**Data Preparation/Feature Engineering**

**Project Title: AI-Powered Early Detection of Crop Diseases for Smallholder Farmers**

**Team Members**

1. Mubarek Jemal
2. Nahom Tibebu
3. Natnael Meles

Table of Contents

[**List of Figures** 3](#_Toc196947773)

[**Data Preparation/Feature Engineering** 1](#_Toc196947774)

[**1.** **Overview** 1](#_Toc196947775)

[**2.** **Data Collection** 2](#_Toc196947776)

[**3.** **Data Cleaning** 2](#_Toc196947777)

[**4.** **Exploratory Data Analysis (EDA)** 5](#_Toc196947778)

[**5.** **Data Transformation** 9](#_Toc196947779)

[**5.1.** **Resizing and Format Standardization** 9](#_Toc196947780)

[**6.** **Model Exploration** 11](#_Toc196947781)

[**6.1.** **Model Selection** 11](#_Toc196947782)

[**6.2.** **Model Training** 12](#_Toc196947783)

[**6.3.** **Model Evaluation** 14](#_Toc196947784)

[**7.** **Code Implementation** 16](#_Toc196947785)

# **List of Figures**

[Figure 1 Image with heavy shadows, blurred, and distracting background 4](#_Toc196947588)

[Figure 2 Class distribution 5](#_Toc196947589)

[*Figure 3 Image Dimension Distribution* 6](#_Toc196947590)

[Figure 4 Brightness Distribution of images 7](#_Toc196947591)

[Figure 5 Sample images from each class 8](#_Toc196947592)

[Figure 6 Model Validation Accuracy 14](#_Toc196947593)

[Figure 7 Model loss 15](#_Toc196947594)

# **Data Preparation/Feature Engineering**

# **Overview**

The data preparation and feature engineering phase are a critical step in any machine learning project, as it lays the foundation for model performance and accuracy. This phase begins with data collection, where raw data is gathered from various sources such as databases, APIs, or user inputs. Once collected, the data must undergo cleaning to address inconsistencies, missing values, outliers, and duplicates that could bias or degrade the model's performance. Standard techniques like imputation, normalization, and encoding are applied to ensure the dataset is consistent, complete, and suitable for analysis.

In developing an AI-powered tool using the PlantVillage dataset to help smallholder farmers detect crop diseases early, the data preparation and feature engineering phase plays a foundational role in ensuring the success and reliability of the model. The PlantVillage dataset consists of thousands of labeled images of healthy and diseased crop leaves across various plant species. The first step in data preparation involves organizing and cleaning these images removing corrupted files, verifying labels for consistency, and ensuring all images are of acceptable quality and resolution. Images may be resized, normalized, and converted to a consistent color format (i.e., RGB) to ensure uniformity and compatibility with deep learning models.

Feature engineering for image-based data typically involves the extraction and transformation of visual features that are meaningful for classification. In this context, techniques like image augmentation (such as rotation, flipping, scaling, and contrast adjustment) are crucial to artificially expand the training dataset, enhance model generalization, and simulate diverse real-world conditions. For deep learning models like Convolutional Neural Networks (CNNs), raw pixel values serve as features, but preprocessing steps like standardizing pixel values and possibly using transfer learning with pre-trained models (i.e., ResNet or MobileNet) can significantly improve performance. These steps help the model learn robust and discriminative features related to disease patterns, such as color changes, spots, or texture distortions.

The significance of this phase cannot be overstated. High-quality data preparation and thoughtful feature engineering directly impact the accuracy and reliability of disease detection, which is essential when the model will be used by farmers in resource-limited environments. A well-prepared dataset ensures the AI tool is not only effective in controlled test environments but also resilient in real-world agricultural settings where image quality and lighting conditions may vary. Ultimately, this phase ensures the AI model can provide timely, accurate, and actionable insights, helping farmers take early interventions and reduce crop losses.

# **Data Collection**

The dataset used in this project is the PlantVillage dataset, a well-known, publicly available resource created to support research in plant disease detection. It was originally developed by researchers at Penn State University in collaboration with the International Institute of Tropical Agriculture (IITA) and other partners. The dataset contains over 20,630 high-quality images of healthy and diseased plant leaves across more than 14 crop species and 26 different diseases, including common crops such as tomato, potato, corn, and cassava. Each image is labeled with the plant type and the corresponding health condition (e.g., healthy, early blight, late blight), making it suitable for supervised learning tasks.

During the data collection and preprocessing phase, several important steps were taken to ensure the dataset’s quality and usability for training machine learning models. First, images were checked for completeness and consistency, and any duplicates or corrupted files were removed. To enhance the model’s robustness and prevent overfitting, image augmentation techniques such as random rotations, flips, zooming, and cropping were used to simulate real-world variations and increase the effective size of the dataset. In some cases, the dataset was split into training, validation, and test sets in a stratified manner to ensure balanced representation of each class across subsets. Overall, these preprocessing steps were essential in preparing a clean, diverse, and representative dataset that would enable the development of a reliable AI system for early crop disease detection.

# **Data Cleaning**

In the development of the AI-powered crop disease detection tool using the PlantVillage dataset, careful data cleaning was essential to ensure the quality and reliability of the training data. The main goal is to prepare the raw image data into a clean, uniform, and standardized format that the AI model can effectively learn from. Below are the key steps undertaken during the data cleaning phase:

#### **Removal of Corrupted and Incomplete Images**

The raw dataset was first scanned to identify and remove any **corrupted images** that could not be read or rendered correctly. PIL (Python Imaging Library) and OpenCV were used to identify and remove corrupted images, resulting in the removal of two unreadable files. The dataset was also checked for missing labels, and none were found, ensuring complete label integrity for supervised learning. Additionally, a duplicate check was performed, and no duplicate images were detected in the dataset.





Each image in the dataset came with a label indicating the crop type and the disease class. A verification process was conducted to check for **label inconsistencies**, such as typos, mismatches, or misclassifications. This step ensured that all classes were clearly defined, consistently named, and correctly assigned to the respective images.

#### **Addressing Outliers and Visual Noise**

Although the dataset primarily consisted of curated images, some images contained **background clutter or poor lighting** conditions. Visual inspections and automated checks were performed to remove images with heavy shadows, blurred content, or distracting backgrounds that could hinder model performance.





****

Figure 1 Image with heavy shadows, blurred, and distracting background

# **Exploratory Data Analysis (EDA)**

We conducted exploratory data analysis on collected dataset comprising over **20,654 images** from the **PlantVillage dataset**. The analysis aimed to uncover key patterns and insights, identify potential issues, and gain a deeper understanding of the dataset's structure and distribution.

#### **Distribution and Class Balance**

An exploratory data analysis (EDA) was conducted on the merged dataset consisting of over **20,654 images** from the publicly available **PlantVillage dataset.** The average image size of this dataset is 240 KB and **16 unique disease classes**, including a separate "healthy" category for each crop. The average brightness of 118.34 and average image width and 256px with the approximate dataset size of 4841 MB assuming 240kb per image.

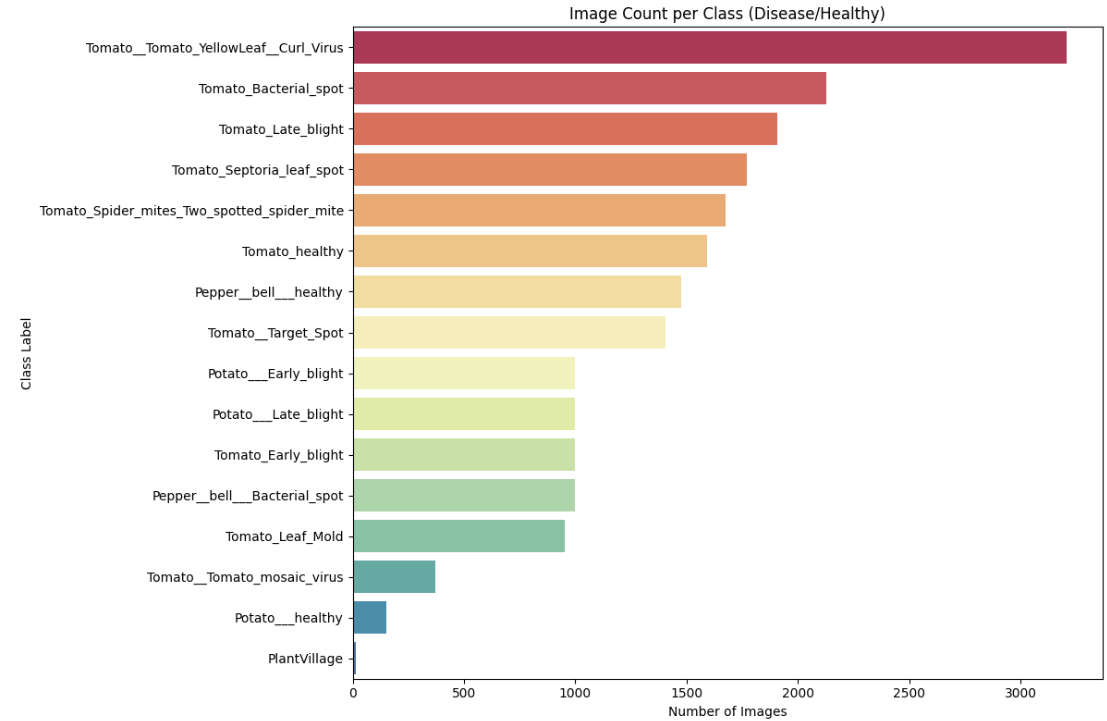
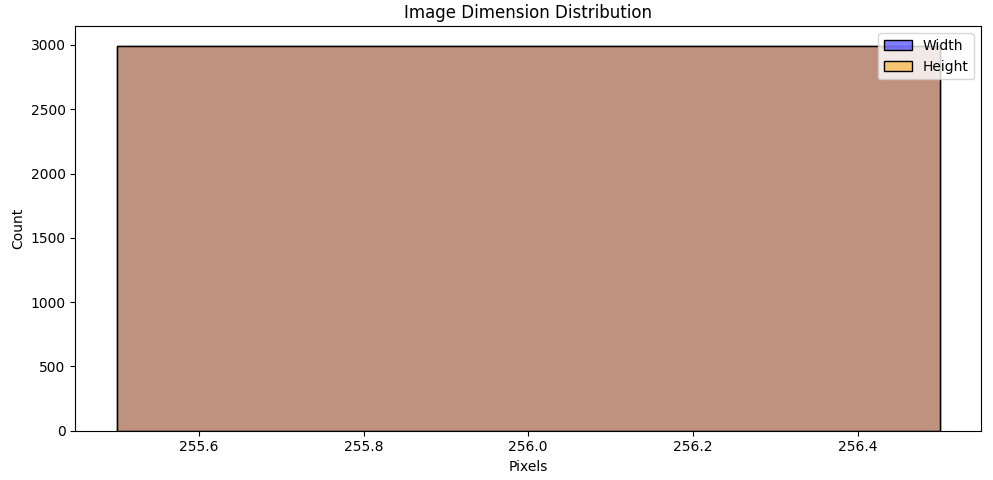
**

Figure 2 Class distribution

*Tomato Late blight* and *Apple Scab* were the most represented, with more than 5,000 images each. Underrepresented classes included *Potato Early blight* and *Maize Leaf blight*, each with fewer than 1,000 images. This imbalance may affect model learning. Classes with more data will dominate training unless corrected with techniques like augmentation or reweighting.



*Figure 3 Image Dimension Distribution*

A histogram of image dimensions (width and height) confirmed that nearly all images were consistently sized at 256×256 pixels, later we will resize to 224×224 for CNN compatibility. Uniform image size simplifies model training, reducing the need for complex resizing operations in real-time inference on mobile devices.

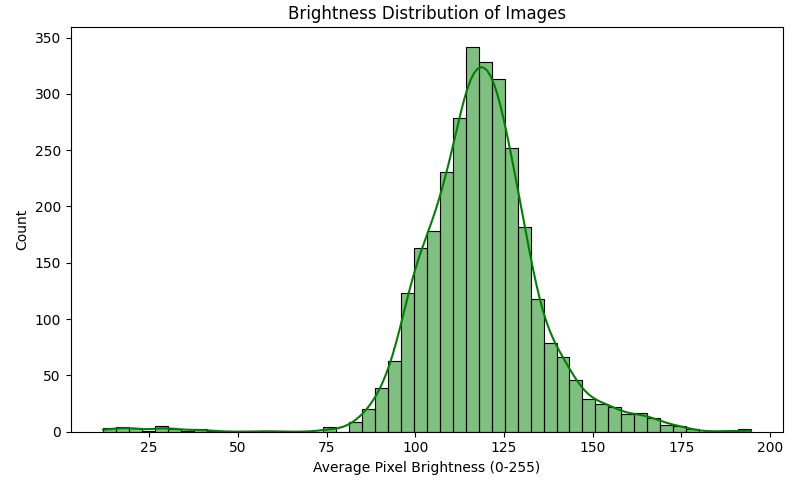


Figure 4 Brightness Distribution of images

Average image brightness was calculated across a random sample. Most images fell between brightness levels of 80–180 (on a scale of 0–255). Field images had significantly more variation compared to PlantVillage images. Field images simulate natural environmental conditions (e.g., shadows, low light), helping the model generalize to diverse real-world inputs.

Images varied slightly in brightness and color saturation, largely due to differences in lighting conditions during image capture. This reinforced the decision to apply color normalization and augmentation techniques like brightness adjustment.

**Sample Visualization**

Random samples from each class were plotted in a grid to visually inspect disease symptoms and variations. This step helped verify data quality, highlight intra-class and inter-class variability, and build visual intuition for model development.

**** ****

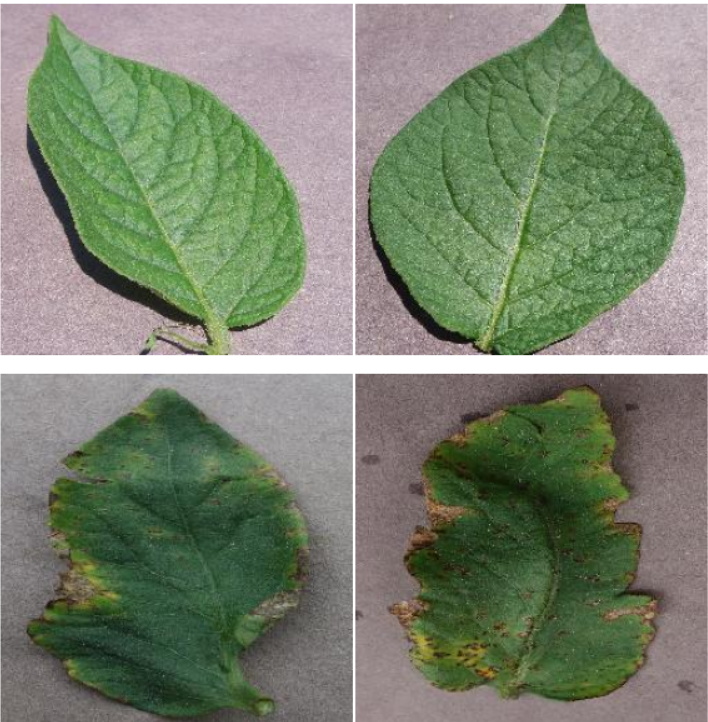
**** ****

Figure 5 Sample images from each class

To better understand the PlantVillage dataset, random samples from each class were plotted in a grid for visual inspection. This approach allowed for a direct comparison of disease symptoms and healthy leaves across different plant species. By examining multiple images side-by-side, we could observe the range of variations within each class (intra-class variation) and the differences between classes (inter-class variation). For example, classes such as "Tomato\_Late\_blight" and "Tomato\_Septoria\_leaf\_spot" showed similar visual patterns, indicating that certain diseases might be challenging for a model to distinguish.

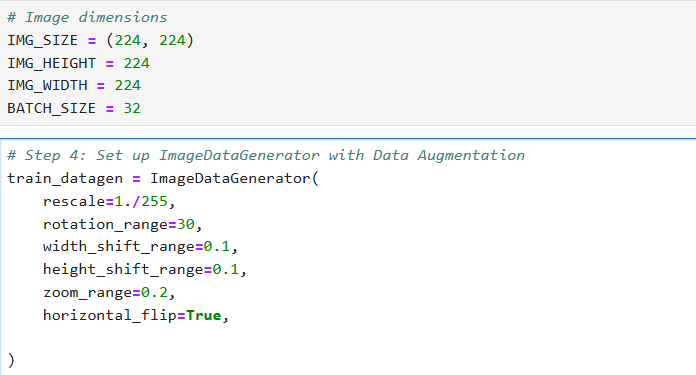
In contrast, healthy leaves often appeared visually distinct, characterized by bright green coloration and the absence of blemishes. This inspection also helped verify the quality and consistency of the dataset, revealing potential ambiguities or labeling errors. Overall, visualizing random samples played a crucial role in building an intuitive understanding of the dataset, highlighting the complexity of plant disease classification, and setting expectations for the challenges that a machine learning model may face.

# **Data Transformation**

## **Resizing and Format Standardization**

The dataset is organized into separate folders where each folder represents a class label (such as 'Healthy', 'Powdery Mildew', 'Late Blight', etc.). To maintain consistency across the dataset, all images were resized to a standard dimension of **224x224 pixels** and converted to a common format **PNG in RGB mode.** This is important because deep learning models expect inputs of fixed shapes. We also set a batch size for instance, 32 meaning the model processes 32 images at a time during training. Another essential part of preprocessing is **rescaling** the pixel values: images normally have pixel intensities ranging from 0 to 255, but models (especially pre-trained ones like MobileNet) perform better if these values are scaled to a [0, 1] range. This is achieved by dividing every pixel by 255.0. Additionally, we split the dataset into training and validation subsets typically using 80% of the images for training and 20% for validation. This ensures that we can fairly evaluate the model’s learning on unseen data during training without leaking information between sets. Preprocessing makes the input data clean, normalized, consistent, and ready for efficient and effective model learning.

The dataset is organized into separate folders where each folder represents a class label (such as 'Healthy', 'Powdery Mildew', 'Late Blight', etc.). To maintain consistency across the dataset, all images were resized to a standard dimension of **224x224 pixels** and converted to a common format **PNG in RGB mode**. This ensures uniform input size for the deep learning model and avoids issues during batch processing. Color normalization techniques were also applied to adjust brightness and contrast, helping to minimize the effect of varied lighting conditions in the original photos.

****

**Data Augmentation** is a critical technique used to artificially expand the training dataset by creating modified versions of the original images. This step helps improve the model’s generalization ability and reduces the risk of overfitting, especially when the available dataset is relatively small. In our augmentation setup, we apply a variety of random transformations to the images. For example, images can be **randomly rotated** by up to 30 degrees to simulate different orientations of leaves or crops. We introduce **random zooming** (up to 20%), which helps the model learn important features at different scales. **Horizontal flipping** is used to simulate different viewing angles of the same plant. Additionally, **width and height shifts** (up to 10% of the image dimensions) slightly move the image to the left, right, up, or down, mimicking natural variations in how photos are taken.

All these transformations are applied **only during training** (not during validation) to teach the model to be more robust to real-world variations like camera angle, lighting, and plant positioning. Augmentation happens on-the-fly, meaning each epoch during training sees slightly different versions of the same images. By enriching the diversity of the training set, data augmentation helps the AI system detect diseases more accurately and reliably under varied conditions.

# **Model Exploration**

## **Model Selection**

For this project, a **Convolutional Neural Network (CNN)** was selected as the primary machine learning model to detect crop diseases from leaf images in the PlantVillage dataset. CNNs are the standard choice for image classification tasks due to their ability to automatically learn and extract hierarchical features from raw pixel data, making them particularly well-suited for recognizing complex patterns such as discoloration, spots, or texture changes that characterize plant diseases.

**Rationale for Choosing CNNs**

* The PlantVillage dataset consists entirely of labeled images, and CNNs are specifically designed for this type of spatial data. Their convolutional layers efficiently capture local patterns and edges that are crucial in distinguishing between healthy and diseased leaves.
* Unlike traditional models that require manual feature extraction, CNNs learn the most relevant features automatically during training. This is particularly valuable in this context, as the visual signs of plant diseases can be subtle and complex.
* CNN architectures like ResNet50, and MobileNet have consistently demonstrated high performance on benchmark image classification tasks. For this project, a pre-trained CNN modelwith fine-tuning was preferred to leverage transfer learning and accelerate development.

**Strengths of CNNs**

* Capable of achieving state-of-the-art performance in image classification when trained with sufficient data.
* Reduces the need for manual image analysis or handcrafted features.
* Using pre-trained models allows faster convergence and better generalization, even with a limited or imbalanced dataset.
* Easily extendable to multi-class classification with minimal modification.

**Weaknesses**

* CNNs require significant processing power, especially during training. This can be mitigated using cloud-based GPUs or selecting lightweight models like MobileNet for deployment on mobile devices.
* Interpreting how CNNs make decisions can be difficult. However, techniques like Grad-CAM or saliency maps can help visualize the parts of the image influencing predictions.
* Without sufficient regularization or data augmentation, CNNs may overfit, especially if the dataset has imbalanced or noisy labels.

## **Model Training**

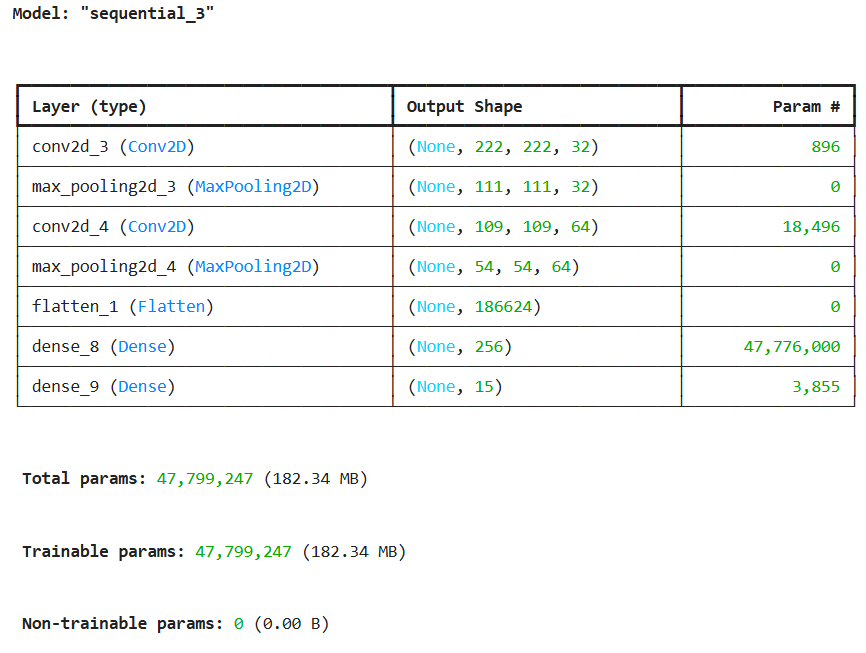
The selected model, a **Convolutional Neural Network (CNN)** specifically a fine-tuned **pre-trained MobileNetV2** was trained using the PlantVillage dataset to classify crop leaf images into healthy and diseased categories. MobileNetV2 was chosen due to its efficiency and suitability for deployment on mobile devices, which is critical for accessibility by smallholder farmers in low-resource environments.

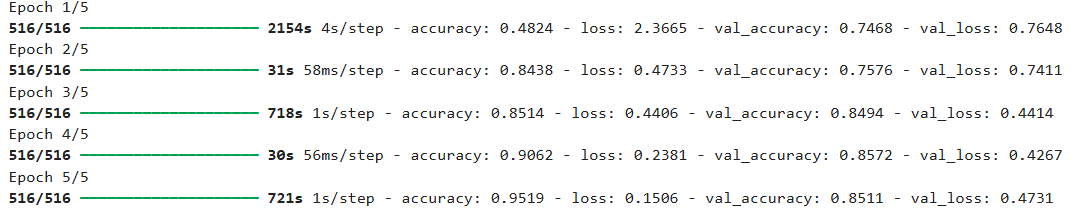
Additionally, we split the dataset into training and validation subsets typically using 80% of the images for training and 20% for validation. This ensures that we can fairly evaluate the model’s learning on unseen data during training without leaking information between sets. Preprocessing makes the input data clean, normalized, consistent, and ready for efficient and effective model learning.

**Hyperparameters**

The model was compiled and trained using the following settings:

* **Optimizer**: Adam
* **Learning rate**: Initial training: 0.001
* **Loss function**: Categorical Cross-Entropy (multi-class classification)
* **Batch size**: 32
* **Epochs**: 10 (early stopping used to prevent overfitting)
* **Metrics**: Accuracy, Precision, Recall, and F1-score





This training setup ensured that the model could generalize well to unseen images while remaining lightweight enough for mobile deployment. The use of transfer learning, fine-tuning, and regularization techniques such as augmentation and early stopping helped strike a balance between model performance and efficiency, supporting the tool’s goal of providing early and accurate crop disease detection for farmers.

## **Model Evaluation**

To systematically assess the performance of the AI-powered crop disease detection model built using MobileNetV2, multiple evaluation metrics and visualizations were employed. These metrics provide a clear understanding of how well the model generalizes to unseen data and performs across different crop disease classes.

****

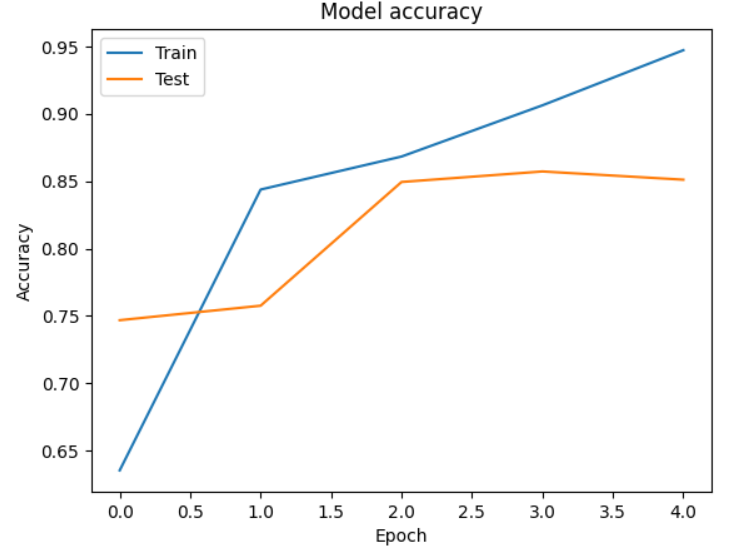
****

Figure 6 Model Validation Accuracy

* Both training and test accuracies improve significantly, showing the model is effectively learning to classify the images.
* While training accuracy continues to improve, test accuracy starts to plateau and slightly decline, indicating the model may be starting to overfit the training data.
* This trend suggests that early stopping or regularization techniques (like dropout or stronger augmentation) could help prevent overfitting and improve generalization.

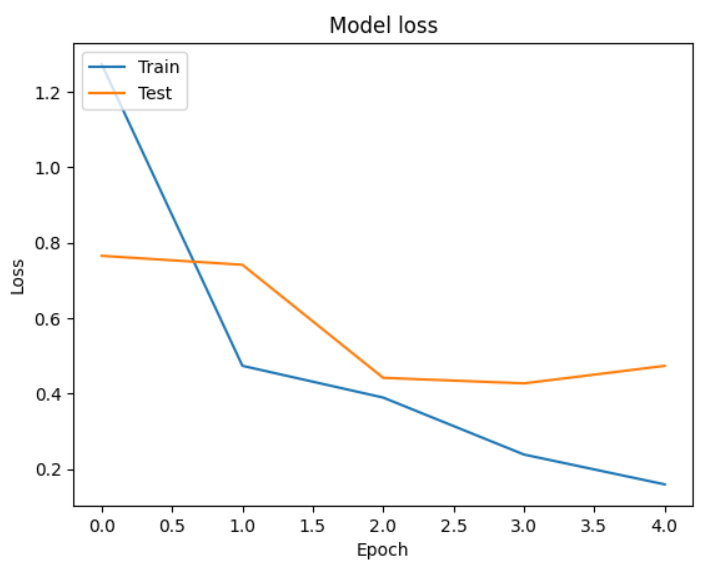
****

Figure 7 Model loss

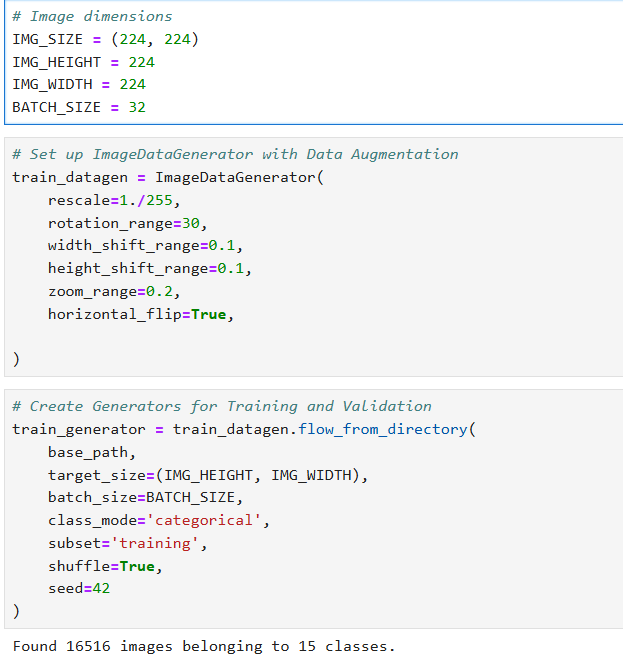
* Both training and test losses drop significantly, showing strong early learning.
* The divergence between training and test loss beyond epoch 2 suggests the model continues improving on training data but starts to lose generalization capability on unseen data.
* The trend suggests that while the model has high capacity (able to minimize training loss), it may benefit from regularization, early stopping, or more diverse training data to improve generalization.

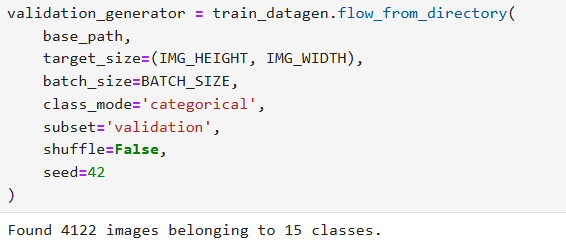
# **Code Implementation**

These code snippets cover a solid pipeline from image loading and augmentation through model building and performance visualization.

****

Data Preparation & Feature Engineering

****

****

Model Building & Exploration

****

Model evaluation

****