# Crop Disease Detection Using Deep Learning

Kuppusamy P
*School of Computer Science and Engineering*
VIT-AP University
Amaravathi, India
drpkscse@gmail.com

Maturi Nikhita
*School of Computer Science and Engineering*
VIT-AP University
Amaravathi, India
nikhita.22bce7925@vitapstudent.ac.in

Moumita Mandal
*School of Computer Science and Engineering*
VIT-AP University
Amaravathi, India
moumita.22bce7920@vitapstudent.ac.in

Abi Karimireddy
*School of Computer Science and Engineering*
VIT-AP University
Amaravathi, India
abi.22bce7140@vitapstudent.ac.in

***Abstract*—Crop diseases pose a major risk to food security worldwide, leading to considerable declines in agricultural output. Timely identification and diagnosis of these diseases are essential for their effective control and management. Traditional approaches to identifying diseases often take a lot of time and necessitate specialized expertise. However, with the progress in deep learning, it is now possible to create automated systems that can accurately detect and categorize crop diseases. The creation and assessment of a deep learning model are presented in this study for crop disease detection using the PlantVillage dataset. Two convolutional neural network models are utilized to categorize images of crops, distinguishing between those that are diseased and those that are healthy, into various groups.**

## I. Introduction

Crop diseases pose a serious threat to agricultural productivity and global food security, resulting in significant economic losses. Traditionally, identifying these diseases has depended on expert knowledge, which can often be time-consuming and hard to access in many areas. This highlights the urgent need for automated and efficient solutions.

Recently, deep learning has made impressive strides, especially with convolutional neural networks (CNNs), which have proven to be effective in detecting diseases through images. CNNs are particularly good at picking up complex patterns from data, allowing for accurate differentiation between healthy and diseased crops. This study delves into the potential of CNNs for detecting crop diseases, utilizing the PlantVillage dataset.

## II. Literature Review

Recent studies have explored various deep learning architectures for plant disease detection:

1) *Construction of Deep Learning-Based Disease Detection Model in Plants (2023): This study developed a CNN-based model to detect diseases in plant leaves, achieving high accuracy and demonstrating the potential of deep learning in agricultural applications.*
2) *Using Deep Learning for Image-Based Plant Disease Detection (2016): This research utilized deep learning techniques to identify plant diseases from images, highlighting the effectiveness of CNNs in plant pathology.*
3) *Classification of Various Plant Leaf Diseases Using CNN (2024): This study applied CNNs to classify different plant leaf diseases, demonstrating the model's capability in handling complex image data for accurate disease identification.*
4) *Leveraging Edge Artificial Intelligence for Sustainable Agriculture (2024): This research discussed the integration of edge AI technologies in agriculture, emphasizing the role of AI in enhancing crop disease detection and management.*
5) *From Detection to Protection: The Contribution of Optical Sensors, Robotics, and Artificial Intelligence in Contemporary Plant Disease Management (2024): This comprehensive review highlighted the partnership between AI and optical sensors in enhancing approaches for handling plant diseases.*
6) *Research Priorities to Leverage Smart Digital Technologies for Sustainable Crop Production (2024): This study identified key research areas where digital technologies, including AI, can contribute to sustainable crop production and disease management.*
7) *Assessing Urban Forest Biodiversity Through Automatic Taxonomic Identification of Street Trees from Citizen Science Applications and Remote-Sensing Imagery (2024): This research utilized AI-based identification to assess urban forest biodiversity, showcasing the application of AI in plant species identification.*
8) *Evaluating Image-Based Species Recognition Models Suitable for Citizen Science Application to Support European Invasive Alien Species Policy (2022): This study evaluated various AI models for species recognition, providing insights into the effectiveness of different*

*architectures in plant identification tasks.*

9) *A Comprehensive Review of Data Mining Techniques in Smart Agriculture (2019): This review discussed the application of data mining and AI techniques in smart agriculture, including plant disease detection.*
10) *Defect Detection in Fruit and Vegetables by Using Machine Vision Systems and Image Processing (2022): This research applied machine vision and image processing techniques for defect detection in produce, relevant to disease detection methodologies.*

## III. Proposed Work

This project develops and compares two deep learning models for crop disease detection:

### A. Model 1: User-Defined CNN Model

A custom-designed CNN optimized for the characteristics of the leaf disease dataset:

- Comprises 6 convolutional layers, each followed by MaxPooling layers for feature extraction.
- Ends with a dense layer and fully connected layer for classifying plant images into respective disease categories.
- Uses ReLU activation function and softmax for output layer activation.
- After the model is built, the models are trained and evaluated on a dataset of diseased and healthy leaf images to assess their accuracy and robustness and classify the plant images with their respective diseases.

### B. Model 2: ResNet50 Model

A deep residual network designed for effective image classification:

- Comprises 50 layers, leveraging residual connections to mitigate the vanishing gradient problem.
- Utilizes GlobalMaxPooling for feature extraction.
- Includes dense and fully connected layers for final classification.
- Employs ReLU activation and softmax at the output layer.
- After the model is built, the models are trained and evaluated on a dataset of diseased and healthy leaf images to assess their accuracy and robustness and classify the plant images with their respective diseases.

### C. Training Details

- Both models are trained using categorical cross-entropy loss.
- Adam optimizer is used to minimize the loss and optimize trainable parameters.
- Performance is evaluated over multiple epochs to ensure accuracy and efficiency.
- After the model is built, the models are trained and evaluated on a dataset of diseased and healthy leaf images to assess their accuracy and robustness and classify the plant images with their respective diseases.

## IV. Dataset

The PlantVillage dataset is used for this project. It contains images of various crops (e.g., pepper, potato, tomato) with different diseases and healthy conditions. The dataset is divided into 15 classes, including:

- Pepper: Bacterial spot, Healthy
- Potato: Early blight, Late blight, Healthy
- Tomato: Bacterial spot, Early blight, Late blight, Leaf Mold, Septoria leaf spot, Spider mites, Target Spot, Yellow Leaf Curl Virus, Mosaic Virus, Healthy

### A. Dataset Statistics

- Total images: 645
- Image size: 256x256 pixels
- Channels: 3 (RGB)
- Classes: 15

## V. Methodology

### A. Data Preprocessing

- **Image Resizing and Rescaling:** All images are resized to 256x256 pixels and normalized by rescaling pixel values to the range [0, 1].
- **Data Augmentation:** To improve model generalization, data augmentation techniques such as random flipping (horizontal and vertical) and rotation (0.2) are applied.
- **Dataset Splitting:** The dataset is split into training, validation, and test sets:
  - Total samples: 645
  - Training set: 516 samples
  - Validation set: 64 samples
  - Test set: 65 samples

### B. Model Architecture

#### 1) Model-1: User-Defined CNN Model:

- **Input Layer:** Takes images that are 256x256 in size and have 3 channels (RGB color) as input.
- **Convolutional Layers:** Six Conv2D layers with ReLU activation and MaxPooling2D layers for feature extraction with filter sizes of (3,3) and (2,2), respectively.
- **Flatten Layer:** Converts 2D feature maps into a single-dimensional vector.
- **Dense Layers:** Fully connected layers containing 64 units with ReLU activation for classification purposes.
- **Output Layer:** A softmax layer with 15 units (one for each class).
  **Model Summary:**
  - Total parameters: 184,527
  - Trainable parameters: 184,527

#### 2) Model-2: ResNet50 Model:

- **Input Layer:** Receives images that are 256x256 in size and have 3 channels (RGB color).
- **Convolutional Layers:** ResNet50 layer with ReLU activation and GlobalAveragePooling2D layers for feature extraction.

- **Dense Layers:** Fully connected layers containing 1024 units with ReLU activation for classification purposes.
- **Output Layer:** A softmax layer with 15 units (one for each class).
  **Model Summary:**
  - Total parameters: 25,701,263
  - Trainable parameters: 25,648,143
  - Non-Trainable parameters: 53,120

### C. Architectures

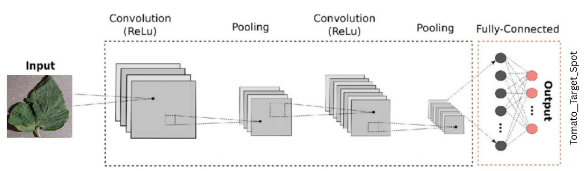


Fig. 1. CNN Architecture

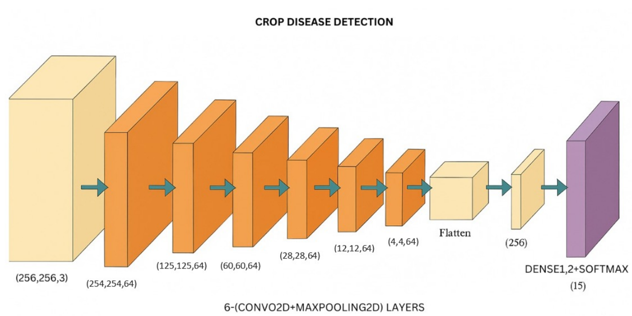


Fig. 2. CNN Representation

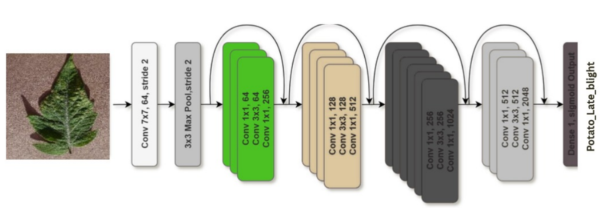


Fig. 3. ResNet50 Architecture

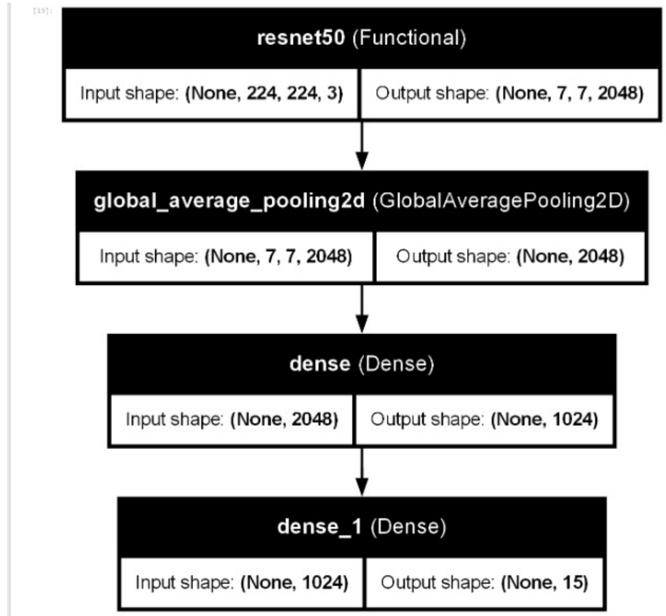


Fig. 4. ResNet50 FlowChart

### D. Equations

*1) Preprocessing:* The image data is preprocessed through resizing and normalization. Given an image tensor $x$, the transformation is:

$$x_{\text{rescaled}} = \frac{x}{255} \tag{1}$$

This ensures all pixel values lie in the range $[0, 1]$.

*2) Equations Used in CNN:* Convolutional Neural Networks (CNNs) have convolution and pooling operations:

$$y_{i,j}^{(k)} = f\left(\sum_{m=1}^{M}\sum_{n=1}^{N} x_{i+m,j+n} \cdot w_{m,n}^{(k)} + b^{(k)}\right) \tag{2}$$

Where:

- $x$ is the input image or feature map
- $w$ is the kernel/filter
- $b$ is the bias
- $f$ is an activation function (e.g., ReLU)

MaxPooling operation:

$$y_{i,j} = \max_{(m,n)\in\mathcal{R}} x_{i+m,j+n} \tag{3}$$

*3) Equations Used in ResNet:* ResNet introduces skip (residual) connections. If $x$ is the input to a block and $\mathcal{F}(x)$ is the transformation (typically convolution + BN + ReLU), the output is:

$$y = \mathcal{F}(x) + x \tag{4}$$

This helps in training deeper networks by mitigating vanishing gradient issues.

Let $f_{\text{ResNet50}} : \mathbb{R}^{224\times224\times3} \rightarrow \mathbb{R}^d$ be the pretrained ResNet50 feature extractor (excluding the top layer). The forward pass of the model using transfer learning is defined as:

$$z = \text{Dense}_{1024}\left(\text{GlobalAvgPool}\left(f_{\text{ResNet50}}(I)\right)\right) \tag{5}$$
$$\hat{y} = \text{Softmax}\left(\text{Dense}_{15}(z)\right) \tag{6}$$

Where:

- $I$ is the input image of shape $224 \times 224 \times 3$
- $f_{\text{ResNet50}}$ extracts deep features
- GlobalAvgPool performs global average pooling
- $\text{Dense}_{1024}$ is a fully connected layer with 1024 units
- $\text{Dense}_{15}$ maps to 15 output classes
- Softmax converts the logits to class probabilities

*4) Common Evaluation Metrics:* Let $y_i$ be the true label and $\hat{y}_i$ the predicted label for class $c$.

*a) Accuracy::*

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} = \frac{1}{n}\sum_{i=1}^{n} \nVdash (y_i = \hat{y}_i) \tag{7}$$

*b) Precision (per class)::*

$$\text{Precision}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c} \tag{8}$$

*c) Recall (per class)::*

$$\text{Recall}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c} \tag{9}$$

*d) F1 Score::*

$$F1_c = 2 \cdot \frac{\text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c} \tag{10}$$

*5) Optimizer: Adam:* Adam optimizer combines momentum and RMSProp. The updates are:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \tag{11}$$
$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \tag{12}$$
$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \tag{13}$$
$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{14}$$
$$\theta_{t+1} = \theta_t - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{15}$$

Where:

- $g_t$ is the gradient at time $t$
- $\beta_1, \beta_2$ are exponential decay rates
- $\eta$ is the learning rate

*6) Confusion Matrix:* A confusion matrix $C$ for multi-class classification is defined as:

$$C_{ij} = \text{Number of instances of class } i \text{ predicted as class } j$$

From this, all other metrics (precision, recall, etc.) can be derived.

### *E. Training*

- **Optimizer:** Adam with learning rate = 0.0001
- **Loss Function:** Sparse Categorical Crossentropy
- **Metrics:** Accuracy
- **Epochs:** 30 for Model-1 and 20 for Model-2
- **Batch Size:** 32

## VI. Results

### *A. Training and Validation Performance*

*1) Model-1:*

- **Training Accuracy:** Increased from 27.31% (Epoch 1) to 96.81% (Epoch 30).
- **Validation Accuracy:** Increased from 66.85% (Epoch 1) to 96.88% (Epoch 30).
- **Training Loss:** Decreased from 2.7496 (Epoch 1) to 0.1003 (Epoch 30).
- **Validation Loss:** Decreased from 0.9751 (Epoch 1) to 0.1078 (Epoch 30).

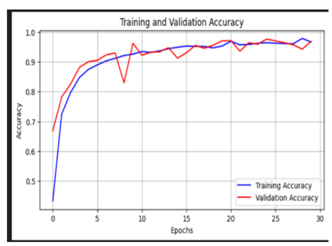


Fig. 5. Accuracy

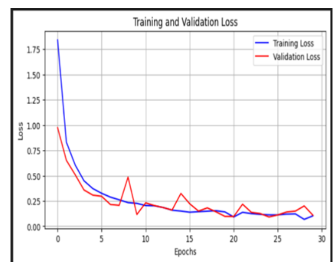


Fig. 6. Loss

*2) Model-2:*

- **Training Accuracy:** Increased from 86.10% (Epoch 1) to 99.77% (Epoch 20).
- **Validation Accuracy:** Increased from 96.04% (Epoch 1) to 98.44% (Epoch 20).
- **Training Loss:** Decreased from 46.09% (Epoch 1) to 0.72% (Epoch 20).
- **Validation Loss:** Decreased from 12.21% (Epoch 1) to 4.06% (Epoch 20).

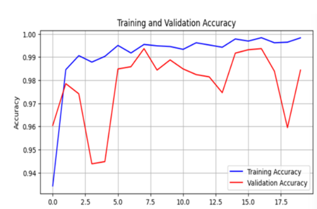


Fig. 7. Accuracy

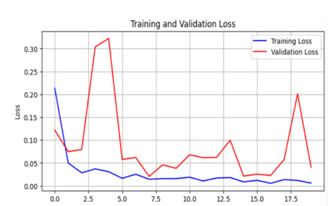


Fig. 8. Loss

## B. Test Performance

*Model-1:*

- **Accuracy:** 96.88%
- **Precision:** 97.00%
- **Recall:** 94.60%
- **F1 Score:** 95.64%
- **Test Loss:** 13.05%
- **Test Accuracy:** 96.15%

*Model-2:*

- **Accuracy:** 98.49%
- **Precision:** 98.14%
- **Recall:** 97.97%
- **F1 Score:** 98.01%
- **Test Loss:** 7.70%
- **Test Accuracy:** 97.54%

## C. Input and Output Examples

### 1) Model-1:

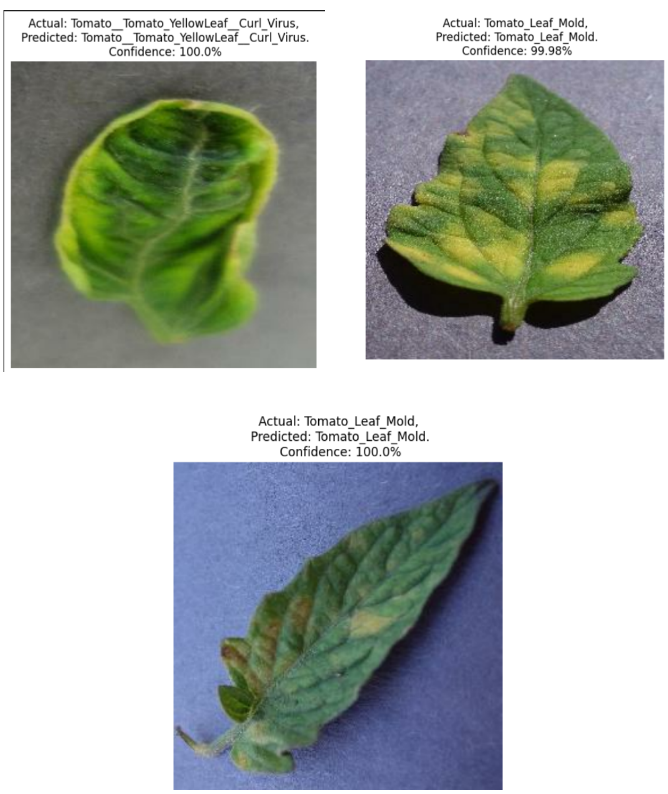


Fig. 9. Visualizations of model-1 performance

### 2) Model-2:

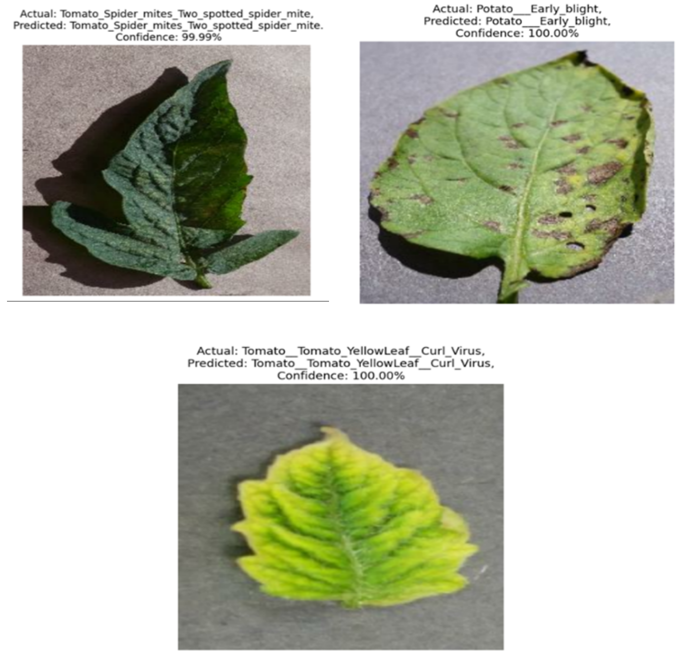


Fig. 10. Visualizations of model-2 performance

## D. Confusion Matrix

The confusion matrix illustrates the model's performance across all 15 classes. Both models perform well for most classes, with minor misclassifications observed, particularly in visually similar classes.

*1) Model-1:* The confusion matrix for Model-1 shows strong performance overall but with occasional misclassifications between similar disease categories, such as *Tomato Bacterial spot* and *Tomato Early blight*.

```
Accuracy: 0.96875
Precision: 0.9700260078130081
Recall: 0.9459838444194495
F1 Score: 0.9563931815631666
Confusion Matrix:
 [[ 83   0   1   1   0   0   1   0   0   4   0   0   0   0   0]
 [  0 160   0   2   0   0   0   0   0   0   0   1   0   0   0]
 [  0   0  94   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0 106   0   0   0   2   0   2   0   0   0   0   0]
 [  0   3   0   0  16   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0 194   2   0   0   1   0   0   1   0   0]
 [  2   0   0   0   0   1  87   4   0   0   1   0   0   0   0]
 [  0   0   0   2   0   0   6 187   0   0   1   0   0   0   0]
 [  0   1   0   0   0   0   0   0  76   3   3   0   0   0   0]
 [  1   1   0   0   0   0   0   1   1 157   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0 170   1   0   0   0]
 [  0   0   0   0   0   0   0   0   0   2   2 151   0   0   1]
 [  0   0   0   0   0   0   0   0   0   0   0   0 313   0   0]
 [  0   0   0   0   0   0   0   0   0   5   3   0   0  29   0]
 [  0   0   0   0   0   0   0   0   0   0   1   0   0   0 161]]
```

*2) Model-2:* The confusion matrix for Model-2 exhibits improved classification accuracy with fewer misclassifications compared to Model-1, showcasing the robustness of the ResNet50 architecture.

```
Accuracy: 0.984973339796413
Precision: 0.9814715102046214
Recall: 0.979711389442886
F1 Score: 0.9801384179993764
Confusion Matrix:
 [[195   0   0   0   0   0   2   0   0   2   0   1   1   0   0]
 [  0 301   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0 187   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   1 174   1   0  11   2   0   0   1   3   0   0   0]
 [  0   0   0   0  29   0   1   0   0   0   0   1   0   0   0]
 [  0   0   0   0   0 444   0   0   0   0   0   0   2   0   0]
 [  0   0   0   0   0   2 205   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   9 385   0   0   0   1   0   0   0]
 [  0   0   0   0   0   0   0   0 195   0   0   0   0   0   0]
 [  0   0   1   0   0   0   2   0   0 344   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   2 324   7   0   1   0]
 [  0   0   0   0   0   0   1   0   0   1   1 270   0   0   0]
 [  0   0   0   0   0   1   1   0   0   0   0   0 648   1   0]
 [  0   0   0   0   0   0   0   0   0   1   0   0   0  68   0]
 [  0   0   0   0   0   0   0   0   0   0   0   1   0   0 295]]
```

## VII. Discussion

- **Model Comparison:** The Model-1 (Sequential CNN) achieves an accuracy of 96.88% on the test set, while Model-2 (ResNet50) achieves an accuracy of 98.49%. This demonstrates the superior performance of the ResNet50 model in detecting and classifying crop diseases.
- **Impact of Data Augmentation:** Data augmentation, which increases dataset diversity by modifying existing samples, along with preprocessing to resize images, significantly improves model generalization and robustness.
- **Misclassifications:** Misclassifications are primarily observed in visually similar classes, such as different tomato diseases. Addressing this issue may involve incorporating more diverse data or leveraging advanced architectures like transfer learning.

## VIII. Conclusion

The deep learning models developed in this project effectively detect and classify crop diseases using the PlantVillage dataset. Due to the large number of layers, ResNet50 required more time to train compared to conventional CNN models but outperformed them in terms of precision and recall. With an accuracy of 98.49%, the ResNet50 model demonstrates its potential for deployment as part of an automated system to assist farmers in early disease detection and management.

## IX. Future Work

- **Transfer Learning:** Explore the use of pre-trained models (such as GoogleNet, VGGNet, and EfficientNet) to enhance performance.
- **Larger Dataset:** Incorporate more diverse and larger datasets to improve model robustness and adaptability.
- **Real-Time Deployment:** Develop a mobile or web application for real-time disease detection using the trained model.
- **Multi-Language Support:** Add support for multiple languages to make the system accessible to farmers worldwide.

## X. References


1) PlantVillage Dataset: https://plantvillage.psu.edu/
2) TensorFlow Documentation: https://www.tensorflow.org/
3) Keras Documentation: https://keras.io/
4) Deep Learning for Agriculture: Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using Deep Learning for Detecting Plant Diseases via Image Analysis. *Frontiers in Plant Science*, 7, 1419. DOI:10.3389/fpls.2016.01419
5) Convolutional Neural Networks (CNNs): LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. *Nature*, 521(7553), 436–444. DOI:10.1038/nature14539
6) Data Augmentation in Deep Learning: Shorten, C., & Khoshgoftaar, T. M. (2019). A Survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1), 60. DOI:10.1186/s40537-019-0197-0
7) Precision Agriculture: Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep Learning in Agriculture: A Survey. *Computers and Electronics in Agriculture*, 147, 70–90. DOI:10.1016/j.compag.2018.02.016
8) Evaluation Metrics for Classification: Sokolova, M., & Lapalme, G. (2009). A Systematic Analysis of Performance Measures for Classification Tasks. *Information Processing & Management*, 45(4), 427–437. DOI:10.1016/j.ipm.2009.03.002
9) Existing Work: https://www.nature.com, https://www.frontiersin.org, https://openagriculturejournal.com, https://de.wikipedia.org, https://en.wikipedia.org