■ Plant Disease Detection - AI/ML Training Process

1. Dataset Preparation (PlantVillage)

You used the PlantVillage dataset containing images of healthy and diseased plant leaves. The dataset was split into training, validation, and test sets. Sometimes data augmentation (rotations, flips, zooms) is applied to increase variety and robustness.

2. Data Preprocessing

Images were resized (e.g., 160×160 or 224×224) and normalized (pixel values scaled to [0,1]). This ensures consistency and stability during training.

3. Model Architecture (CNN – Convolutional Neural Networks)

You used Convolutional Neural Networks (CNNs), which are highly effective for image recognition. They automatically learn spatial patterns such as leaf texture, shape, and color.

4. Transfer Learning (Optional but Common)

Pre-trained models like MobileNetV2 or ResNet50 can be fine-tuned on the PlantVillage dataset. This saves training time, requires less data, and improves accuracy.

5. Training (Forward + Backpropagation)

The model predicts outputs and compares them to true labels using a loss function (e.g., categorical crossentropy). The optimizer (e.g., Adam) updates weights by backpropagation.

6. Evaluation & Metrics

Performance was evaluated on validation/test sets using metrics such as Accuracy, Precision, Recall, F1-score, and Confusion Matrix.

7. Model Export & Deployment

The trained model was saved in .keras or .h5 format. It is now deployed in your Flask app for real-world predictions.

■ In Summary: Your project followed a Supervised Learning pipeline with Deep Learning (CNNs): Dataset Preparation → Preprocessing → CNN/Transfer Learning → Training → Evaluation → Model Export & Deployment.