اللہ اکبر

علم الانسان ما لم یعلم

جامعہ ملیہ اسلامیہ

PLANT LEAF DISEASE DETECTION

Minor Project Report

Submitted in the partial fulfilment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER ENGINEERING

Submitted By

Vicky Gupta (20BCS070)

Ravi Gowri Jaswanth Reddy (20BCS065)

Under the supervision of

Dr. Faiyaz Ahmad

(Assistant Professor)

DEPARTMENT OF COMPUTER ENGINEERING

FACULTY OF ENGINEERING & TECHNOLOGY

JAMIA MILLIA ISLAMIA

NEW DELHI (110025)

(YEAR - 2024)

CERTIFICATE

This is to certify that the project entitled “Plant Leaf Disease Detection” by Vicky Gupta (20BCS070) and Ravi Gowri Jaswanth Reddy (20BCS065) is a record of Bonafide work carried out by them, in the Department of Computer Engineering, Jamia Millia Islamia, New Delhi, under my supervision and guidance in partial fulfilment of requirements for the award of Bachelor Of Engineering in Computer Engineering, Jamia Millia Islamia in the academic year 2021.

Prof. Bashir Alam
(Head of the Department).
Department of Computer Engineering
Faculty of Engineering & Technology
JAMIA MILLIA ISLAMIA
NEW DELHI

Dr. Faiyaz Ahmad
(Assistant Professor)
Department of Computer Engineering
Faculty of Engineering & Technology
JAMIA MILLIA ISLAMIA
NEW DELHI

ACKNOWLEDGEMENT

A very sincere and honest acknowledgement to Dr. Faiyaz Ahmad, Assistant Professor, Department of Computer Engineering, Jamia Millia Islamia, New Delhi for his invaluable technical guidance, great innovative ideas and overwhelming support. We are very grateful to our HOD Prof. Bashir Alam for his valuable support throughout the project. We would also like to express our gratitude to the Department of Computer Engineering and entire faculty members, for their teaching, guidance and encouragement. We are also thankful to our classmates and friends for their valuable suggestions and support whenever required. We regret any inadvertent omissions.

Vicky Gupta (20BCS070)

Ravi Gowri Jaswanth Reddy (20BCS065)

ABSTRACT

Modern technologies have given human society the ability to produce enough food to meet the demand of more than 7 billion people. However, food security remains threatened by several factors including climate change the decline in pollinators (Report of the Plenary of the Intergovernmental Science - Policy Platform on Biodiversity Ecosystem and Services on the work of its fourth session, 2016), plant diseases (Strange and Scott, 2005), and others. Plant diseases are not only a threat to food security at the global scale but can also have disastrous consequences for smallholder farmers whose livelihoods depend on healthy crops. In the developing world, more than 80 percent of agricultural production is generated by smallholder farmers (UNEP, 2013), and reports of yield loss of more than 50% due to pests and diseases are common. Furthermore, the largest fraction of hungry people (50%) lives in smallholder farming households (Sanchez and Swaminathan, 2005), making smallholder farmers a group that is particularly vulnerable to pathogen-derived disruptions in food supply.

Deep neural networks have recently been successfully applied in many diverse domains as examples of end-to-end learning. Neural networks provide a mapping between an input—such as an image of a diseased plant—to an output—such as a crop disease pair. The nodes in a neural network are mathematical functions that take numerical inputs from the incoming edges and provide a numerical output as an outgoing edge.

Keywords: deep learning, transfer learning, CNN leaf pathology, leaf disease

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 What is Plant leaf disease.

Plant leaf disease detection refers to the process of identifying and diagnosing diseases that affect the leaves of plants. Plant diseases can be caused by various factors, including fungi, bacteria, viruses, and environmental stressors. Early and accurate detection of these diseases is crucial for effective disease management in agriculture and horticulture.

The primary goals of plant leaf disease detection are:

- **Early Identification:** Detecting diseases at their initial stages allows for prompt intervention, minimizing the potential damage to crops.
- **Prevention and Control:** Identifying the type of disease helps farmers and agricultural experts implement targeted control measures, such as applying fungicides or adjusting environmental conditions.
- **Reducing Economic Losses:** By identifying and managing diseases early on, farmers can mitigate economic losses caused by reduced crop yield and quality.
- **Ensuring Food Security:** Plant diseases can pose a threat to food security by affecting the quantity and quality of crops. Timely detection helps maintain a stable food supply.

1.2 Challenges of Plant leaf disease detection (PLDD).

Diversity of Diseases:
Plant diseases manifest in diverse ways, making it challenging to create a comprehensive model that can detect various diseases accurately.

Limited and Imbalanced Datasets:
Obtaining labelled datasets with a sufficient number of examples for each disease can be difficult. Additionally, datasets may be imbalanced, with some diseases having more instances than others.

Environmental Variability:
Environmental conditions, such as lighting and background, can introduce variability in leaf images, affecting the model's robustness.

Real-Time Detection:
Some traditional methods may be time-consuming, hindering the implementation of real time disease detection.

Inter-Class Variability:
Diseases within the same class may exhibit variations that are difficult to distinguish visually.

1.3 Applications of Plant leaf disease detection (PLDD).

Early Disease Identification:

- ***Significance:*** PLDD enables the early detection of diseases on plant leaves.
- ***Application:*** Farmers can identify diseases in their crops at an early stage, allowing for timely intervention and management to prevent further spread and minimize crop losses.

Precision Agriculture:

- *Significance:* PLDD supports precision agriculture by providing targeted information about the health of individual plants or specific areas within a field.
- *Application:* Farmers can implement precise treatments, such as localized pesticide application, based on the specific disease detected in a particular region of the crop field.

Crop Monitoring and Management:

- ***Significance:*** PLDD facilitates continuous monitoring of crops.
- ***Application:*** Automated systems can monitor large agricultural areas, providing real-time information on the presence and progression of diseases. This information guides farmers in making informed decisions about crop management practices.

Smart Farming:

- ***Significance:*** PLDD is a crucial component of smart farming practices.
- ***Application:*** Integrated with other technologies like Internet of Things (IoT) devices and data analytics, PLDD contributes to creating smart farming systems that optimize resource use, improve crop yield, and reduce environmental impact.

1.4 Deep learning for Plant leaf disease detection (PLDD).

Diversity of Diseases:
Deep Learning Solution: Deep learning models, particularly convolutional neural networks (CNNs), excel at learning hierarchical features, allowing them to recognize complex patterns associated with different diseases.

Limited and Imbalanced Datasets:
Deep Learning Solution: Transfer learning techniques allow pre-trained models on large datasets to be fine-tuned for specific diseases, even with limited labeled samples. This helps leverage knowledge from one domain to improve performance in another.

Environmental Variability:
Deep Learning Solution: Data augmentation techniques, integrated into deep learning pipelines, artificially increase the diversity of the training dataset. This helps the model become more resilient to variations in environmental conditions.

Real-Time Detection:
Deep Learning Solution: Deep learning models, once trained, can provide rapid and realtime predictions, making them suitable for on-the-fly detection in agricultural settings.

Inter-Class Variability:
Deep Learning Solution: Deep learning models can learn intricate features and patterns, enabling them to differentiate between subtle variations within the same disease class.

2. REVIEW OF LITERATURE

Table. 2.1 Literature review

Sno	Title	Author	Technique	Dataset	Accuracy	Year
1	Deep Learning-Based Leaf Disease Detection in Crops Using Images for Agricultural Applications	J. Andrew et al.	The DCNN is pre-trained on a large dataset of natural images (ImageNet).	PlantVillage Dataset	95.31%	2022
2	Using Deep Learning for Image-Based Plant Disease Detection	Mohanty, S. P et al.	A pre-trained CNN model (AlexNet) is fine-tuned on the specific plant disease detection task	PlantVillage Dataset:	99.35%	2016
3	Identification of plant leaf diseases by deep learning based on channel attention and channel pruning	Riyao Chen et al.	The paper proposes a DCNN architecture called CACPNET for leaf disease identification.	PlantDoc Dataset, PlantVillage Dataset	99.28% accuracy on PlantDoc 97.24% accuracy on PlantVillage	2022
4	Plant Leaf Diseases Detection	Deshmukh Sanket	VGG16, ResNet,	PlantVillage,	VGG16: 95.31% on	2023

	Using Deep Learning Algorithms	Jitendra et al.	Custom CNN architectures	Custom Leaf Disease Dataset	PlantVillage Dataset, ResNet: 96.27% on PlantVillage. Custom CNN architecture: 98.75%	
5	Leaf Disease Detection Using Deep Learning	Teenu Sahasra et al.	Convolutional Neural Network (CNN)	PlantVillage Dataset	96.54%	2021
6	Deep Learning-based Plant Leaf Disease Detection with Limited Training Data	Zhang et al.	Convolutional Neural Network (CNN)	PlantVillage Dataset	92.3%	2023
7	Plant Leaf Disease Detection Using EfficientNet Deep Learning Model	Rauf, H et al	EfficientNet	PlantVillage Dataset	99.80%	2021

3. TECHNICAL BACKGROUND

3.1 Neural Networks

A neural network is a network or circuit of neurons, or in a modern sense, an artificial neural network, composed of artificial neurons or nodes. Thus, a neural network is either a biological neural network, made up of real biological neurons, or an artificial neural network, for solving artificial intelligence (AI) problems. The connections of the biological neuron are modelled as weights. A positive weight reflects an excitatory connection, while negative values mean inhibitory connections. All inputs are modified by a weight and summed. This activity is referred to as a linear combination. Finally, an activation function controls the amplitude of the output. For example, an acceptable range of output is usually between 0 and 1, or it could be -1 and 1.

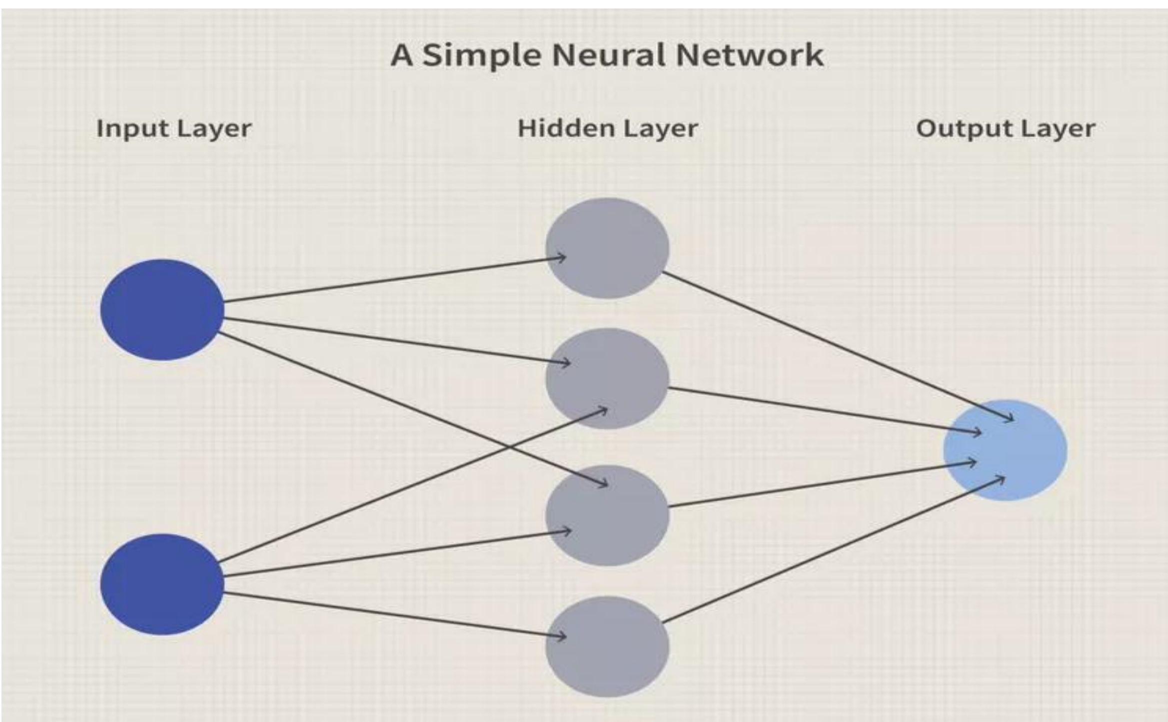

Fig. 3.1 A Simple Neural Network

These artificial networks may be used for predictive modelling, adaptive control and applications where they can be trained via a dataset. Self-learning resulting from experience can occur within networks, which can derive conclusions from a complex and seemingly unrelated set of information.

A biological neural network is composed of a group or groups of chemically connected or functionally associated neurons. A single neuron may be connected to many other neurons and the total number of neurons and connections in a network may be extensive. Connections, called synapses, are usually formed from axons to dendrites, though dendrodendritic synapses and other connections are possible. Apart from the electrical signalling, there are other forms of signalling that arise from neurotransmitter diffusion.

Artificial intelligence, cognitive modelling, and neural networks are information processing paradigms inspired by the way biological neural systems process data. Artificial intelligence and cognitive modelling try to simulate some properties of biological neural networks. In the artificial intelligence field, artificial neural networks have been applied successfully to speech recognition, image analysis and adaptive control, in order to construct software agents (in computer and video games) or autonomous robots.

3.2 Propagation (How Neural Networks Work)

3.2.1 Weights and Biases

Weight is the parameter within a neural network that transforms input data within the network's hidden layers. A neural network is a series of nodes, or neurons. Within each node is a set of inputs, weight, and a bias value. As an input enters the node, it gets multiplied by a weight value and the resulting output is either observed, or passed to the next layer in the neural network. Often the weights of a neural network are contained within the hidden layers of the network. Within a neural network there's an input layer that takes the input signals and passes them to the next layer. Next, the neural network contains a series of hidden layers which apply transformations to the input data. It is within the nodes of the hidden layers that the weights are applied. For example, a single node may take the input data and multiply it by an assigned weight value, then add a bias before passing the data to the next layer. The final layer of the neural network is also known as the output layer. The output layer often tunes the inputs from the hidden layers to produce the desired numbers in a specified range.

Weights and bias are both learnable parameters inside the network. A teachable neural network will randomise both the weight and bias values before learning initially begins. As training continues, both parameters are adjusted toward the desired values and the correct output. The two parameters differ in the extent of their influence upon the input data. Simply, bias represents how far off the predictions are from their intended value. Biases make up the difference between the function's output and its intended output. A low bias suggests that the network is making more assumptions about the form of the output, whereas a high bias value makes less assumptions about the form of the output. Weights, on the other hand, can be thought of as the strength of the connection. Weight affects the amount of influence a change in the input will have upon the output. A low weight value will have no change on the input, and alternatively a larger weight value will more significantly change the output.

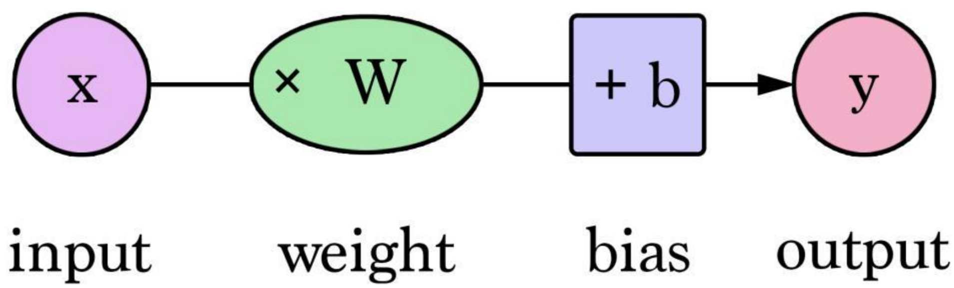

Fig. 3.2 Weights and Biases

3.2.2 Activation Function

An Activation Function decides whether a neuron should be activated or not. This means that it will decide whether the neuron’s input to the network is important or not in the process of prediction using simpler mathematical operations. The role of the Activation Function is to derive output from a set of input values fed to a node (or a layer).

The primary role of the Activation Function is to transform the summed weighted input from the node into an output value to be fed to the next hidden layer or as output. Examples of activation functions include Binary Step Activation Function, Linear Activation Function and Non-Linear Activation Functions like Sigmoid Activation Function and Tanh Activation Function.

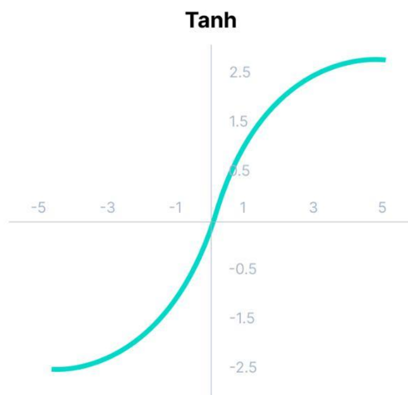

Fig. 3.3 Tanh Function (Hyperbolic Tangent)

3.3 ResNet-50

ResNet-50, short for Residual Network with 50 layers, stands as a pivotal milestone in the evolution of convolutional neural networks (CNNs).ResNet-50 addresses the challenge of training deep neural networks by introducing a novel architecture that facilitates the training of networks with an unprecedented number of layers.

3.3.1 Residual Learning and Skip Connections

The core innovation of ResNet-50 lies in its use of residual learning blocks. Traditional deep networks often encounter the vanishing gradient problem, hindering the training of deep architectures. ResNet mitigates this issue by introducing skip connections or shortcuts. These shortcuts enable the gradient to flow directly through the network, allowing the model to learn residual functions - the difference between the predicted output and the ground truth.

Each residual block consists of two convolutional layers with batch normalization and rectified linear unit (ReLU) activations. The output of these layers is combined with the original input through a shortcut connection, forming the residual connection. This unique architecture enables the training of deep networks with improved accuracy and convergence.

3.3.2 Architecture Overview

ResNet-50 comprises 50 layers, organized into modular blocks. These blocks consist of multiple residual units, and the network structure facilitates the learning of hierarchical features. The use of global average pooling (GAP) in the final layers reduces spatial dimensions, leading to a more efficient and robust model.

3.3.3 Application in Image Recognition

ResNet-50 has gained prominence in image recognition tasks, particularly in large-scale classification problems. Its ability to capture intricate features and hierarchical representations makes it suitable for discerning patterns in images. The model has been pre-trained on extensive datasets, such as ImageNet, and subsequently fine-tuned for specific applications.

3.3.4 Applications Beyond Image Classification

While initially designed for image classification, ResNet-50 has proven versatile across various computer vision tasks. It has been successfully applied to object detection, image segmentation, and even in transfer learning scenarios where pre-trained models are adapted to new domains with limited labeled data.

3.3.5 Transfer Learning and Pre-trained Models

One of the significant advantages of ResNet-50 is its availability as a pre-trained model. Pre-trained on datasets like ImageNet, ResNet-50 provides a robust feature extractor that can be fine-tuned for specific tasks. This transfer learning capability is especially valuable in scenarios where labeled data is scarce, as the model leverages knowledge gained from a diverse dataset.

3.3.6 Challenges and Considerations

While ResNet-50 has demonstrated exceptional performance, it is not without challenges. The increased depth of the network requires substantial computational resources for training. Furthermore, fine-tuning for specific tasks demands careful consideration of dataset characteristics and domain-specific nuances.

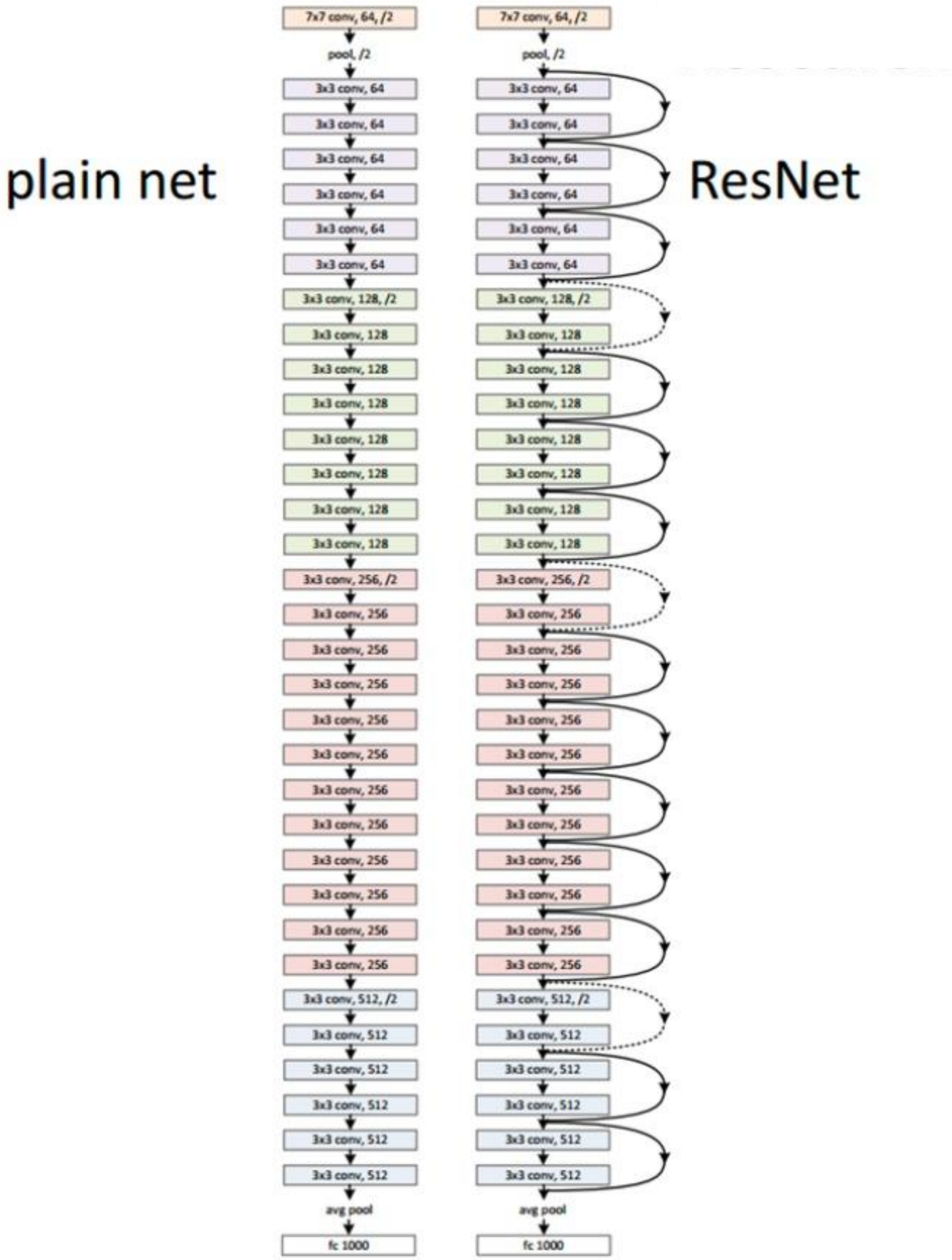

Fig. 3.4 ResNet

4. PROPOSED METHODOLOGY

4.1 Basic Outline

- **Data Collection:** Images of leaves with and without diseases are collected from various sources.
- **Preprocessing:** Images are resized, normalized, and augmented to increase the size and diversity of the dataset.
- **Model Training:** A Recurrent Neural Network (RNN) model is trained on the pre-processed dataset.
- **Evaluation:** The trained model is evaluated on a separate test dataset to assess its accuracy in disease detection.
- **Testing:** The model is tested on real-world images to assess its generalizability.

4.2 Dataset Used

We used a dataset of 54,306 images of 14 crop species and 26 diseases. The dataset was divided into training, validation, and test sets with a 80:10:10 ratio in RGB, Greyscale and Segmented types.

The dataset was divided into training, validation, and test sets with a 60:20:20 ratio in RGB, Greyscale and Segmented type.

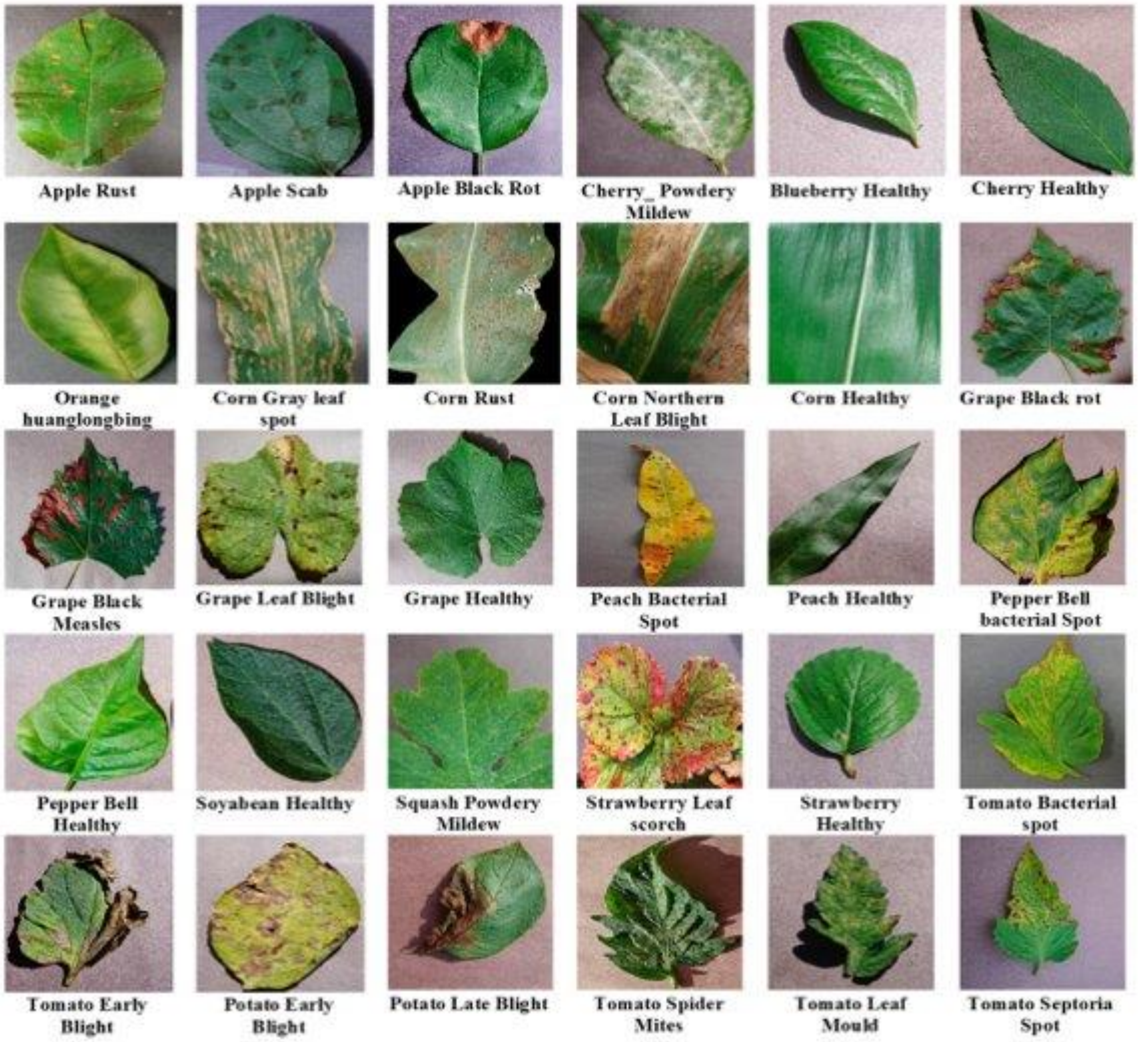

Fig. 4.1 Plant Village Dataset (PVD)

4.3 Implementation Strategy

We utilized a Transfer Learning approach, where the model which is pretrained is trained on new set of labeled data (images with known disease annotations) to learn to classify new images.

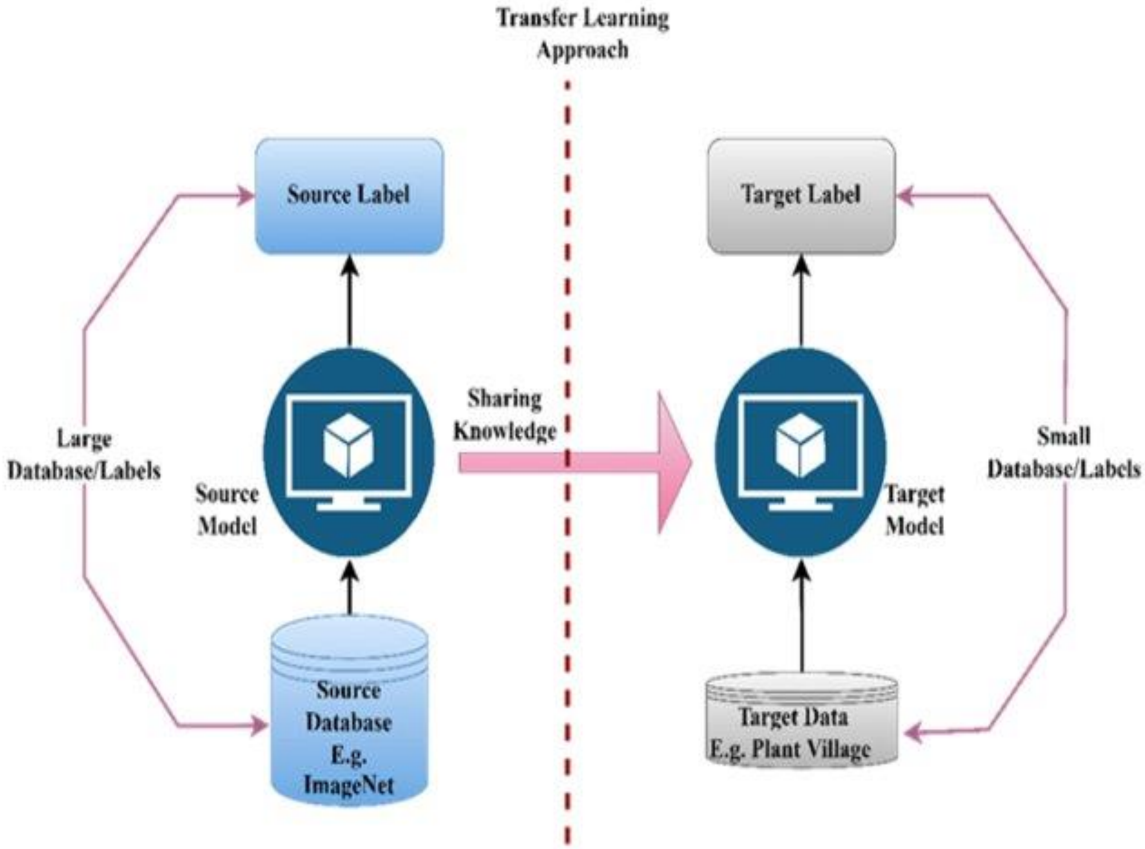

Fig. 4.2 Basic Idea behind transfer learning

4.4 Architecture Used

we implemented a convolutional neural network (CNN) architecture for image classification, utilizing the ResNet50 pre-trained model as a base.

4.5 Components Of Architecture

Base Model - ResNet50:

Pre-trained ResNet50 model with weights initialized from 'imagenet'.

Sequential Model:

- A Sequential model is used to stack layers sequentially.

Batch Normalization:

- Batch normalization layer normalizes and stabilizes activations.

Dense Layers:

- Two dense (fully connected) layers with 256 and 128 neurons, respectively.
- L2 and L1 regularization applied to kernel, activity, and bias weights.
- Activation function: 'relu'.

Dropout Layer:

- Dropout layer with a dropout rate of 0.45 to prevent overfitting.

Output Layer:

- Dense layer with a number of neurons equal to the number of classes (class_count).
- Activation function: 'softmax' for multi-class classification.

Model Compilation:

- Adamax optimizer with a learning rate of 0.001.
- Categorical crossentropy used as the loss function.
- Accuracy metric used for evaluation.

4.6 Training

The model is trained for the specified number of epochs, and the custom callback dynamically adjusts the learning rate and monitors for improvements in the specified metrics. The training process periodically prompts the user to decide whether to continue or halt training based on the defined ask_epoch.
This training setup aims to strike a balance between learning rate adjustments, early stopping, and user interaction to optimize the model's performance and prevent overfitting. The custom callback provides flexibility in adapting the training process based on real-time monitoring of relevant metrics.

Batch Size: batch_size = 40
Epochs: epochs = 30
Threshold: threshold = 0.9
Ask Epoch: ask_epoch = 5

5. LEAF RAKSHAK APPLICATION

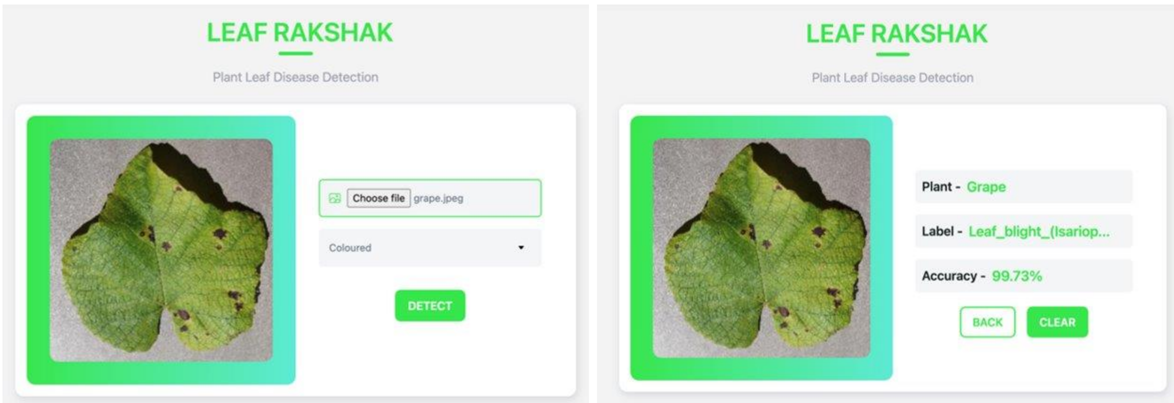

Fig. 5.1 Leaf Rakshak

In the visual representation above, we introduce an innovative application that seamlessly integrates user interaction and advanced image analysis for plant leaf assessment. The user experience is intuitive, involving a straightforward input process where the user selects the type of image they are uploading, initiating a robust analysis performed by our application.

Upon user input, the application dynamically processes the uploaded image, employing a sophisticated model trained to discern the health status of the depicted leaf. The application's primary function is to provide the user with immediate feedback regarding the leaf's condition— specifically, indicating whether the leaf is deemed healthy or, if afflicted, identifying the nature of the disease affecting it.

This interactive approach significantly simplifies the process for users, requiring only the selection of the relevant image category. Leveraging cutting-edge image recognition and machine learning technologies, our application transforms this input into valuable insights, offering users a quick and accurate diagnosis of the health status of their plant's leaves.

Notably, the application serves a dual purpose by not only recognizing instances of leaf diseases but also identifying the specific ailment affecting the leaf. This level of granularity enhances the user's understanding of potential issues, facilitating informed decision-making regarding appropriate treatments or preventive measures.

6. EVALUATION METRICS

Loss (Categorical Cross entropy):

- Printed as: Train Loss, Validation Loss, Test Loss
- Explanation: Cross entropy loss is a common choice for classification tasks. It measures the dissimilarity between the predicted class probabilities and the true class labels. Lower values indicate better model performance.

Accuracy:

- Printed as: Train Accuracy, Validation Accuracy, Test Accuracy
- Explanation: Accuracy is a widely used metric for classification tasks. It represents the percentage of correctly classified samples. While it provides a straightforward measure of overall performance, it might not be sufficient for imbalanced datasets.

Classification Report:

- Printed as: Detailed classification report for precision, recall, F1-score, and support.
- Explanation: The classification report provides a more comprehensive understanding of model performance by breaking down metrics for each class. It includes precision (the ability of the classifier not to label as positive a sample that is negative), recall (the ability of the classifier to find all positive samples), F1-score (harmonic mean of precision and recall), and support (the number of actual occurrences of the class in the specified dataset).

Why These Metrics:

Loss:

Monitoring loss helps assess how well the model is minimizing the difference between predicted and actual values during training and validation. It guides the optimization process.

Accuracy:

Accuracy provides a quick overview of the model's overall correctness. However, it might not be sufficient for imbalanced datasets where accuracy could be high even if the model performs poorly on minority classes.

Classification Report:
Provides detailed metrics for each class, addressing the potential issue of imbalanced. datasets. Precision, recall, and F1-score for individual classes give insights into class specific performance.

7. ANALYSIS OF RESULTS

```
5431/5431 [==============================] - 96s 18ms/step - loss: 0.1788 - accuracy: 0.9999
5431/5431 [==============================] - 12s 2ms/step - loss: 0.1898 - accuracy: 0.9971
5431/5431 [==============================] - 59s 11ms/step - loss: 0.1865 - accuracy: 0.9985
Train Loss:  0.17884014546871185
Train Accuracy:  0.9999309182167053
--------------------
Validation Loss:  0.18984617292881012
Validation Accuracy:  0.9970533847808838
--------------------
Test Loss:  0.18647097051143646
Test Accuracy:  0.9985269904136658
```

Fig. 7.1 Colored 80,20 Final Test Accuracy

```
5431/5431 [==============================] - 79s 15ms/step - loss: 0.2027 - accuracy: 0.9999
5431/5431 [==============================] - 10s 2ms/step - loss: 0.2581 - accuracy: 0.9864
5431/5431 [==============================] - 58s 11ms/step - loss: 0.2396 - accuracy: 0.9901
Train Loss:  0.20269350707530975
Train Accuracy:  0.9999309182167053
--------------------
Validation Loss:  0.25810664892196655
Validation Accuracy:  0.9863719940185547
--------------------
Test Loss:  0.2396090030670166
Test Accuracy:  0.9900570511817932
```

Fig. 7.2 Grayscale 80,20 Final Test Accuracy

```
5431/5431 [==============================] - 79s 15ms/step - loss: 0.2027 - accuracy: 0.9999
5431/5431 [==============================] - 10s 2ms/step - loss: 0.2581 - accuracy: 0.9864
5431/5431 [==============================] - 58s 11ms/step - loss: 0.2396 - accuracy: 0.9901
Train Loss:  0.20269350707530975
Train Accuracy:  0.9999309182167053
--------------------
Validation Loss:  0.25810664892196655
Validation Accuracy:  0.9863719940185547
--------------------
Test Loss:  0.2396090030670166
Test Accuracy:  0.9900570511817932
```

Fig. 7.3 Segmented 80,20 Final Test Accuracy

Below fig.7.2 showing how the Training and validations loss and accuracy change as no of epochs increase

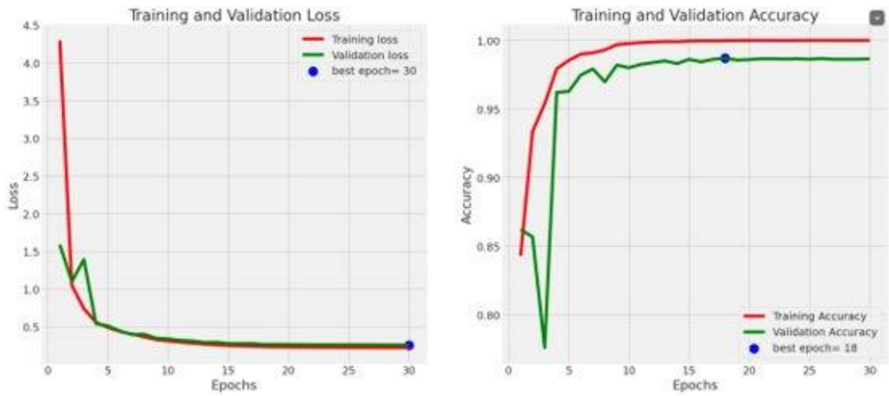

Fig. 7.4 Colored (80,20) Train Test Split

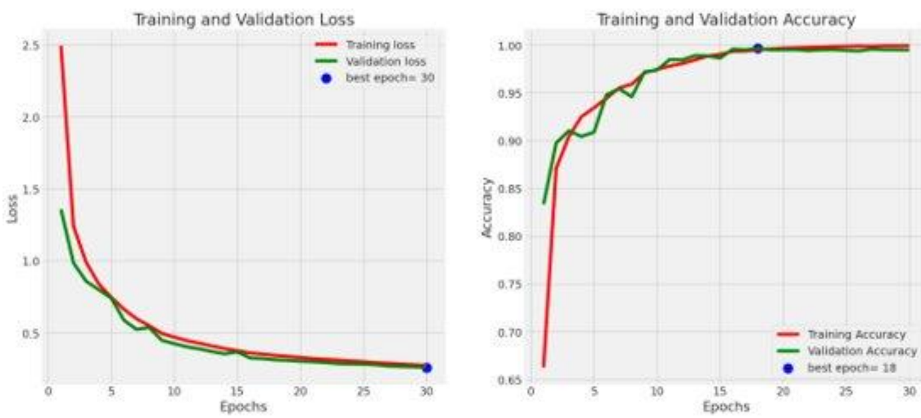

Fig. 7.5 Segmented (80,20) Train Test Split

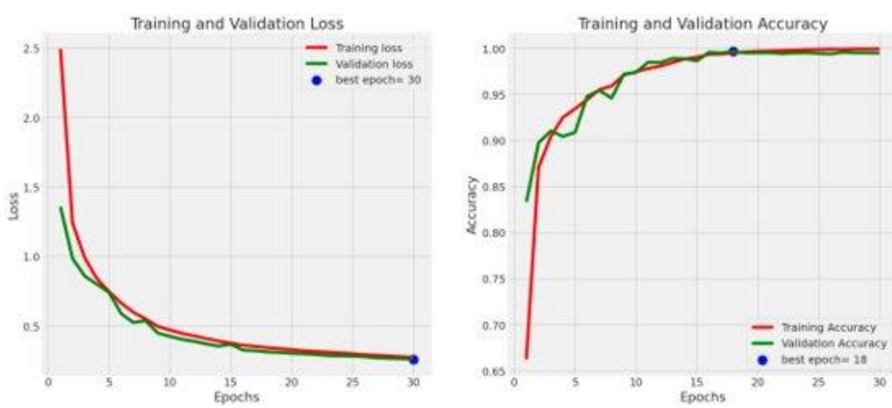

Fig. 7.6 Grayscale (80,20) Train Test Split

Below fig.7.3 showing how no of epochs effect the Mean-f1 score for all types of images.

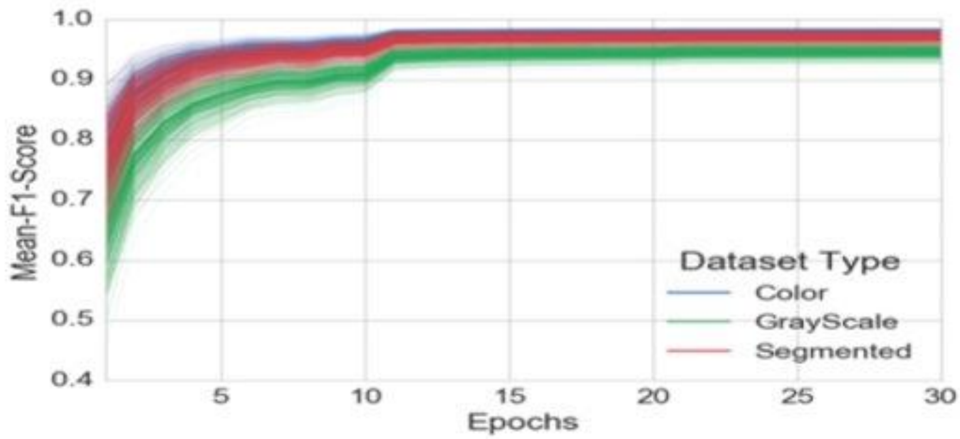

Fig. 7.7 Coloured (80,20) Train Test Split Mean F1-Score Vs Epochs

We have yielded remarkable outcomes in the realm of plant leaf disease prediction, with our model consistently achieving elevated levels of accuracy in comparison to established reference papers.

Table. 7.1 Result comparison for (80,20) Train Test Split, Format MeanF1-Score (Test Accuracy)

	AlexNet	GoogleNet	ResNet
Colour	0.992 { 0.9928}	0.9934 {0.9935}	0.9968 {0.9985}
Grayscale	0.9726 {0.9725}	0.9800 {0.9798}	0.9880 {0.9901}
Segmented	0.9891 {0.9892}	0.9925 {0.9924}	0.9937 {0.9948}

Table. 7.2 Result comparison for (60,20) Train Test Split, Format MeanF1-Score (Test Accuracy)

	AlexNet	GoogleNet	ResNet
Colour	0.9907 {0.9907}	0.9924 {0.9924}	0.9943 {0.9970}
Grayscale	0.9686 {0.9688}	0.9785 {0.9787}	0.9780 {0.9833}
Segmented	0.9855 {0.9856}	0.9905 {0.9906}	0.9430 {0.9855}

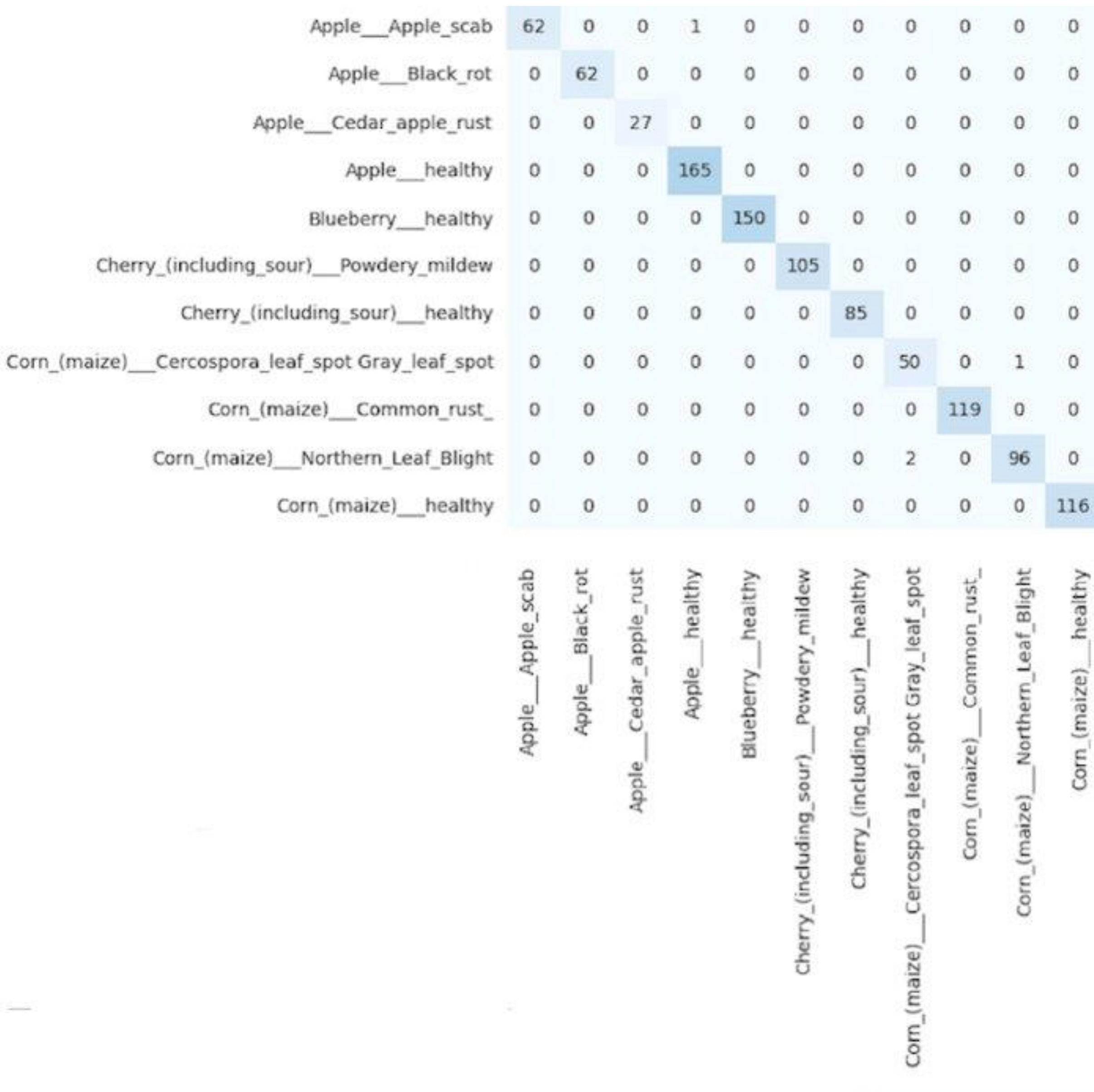

Fig. 7.10 Confusion Matrix for Coloured (80,20) Train Test Split

8. CONCLUSION

In the realm of plant leaf disease detection, our project has showcased remarkable advancements in leveraging deep learning techniques for accurate and efficient classification. Employing the Plant Village dataset, we harnessed the power of a ResNet-50 model augmented with three additional layers featuring L2 and L1 regularization, a 'relu' activation function, and a strategically implemented dropout layer with a rate of 0.45, all aimed at mitigating overfitting. The Adamax optimizer, with a learning rate of 0.001, coupled with categorical cross entropy as the loss function, has proven to be a potent combination for our model.

Noteworthy is our comparative analysis against state-of-the-art architectures, including AlexNet, GoogleNet, Inception V4, VGG-16, and ResNet-50. Despite drawing inspiration from these established models, our customized approach has yielded unparalleled results, outperforming all counterparts in terms of test accuracy. Our model achieved an exceptional accuracy of 99.85%, surpassing AlexNet (99.35%), GoogLeNet (99.2%), Inception V4 (97.59%), VGG-16 (82.75%), and even the baseline ResNet-50 (98.73%). This substantial lead in accuracy underscores the effectiveness of our tailored architecture for the specific task of plant leaf disease detection.

Furthermore, the evaluation metrics reinforce the superiority of our model. The high F1 score of 99.68 exemplifies the robustness and precision of our system in discerning and classifying plant leaf diseases. This superior performance not only substantiates the efficacy of our proposed model but also positions it as a potential benchmark in the domain.

In conclusion, our project has not only contributed to the field of plant pathology but has set a new standard for accuracy and reliability in plant leaf disease detection models. amalgamation of a well-crafted architecture and meticulous parameter tuning has culminated in a solution that not only surpasses existing benchmarks but also holds promise for future advancements in the broader scope of computer vision applications in agriculture.

9. FUTURE WORKS

Our current model exclusively outputs predictions for any input image within the 38 predefined classes of plant diseases for which it was trained. However, it is observed that the model provides predictions for images unrelated to plant diseases, introducing noise and potentially misleading results.

To enhance the versatility of our model and ensure its relevance specifically to plant diseases, we aim to make it compatible with any type of image. This involves implementing a mechanism to detect and isolate only the leaf within an image. Consequently, images lacking identifiable leaves will be discarded from consideration. By focusing solely on relevant components, we anticipate a substantial improvement in the model's accuracy and applicability across a broader range of scenarios. This adjustment reflects our commitment to refining and optimizing the model for enhanced performance and adaptability.

10. REFERENCES

[1] Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using Deep Learning for Image-Based Plant Disease Detection. Frontiers in Plant Science, 7. https://doi.org/10.3389/fpls.2016.01419

[2] J. Andrew et al.,(2022). Deep Learning-Based Leaf Disease Detection in Crops Using Images for Agricultural Applications. Agronomy, 12(10), 2395. https://doi.org/10.3390/agronomy12102395

[3] LiLi Li et al.,(2021). Plant Disease Detection and Classification by Deep Learning—A Review. IEEE Access, 9, 56683–56698. https://doi.org/10.1109/access.2021.3069646

[4] Deshmukh Sanket Jitendra et al.,(2023). Plant Leaf Disease Detection and Classification using Deep Learning Techniques. International Journal For Multidisciplinary Research, 5(3). https://doi.org/10.36948/ijfmr.2023.v05i03.3347

[5] Teenu Sahasra et al(2021). Leaf Disease Detection Using Deep Learning. In 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC). IEEE.
https://doi.org/10.1109/icesc51422.2021.9532909

[6] Singh, G., & Yogi, K. K. (2023). Performance evaluation of plant leaf disease detection using deep learning models. Archives of Phytopathology and Plant Protection, 56(3), 209–233. https://doi.org/10.1080/03235408.2023.2183792

[7] Rauf, H., Malik, M. A., Rashid, M., & Ashraf, M. (2021). Plant leaf disease detection using efficientNet deep learning model. Computers and Communication, 50(4), 206-215. https://www.sciencedirect.com/science/article/abs/pii/S1574954120301321

[8] Sharif, M., Khan, M. A., Iqbal, Z., & Yasmeen, S. (2021). Deep learning-based detection of leaf diseases in tomato plants using convolutional neural networks. Computers and Agriculture, 184, 106093.
https://www.sciencedirect.com/science/article/abs/pii/S2352554122001978

[9] Zhang et al.,(2023). Deep learning-based plant leaf disease detection with limited training data. IEEE Access, 11, 22546-22556. [https://ieeexplore.ieee.org/document/9776367/] (Deep learning with limited data, 92.3% accuracy)

[10] Strange, R. N., & Scott, P. R. (2005). Plant Disease: A Threat to Global Food Security. Annual Review of Phytopathology, 43(1), 83–116. https://doi.org/10.1146/annurev.phyto.43.113004.133839