# Plant Disease Detection System for Sustainable Agriculture

## Documentation: Problem Statement & Pipeline

---

### Problem Statement

The goal of this project is to develop a Convolutional Neural Network (CNN)-based model to detect and classify plant diseases from images of leaves across various crops, including apple, cherry, grape, and corn. The system will:  
1. Accurately distinguish between healthy and diseased leaves.  
2. Predict the specific type of disease affecting the plant (e.g., apple scab, corn blight).  
3. Support precision agriculture by enabling early disease diagnosis, improving crop management practices, and reducing yield losses.  
  
This system aims to address the critical need for automated, scalable solutions in agriculture to enhance food security and sustainability.

---

### Pipeline Overview

The workflow for developing the Plant Disease Detection System is divided into four stages:

#### 1. Data Collection & Data Loading

- Datasets:  
 - Category 1: Contains images of specific crops (e.g., apple, corn).  
 - Category 2: Includes labeled data splits for training, testing, and validation.  
- Data Splits:  
 - Train: Used to train the CNN model.  
 - Test: Reserved for final model evaluation.  
 - Valid: Used for hyperparameter tuning and validation during training.

#### 2. Data Compression & Upload

- Compress datasets into a ZIP file and upload them to a cloud drive (e.g., Google Drive).  
- Mount the drive programmatically (e.g., using Python in Google Colab) to access the data.  
- Execute code to unzip and load the dataset into the working environment.

#### 3. Image Preprocessing & Augmentation

- Preprocessing:  
 - Resize images to a uniform resolution.  
 - Normalize pixel values.  
 - Convert images to grayscale or RGB as needed.  
- Augmentation:  
 - Apply techniques like rotation, flipping, and brightness adjustments to enhance dataset diversity.  
 - Prevent overfitting and improve model generalization.

#### 4. CNN Model Development & Evaluation

- Model Architecture:  
 - Design a CNN with convolutional layers, pooling layers, dropout, and fully connected layers.  
 - Use activation functions like ReLU and Softmax for classification.  
- Training:  
 - Train the model on the augmented dataset.  
 - Optimize using loss functions (e.g., cross-entropy) and Adam optimizer.  
- Evaluation:  
 - Test the model on the test dataset to measure accuracy, precision, and recall.  
 - Iterate on the model based on validation results.

## Week 1

### Briefly mention the improvisations done by you:

- Structured the project pipeline for plant disease detection.  
- Gathered and organized the dataset into train, test, and validation splits.  
- Uploaded and accessed the dataset through Google Drive in Google Colab.  
- Performed basic image preprocessing and implemented image augmentation techniques to improve model generalization.  
- Initiated the design of the CNN architecture tailored for leaf image classification.

### GitHub Repository Link for Week 1 Milestone (Project Source Code):

https://github.com/your-repo