Project Documentation: Plant Disease Detection (4-Class Model)

---

1. Project Overview and Goal

The primary goal of this project was to develop a high-accuracy deep learning model capable of classifying the health status of two key crops: **Pepper Bell** and **Tomato**. The solution leverages the power of **Transfer Learning** to create an effective and fast model for agricultural diagnosis.

Metric	Result	Note
Final Test Accuracy	**99%**	Achieved during the initial training phase (frozen base).
Model Type	Multi-Class Image Classification (4 classes).	
Base Architecture	MobileNetV2 (Pre-trained on ImageNet).	
Deployment Target	Web/API (Flask/FastAPI).	

2. Dataset and Preprocessing

The model was trained on a specific subset of the public **PlantVillage** dataset, focusing only on two conditions (Healthy and Bacterial Spot) across two plants.

2.1 Final Class Labels (4)

The model's output is one of the following four distinct classes, read directly from the dataset folder names:

1. Pepper_bell___Bacterial_spot
2. Pepper_bell___healthy
3. Tomato___Bacterial_spot
4. Tomato___healthy

2.2 Data Splitting Strategy

To ensure a rigorous and unbiased evaluation, the data was organized using a **stratified split** to maintain the class distribution across all subsets:

Split Set	Ratio	Purpose
Training (Train)	80%	Used for training the model's weights.
Validation (Valid)	10%	Used *during* training for monitoring loss and preventing overfitting.
Testing (Test)	10%	Used *once* for the final, unbiased accuracy assessment.

2.3 Data Augmentation

To enhance generalization and prevent overfitting, the training images were dynamically augmented using ImageDataGenerator with transformations including rotation, horizontal flips, and shifts. Validation and test data were only rescaled.

3. Model Architecture (Transfer Learning)

The model is built on the **Transfer Learning** paradigm, using the **MobileNetV2** Convolutional Neural Network (CNN) as a feature extractor.

3.1 MobileNetV2 Base (Feature Extractor)

- **Model:** MobileNetV2
- **Input Size:** $224 \times 224 \times 3$ (RGB)
- **Weights:** imagenet (pre-trained weights).
- **Trainability: Frozen (trainable = False)**. This prevented the ImageNet-learned features from being destroyed and led to the high $\sim99\%$ accuracy.

3.2 Custom Classification Head

The custom layers added on top of the frozen base model adapt the output to the 4-class problem:

Layer Type	Parameters/Notes
GlobalAveragePooling2D	Reduces feature maps to a single vector.
Dropout(0.5)	Randomly ignores 50% of neurons during training (Regularization).
Dense(512, activation='relu')	Fully connected layer for pattern recognition.
Dense(4, activation='softmax')	**Output Layer:** 4 neurons (one per class) with Softmax activation for probability distribution.

4. Training Parameters

Parameter	Value
Optimizer	Adam (learning_rate=0.001)
Loss Function	categorical_crossentropy (Standard for multi-class)
Metrics	accuracy
Batch Size	32
Epochs	10 (Early stopped due to high validation accuracy)
Callbacks	EarlyStopping (patience=5), ModelCheckpoint

5. Evaluation and Results

The high accuracy of the frozen-base model on the validation set indicated that **Fine-Tuning was unnecessary and was thus skipped**.

- **Final Test Accuracy:** 99%
- **Confirmation:** The model's low validation loss and high test accuracy confirm that the pre-trained features (MobileNetV2) were sufficient for this task, resulting in an efficient, non-overfit solution.
