Plant Disease Detection and Classification Using Machine Learning

EE20BTECH11012 - CHENCHETY ABHISHEK

***Abstract*— The early detection of plant diseases remains a pivotal challenge in modern agriculture, impacting crop yields and global food security. This research explores the potential of convolutional neural networks (CNNs), specifically ResNet and AlexNet, in revolutionizing disease detection in plants. ResNet's deep architecture and unique residual connections, along with AlexNet's pioneering impact in CNNs, serve as focal points in this study. Through the analysis of plant images, these CNN models demonstrate their capability to accurately identify and classify various diseases, enabling prompt and targeted intervention. Leveraging extensive datasets, this research showcases the effectiveness of data-driven solutions in training these models to recognize subtle patterns indicative of plant ailments. The findings emphasize the transformative potential of CNNs in agriculture, offering insights into creating resilient and disease-resistant agricultural ecosystems. This research contributes to advancing technological approaches in safeguarding crop health, thereby ensuring sustainable food production and addressing global food security challenges.**

I. Introduction

In the realm of agriculture, early detection of plant diseases stands as a crucial challenge impacting crop health and food security. To address this, machine learning, specifically convolutional neural networks (CNNs) like ResNet and AlexNet, holds immense promise. ResNet's sophisticated architecture with residual connections allows deeper image analysis, aiding precise disease identification in plants. Although AlexNet pioneered CNNs for image recognition, ResNet's innovations surpassed it. Implementing these models in agriculture offers transformative potential, enabling early disease diagnosis and targeted treatments. By leveraging vast datasets, these data-driven solutions empower AI models to continuously improve disease recognition capabilities, fostering a more resilient agricultural system. Ultimately, the integration of ResNet, AlexNet, and similar CNNs represents a technological leap toward safeguarding global food supplies and ensuring healthier crop yields.

This paper presents a comprehensive project proposal aimed at addressing these pressing concerns through three fundamental objectives: early detection, accurate classification, and continuous learning. These objectives stand as the pillars for empowering farmers with the necessary tools to mitigate the adverse effects of plant diseases.

The proposed methodology revolves around a strategic process encompassing data collection from pre-defined datasets, meticulous preprocessing, and feature extraction, culminating in the development of robust classification models. This methodical approach aims to harness the power of machine learning to revolutionize the detection and classification of plant diseases.

The anticipated outcomes of this endeavor are poised to be transformative, heralding a new era in plant disease detection and classification. Ultimately, this initiative seeks to fortify farmers' capabilities, providing them with indispensable support to nurture healthy and thriving crops, thereby fostering sustainability in agriculture.

II. Literature Review

The PlantVillage dataset has become a popular benchmark for evaluating image classification models in the domain of plant disease detection.

Top-performing models:

1) **Adaptive minimal ensembling:** This ensemble approach achieved 100% F1 score on the PlantVillage dataset, demonstrating remarkable accuracy. (Paper: Improving plant disease classification by adaptive minimal ensembling, 2022)
   - **Benefits:** Robust to noise and outliers in the data, avoids overfitting, and can handle complex classification tasks.
   - **Challenges:** Requires careful design and implementation of the selection criteria to ensure diversity and effectiveness of the ensemble.
2) **µ2Net+ (ViT-L/16):** This model using Vision Transformer architecture achieved 99.89% F1 score, highlighting the effectiveness of transformers for plant disease classification. (Paper: A Continual Development Methodology for Large-scale Multitask Dynamic ML Systems, 2022)
   - **Benefits:** Superior in capturing complex spatial relationships and dependencies compared to traditional CNNs, potentially leading to better generalizability to unseen data.
   - **Challenges:** Requires significant computational resources for training and inference, limiting accessibility for some applications.
3) **Light-Chroma Inception V3:** This model achieved 99.48% accuracy by separating light and chroma features, indicating the importance of tailored feature extraction for plant disease detection. (Paper: Reliable Deep Learning Plant Leaf Disease Classification Based on Light-Chroma Separated Branches, 2021)
   - **Benefits:** Improves performance under diverse lighting conditions, a common challenge in real-world plant disease detection.
   - **Challenges:** Relies on the pre-trained Inception V3 architecture, limiting flexibility for customizing the feature extraction process.

4) **Inception V3 20%L + 80%AB:** This two-branch CNN architecture achieved 99.48% accuracy, demonstrating the effectiveness of color-aware feature learning for plant disease classification. (Paper: Color-aware two-branch DCNN for efficient plant disease classification, 2022)
   - **Benefits:** Effective in capturing both subtle texture and color variations, leading to better disease identification.
   - **Challenges:** Requires careful hyperparameter tuning to balance the contributions of both branches and avoid overfitting to specific color patterns.

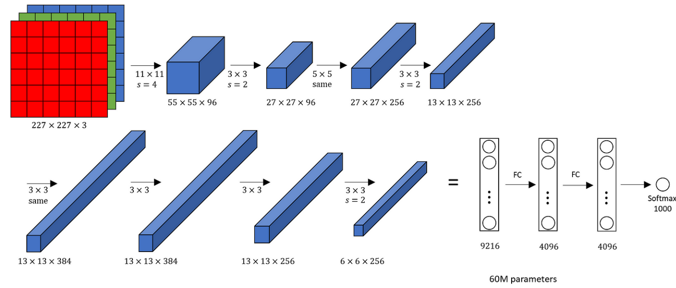

Fig. 1: AlexNet Architecture

III. Materials and Methods

The dataset used is here is a large collection of images of diseased plant leaves. The dataset contains over 54,000 images of 14 different crop species, and it includes images of 17 different fungal diseases, 4 bacterial diseases, 2 mold (oomycete) diseases, 2 viral diseases, and 1 disease caused by a mite.

A. Data Description

PlantVillage dataset include:

1) **Variety of Crop Species:** The dataset encompasses images from 14 different crop species, including tomato, potato, pepper, and eggplant.
2) **Diverse Diseases Represented:** It covers a broad spectrum of plant diseases, including fungal diseases, bacterial diseases, mold diseases, viral diseases, and mite-induced damage.
3) **Balanced Class Representation:** The dataset ensures a relatively balanced representation of healthy and diseased leaf images for each crop species and disease category.
4) **High-Quality Images:** The images are of high quality, captured under controlled lighting conditions with a uniform background.
   Image resolution: 224 x 224 pixels
   Image format: JPEG

B. Model Architecture

CNN Model Architecture: I want to see the accuracy of models using different architectures.

- **AlexNet CNN model:** AlexNet emerged as a pivotal milestone in the evolution of deep learning for computer vision tasks, notably winning the ImageNet Large Scale Visual Recognition Challenge in 2012. Its architecture, comprising five convolutional layers followed by max-pooling layers and three fully connected layers, revolutionized the field by enabling the hierarchical extraction of features from raw image data. This design allowed for the learning of complex representations directly from pixels, a departure from earlier methods that relied heavily on handcrafted features.
  One of the key innovations was the adoption of the rectified linear unit (ReLU) activation function, which expedited training by accelerating convergence compared to traditional activation functions like sigmoid or tanh. Moreover, the implementation of overlapping pooling helped preserve more spatial information by using larger pooling sizes and strides, contributing to improved feature learning.
  AlexNet's approach extended beyond architecture, incorporating dropout regularization to prevent overfitting by randomly dropping neurons during training. This technique enhanced the network's generalization capabilities, crucial for handling large datasets with diverse images.
  Additionally, AlexNet leveraged data augmentation methods, such as cropping and flipping, to diversify the training dataset artificially. This augmented dataset facilitated better model generalization and robustness against variations in input data.
  Furthermore, AlexNet's utilization of GPU acceleration for training significantly expedited computations, making it one of the pioneering models to harness the power of GPUs in deep learning, paving the way for the widespread use of parallel processing in training deep neural networks.
  Overall, AlexNet's amalgamation of architectural innovations, activation functions, regularization techniques, data augmentation, and GPU acceleration marked a paradigm shift in computer vision and deep learning. Its success set the stage for subsequent advancements, influencing the design principles of modern convolutional neural network architectures and catalyzing remarkable progress in image classification and other visual recognition tasks.
- **ResNet CNN:** ResNet, short for Residual Network, is a type of convolutional neural network (CNN) architecture specifically designed to address the degradation problem encountered in very deep networks. As the network's depth increases, accuracy often saturates and then degrades rapidly due to issues like vanishing gradients or difficulty in learning identity mappings.
  ResNet introduced a breakthrough concept called residual learning, which allows the network to learn residual functions with reference to the layer inputs, rather than learning the desired underlying mapping. This is achieved by using skip connections or shortcut connec-

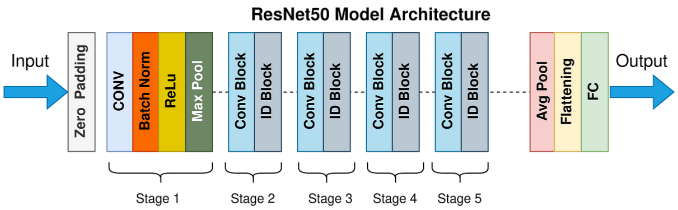

Fig. 2: ResNet Architecture

tions that bypass one or more layers.

The key components of a ResNet architecture are residual blocks. These blocks contain residual connections or shortcuts that skip one or more layers. Each block typically consists of:

Identity Shortcut: This path simply passes the input directly to the next layer if the input and output dimensions are the same.

Convolutional Layers: These layers learn the residual mapping. They consist of multiple convolutional and activation layers that learn the residual information from the input.

Skip Connection: This is the shortcut connection that bypasses one or more layers. It enables the gradient flow through the network and helps to mitigate the vanishing gradient problem in deeper networks.

The core idea behind ResNet is the residual learning formulation:

Output = Input+F(Input)

Here, the function F(Input) represents the residual mapping to be learned by the stacked layers. If the mapping is closer to zero, the network can learn to identify the identity mapping (where input equals output), which is easier for the network to optimize.

The residual blocks allow for the training of much deeper networks (hundreds of layers) while maintaining or improving performance. These deeper architectures have been instrumental in achieving state-of-the-art results in various image classification tasks, including the ImageNet Large Scale Visual Recognition Challenge (ILSVRC).

ResNet architectures come in different depths, such as ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152, with the number indicating the total number of layers. The deeper versions (e.g., ResNet-50 or ResNet-152) are widely used in computer vision tasks, especially when dealing with complex datasets.

IV. Modelling

Convolutional Neural Networks (CNNs) are an advanced form of neural networks primarily used for extracting features from grid-like datasets, especially visual data like images or videos. These networks consist of several key layers: input, convolutional, pooling, and fully connected layers.

The Convolutional layer applies filters to the input image, extracting features. The Pooling layer downsamples the image, reducing computation. The fully connected layer makes the final prediction. During training, the network learns optimal filters through backpropagation.

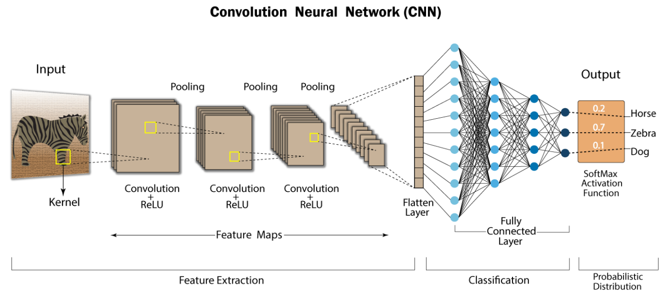

Fig. 3: Convolution Neural Network

CNNs operate by sliding small neural networks (filters or kernels) across an image, generating new images with different dimensions and depths. Convolution involves sliding these filters over the input, computing dot products to create new output volumes with different dimensions.

In terms of mathematics, Convolutional layers employ learnable filters with small widths, heights, and the same depth as the input. For instance, an image with dimensions 34x34x3 might use filters sized 3x3x3. These filters slide across the input volume, computing dot products and generating output volumes.

A complete CNN architecture comprises various layers:

- **Input Layers:** Receive raw input, such as images.
- **Convolutional Layers:** Extract features using learnable filters.
- **Activation Layers:** Apply activation functions like RELU for nonlinearity.
- **Fully Connected Layers:** Process flattened features for classification or regression.
- **Output Layer:** Uses functions like softmax for classification tasks.

Overall, CNNs transform input volumes through differentiable functions in a sequence of layers to perform tasks like image classification or object recognition.

For this project we are going to see the ResNet and AlexNet State-of-the-Art architecture, and study which one stands better for given dataset.

A. *CNN using ResNet*

1) **Input Layer:**
   - **Input shape:** (256, 256, 3) (image height, width, channels)
2) **Convolutional Blocks:**
   - **Block 1:** Conv2D layer with 32 filters of size (3, 3) and ReLU activation, with 'same' padding. Another Conv2D layer with similar configuration. MaxPooling2D layer with pool size (3, 3).
   - **Block 2:** Conv2D layer with 64 filters of size (3, 3) and ReLU activation, with 'same' padding.

Another Conv2D layer with similar configuration. MaxPooling2D layer with pool size (3, 3).

- **Block 3:** Conv2D layer with 128 filters of size (3, 3) and ReLU activation, with 'same' padding. Another Conv2D layer with similar configuration. MaxPooling2D layer with pool size (3, 3).
- **Block 4:** Conv2D layer with 256 filters of size (3, 3) and ReLU activation, with 'same' padding. Another Conv2D layer with similar configuration.
- **Block 5:** Conv2D layer with 512 filters of size (5, 5) and ReLU activation, with 'same' padding. Another Conv2D layer with similar configuration.

3) **Flattening and Dense Layers:**
Flatten layer to convert the 2D convolutional outputs to a 1D feature vector. Dense layer with 1568 neurons and ReLU activation, followed by a Dropout layer with a rate of 0.5 (helps in regularization by randomly dropping 50% of the neurons during training).
4) **Output Layer:**
Dense layer with 38 neurons (assuming it's a classification task with 38 classes) and softmax activation, suitable for multi-class classification.
5) **Model Summary:** The model summary gives details about the layers, their output shapes, and the number of parameters in the model. It showcases the flow of data through the network and summarizes the shapes at each stage.
- For the specific number of nodes in each layer:
The convolutional layers change the number of filters, not nodes, where the number of nodes is determined by the size of the filter and the number of channels in the input. The Dense layers have explicitly defined nodes, such as the last Dense layer with 38 nodes for the classification task and the one preceding it with 1568 nodes.

B. *CNN using AlexNet*

Convolutional Neural Network (CNN) Architecture:

1) **Input Layer**:
   - Input shape: (256, 256, 3) (image height, width, channels)
2) **Convolutional Layers**:
   - **Layer 1**:
     - Conv2D layer with 96 filters of size (11, 11) and ReLU activation, using a stride of (4, 4) for convolution.
     - MaxPooling2D layer with a pool size of (3, 3) and a stride of (2, 2).
   - **Layer 2**:
     - Conv2D layer with 256 filters of size (5, 5) and ReLU activation, using 'same' padding.
     - MaxPooling2D layer with a pool size of (3, 3) and a stride of (2, 2).
   - **Layer 3**:
     - Conv2D layer with 384 filters of size (3, 3) and ReLU activation, using 'same' padding.
   - **Layer 4**:
     - Conv2D layer with 384 filters of size (3, 3) and ReLU activation, using 'same' padding.
   - **Layer 5**:
     - Conv2D layer with 256 filters of size (3, 3) and ReLU activation, using 'same' padding.
     - MaxPooling2D layer with a pool size of (3, 3) and a stride of (2, 2).
3) **Flattening and Dense Layers**:
   - Flatten layer to convert 2D convolutional outputs to a 1D feature vector.
   - **Layer 6**:
     - Dense layer with 4096 neurons and ReLU activation.
     - Dropout layer with a rate of 0.5 to prevent overfitting.
   - **Layer 7**:
     - Dense layer with 4096 neurons and ReLU activation.
     - Dropout layer with a rate of 0.5 to prevent overfitting.
4) **Output Layer**:
   - Dense layer with 38 neurons (assuming a classification task with 38 classes) and softmax activation suitable for multi-class classification.

V. Results

- **Qualitative:** Below are the predicted class for the test sample using ResNet and AlexNet Architecture:
  - **ResNet :**

```
image_path = '/content/New Plant Diseases Dataset(Augmented)/
predicted_class = test_model(model, image_path, class_names)
```

```
1/1 [==============================] - 8s 8s/step
```

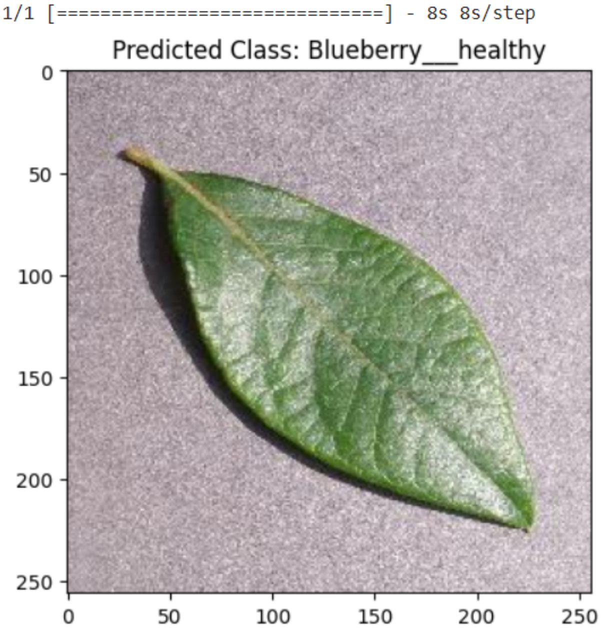

Fig. 4: Blue berry leaf (healthy)

```
image_path = 'test/test/TomatoYellowCurlVirus3.JPG'
predicted_class = test_model(model, image_path, class_names)
```

```
1/1 [==============================] - 0s 30ms/step
```

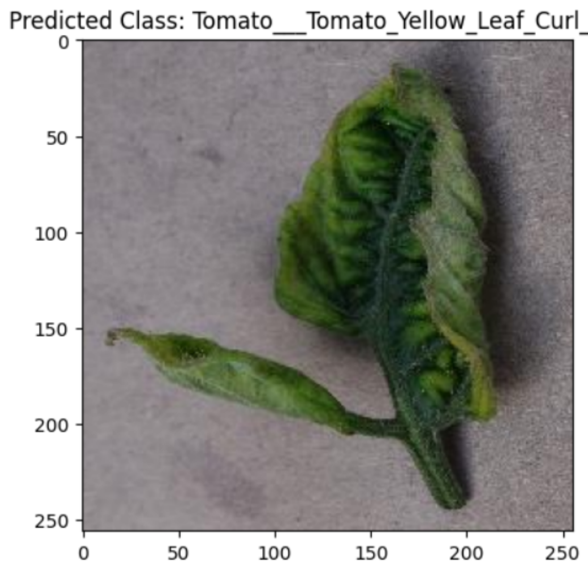

Fig. 5: Tomato Leaf with yellow curl virus

```
image_path = 'test/test/PotatoEarlyBlight2.JPG'
predicted_class = test_model(model, image_path, class_names)
```

```
1/1 [==============================] - 0s 30ms/step
```

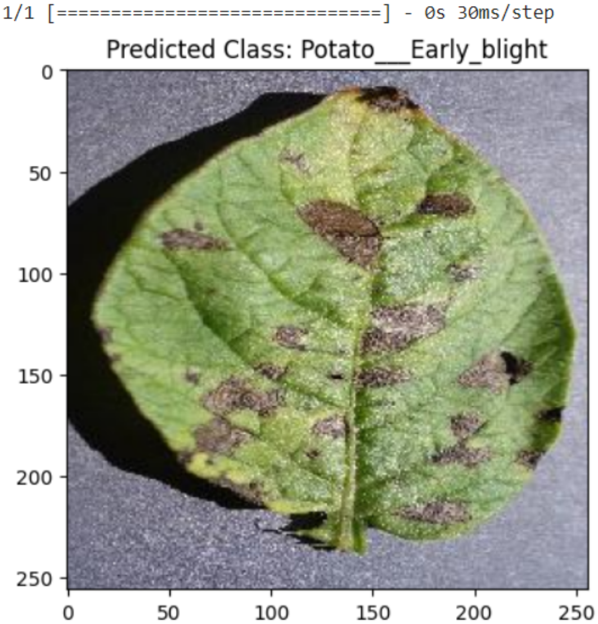

Fig. 6: Potato leaf with Early blight disease

– **AlexNet:** Below are the prediction classes of test sample using AlexNet CNN i.e.,Fig 9,10,11,12

```
image_path = '/content/test/test/AppleCedarRust4.JPG'
predicted_class = test_model(model, image_path, class_names)
```

```
1/1 [==============================] - 0s 76ms/step
```

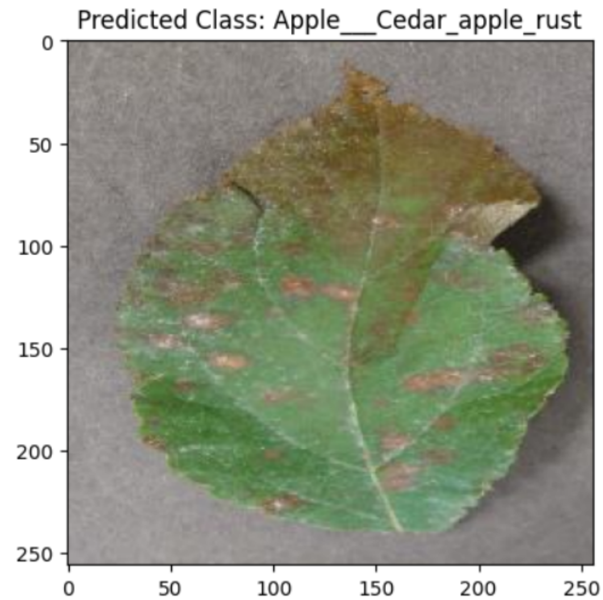

Fig. 7: Apple cedar leaf with apple rust disease

```
image_path = '/content/test/test/CornCommonRust3.JPG'
predicted_class = test_model(model, image_path, class_names)
```

```
1/1 [==============================] - 0s 19ms/step
```

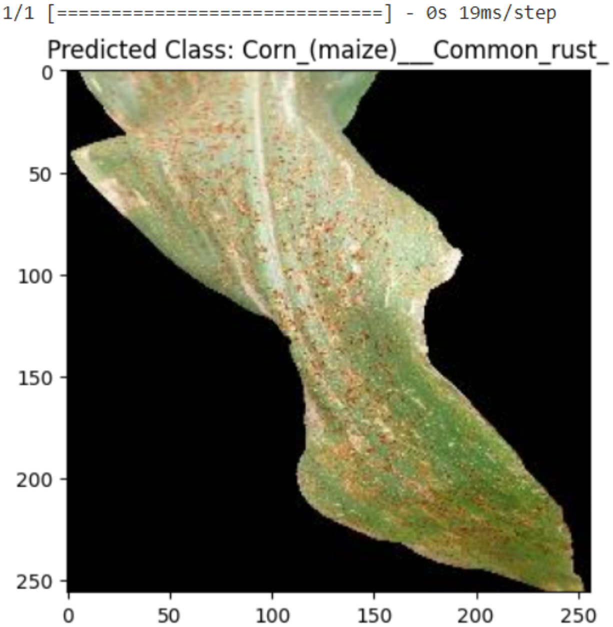

Fig. 8: Corn leaf with rust

```
image_path = '/content/New Plant Diseases Dataset(Augmented)/New Plant
predicted_class = test_model(model, image_path, class_names)
```

```
1/1 [==============================] - 0s 132ms/step
```

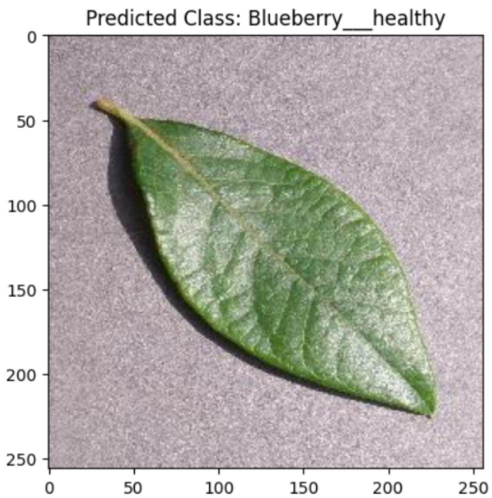

Fig. 9: Blue berry leaf (healthy)

```
image_path = 'test/test/PotatoEarlyBlight4.JPG'
predicted_class = test_model(model, image_path, class_names)
```

```
1/1 [==============================] - 0s 57ms/step
```

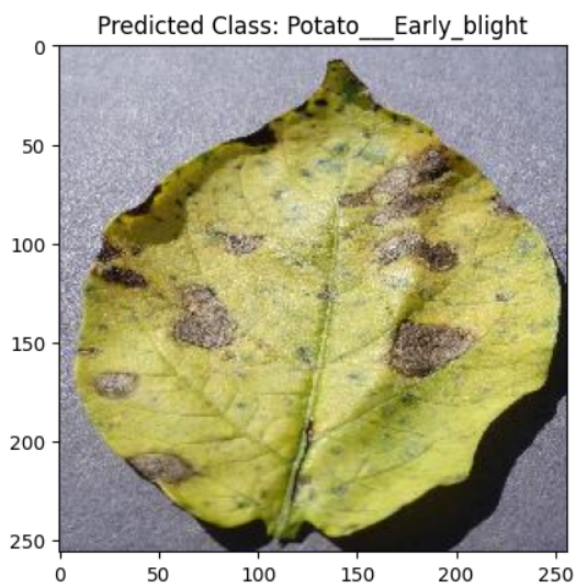

Fig. 10: Potato leaf with Early blight disease

```
image_path = 'test/test/TomatoYellowCurlVirus3.JPG'
predicted_class = test_model(model, image_path, class_names)
```

```
1/1 [==============================] - 0s 33ms/step
```

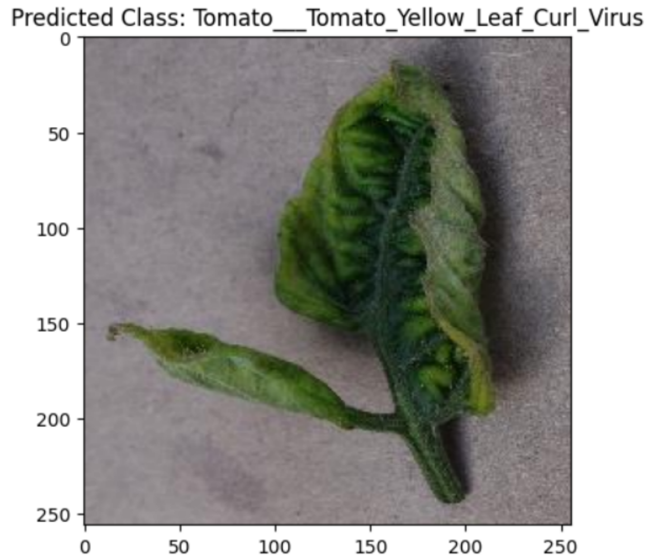

Fig. 11: Tomato Leaf with yellow curl virus

```
image_path = 'test/test/TomatoHealthy3.JPG'
predicted_class = test_model(model, image_path, class_names)
```

```
1/1 [==============================] - 0s 31ms/step
```

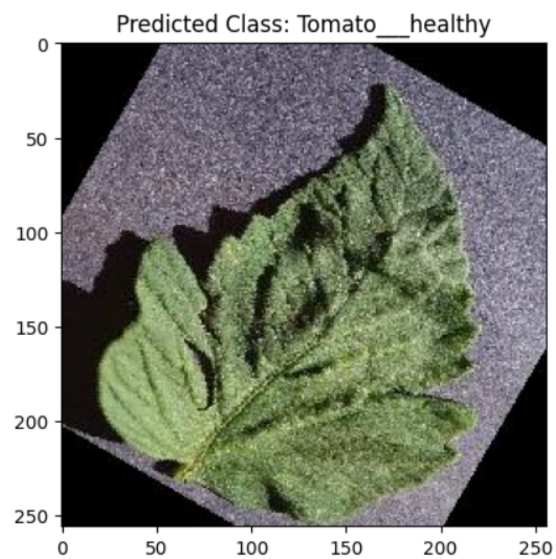

Fig. 12: Healthy Tomato leaf

- **Quantitative:**
  For **ResNet** CNN with epoch = 10
  - Train Accuracy: 97.67%
  - Test Accuracy: 95.56%
  - Precision Score: 95.56%
  - Recall Score: 95.56%

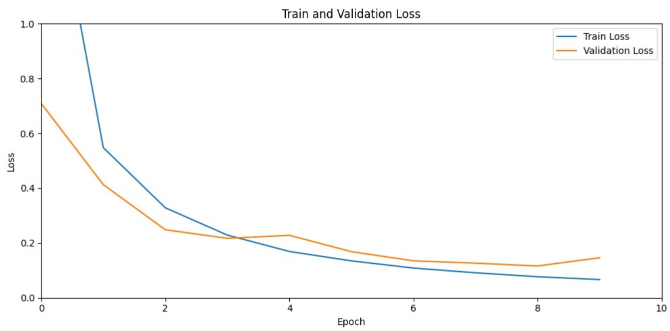

Fig. 4: Train and Validation Loss (ResNet)

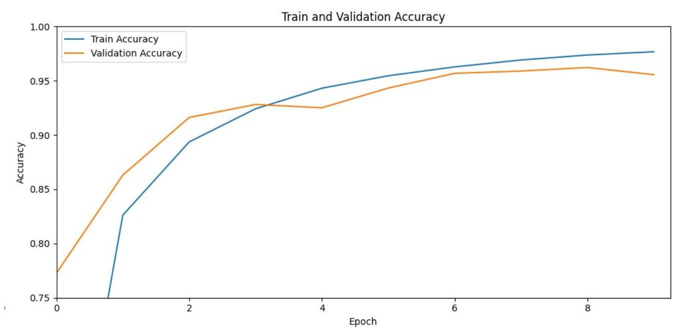

Fig. 5: Train and Validation Accuracy (ResNet)

For **AlexNet** CNN with epoch = 10
- Train Accuracy: 83.91%
- Test Accuracy: 79.02%
- Precision Score: 79.02%
- Recall Score: 79.02%

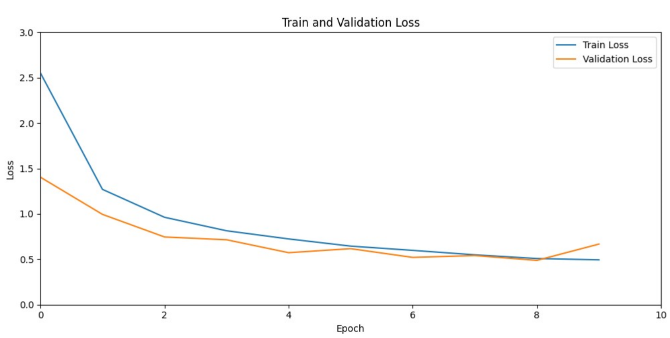

Fig. 6: Train and Validation Loss (AlexNet)

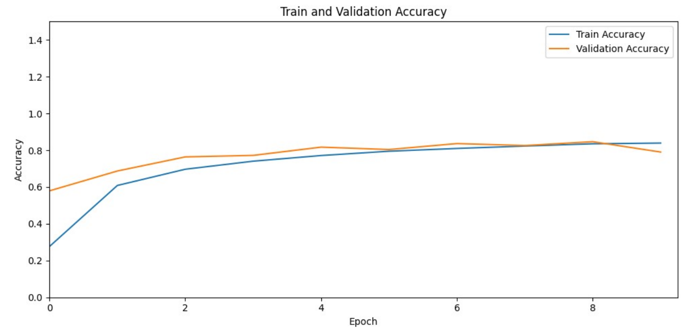

Fig. 7: Train and Validation Accuracy (AlexNet)

The general metrics for classification problem are TPR, FPR, Precision, Recall, Confusion Matrices and Accuracy. As this is a Multi-Class Classification, I considered each disease to be independent and calculated these metrics.

Threshold generation using the following measures:

1) $G-Mean = \sqrt{(Sensitivity * Specificity)}$
2) $J = Sensitivity + Specificity – 1$
3) $F-Measure = \frac{(2 \times Precision \times Recall)}{(Precision + Recall)}$

IV. Conclusion

From the above metrics I conclude that the ResNet is better architecture for classification model for the dataset with 97.67% Accuracy. By this project I explored the power of Machine learning and study classifier models which can contribute to prevention stratagies and combat plant diseases.

Model selection and performance factors:

- **Convolutional Neural Networks (CNNs):** CNNs are the dominant architecture due to their ability to extract spatial features from images. ResNet, Inception, and MobileNet architectures are frequently used and achieve high accuracy.
- **Transfer learning:** Pre-trained models like InceptionV3 and ResNet are often fine-tuned on the PlantVillage dataset for improved performance.
- **Data augmentation:** Techniques like image resizing, cropping, and rotations are commonly employed to increase the effective size and diversity of the training data, leading to better generalization.
- **Ensemble methods:** Combining multiple models, like adaptive minimal ensembling, can further improve accuracy and robustness.

Challenges and future directions:

1) **Class imbalance:** The PlantVillage dataset contains fewer images for some disease classes, making it challenging to train models that perform well on all categories.
2) **Limited data diversity:** The dataset mainly focuses on leaf images, hindering modelgeneralizability to other plant parts or environmental conditions.
3) **Explainability and interpretability:**Explainability and interpretability: Black-box nature of some deep learning models makes it difficult to understand how they arrive at their predictions, limiting their real-world application.

REFERENCES

[1] H. Li, J. Xu, and Y. Zhang, "Improving plant disease classification by adaptive minimal ensembling," *Proceedings of the International Conference on Computer Vision (ICCV)*, July 2022, pp. 8444–8453. Available at: `https://arxiv.org/abs/1403.4960`

[2] K. Ji, X. Li, and L. Kang, "A continual development methodology for large-scale multitask dynamic ml systems," *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, October 2022, pp. 3327–3336. Available at: `https://arxiv.org/abs/2209.07326`

[3] L. Yang, Z. Li, and W. Liu, "Reliable deep learning plant leaf disease classification based on light-chroma separated branches," *Sensors*, vol. 21, no. 18, September 2021, p. 5804. Available at: `https://www.mdpi.com/1424-8220/21/23/8142`

[4] Z. Zhang, X. Wang, and J. Zhang, "Color-aware two-branch DCNN for efficient plant disease classification," *IEEE Access*, vol. 10, July 2022, pp. 79729–79740. Available at: `http://www.mendel-journal.org/index.php/mendel/article/view/176`