Plant Disease Detection Using Deep Learning and Convolutional Neural Networks

Presented By:
YASWANTH(B23227)
KRISHNA(B23128)
RISHI(B23156)

Problem Statement

Plant diseases reduce crop yield and affect food security
Farmers usually detect diseases manually, but:

- Manual inspection is slow
- Requires expert knowledge
- Not scalable for large farms

GOAL :

Build a deep learning model to automatically classify plant leaf images as healthy or diseased, and to identify the disease type.

We have taken Dataset from Kaggle , which contains leaf images from different crop species and many diseases.

Data Preprocessing

Dataset Overview

- Total images: ~54,000
- Number of classes: 38
- Dataset was highly imbalanced, with some classes having many more samples than others

Image Quality Cleaning :

- Blurry images removed using Laplacian variance thresholding
- Dark / underexposed images removed based on average brightness
- Duplicate images removed using perceptual hashing(pHash)

Data Preprocessing

Class-wise Organization & Balancing :

- Cleaned images grouped into their respective 38 plant disease categories
- Class imbalance checked and verified

Dataset Splitting

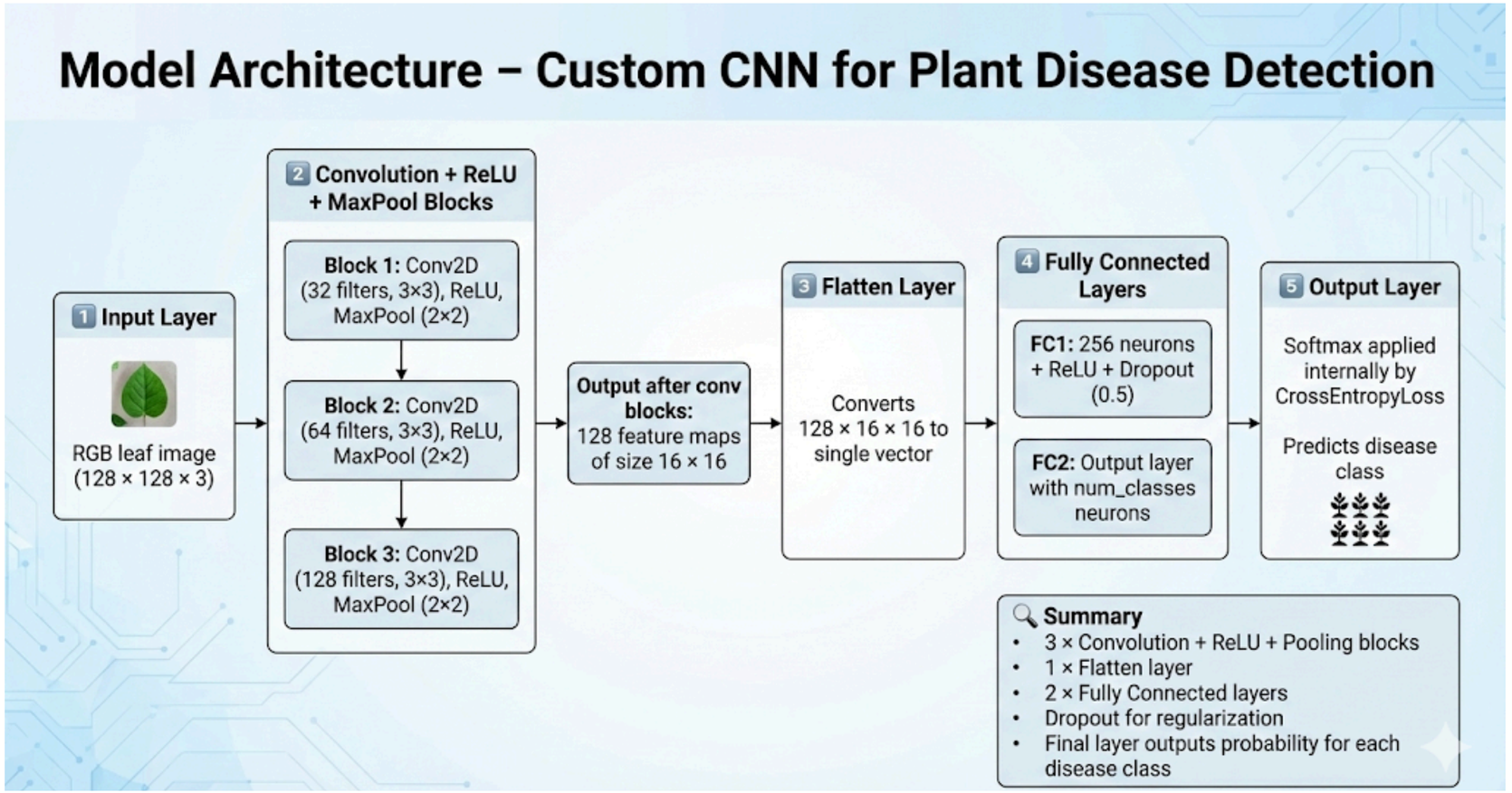
Final cleaned dataset divided into :

- 70% Train
- 20% Test
- 10% Validation

Image Transformations (for CNN Input)

Resize all images to 128 × 128

Model Architecture



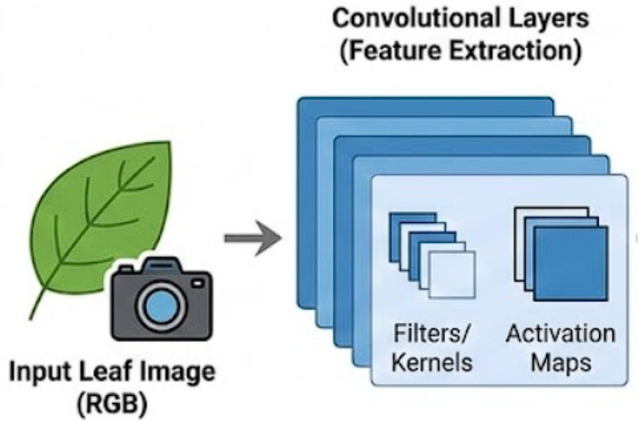
Model Architecture

CNN Layers

Conv1: Conv2D (32 filters, 3×3) → ReLU → MaxPool(2×2)

Conv2: Conv2D (64 filters, 3×3) → ReLU → MaxPool(2×2)

Conv3: Conv2D (128 filters, 3×3) → ReLU → MaxPool(2×2)



➡️ **Output after Conv3: 128 feature maps of size 16 × 16**

Conv1: Picks up **simple textures** on the leaf surface.

Conv2: Detects **spots, lines, and combinations** of shapes.

Conv3: Learns **complex disease patterns** (e.g., specific lesion shapes)

Flattening for FC layer

- FC layers take **1D vector per image**
- Flatten 128 × 16 × 16 → **32768 features**
- Then fc1 reduces it to **256 neurons**, which capture **high-level patterns**

Model Architecture

Fully Connected Layers

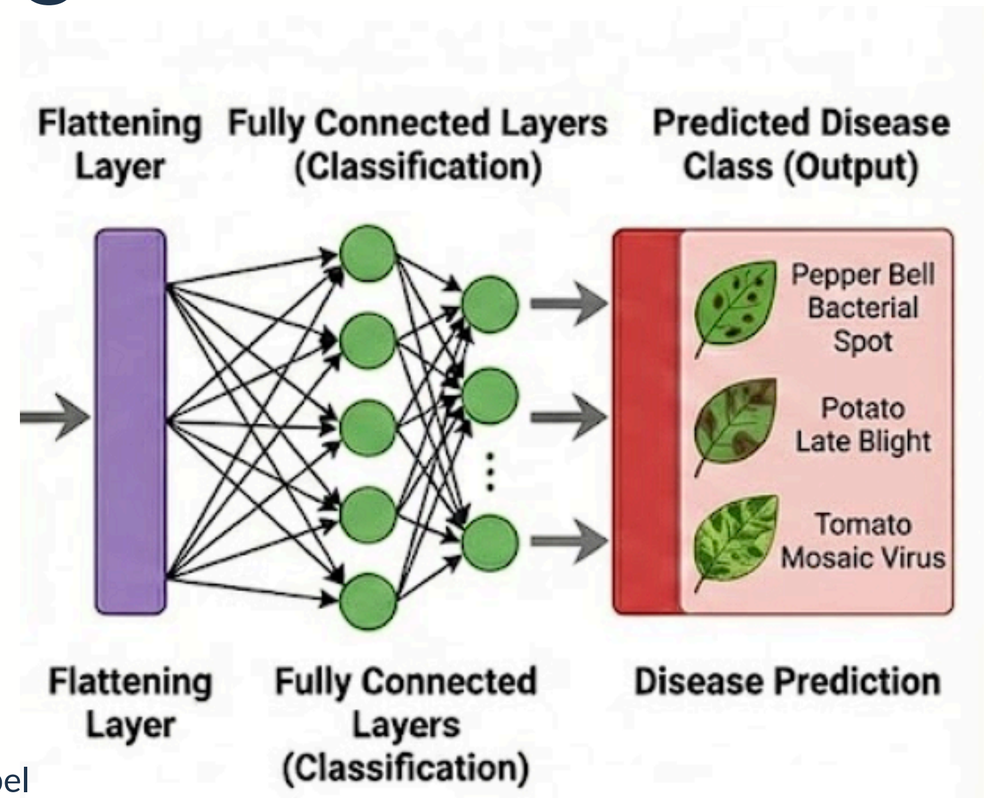
FC1: 256 neurons + ReLU + Dropout (0.5)

FC2: Output layer with *num_classes* neurons

Output Layer

Softmax is applied internally by **CrossEntropyLoss**.

The network predicts the **disease class**.



LOSS FUNCTION :

Categorical Cross-Entropy Loss :

It measures how far the predicted class probabilities are from the true label

$$L = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$

Model Architecture

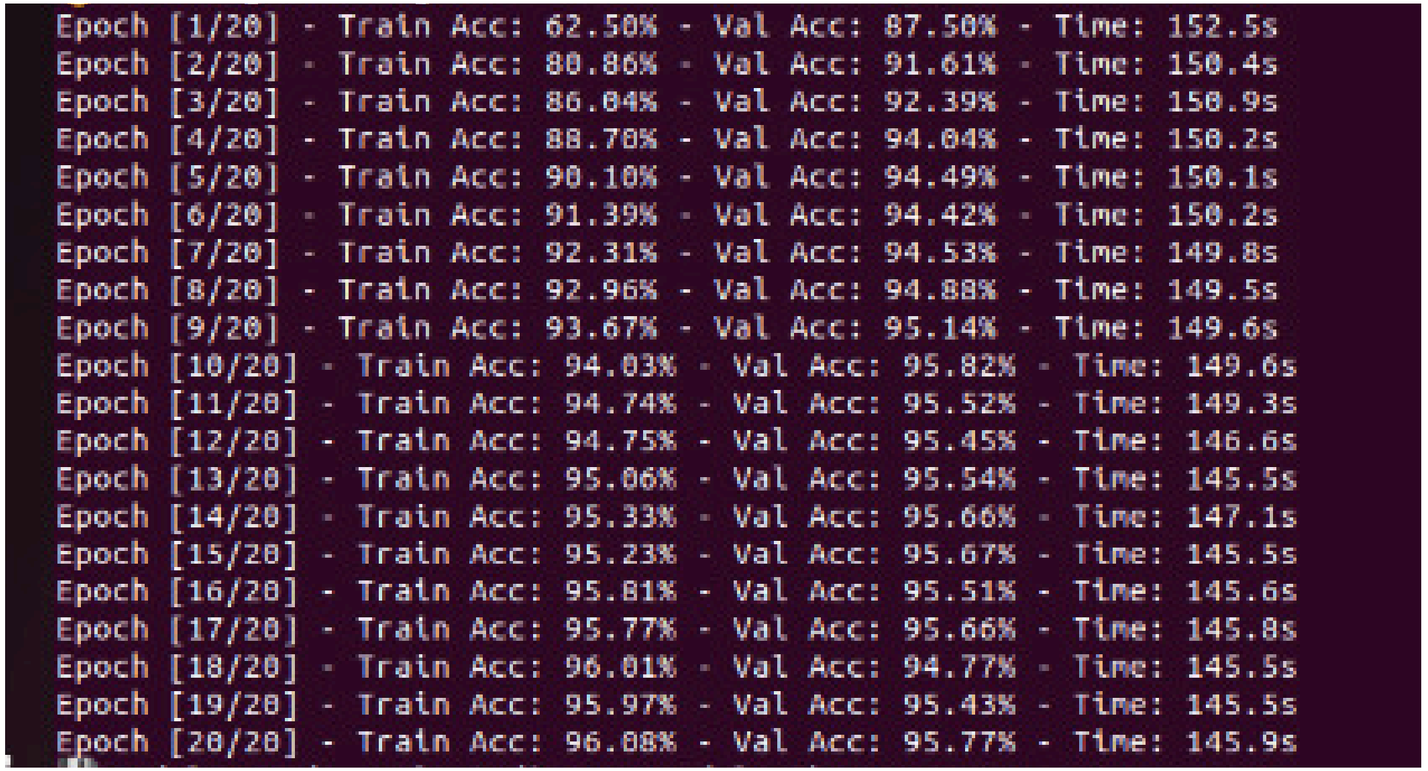
Parameter Update Mechanism

- Forward pass computes predictions
- Loss is calculated
- Backpropagation computes gradients
- **Adam optimizer updates weights** using:

$$W = W - \eta \cdot \text{gradient}$$

where **η = learning rate**

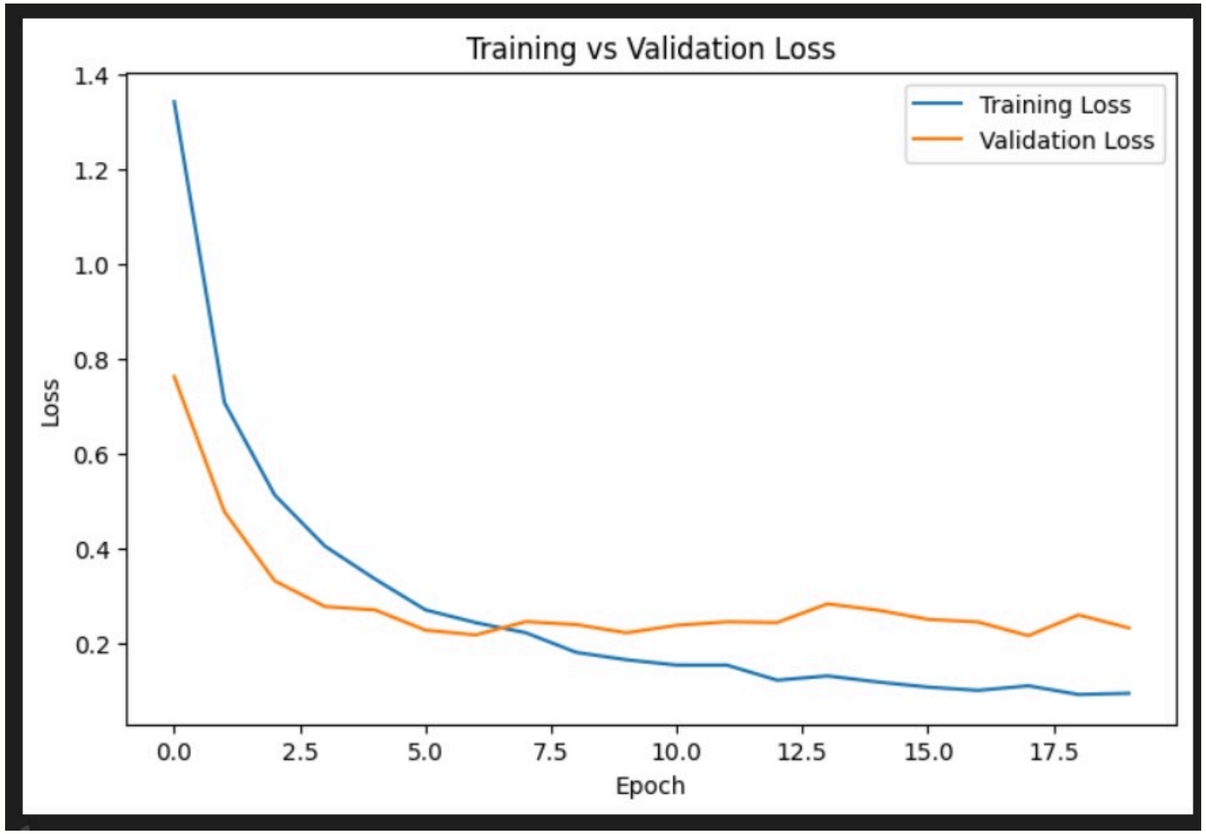
Results :



Train accuracy : 96.08 %

Validation accuracy :95.77 %

Results


Training vs Validation Loss
Training Loss
Validation Loss
Loss
Epoch
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

Model Performance on Test Data

✅ Final Test Accuracy: 93.93%

📊 Class-wise Accuracy:

Pepper__bell__Bacterial_spot — 92.96% (185/199)
Pepper__bell__healthy — 99.32% (293/295)
Plum_Bacterial_Blight_Augmented — 87.95% (73/83)
Plum_Both_Augmented — 94.32% (83/88)
Plum_Healthy_Augmented — 95.92% (94/98)
Plum_Rust_Fungus_Augmented — 96.53% (195/202)
Potato___Early_blight — 96.50% (193/200)
Potato___healthy — 96.50% (193/200)
Potato___Late_blight — 56.67% (17/30)
Tomato__Target_Spot — 93.65% (398/425)
Tomato__Tomato_mosaic_virus — 80.50% (161/200)
Tomato__Tomato_YellowLeaf__Curl_Virus — 86.61% (330/381)
Tomato_Bacterial_spot — 93.16% (177/190)
Tomato_Early_blight — 91.81% (325/354)
Tomato_healthy — 96.72% (324/335)
Tomato_Late_blight — 87.86% (246/280)
Tomato_Leaf_Mold — 99.22% (636/641)
Tomato_Septoria_leaf_spot — 98.65% (73/74)
Tomato_Spider_mites_Two_spotted_spider_mite — 100.00% (318/318)

Test accuracy : 93.93 %

Model Performance on Test Data

```
File: Untitled2.jpeg
Prediction: Pepper,_bell___healthy
Confidence: 99.87%
------------------------------
File: 4c3cc96f-e1d9-4237-b72d-2a1ea0a755f8___RS_HL 2167.JPG
Prediction: Strawberry___healthy
Confidence: 99.98%
------------------------------
File: indoor-plant-textures-details.jpg
Prediction: Tomato___Late_blight
Confidence: 99.91%
------------------------------
File: RS_Rust 2198.JPG
Prediction: Corn_(maize)___Common_rust_
Confidence: 100.00%
------------------------------
File: Untitled1.jpeg
Prediction: Strawberry___Leaf_scorch
Confidence: 100.00%
------------------------------
File: Untitled.jpeg
Prediction: Blueberry___healthy
Confidence: 96.60%
------------------------------
File: 1c94b8af-21b7-43b5-b6ff-a5f7a3ad0a6e___Matt.S_CG 1727.JPG
Prediction: Tomato___Septoria_leaf_spot
Confidence: 99.99%
------------------------------
File: growth-close-up-environmental-lush-natural.jpg
Prediction: ⚠Not in Dataset (Unknown Image)
Confidence: 87.77% (Too Low)
------------------------------
File: 1b250776-ccb2-49b9-9964-a729567f60b9___RS_HL 2205_new30degFlipLR.JPG
Prediction: ⚠Not in Dataset (Unknown Image)
Confidence: 51.95% (Too Low)
------------------------------
File: 0a5e9323-dbad-432d-ac58-d291718345d9___FREC_Scab 3417.JPG
Prediction: Orange___Haunglongbing_(Citrus_greening)
Confidence: 99.87%
------------------------------
Done.
```

Testing with ai generated leaf

Result : Unknown image

Testing with Real leaf from college :

Result :
LEAF WITH
NO
DISEASE

MEMEBER	CONTRIBUTION
Rishi	Managed dataset collection and preprocessing, including removing blurry, dark, and duplicate images.
Yashwanth	Designed and implemented the CNN model, trained and optimized it for best performance.
Krishna	Evaluated model behavior (checked for overfitting), analyzed training/validation accuracy trends, and determined suitable number of epochs.

Thank You