**1. Problem Statement**

**Title:** *Plant Disease Detection System for Sustainable Agriculture*

**Objective:**  
To build an AI-based image classification system using Convolutional Neural Networks (CNN) to detect and identify various plant diseases from leaf images. The goal is to help farmers and agriculturalists take early action, reduce crop losses, and promote sustainable farming practices through automation and technology.

**2. AI/ML Pipeline Explanation**

The pipeline we follow for this project includes the following steps:

**Step 1: Data Collection & Data Loading**

* Gather leaf images from different plant categories.
* Split the data into three sets:
  + **Train** (used to train the model),
  + **Validation** (used to fine-tune the model),
  + **Test** (used to evaluate final accuracy).

**Step 2: Dataset Structure**

Organized as:

bash

CopyEdit

Dataset/

├── train/

│ ├── category1/

│ └── category2/

├── valid/

│ ├── category1/

│ └── category2/

└── test/

├── category1/

└── category2/

**Step 3: Data Upload & Mount**

* Zip the dataset folder.
* Upload the .zip file to **Google Drive**.
* Mount the Google Drive in **Google Colab** using Python code.
* Unzip the file and access the dataset for training.

**Step 4: Image Processing & Image Augmentation**

* Resize all images to a standard dimension (e.g., 128×128).
* Apply **Image Augmentation** to improve model performance:
  + Rotation
  + Flipping
  + Zooming
  + Color shifting  
    This helps make the model more robust and accurate.

**Step 5: Model Building (CNN)**

* Use a **Convolutional Neural Network (CNN)** model for image classification.
* CNN is ideal for tasks involving image data due to its ability to detect spatial hierarchies and patterns.

**Step 6: Testing & Evaluation**

* Evaluate the model using the test set.
* Measure performance using accuracy, loss, confusion matrix, etc.

**Neural Networks Summary**

| **Type** | **Full Form** | **Use Case** |
| --- | --- | --- |
| **ANN** | Artificial Neural Network | For general data processing |
| **RNN** | Recurrent Neural Network | For sequences, language, and emotions |
| **CNN** | Convolutional Neural Network | For image classification and computer vision |