly classified leaf images.

# 6. Conclusion

The CNN model effectively detects plant diseases from images using Google Colab. The approach can be extended to real-time disease monitoring systems in agriculture.

# 7. References

[1] PlantVillage Dataset - https://www.kaggle.com/datasets/emmarex/plantdisease  
[2] TensorFlow & Keras Documentation - https://www.tensorflow.org/  
[3] Google Colab - https://colab.research.google.com/