A white sign with black text

Description automatically generated****

**PlantShield: AI-Driven Disease Diagnosis and**

**Smart Sensor-Based Irrigation System**

Data Preparation Feature Engineering and Model: Capstone Project (Third Assignment)

Group4 Members: Mohammad Sameer Naeemi, Tahmina Rafat, Shoaib Fateh, Safiullah Arian, Abdul Wahab Achakzai, Nasratullah Shafiq.

Date: 01/10/2025

Instructed by: Eda nur Saruhan

Contents

[Data Preparation/Feature Engineering 3](#_Toc188213945)

[1. Overview 3](#_Toc188213946)

[2. Data Collection 3](#_Toc188213947)

[3. Data Cleaning 3](#_Toc188213948)

[4. Exploratory Data Analysis (EDA) 4](#_Toc188213949)

[Key Visualizations: 5](#_Toc188213950)

[5. Feature Engineering 7](#_Toc188213951)

[6. Data Transformation 8](#_Toc188213952)

[Model Exploration 9](#_Toc188213953)

[1. Model Selection 9](#_Toc188213954)

[2. Model Training 9](#_Toc188213955)

[3. Model Evaluation 9](#_Toc188213956)

[4. Code Implementation 10](#_Toc188213957)

Data Preparation/Feature Engineering

## Overview

Data preparation and feature engineering are crucial steps in any machine learning project. These phases involve transforming raw data into a format suitable for modeling, which directly impacts the model's performance. It involves collecting, cleaning, and transforming data into a format suitable for analysis and modeling. Properly prepared data can highlight important patterns and relationships within the dataset, leading to more accurate predictions. These steps help enhance data quality, reduce noise, and improve model performance by making the dataset more informative and structured.

## Data Collection

The dataset used in this project was sourced from the New Plant Diseases Dataset. It contains 70,295 labeled images with different healthy and unhealthy crop leaves, categorized by plant type and disease.

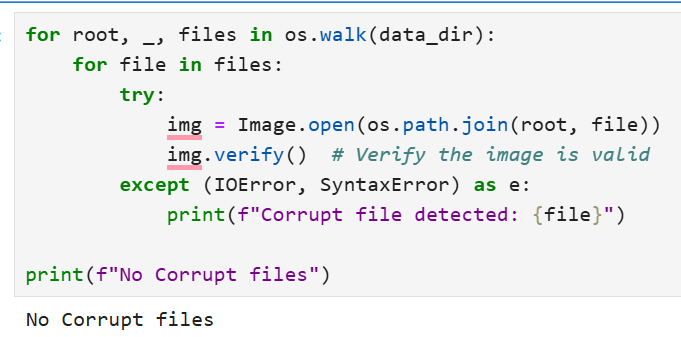
**Preprocessing during collection:**

* **Downloaded the Dataset:** The dataset was inside a large zip file with size of 3GBs
* **Dataset structure adjustment:** overviewed the images and folders based on their class labels.
* **Duplicates removal:** Scanned and eliminated duplicate images to prevent bias.

## Data Cleaning

To prepare the raw data for analysis, the following steps were undertaken:

* **Missing values:** Verified that no images or labels were missing from the dataset.
* **Outliers:** Reviewed image dimensions and removed corrupt or incomplete files.

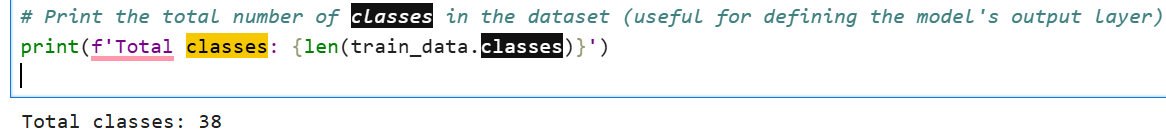


* **Duplicates removal:** Removed the duplicate images which resulted to reduce the size of dataset from 3GB to 1.4GB.

## Exploratory Data Analysis (EDA)

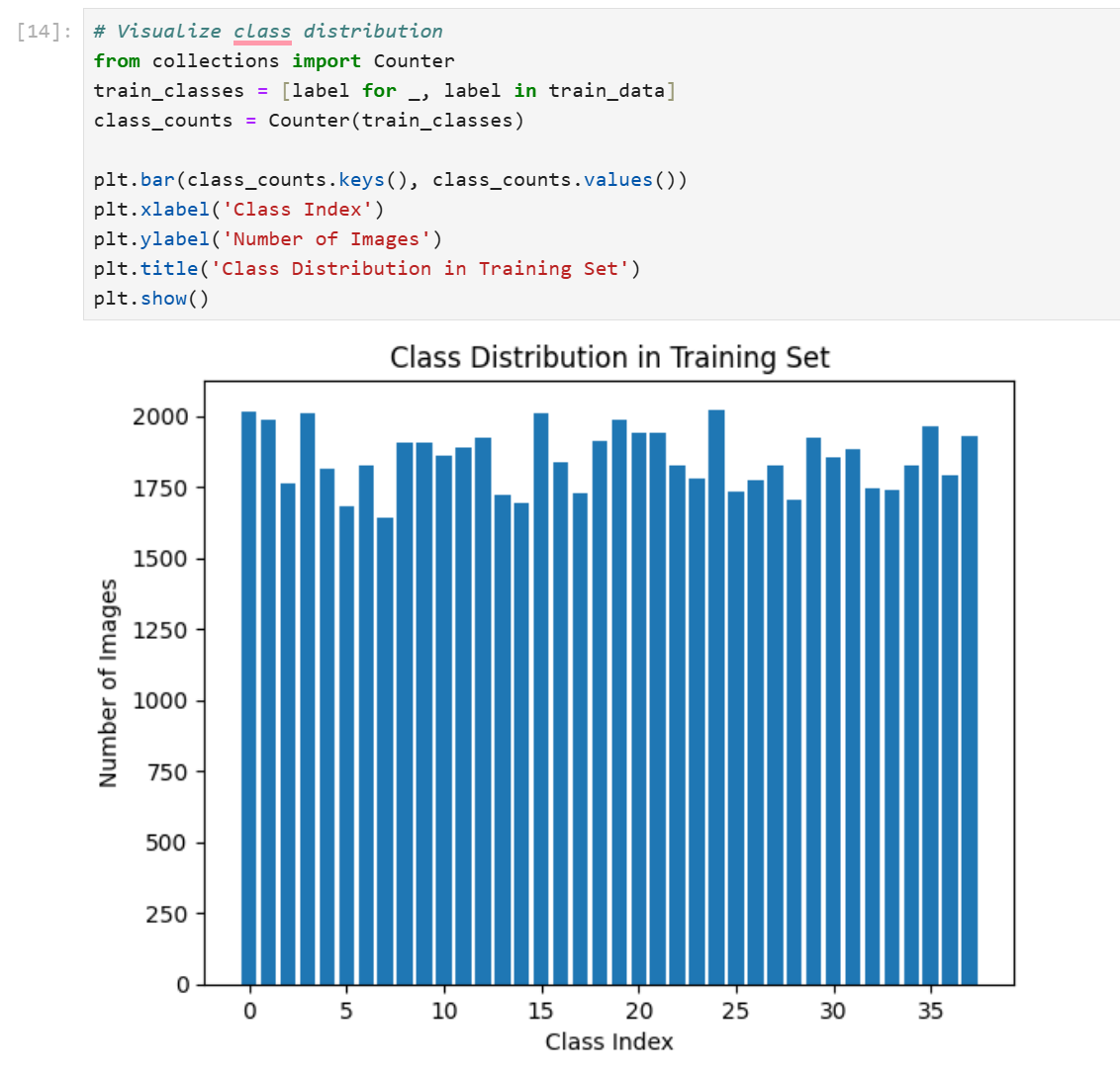
**Summary Statistics:**

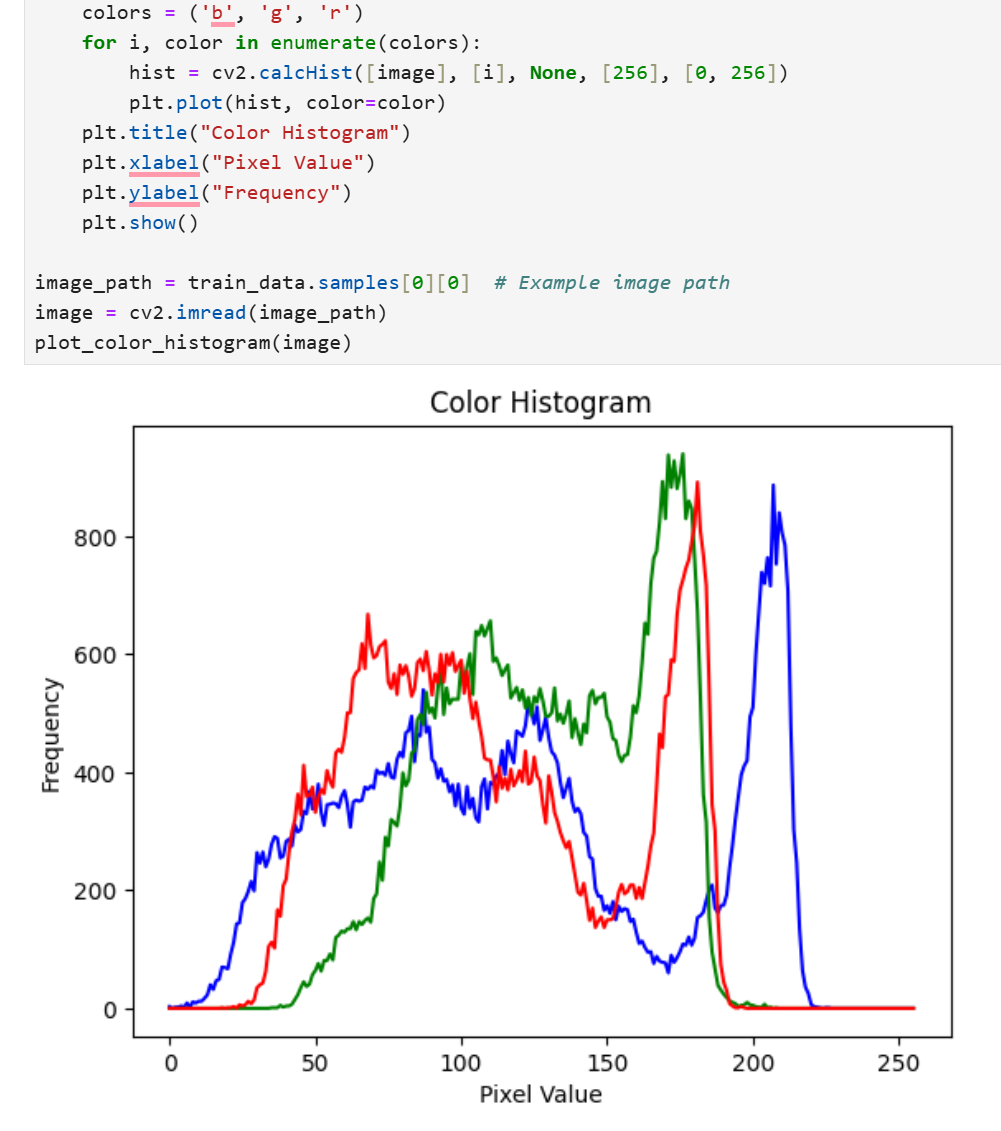
* **Number of classes**: 38 (healthy and diseased categories across various plants).



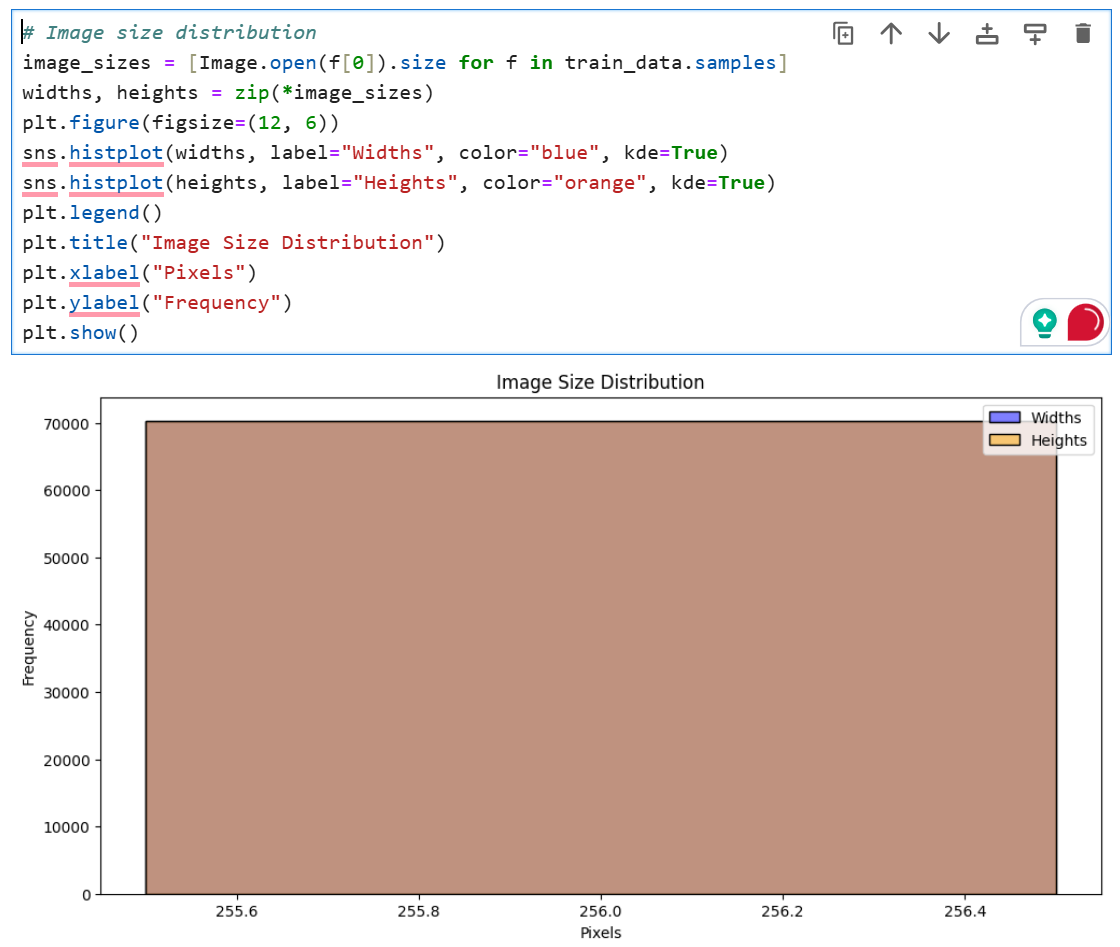
* **Sample distribution**: Balanced across most classes, with minor imbalances addressed later through augmentation.

### Key Visualizations:

* **Class distribution:** A bar plot showing the count of samples per class to understand dataset balance. 
* **Sample images:** Displayed example images for each class, highlighting variations in disease appearance.
* **Color analysis:** Histograms of RGB values to analyze color intensity distribution across images.



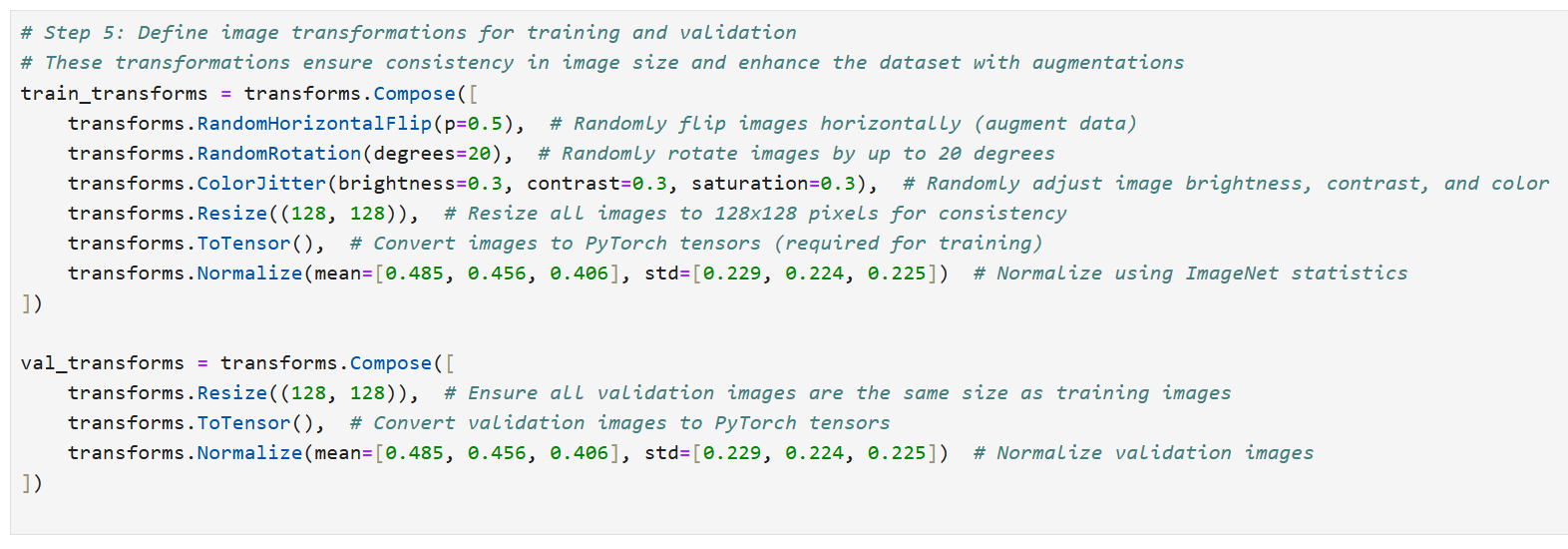
* **Image size distribution:** Visualized the pixel dimensions to ensure consistency before resizing.



## Feature Engineering

**Techniques used:**

* **Augmentation:**
  + Applied random horizontal flips, rotations (up to 30 degrees), and color jitter to increase dataset diversity and prevent overfitting.



* **Label encoding:**
  + Converted class names into numerical values for model compatibility.

## Data Transformation

* **Resizing:** Rescaled all images to 128x128 pixels to ensure uniformity.
* **Normalization:** Applied mean subtraction and scaling to range [0, 1] using dataset-wide mean and standard deviation.
* **One-hot encoding:** Encoded class labels into one-hot format for multi-class classification.

Model Exploration

## Model Selection

The **Convolutional Neural Network (CNN)** architecture was chosen for this project, specifically using a **ResNet-18** pretrained model from the PyTorch library.

**Rationale:**

* CNNs are highly effective for image classification tasks due to their ability to capture spatial features.
* ResNet-18’s residual blocks mitigate vanishing gradient issues, making it suitable for deep learning tasks. (Worked better than ResNet50 in my case)

**Strengths:**

* Excellent feature extraction capabilities.
* Pretrained weights allow faster convergence.

**Weaknesses:**

* Computationally expensive compared to simpler models.

## Model Training

**Training details:**

* **Optimizer:** Adam with a learning rate of 0.001.
* **Loss function:** CrossEntropyLoss.
* **Batch size:** 32. (64 somehow made it slower)
* **Data split:** The data was already split into three folders, train, validate, test

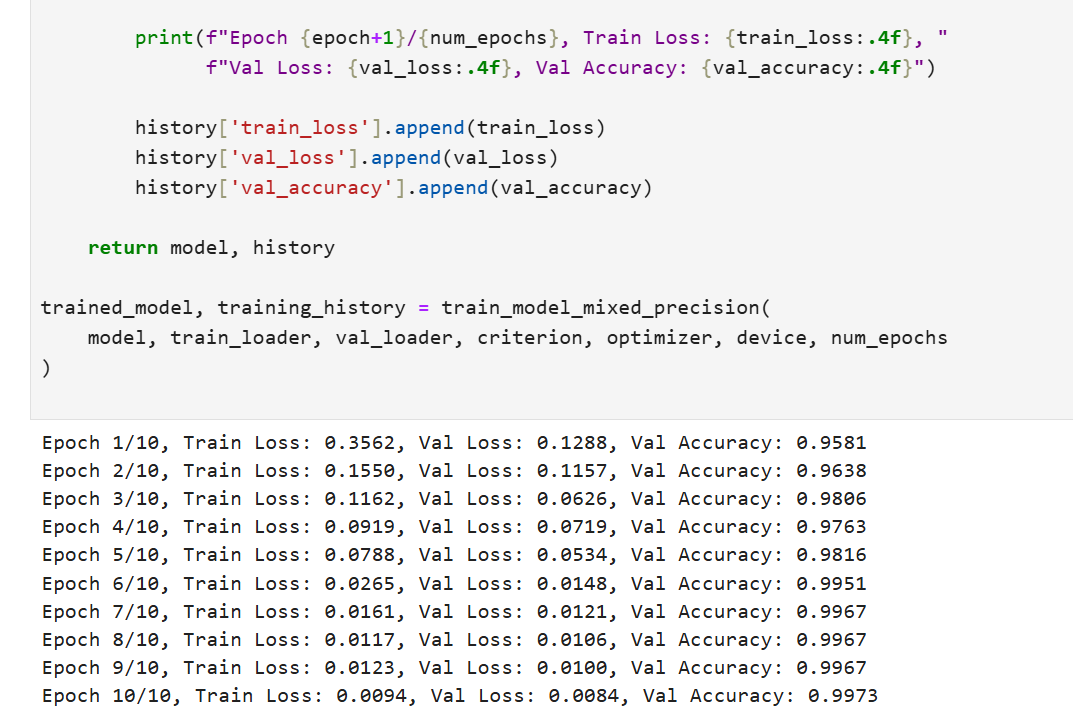
## Model Evaluation

**Metrics used:**

* **Accuracy:** Overall percentage of correctly classified samples.
* **Confusion Matrix:** Provided class-wise performance insights.
* **Precision, Recall, F1-Score:** Evaluated per-class effectiveness.
* **ROC Curve:** Analyzed the trade-off between sensitivity and specificity.

**Key Results:**

* Achieved up to **99.73% accuracy** on the test set using Epoch number of 10.
* High precision and recall across all classes, indicating minimal false positives and negatives.



## Code Implementation

I have attached the codes and their results above on their respective places.