# Automatic Detection and Recognition of Citrus Fruits Diseases Using Deep Learning Model

Author

Shah Faisal

Registration Number

00000330796

Supervisor

Dr. Kashif Javed

DEPARTMENT OF ROBOTICS AND ARTIFICIAL INTELLIGENCE

SCHOOL OF MECHANICAL & MANUFACTURING ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD

OCTOBER, 2022

# Automatic Detection and Recognition of Citrus Fruits Diseases Using Deep Learning Model

Author

Shah Faisal

Registration Number

00000330796

A thesis submitted in partial fulfillment of the requirements for the degree of

MS Robotics and Intelligent Machine Engineering

Thesis Supervisor:

Dr. Kashif Javed

Thesis Supervisor's Signature: ________________________________________

DEPARTMENT OF ROBOTICS AND ARTIFICIAL INTELLIGENCE

SCHOOL OF MECHANICAL & MANUFACTURING ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD

OCTOBER, 2022

## Declaration

I certify that this research work titled "*Automatic Detection and Recognition of Citrus Fruits Diseases using Deep Learning Model"* is my own work. The work has not been presented elsewhere for assessment. The material that has been used from other sources it has been properly acknowledged / referred.

Signature of Student

Shah Faisal

00000330796

**Plagiarism Certificate (Turnitin Report)**

This thesis has been checked for Plagiarism. Turnitin report endorsed by Supervisor is attached.

Signature of Student

Shah Faisal

00000330796

Signature of Supervisor

## Copyright Statement

## Acknowledgements

I am thankful to my Creator Allah Subhana-Watala to have guided me throughout this work at every step and for every new thought that Allah setup in my mind to improve it. Indeed I could have done nothing without Allah priceless help and guidance. Whosoever helped me throughout the course of my thesis, whether my parents or any other individual was Allah will, so indeed none be worthy of praise but Allah.

I am profusely thankful to my beloved parents who raised me when I was not capable of walking and continued to support me throughout every department of my life.

I would also like to express special thanks to my supervisor Dr. Kashif Javed for his help throughout my thesis and also for the Advanced Digital Design course which he has taught me. I can safely say that I haven't learned any other engineering subject in such depth than the ones which he has taught.

I would also like to pay special thanks to my Co-Supervisor Dr. Saddaf Rubab for her tremendous support and cooperation. Each time I got stuck in something, she came up with the solution. Without her help, I wouldn’t have been able to complete my thesis. I appreciate her patience and guidance throughout the whole thesis.

I would also like to thank my evaluation committee members Dr. Khwaja Fahad Iqbal and Dr. Sara Babar for being on my thesis guidance and for their support and cooperation throughout my course work and research work.

Finally, I am very thankful to my parents for supporting and loving me throughout my life journey. I would also like to express my gratitude to all the individuals, especially to MS scholar Engr. Dil Jan who has rendered valuable assistance to my study.

*Dedicated to my exceptional parents and adored siblings whose tremendous support and cooperation led me to this wonderful accomplishment.*

**Abstract**

In a country's economy, agriculture plays a very vital role. Agriculture's yield and production are reduced by plant diseases, resulting in significant economic losses and instability in the food market. In plants, the citrus fruit crop is one of the most important agricultural products in the world, produced and grown in around 140 countries. It has a lot of nutrients, such as vitamin C. However, due to pests and diseases, citrus cultivation is widely affected and suffers significant losses in both yield and quality. The majority of plant diseases exhibit visible symptoms, and the accepted method used today is for a skilled plant pathologist to detect the diseases by examining affected plant leaves under a microscope, which is a costly and time-consuming method. During the last decade, computer vision and machine learning have been widely adopted to detect and classify plant diseases, providing opportunities for early disease detection and bringing improvements to agricultural production. The early detection and accurate diagnosis of plant diseases are essential for reducing their spread and damage to crops. In this work, we presented an automatic system for early detection and recognition of citrus plant diseases based on a deep learning (DL) model to improve accuracy and reduce computational complexity. The most recent transfer learning-based models were applied to our dataset in order to increase classification accuracy. In this work, we successfully proposed a CNN-based pre-trained model (EfficientNetB3, ResNet50, MobiNetV2, (InceptionV3) for the identification and classification of citrus plant diseases using transfer learning. In order to assess the performance of the model, we found that the transfer of an EfficientNetb3 model led to the highest training, validating, and testing accuracies, which were 99.43%, 99.48%, and 99.58%, respectively. The proposed CNN model exceeds other cutting-edge CNN network architectures developed in earlier literature in the identification and categorization of citrus plant diseases.

**Key Words:** *Citrus Diseases Classification, Deep Learning, Convolutional Neural Network, Transfer Learning, EfficientNetB3, MobiNetV2, ResNet50, Inception*

**Table of Contents**

## List of Figures

## List of Tables

## List of Equations

## List of Tables

| | |
|---|---|
| Artificial Intelligence | AI |
| Machine Learning | ML |
| Deep Learning | DL |
| Computer Vision | CV |
| Artificial Neural Network | ANN |
| Convolutional Neural Network | CNN |
| Rectified Linear Unit | Relu |
| Stochastic Gradient Descent | SGD |
| Batch Size | BS |
| Data Spilit | DS |

# CHAPTER 1: INTRODUCTION

## 1.1. Background and Motivation

A nation's economy's growth and improvement are greatly influenced by agriculture. It is the main source of the world economy. The goal of agricultural research is to rise food quality and production while decreasing costs and improving profitability[1]. Any state's economic development depends on its fruit trees. The citrus tree is one of the most identifiable fruit species of plants. It is rich in vitamin C and popular throughout the Middle East, Africa, and the Indian subcontinent [2]. Citrus plants provide a number of health advantages, and the agricultural industry uses them as a raw goods to create a variety of other agro-based products, such as sweets, jams, ice cream, and confectionary, among others.[2], [3]. The most significant fruit plant in Pakistan is citrus, which contributes significantly to the nation's horticultural exports. In 2018, Pakistan produced an estimated 2.5 million tons of citrus annually[1]. Conversely, citrus plants are prone to a number of diseases, including melanose, black spots, cankers, scabs, and greening. Citrus trees can get the canker, which is primarily located on the leaves or fruit and is extremely contagious. According to statistics, crop losses in Kinnow were about 22%, in sweet oranges 25–40%, in grease 15%, in sweet limes 10%, and in lemons 2%. Every year, a large chunk of strong export citrus is discarded due to signs of citrus fruit illnesses. Therefore, early detection of citrus illnesses has the ability to save costs and losses while also raising the quality of the final product.

## 1.2. Problem Statement

For many years, humans have been the main source of disease identification. The diagnostic and recognition processes are prejudicial, expensive, time-consuming, and prone to mistakes. Additionally, previously unknown regions where there is, by necessity, no local expert experience to treat them may see the emergence of new diseases[4]. Consequently, there is a critical need for an automated approach to identifying citrus plant diseases. It is now simpler to scan and automatically identify abnormalities in a plant in actual time because of the advancement of contemporary instruments and fast computer-based assisted processes. [4]. Traditional machine learning approaches have been quite effective at detecting and diagnosing plant diseases, however they can only handle the sequential image processing tasks: segmenting images with clustering and other approaches [5], [6], feature extraction [7], Support Vector Machine (SVM) [8], K-

Nearest Neighbor (KNN) method [9], and Artificial Neural Network (ANN) [10]. It is difficult to choose and extract the finest visible pathogenic features, demanding the use of highly skilled experts and professionals, which is not only random but also inefficient in terms of manpower and economic support. Instead of having to manually create the structural processes of extracting features and classifications, deep learning can automatically recognize the hierarchical characteristics of diseases. Signal processing [10], pedestrian identification [11], facial recognition [12], road fracture detection [13], biological image analysis [14], and many more areas benefit greatly from the use of deep learning techniques. Additionally, deep learning techniques have shown promise in the agricultural sector, assisting more farmers and workers in the food industry, For example, dealing with image analysis has become necessary in the diagnosis of plant diseases [15], analysis of weeds [16], finding of important seeds [17], insect detection [18], fruit processing [15], and other areas. One of the most effective deep learning techniques is convolutional neural networks (CNN) [19]. To recognize and categorize plant diseases, a number of CNN architectures are used, including AlexNet [20], GoogLeNet [21], and others. In addition, numerous researchers have employed deep learning models to recognize and categorize citrus plant diseases (Pourreza et al. [22], Barman and Ridip et al. [23], Xiaoling et al. [24], and Zia Ur Rehman et al. [25]). A few implementations also seek to predict future characteristics, including yield production [26], weather conditions [27], and field soil water contented [28].

The performance of deep learning models is greatly influenced by the dataset that is available for training. These models provide enhanced results and great generalizability on the appropriately large dataset. The datasets for citrus plant diseases that are currently publicly accessible typically don't have enough pictures in a range of conditions that are essential to build high precision models. The models can over-fit because of the small dataset and badly perform on the test dataset derived from real-world data. To overcome this problem, various data augmentation techniques, including rotation, translation, shifting, flipping, and zooming, are used to enhance the dataset.

### 1.3. Objective

The primary objective of this research is to use deep learning techniques to classify citrus plant diseases at a reduced cost with higher prediction accuracy. The most prominent Machine Learning (ML) method currently used in Deep Learning (DL) models is Transfer Learning (TL), which transfers pre-trained model weights to a new classification problem to optimize computer resources and model building. As a consequence, training becomes simpler and more effective. With a limited number of training datasets, the goal of this work is to apply the concept of transfer learning to the problem of image classification.

The significant contributions of this work bring to the field of research are the following:

1. Different pre-trained models are used to train various models with data-augmentation and without data-augmentation.
2. With the use of Proposed DL model, significant production loss and financial loss can be reduced by early detection of citrus plant diseases.
3. The effectiveness of various models is evaluated using multiperformance metrics for identifying and classifying diseases on citrus plants.

### 1.4. Thesis Overview

Following is a chronological breakdown of the article: Chapter 2 is the body of relevant work. Chapter 3 describes the suggested methodology and implementation. Chapter 4 explained the experimental results, and Chapter 5 presented the conclusion of the work. In the last, Chapter 6 described future work.

# CHAPTER 2: LITERATURE REVIEW

For many years, researchers have been trying to diagnose leaf and fruit diseases. Numerous approaches for identifying and classifying plant diseases have been suggested by researchers in the domain of computer vision and machine learning. Due to its enormous production, the citrus plant is given significant importance in agriculture. To protect citrus from diseases, a number of techniques for the identification and categorization of citrus diseases have been presented.

Many innovative techniques are applied to many other types of crops, such as wheat [29], rice [30], maize [31], and corn [32]. Golhani et al. [33] have reported various studies on neural network methods used for the recognition and categorization of plant diseases.

Citrus canker and Huanglongbing (HLB) were detected using SVM and a fluorescent imaging system by Wetterich et al. [34]. This method had a classification accuracy of 97.8% for citrus canker and scab and a 95% detection accuracy for HLB and zinc deficiency.

Infected portions in pre-processed orange photos have been identified using K-Means segmentation, According to Patel et al. [35], The SVM classifier was used to classify the damaged area's colour, texture, and shape based on information from the training dataset. The accuracy achieved by GLCA models was 67.74 %.

Singh et al. [36] applied SVM, K-Nearest Neighbors, Multi-Layer Perceptron and Linear Discriminant Analysis methods for citrus disease classification. The accuracy achieved for MLP, KNN, SVM and LDA was 81.36 %, 77.12 %, 80.93 %, 84.32 %, respectively.

A conventional image processing method for identifying and classifying citrus plant diseases is suggested by M. Sharif et al. [1]. Features are extracted for segmentation using an optimized weighted segmentation approach. Features are selected using entropy and a PCA score-based vector after combining texture, shape, and color features. A multi-class SVM takes the final features and classify them. On plant village dataset, the suggested method achieved an accuracy of 90.4%.

There is much space for improvement in regards to classification accuracy. Deep learning has recently acquired prominence in a range of fields, including image processing, image recognition, and classification, as well as agriculture. Deep learning is a worthy competitor for classifying citrus diseases because it eliminates time-consuming extraction of features and segmentation based on thresholds.

Xing et al. [37] proposed a detection model for citrus illness and pests using weakly thick connected convolutional network. They applied different CNN models on a citrus self-dataset. The NIN-16 model scored a test accuracy of 91.66% compared to the SENet-16 model's 88.36%.

A strong CNN algorithm was proposed by P. Dhiman [38] as a technique for identifying citrus illnesses. The proposed approach is evaluated with a dense model without data-augmentation or pre-processing techniques. The proposed model's prediction accuracy is 89.1%. Results indicate that data-augmentation and pre-processing approaches had effectively reduced estimates of citrus crop losses.

Citrus Leaf miner, Sooty Mold, and Pulvinaria were identified by M. Khanramaki et al. [39] in order to stop these three pests from spreading. Proposed method was evaluated using 1774 images of citrus leaves. In a test study, The 10-fold cross-validation method was employed to assess CNN accuracy. The results of the studies demonstrated that the suggested model outperformed current CNN methods with an accuracy of 99.04%.

According to Hari et al. [40], convolutional neural network is a useful tool for recognizing diseases in plant species like grapes, maize, tomatoes, and apples. The dataset, which was used to build and evaluate the model, contains a total of 15,210 photos of leaves, separated into 10 groups. The proposed convolutional neural network's accuracy was 86%.

A. Khattak, et al. [41] proposed two convolutional layers of a CNN-based leaf disease identification method. Citrus fruit and leaves are classified according to their vulnerability to disease based on the 1st CNN layer, which extracts minimal-level characteristics from the image, and the 2nd CNN layer, which gathers strong-level characteristics. By classifying citrus fruit and leaf diseases with an accuracy of 95.65%, the proposed CNN model beats comparable models.

MobileNetV2 was trained by Liu et al. [42] to classify and detect six common citrus illnesses. Comparing MobiNetV2 to earlier network models in terms of model accuracy, model validation speed, and model size reveals that it is superior in categorizing and identifying citrus diseases.

Barman et al. [43] evaluated the MobiNetV2 and Self Structured (SSCNN) models of two different CNN architectures to identify diseases in citrus leaves. The best training model precision for MobileNetV2 CNN at epoch 10 was 90%, and the validation accuracy was 92%. However, the SSCNN maximum training model accuracy and maximum cross-validation model accuracy at epoch 12 were 98% and 99%, respectively.

A deep CNN-based method described by Pan et al. [3]. The 2,097 photographs in the collection show diseases like black spot, canker, anthracnose, scab, sand rust, and greening (HLB). Data augmentation techniques were employed to increase the number of datasets that were available for training. For training, cross-validation, and testing purposes, the dataset is divided into three parts in a proportion of (6,2,2). The DenseNet is used to extract and categorize features. In this study, the final dense block has been modified to simplify the DenseNet model. The suggested solutions obtained accuracy of 88 percent.

Zhang et al. [44] suggested a technique for identifying canker illness. The 2nd stage, that is dependent on AlexNet, the optimization objective is modified, and the parameters are updated using Siamese training. The proposed model had a recall of 86.5% and an accuracy of 90.9%.

A deep learning method was used by C. Soini et al. [45] for the detection of citrus HLB disease. The suggested method was capable of identifying HLB positive from HLB negative. The final layer of deep learning inception technique is trained rather than the entire model to reduce training time. With an accuracy of 93.3%, the model can distinguish between HLB negative and HLB positive after 4000 iterations, and in the worst instances, it can do so with an accuracy of around 80%.

V. Kukreja et al. [46] presented a deep learning approach that employs preprocessing and data-augmentation to automatically identify and classify citrus illnesses. In this work, 150 original photos were used, and through data augmentation, these 150 images were extended to 1200 with nine features. Their findings showed that the data preprocessing and augmentation techniques increased the dataset's size and quality, which in turn improved classification accuracy. On the fruits dataset, the proposed approach had an overall accuracy of 89.1%.

By using TL and feature fusion, M. Zia Ur Rehman1 et al. [25] suggested a novel method to classify citrus fruit diseases. The visual quality of input photos was improved using a pre-processing method called hybrid contrast stretching. Two different pre-trained algorithms, the MobileNetv2 and DenseNet201 models, were retrained to produce relevant features using transfer learning. Following the fusion of the relevant features array obtained from the retrained network, and the Whale Optimization Algorithm was employed to obtain a smaller set of optimized features. According to the findings, the individual characteristics derived from MobileNetv2 and DenseNet201 are less accurate at classifying data than the fusion and optimum set of features produced by the two retrained classifiers. The suggested model performs better than current methodologies, with accuracy of 95.7%.

A. Elaraby et al. [47] analyzed and compared the effectiveness of SGDM optimization techniques for transfer learning-based automated citrus disease recognition . Two common models, AlexNet and VGG19, were investigated for extracting characteristics from the photos. The datasets for citrus fruit and leaf disease were utilized to achieve the highest classification accuracy, which was 94.3%, which was used to evaluate network performance. In light of the results, they concluded that deep learning was a more developed methodology than other approaches.

This section discusses different methods for classifying a number of citrus diseases. The classification and detection approaches are based on deep learning and conventional image processing. Deep learning approaches have gained popularity in recent years since they are less complicated than image processing methods and produced better results in terms of accuracy.

# CHAPTER 3: METHODOLOGY AND IMPLEMENTATION

Deep neural networks have been increasingly popular in recent years for autonomous disease detection on citrus fruits. We provide a brief overview of the suggested framework for the usage of DL and image processing in the detection and categorization of diseases in citrus plants. Figure 3.1 shows the general scheme of our proposed deep learning models, which includes the input dataset stage, preprocessing stage, the DL model stage, TL stage, diseases classification stage, and model performance assessment stage.

The proposed methodology consists of four main modules:

- Dataset Collection
- Data Augmentation and Image pre-processing
- Data Splitting
- Proposed Architecture.

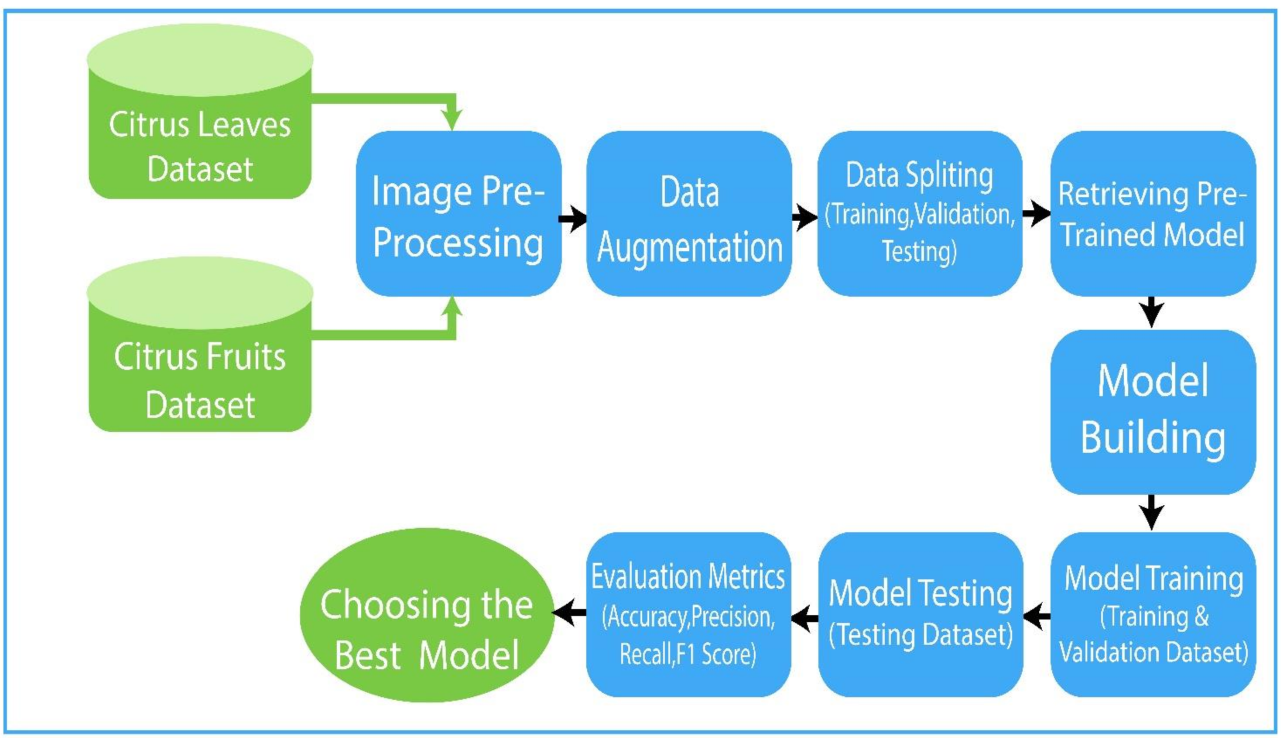


Figure 3.1: General Schematic of Proposed System

### 3.1. Dataset Collection

For deep learning algorithms to effectively learn, a lot of data is needed. In this work, a sample of photos from the Citrus Plant Dataset was utilized [48]. The set of data includes 759 pictures of citrus fruits and leaves, both healthy and unhealthy. With the assistance of a citrus disease domain expert, each image was manually captured using a DSLR from the Sargodha region of Pakistan. All photos had a resolution of 72 dpi, with 256 pixels for width and 256 pixels for height. Four different citrus fruit and leaf diseases were identified from the affected photos. We focused on the diseases Black spot, Scab, Canker, Melanose, and Greening in data sets. Table 3.1 lists the details of the dataset for each type of disease. The samples of Citrus Leaves Diseases and the samples of Citrus Fruits Diseases are presented in Figure 3.2 and Figure 3.3.

| **Citrus Leaves Diseases** | **Images** | **Citrus Fruits Diseases** | **Images** |
|---|---|---|---|
| Black Spot | 171 | Black Spot | 19 |
| Greening | 204 | Greening | 16 |
| Canker | 163 | Canker | 78 |
| Melanose | 19 | Scab | 15 |
| Healthy | 58 | Healthy | 22 |
| Total | 609 | Total | 150 |

Table 3.1: Citrus Plant Dataset

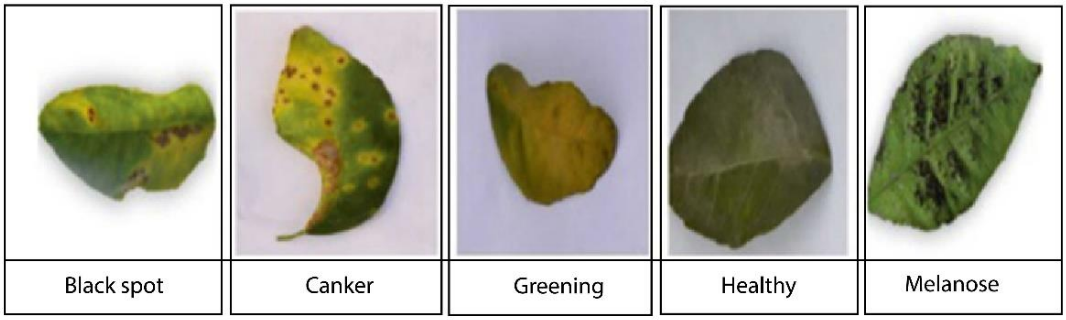


Figure 3.2: Citrus Leaf Diseases


Figure 3.3: Citrus Fruit Diseases

### 3.2. Dataset Augmentation and Resizing

For effective training of DL networks, a huge volume of training data is necessary. Regrettably, the availability of accurate annotated ground truths, the quantity and rarity of currently available citrus disease image collections, and other factors still make it difficult for citrus diseases to be automatically diagnosed. The over-fitting problem that might happen when using a little quantity of training data during the training stage was eliminated by executing augmentation operations on the training set to increase the training photos. Various data augmentation techniques have been used for a variety of effects, including translating, flipping, rotating at various angles, shifting vertically and horizontally, and zooming. Data augmentation can also be used to generate multiple variations of a single photo, as shown in Figure 3.4. There are initially 759 original images but after data augmentation, we acquire 3,383 images, as shown in Table 3.2.

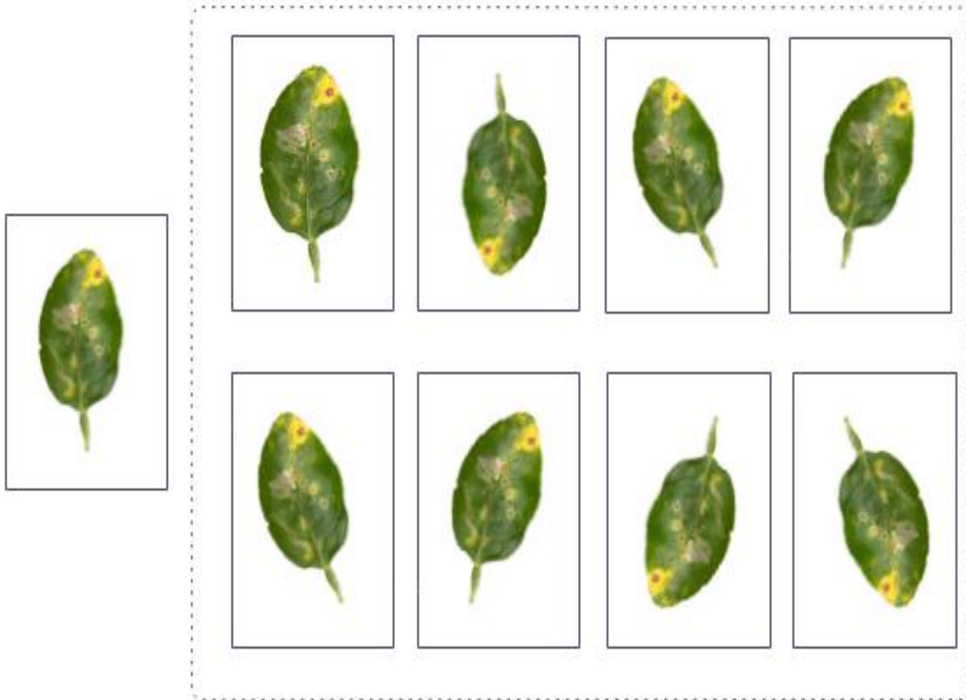

Figure 3.4: Data Augmentation

| Categories | Diseases | Original images | After Data Augmentation |
|---|---|---|---|
| **Leaves** | Black Spot | 171 | 740 |
| | Greening | 204 | 504 |
| | Canker | 163 | 368 |
| | Melanose | 19 | 210 |
| | Healthy | 58 | 369 |
| | | | |
| **Fruits** | Black spot | 19 | 168 |
| | Greening | 16 | 255 |
| | Canker | 78 | 323 |
| | Scab | 15 | 255 |
| | Healthy | 22 | 191 |
| **Total Images** | | 759 | 3383 |

Table 3.2: Augmented Citrus Plant Dataset

### 3.3. Dataset Distribution

The dataset is split into three sections, as shown in Figure 3.5: (i) Training Data, (ii) Validation Data, and (iii) Testing Data. Training, Validation, and Testing are done in batch sizes of 32 and data split (80, 10, 10) respectively.

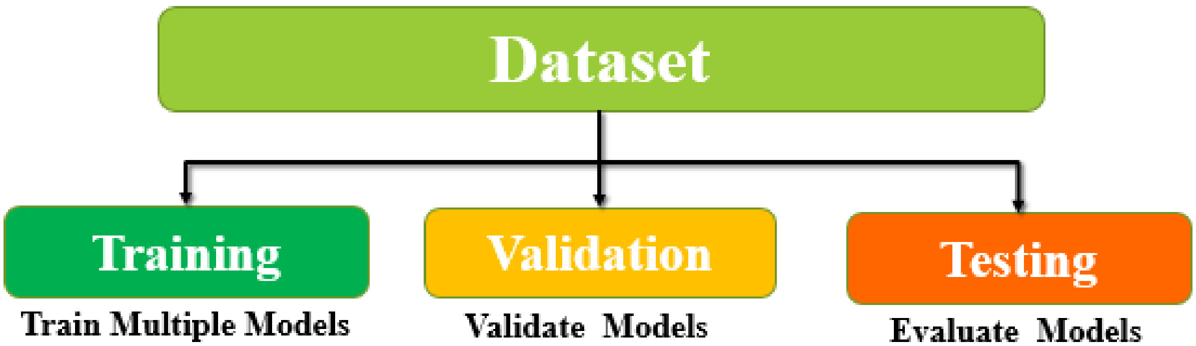


Figure 3.5: Data Distribution

#### *3.3.1. Training Data*

A different deep learning model based on CNN is built utilizing 80% of the training data, although this ratio may alter according to the requirements of the project. The multiple models, which attempts to learn from the training sample, is trained using this data. The training dataset consists both the input and the desired output.

#### *3.3.2. Validation Data*

The validation data is 10% of the original dataset and it is used to validate different CNN based models performance during training. The information obtained from this validation approach can be used to modify the model's hyperparameters and configurations. It works similarly to a critic informing us of the direction our training is heading towards. To avoid overfitting, we split the dataset into a validation dataset. Additionally ranking the model's accuracy and help towards model selection.

### *3.3.3 Testing Data*

The CNN based different models are tested on new data using a test set that represents 10% of the original data. Once the model has received the necessary training, it is used for the evaluation process. It offers a concluding model performance evaluation metric in terms of accuracy precision, recall and F1-Score. Simply put, it offers a response to the question "How effectively does the model work with unseen data?"

## 3.4. Proposed Architecture

This study proposes a unique approach for identifying and classifying citrus fruit diseases. The fundamental architecture consists of four main steps: (a) data acquisition (b) data-augmentation and preprocessing (c) deep CNN feature extraction (d) final classification with a cutting-edge classifier. Figure 3.6 shows a detailed flow of the suggested framework.

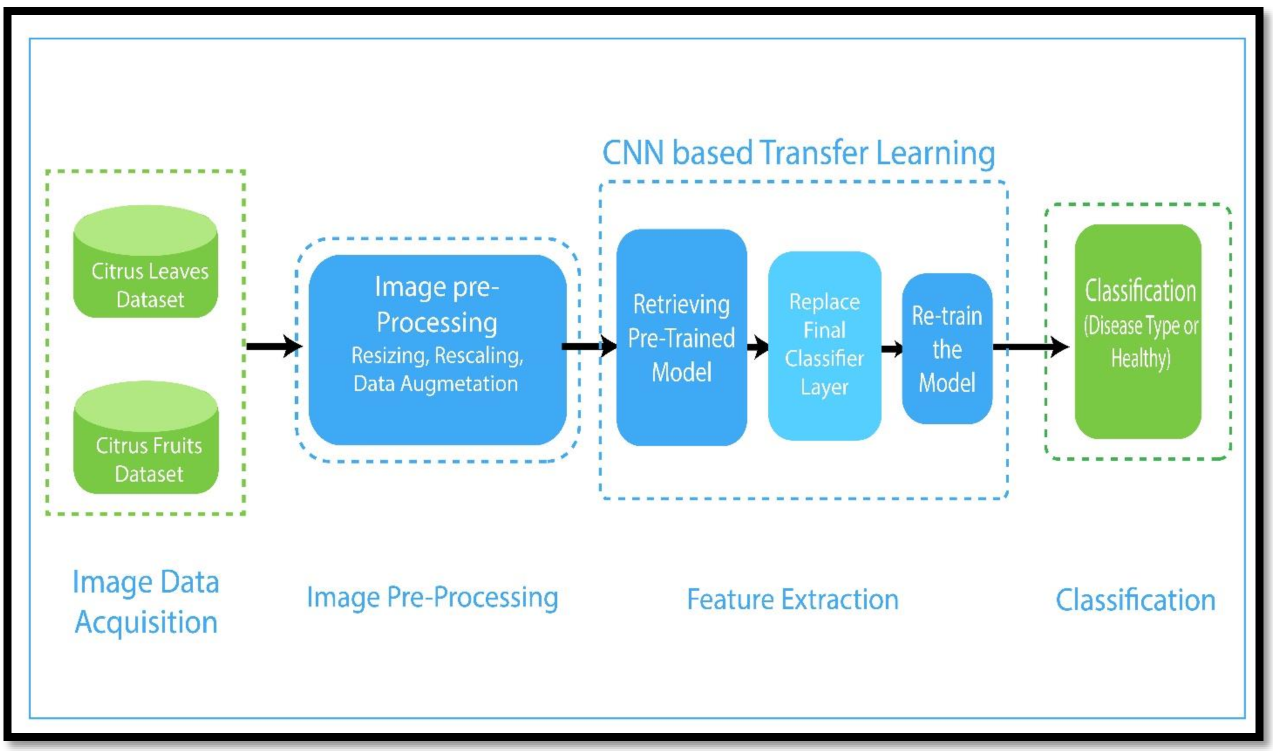


Figure 3.6: Detailed architecture of Transfer Learning based citrus plant disease classification.

### 3.4.1 Feature Extraction Using CNN based Deep Transfer Learning

Four pre-trained CNN architectures (EfficientNetB3, ResNet50, MobiNetV2 and InceptionV3) are used in this work after applying transfer learning, as shown in Figure 3.7.

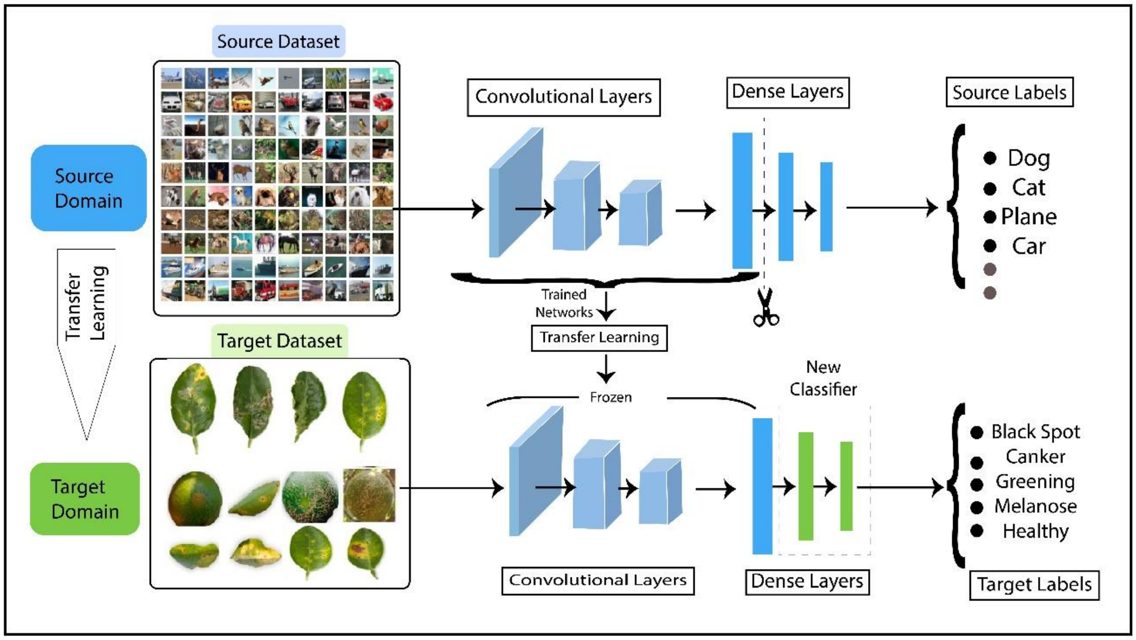


Figure 3.7: Representation of DCNN feature extraction using Transfer Learning

#### *3.4.1.2 Transfer Learning*

TL, which transfers pre-trained model weights to a new classification problem, is a popular deep learning technique. Training consequently becomes easier and more effective. The main benefit of transfer learning is the detection and classification of citrus diseases using pre-trained models like EfficientNetb3, MobiNetV2, ResNet50, and InceptionV3.

#### *3.4.1.2. InceptionV3:*

InceptionV3 main objective is to use less computational resources by changing the Inception architectures that were first introduced in GoogLeNet/InceptionV1 [49]. The Inception V3 model has 48 layers. The network is made up of 11 Inception modules in total, covering five different types. Each module has a convolutional layer, an activation layer, a pooling layer, and a batch normalization layer that are all developed by experts. These modules are combined in the Inception-v3 model to extract the most features possible. The concept of multi-scale is used by inception modules. Each module contains several branches with various kernel sizes (1 x 1, 3 x 3, 5 x 5 and 7 x 7). These filters take different scales of feature maps, extract them, and combine them before sending the result to the following step; before using the computationally complex 3 x 3 and 5 x 5 convolutions, each inception module uses 1 x 1 convolutions to decrease dimensionality. By factoring the (5 x 5), (7 x 7) or asymmetric (1 x 7, 7 x 1) convolutions into smaller (3 x 3) or asymmetric (1 x 7, 7 x 1) convolutions, many DNN parameters are decreased. The general network architecture of InceptionV3 model is shown in Figure: 3.8

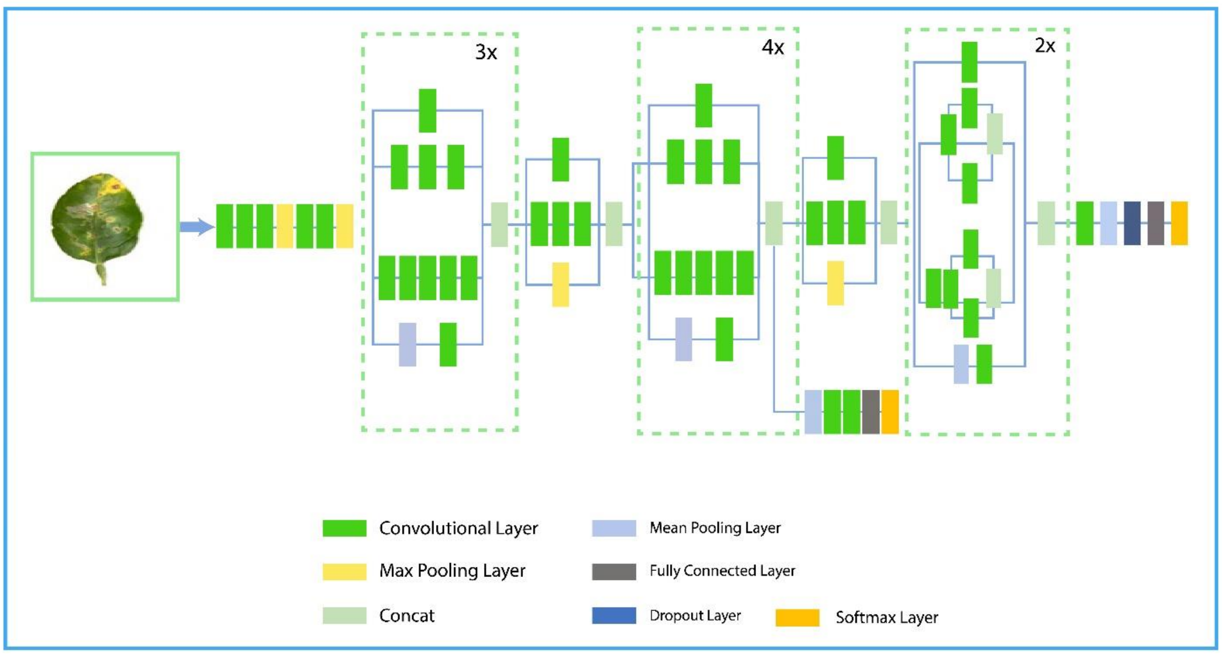


Figure 3.8: InceptionV3 Model Network Architecture

***3.4.1.3. ResNet50:***

A novel architecture known as ResNet [50] was presented in 2015 by scientists at Microsoft Research. To solve the gradient vanishing/exploding problem, this architecture includes the idea of residual blocks. They employ the skip connections method in this network. The skip connection connects layer activations to those of next layers by skipping a few layers in between. This results in a residual block. ResNets are built by stacking these residue blocks. The strategy behind this network is to let the network fit the residual mapping rather than have layers learning the underlying mapping. The ResNet-50 is a 50-layer ResNet variation as shown in Figure 3.9, that was trained using at least one million photos from the ImageNet dataset.

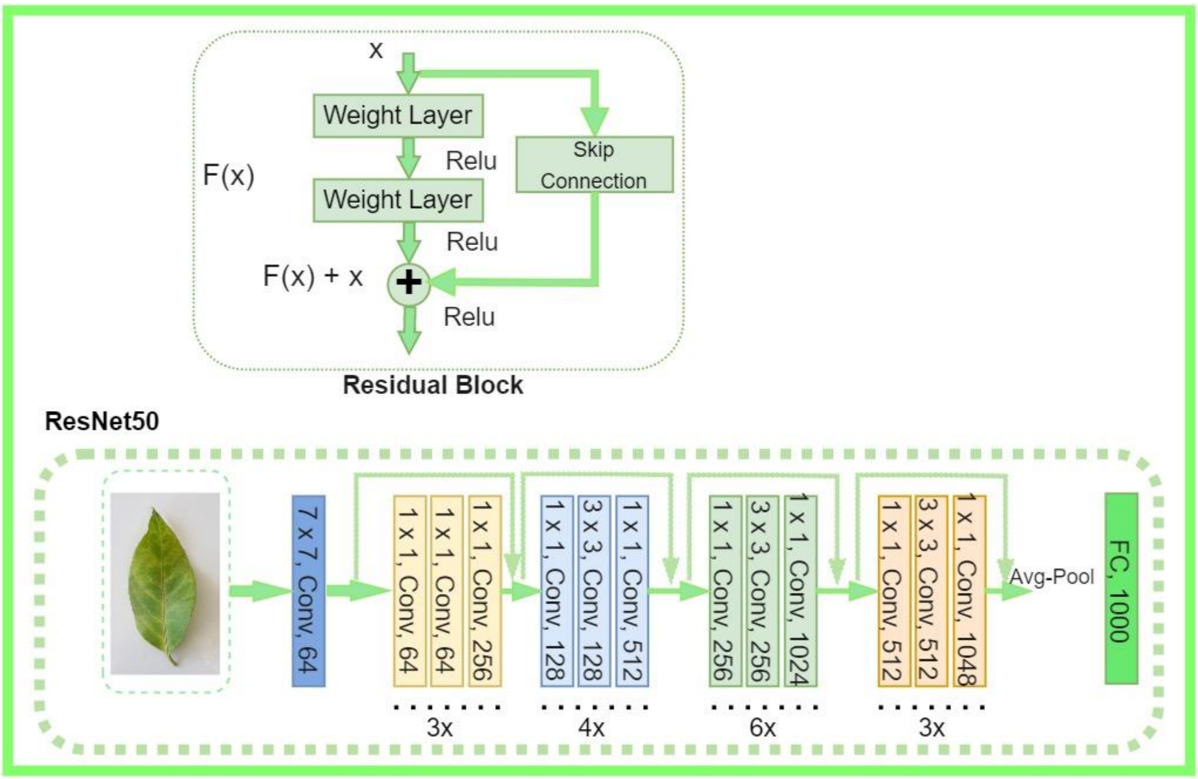


Figure 3.9: ResNet50 Model Network Architecture

### *3.4.1.4. MobiNetV2:*

Google researchers initially presented MobileNet [51]. A convolutional neural network design called MobileNetV2 aims to function well on mobile devices. The ImageNet dataset, a sizable classification dataset, served as the first training ground for the MobileNetv2 model. The bottleneck layers are linked by residual connections, and it is constructed on an inverted residual structure. The middle expansion layer uses simple point-wise and depth-wise convolutions as a source of non-linearity to select characteristics. The MobileNetV2 architecture consists of 19 extra bottleneck layers in addition to the 32-filter initial fully convolution layer. MobileNetv2 is retrained on the citrus plant dataset using the pre-trained weights from ImageNet to hasten feature learning. The general archeticture of MobileNetV2 is illustrated in Figure 3.10.

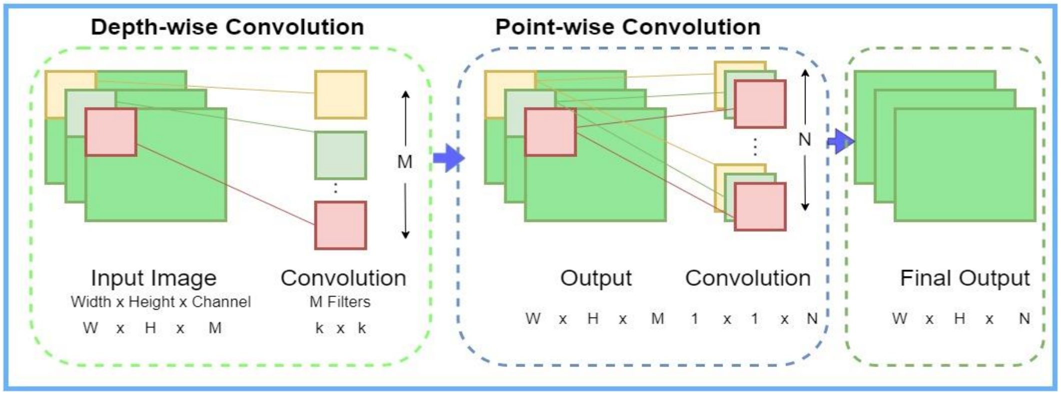


Figure 3.10: MobiNetV2 Network General Architecture.

**3.4.1.5. EfficientB3:**

Tan and Lee originally suggested EfficienNet in 2019 [52], and it is an architecture for enhancing classification networks. The majority of networks typically use three indicators: network expansion, network depth, and improvement in resolution quality. In order to increase the accuracy, the network's width, depth, and resolution are tuned using the combined scaling model as shown in Figure 3.11. By contrasting EfficientNets with other CNNs models that are trained on ImageNet dataset. The EfficientNets model can typically outperform previous CNNs models in terms of accuracy and efficiency.

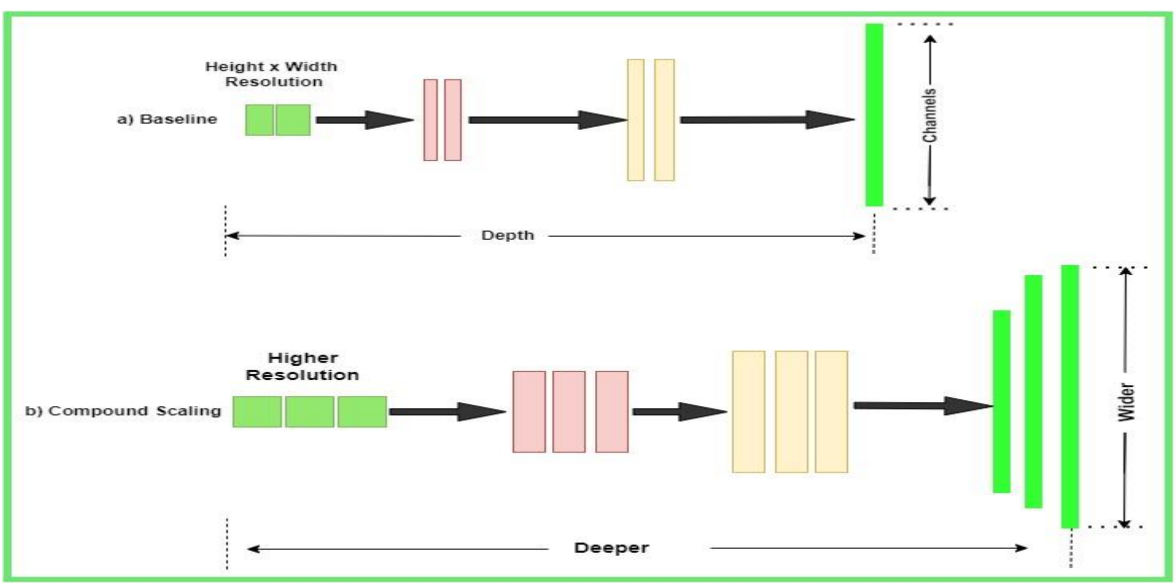


Figure 3.11: Compound Scaling Method used in Network Architecture of EfficientB3 Model.

The pre-trained models utilized in this study was previously trained using an ImageNet dataset. By default, each pre-trained network on the CNN building have thousand fully connected (FC) layer output nodes. The output FC layer was replaced with five nodes based on the number of classes in the dataset for citrus plant diseases and added with softmax activation as shown in Figure 3.7. The flowchart of the entire implementation with algorithm is shown in Figure 3.12-3.13.

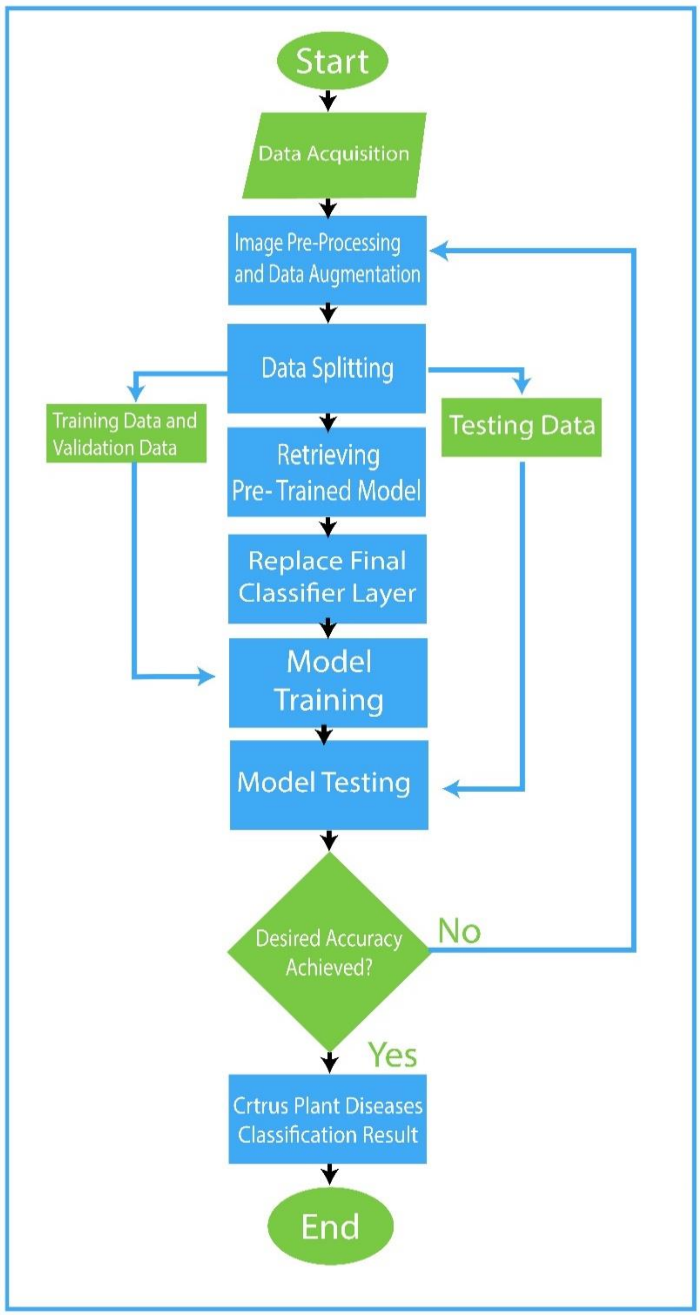


Figure 3.12: Flowchart of our proposed model.

***START***

1. ***Input:*** *Citrus Plant Dataset (Image Files, Class Names)*
2. ***Output:*** *Disease Classification with high prediction accuracy.*
3. *BATCH_SIZE = 32, IMAGE_SHAPE = 256, num_classes = 5;*
4. *dataset = dataset_from_directory("Path of directory ")*
5. *Apply Image Pre-Processing and Data Augmentation*
6. *Def get_dataset_partitions_tf(ds, train_split=0.8, val_split=0.1, test_split=0.1)*
7. *Load Pre-Trained Model*
8. *Freeze features learning layers and add new classifier layer*
9. ***FOR*** *All training examples* ***DO***
10. *Re-Trained the model with a new classifier on training data and validation data*
11. *Test the model with Test Data*
12. ***IF*** *the desired accuracy achieved* ***THEN***
13. *Show citrus plant diseases classification results.*
14. *Show Accuracy, Loss, Precision, Recall, F1-Score*
15. ***ELSE***
16. *Go back to step number 5.*
17. ***END***
18. ***END***

***END***

Figure 3.13: Algorithm of our proposed model.

## CHAPTER 4: EXPEREMENTAL RESULTS AND DISCUSSION

Each experiment's findings are discussed in this section. The CNN based InceptionV3, ResNet50, MobileNetV2, and EfficientNetB3 architectural models were used in experiments using training and validation data. This experimental result aims to measure the accuracy, loss, and computation time needed for each architecture classifier throughout the data training phase. For each CNN based model, Table 4.1 shows the accuracy, loss, and training computational time.

| Pre-Train Model | Data Augmentation | Training Accuracy (%) | Validation Accuracy (%) | Testing Accuracy (%) | Training Loss (%) | Validation Loss (%) | Testing Loss (%) | ETA |
|---|---|---|---|---|---|---|---|---|
| EfficientNetB3 | Un-Augmented Dataset | 97.14 | 96.88 | 92.78 | 0.13 | 0.14 | 0.237 | 19 sec |
| | Augmented Dataset | 99.43 | 99.48 | 99.58 | 0.021 | 0.019 | 0.015 | 47 sec |
| MobiNetV2 | Un-Augmented Dataset | 95.83 | 92.19 | 93.81 | 0.15 | 0.24 | 0.191 | 6 sec |
| | Augmented Dataset | 97.93 | 98.44 | 97.91 | 0.060 | 0.049 | 0.057 | 10 sec |
| ResNet50 | Un-Augmented Dataset | 96.35 | 96.88 | 89.69 | 0.15 | 0.14 | 0.39 | 19 sec |
| | Augmented Dataset | 98.39 | 98.44 | 98.74 | 0.048 | 0.040 | 0.044 | 47 sec |
| GoogleNet (InceptionV3) | Un-Augmented Dataset | 90.10 | 87.50 | 92.97 | 0.32 | 0.36 | 0.23 | 19 sec |
| | Augmented Dataset | 96.39 | 97.40 | 96.23 | 0.109 | 0.104 | 0.092 | 48 sec |

Table 4.1. Accuracy, Loss and time computing of CNN based models.

## 4.1. Models Result Comparison

Experiments show that EfficientNetB3 outperforms all CNN based models in terms of performance, with a training accuracy of 99.43%. ResNet50 model ranked second with accuracy of 98.39%, followed by MobileNetV2 with accuracy of 97.93%, and InceptionV3 with accuracy of 96.39%.

Table 4.1 also shows the training loss for every learned CNN model. With a value of 0.021, EfficientNetB3 is the architecture with the lowest training loss, followed by ResNet50, MobileNetV2, and InceptionV3 with values of 0.048, 0.060, and 0.109, respectively. The accuracy and loss curves achieved during learning phase are shown in Figures 4.1–4.4.

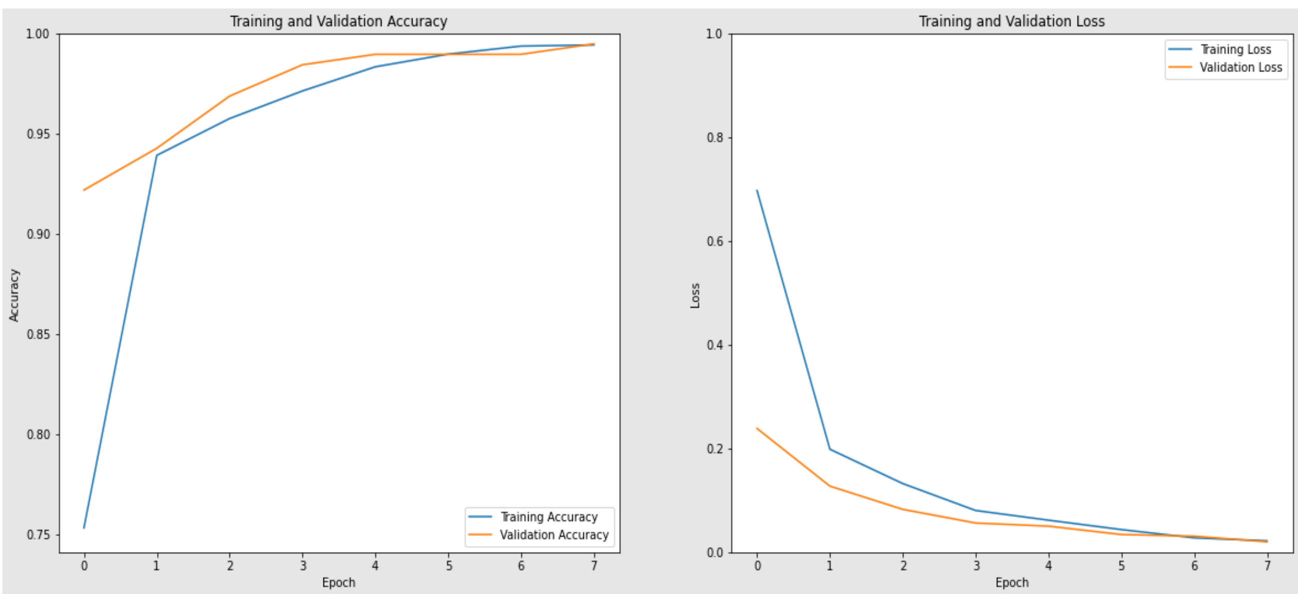


Figure 4.1: Training and Validation Accuracy and Loss graph of the EfficientNetB3 model

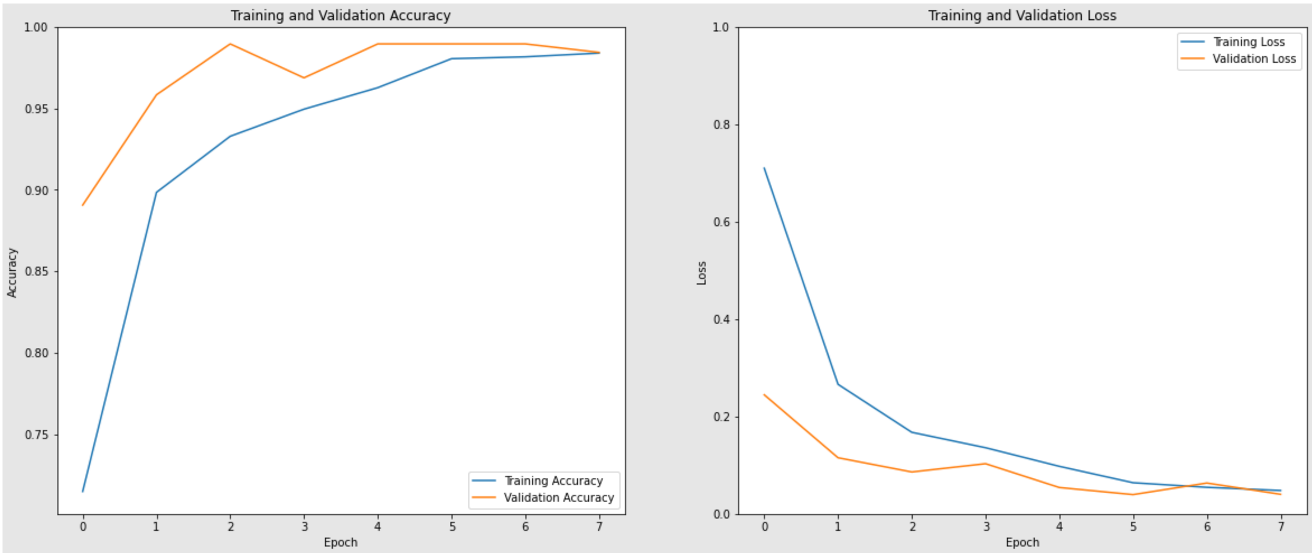


Figure 4.2: Training and Validation Accuracy and Loss graph of the ResNet50 model

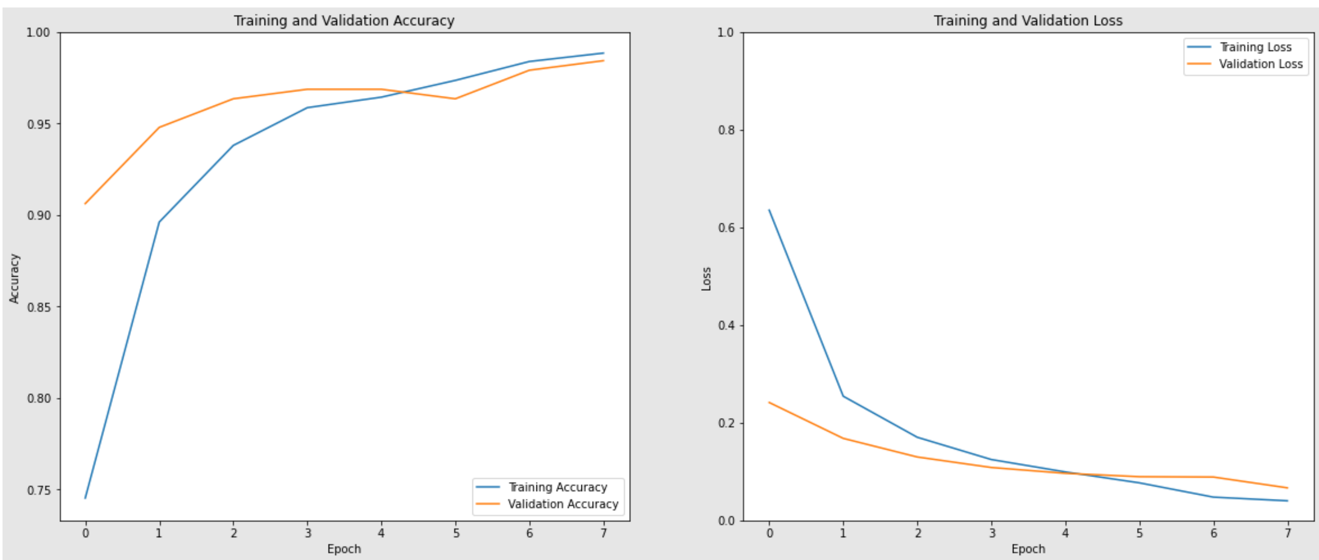


Figure 4.3: Training and Validation Accuracy and Loss graph of the MobiNetV2 model

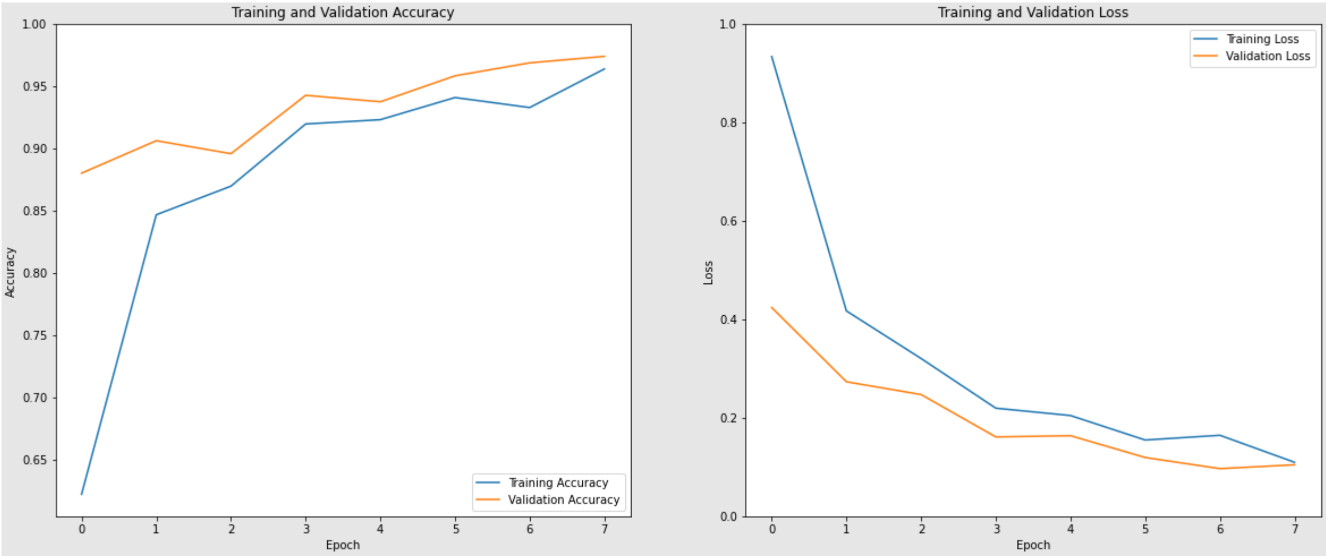


Figure 4.4: Training and Validation Accuracy and Loss graph of the InceptionV3 model

**4.2. Models Performance Evaluation:**

Table 4.2 displays the performance evaluation matrix between the trained network and the test dataset, including each model's accuracy, precision, recall, and F1-Score.

$$\boldsymbol{Accuracy = TP + TN/(TP + TN + FP + FN\ ) - - - - (i)}$$
$$\boldsymbol{Precision = TP/(TP + FP) - - - - (ii)}$$
$$\boldsymbol{Recall = TP/(TP + FN) - - - - (iii)}$$
$$\boldsymbol{F1 - Score = (2 * Precision * Recall)/(Precision + Recall) - - - - (iv)}$$

Where TP = True Positive, TN = True Negative, FP = False Positive, and FN = False Negative.

The test results showed that EfficientNetB3 performed better than all CNN architecture models in regards to accuracy (99.58%), precision (100%), recall (100%), and F1-Score (100%). The ResNet50 architecture came in second with accuracy values of 98.74, precision of 96.00%, recall of 95.00%, and F1-score of 96.00%. Accuracy, precision, recall, and F1-Score for InceptionV2 are all 96.23%, 97.00%, and 97.00%, respectively, whereas MobileNet receives accuracy values of 97.91%, precision 97.00%, recall 98.00%, and F1-Score 99.00%. Figure 4.5 shows the performance evaluation of each model.

| Pre-Trained Model | Data Augmentation | Precision | Recall | F1- Score | Accuracy (%) |
|---|---|---|---|---|---|
| EfficientNetB3 | Un-Augmented Dataset | 0.91 | 0.94 | 0.91 | 92.78 |
| | Augmented Dataset | 1.00 | 1.00 | 1.00 | 99.58 |
| MobiNetV2 | Un-Augmented Dataset | 0.93 | 0.91 | 0.91 | 93.81 |
| | Augmented Dataset | 0.97 | 0.99 | 0.98 | 97.91 |
| ResNet50 | Un-Augmented Dataset | 0.85 | 0.88 | 0.86 | 89.69 |
| | Augmented Dataset | 0.96 | 0.95 | 0.95 | 98.74 |
| GoogleNet (InceptionV3) | Un-Augmented Dataset | 0.91 | 0.91 | 0.91 | 92.97 |
| | Augmented Dataset | 0.97 | 0.97 | 0.97 | 96.23 |

Table 4.2: Performance Evaluation of trained models with testing data

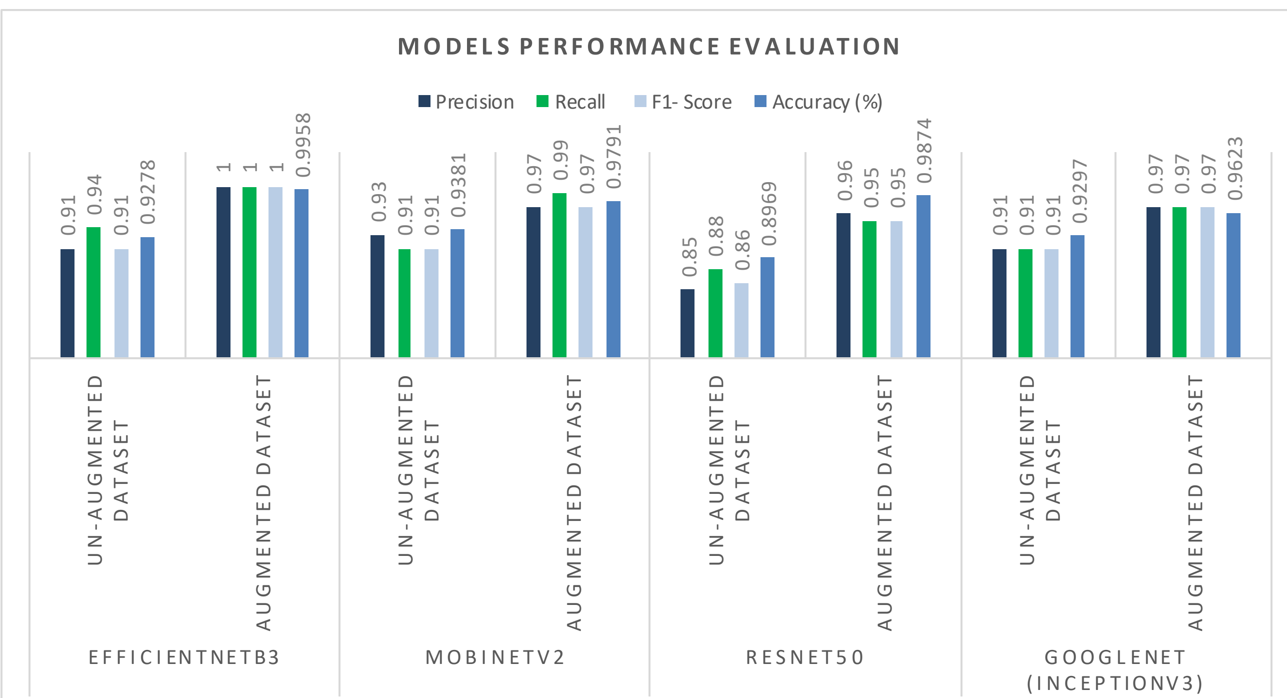


Figure 4.5: Models Performance Evaluation

### 4.3. Confusion Matrix of Models

The confusion matrix with data testing for the EfficientNetB3 architectural model is shown in Figure 4.6. No data from the sample tested were incorrectly categorized as shown in Figure 4.6, all data were correctly categorized. Table 4.2 illustrates the accuracy, precision, recall, and F1 Score values.

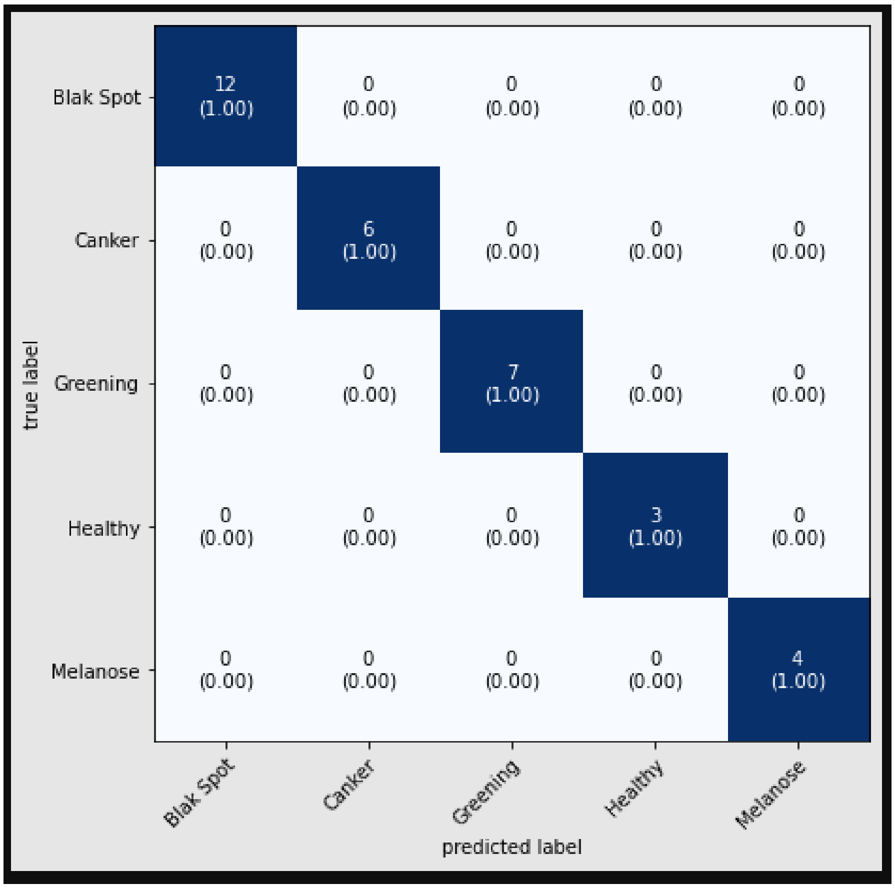


Figure 4.6: Confusion matrix of EfficientNetB3

Figure 4.7 shows the confusion matrix on testing data for the ResNet50 architecture. There are 2 sample data that are incorrectly classified out of the 32 sample data tested. As seen in Figure 4.7, a sample of two Greening class data points were incorrectly classified.

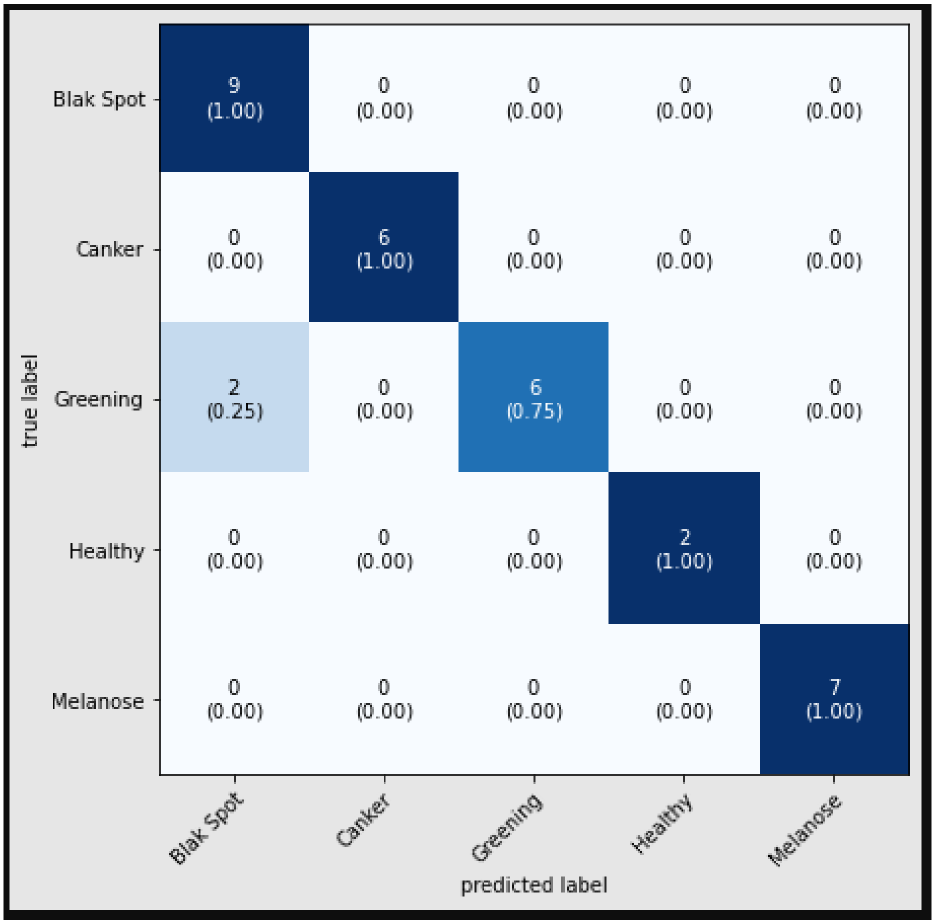


Figure 4.7: Confusion matrix of ResNet50

Figure 4.8 displays the confusion matrix for the MobileNet architecture model on testing data. One sample of the 32 tested data were incorrectly categorized. As shown in Figure 4.8, one sample of incorrectly classified Greening class data.

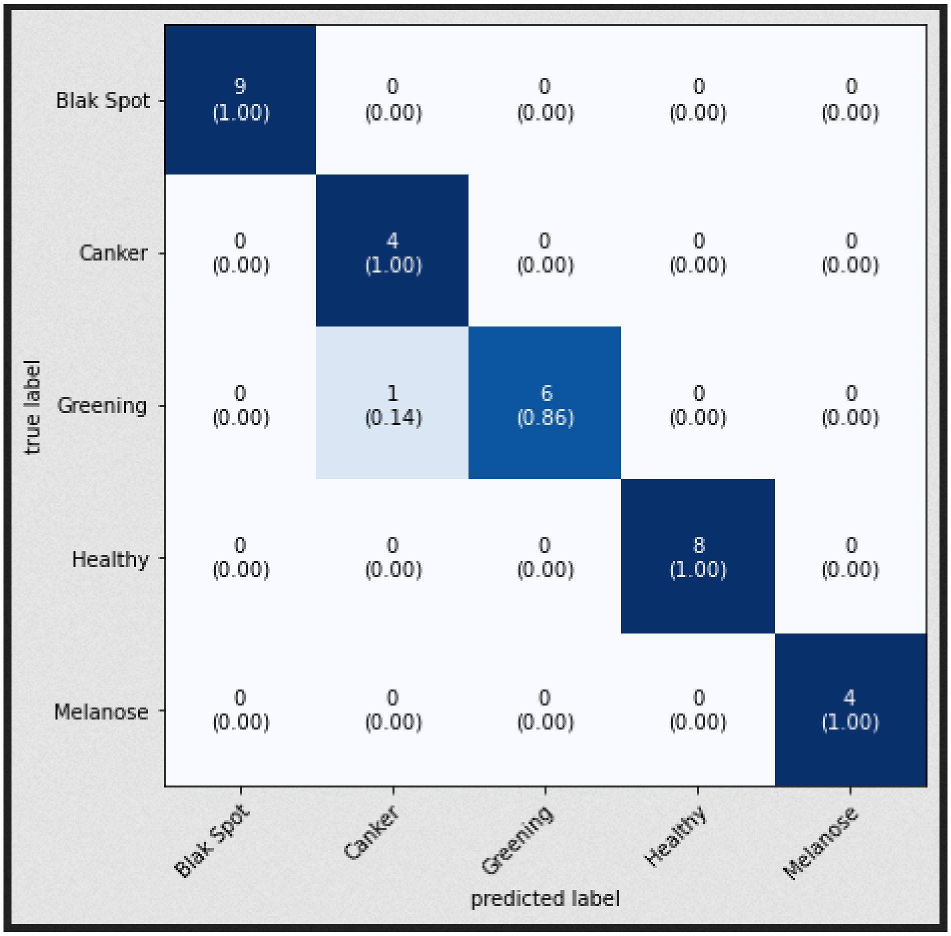


Figure 4.8: Confusion matrix of MobiNetV2

As for the InceptionV3 architectural model, Figure 4.9 presents the confusion matrix with data testing. One sample data is incorrectly categorized out of the 32 sample data analyzed. Figure 4.9 shows 1 sample of incorrectly classified Greening class data.

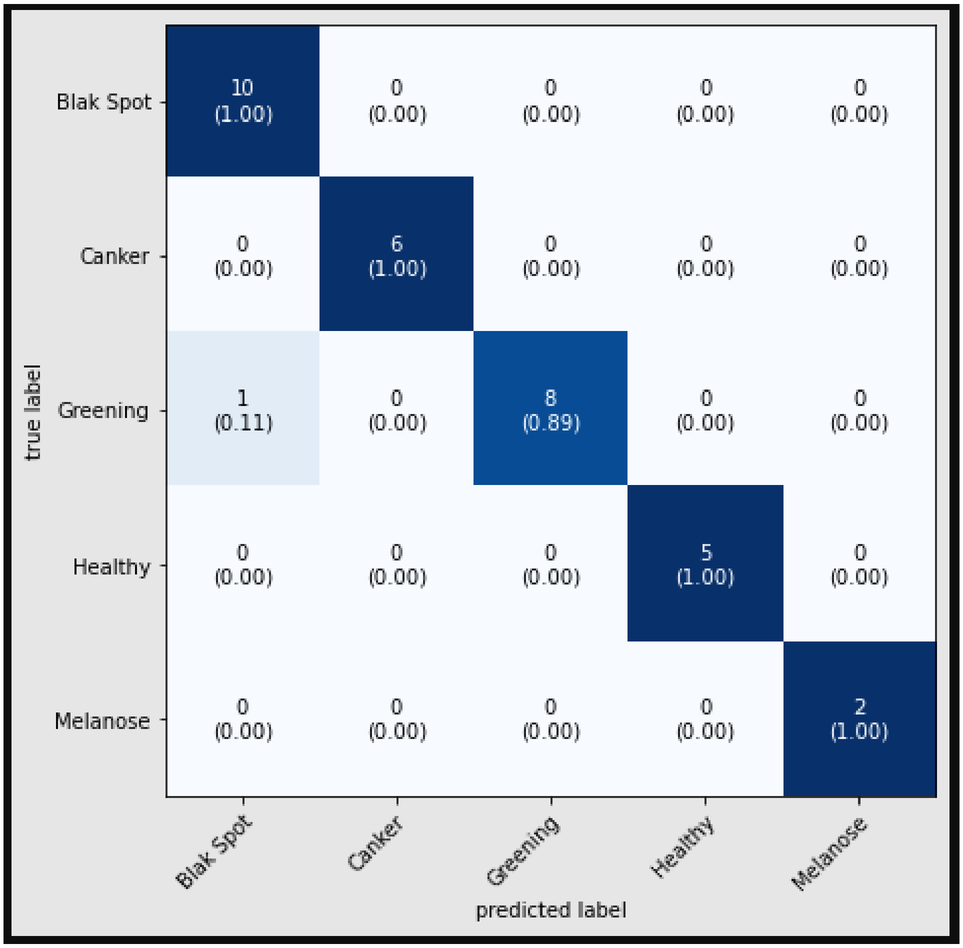


Figure 4.9: Confusion matrix of InceptionV3

## 4.4. Models Testing Accuracy and Loss Comparison

Figure 4.10-4.11 compares the testing accuracy and loss of the EfficientNetB3, MobiNetV3, ResNet50, and InceptionV3 models on augmented and non-augmented dataset. Figure 4.10-4.11 shows that EfficientNetB3 gives the highest performance, with 99.58% testing accuracy and 0.021% loss, which is maximal compared to other models.

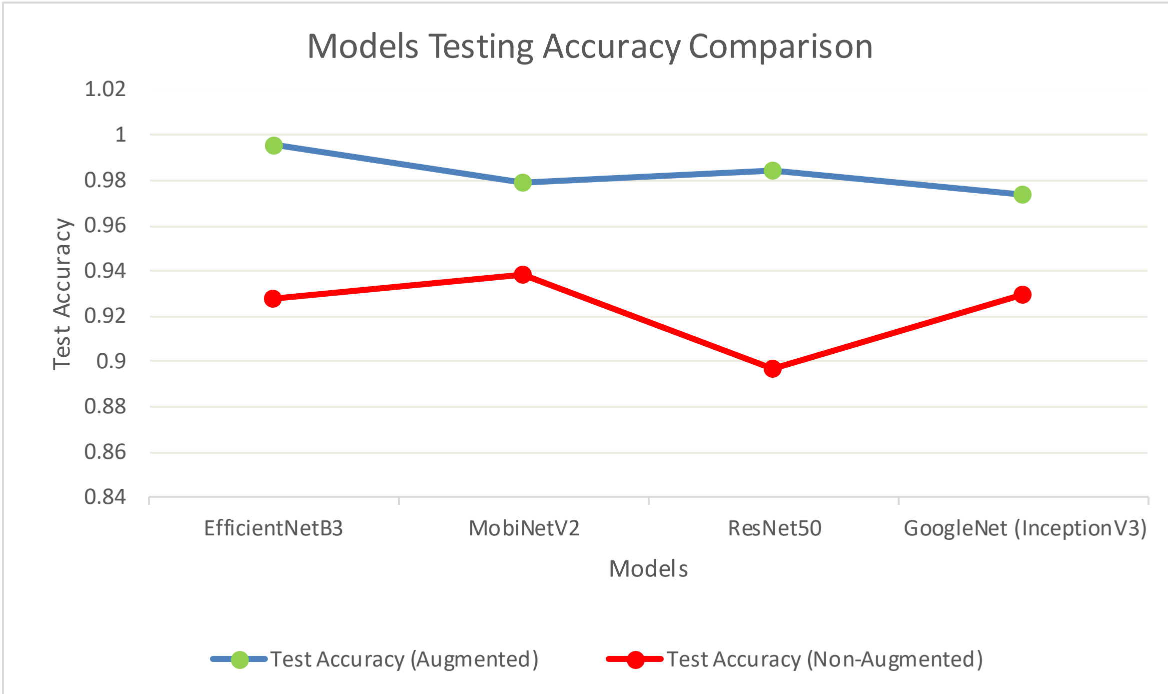


Figure 4.10: Models Testing Accuracy Comparison on Test Data

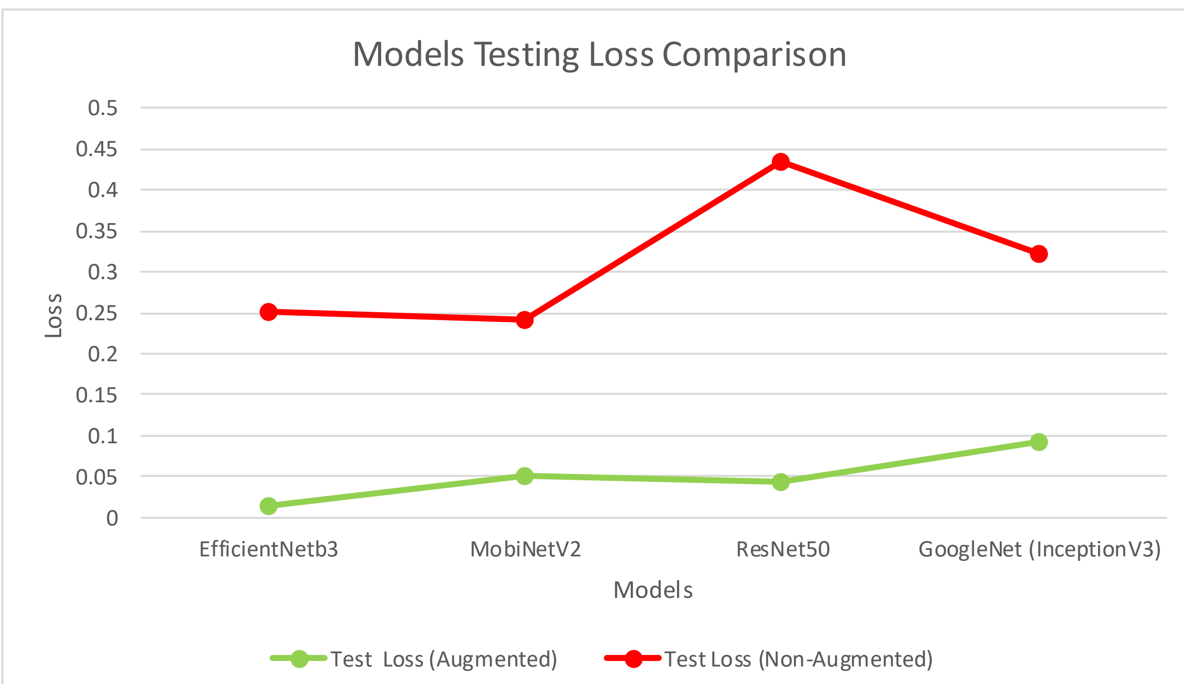


Figure 4.11: Models Testing Loss Comparison on Test Data

With an accuracy of 99.58%, Figure 4.11 demonstrates that the model created using EfficientNetB3 is the most accurate for identifying and classifying citrus plant diseases.

The trained models are also tested on single image and the batch size of 32. Figure 4.12 illustrates the real-time results of the proposed system.

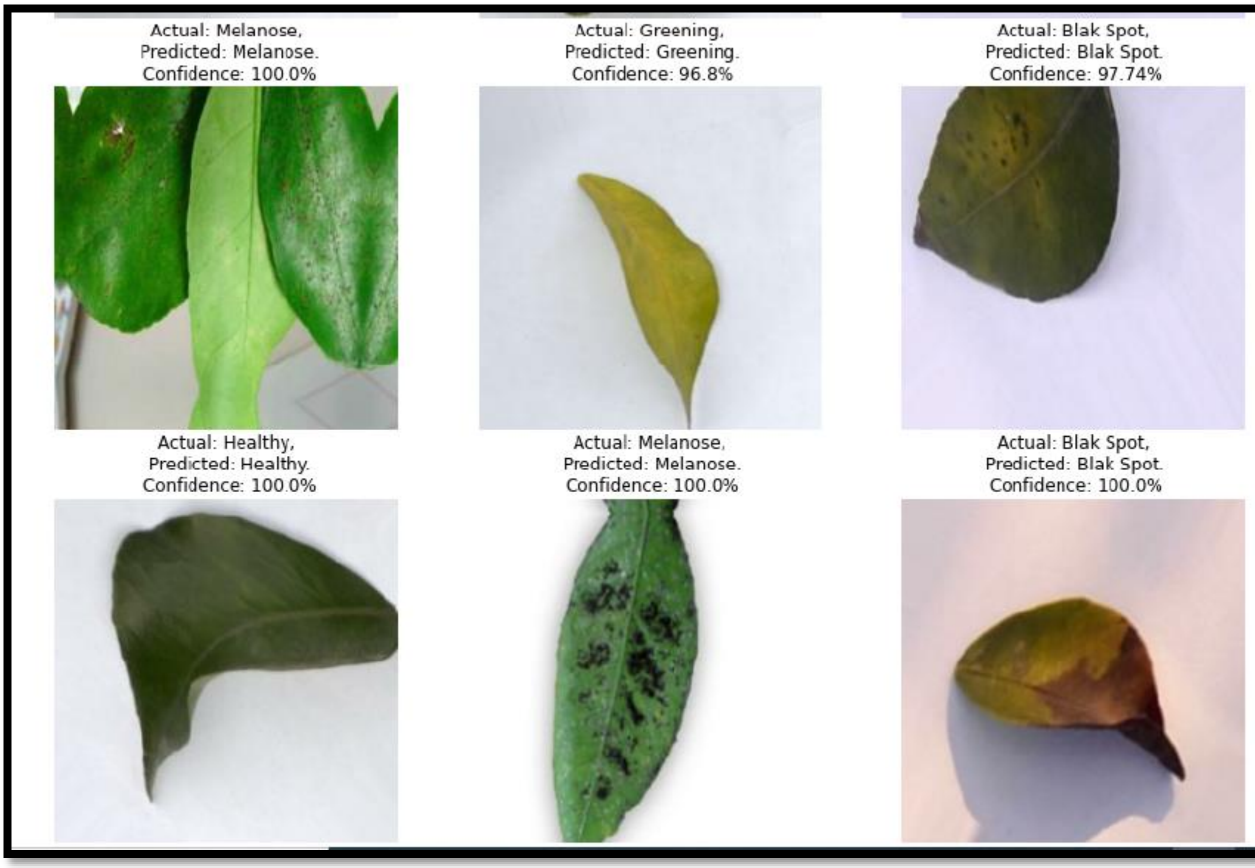


Figure 4.12: Models Test Results

## 4.5. Proposed Model Comparison

Additionally, the proposed approach is compared with current methods for the classification and diagnosis of diseases affecting citrus plants, according to Table 4.3 and Figure 4.12. Based on classification accuracy, the suggested technique outperforms the existing techniques.

| References | Year | Accuracy (% ) |
| --- | --- | --- |
| V. Kukreja et al. [46] | 2020 | 89.1 |
| A. Khattak et al. [41] | 2021 | 95.65 |
| A. Elaraby et al. [47] | 2022 | 94.3 |
| Proposed Model | 2022 | 99.58 |

Table 4.3: Proposed Model Comparison with other studies

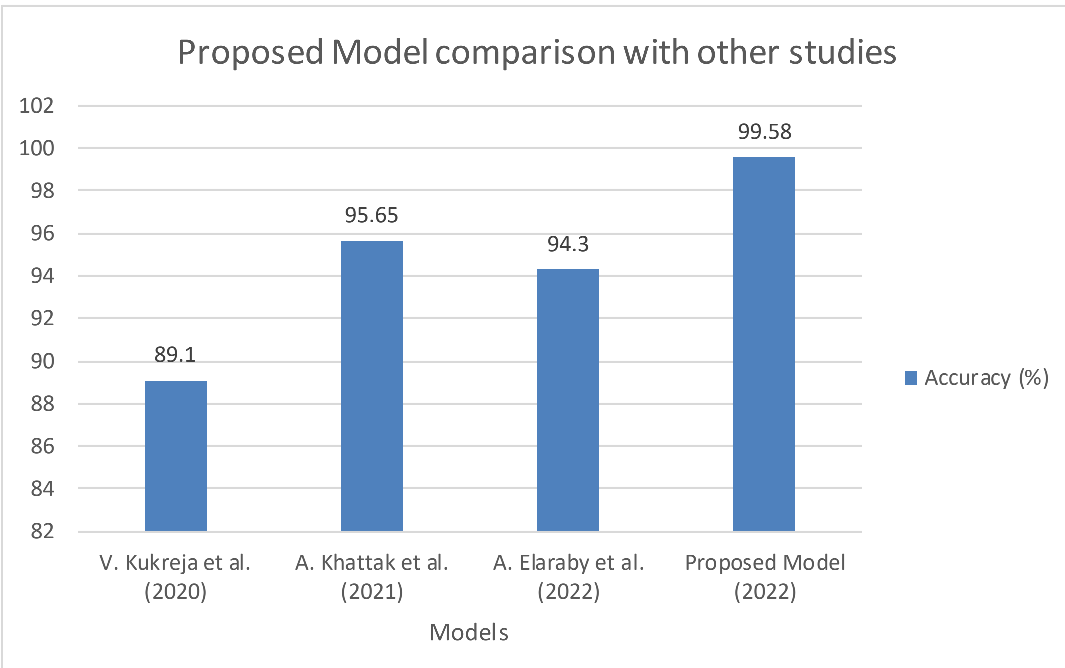


Figure 4.12: Proposed Model Comparison with other studies

# CHAPTER 5: CONCLUSION

To improve citrus plant productivity, it is critically crucial to recognize and classify citrus plant diseases using timely, effective, speedy, automated, less expensive, and precise approaches. Many of the fundamental issues related to the categorization of plant diseases have been successfully addressed by deep learning and CNNs. Transfer learning techniques have proven to be quite useful for identifying and classifying plant diseases. To improve classification accuracy, we have used the most recent transfer learning-based models on our dataset. In this study, we successfully suggested a deep transfer learning based pre-trained CNN model (EfficientNetB3, ResNet50, MobiNetV2, GoogleNet (InceptionV3)) for the recognition and classification of citrus plant diseases. The established model makes a distinction between healthy and unhealthy citrus plant diseases. In order to evaluate network performance, we found that one effective way to create a deep neural network model for the early diagnosis and classification of citrus plant diseases is by transferring an EfficientNetb3 model that has been previously trained on an ImageNet database. With the transfer of an EfficientNetb3 model, we obtained the highest training, validating, and testing accuracies, which were 99.43%, 99.48%, and 99.58%, respectively. Furthermore, the suggested model is contrasted with the other methods for the automated diagnosis and classification of citrus plant diseases. The outcome demonstrates that the recommended CNN model outperforms other state-of-the-art CNN models developed in earlier research in the detection and recognition of citrus plant disease.

# CHAPTER 6: FUTURE WORK

The lack of data is the primary challenge for this study, which is somewhat reduced by the inclusion of the data-augmentation phase. In future studies for this dataset, other data augmentation methods, more training, and other pre-trained models could help to achieve higher accuracy and lower loss. The proposed approach is created using a dataset of five citrus diseases. For more research, different citrus datasets might be investigated for the analysis of other disease classes. In addition, other deep learning models can be used to increase accuracy and computational efficiency.

## APPENDIX I

```
#Data Augmentation and Resizing
from keras.preprocessing.image import ImageDataGenerator
from skimage import io
import numpy as np
import os
from PIL import Image
# Construct an instance of the ImageDataGenerator class
# Pass the augmentation parameters through the constructor.
datagen = ImageDataGenerator(
      rotation_range=45,     #Random rotation between 0 and 45
      width_shift_range=0.2,  height_shift_range=0.2,  shear_range=0.2,
      zoom_range=0.2,horizontal_flip=True,  vertical_flip=True,  fill_mode='reflect',  cval=125
)   #Also try nearest, constant, reflect, wrap
image_directory = 'Citrus\Fruits\scab/'
SIZE = 256, dataset= []
my_images = os.listdir(image_directory)
for i, image_name in enumerate(my_images):
   if (image_name.split('.')[1] == 'jpg'):
      image = io.imread(image_directory + image_name)
      image = Image.fromarray(image, 'RGB')
      image = image.resize((SIZE,SIZE))
      dataset.append(np.array(image))
for batch in datagen.flow (x, batch_size=16,
                  save_to_dir='Augmented_Dataset\Fruits\Scab',
                  save_prefix='aug_scab',
                  save_format='jpg'):
   i += 1
   if i > 16:
      break  # otherwise the generator would loop indefinitely
```

## APPENDIX II

```
#Importing Library and Loading data from disk.
import numpy as np
import PIL.Image as Image
import os
import matplotlib.pylab as plt
import tensorflow as tf
import tensorflow_hub as hub

from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.models import Sequential

BATCH_SIZE = 32
IMAGE_SHAPE = 256
CHANNELS=3
EPOCHS=8

dataset = tf.keras.preprocessing.image_dataset_from_directory(
"Citrus/Leaves",
seed=123,
shuffle=True,
image_size=(IMAGE_SHAPE,IMAGE_SHAPE),
batch_size=BATCH_SIZE
)
class_names = dataset.class_names
class_names
```

## APPENDIX III

```
#Data Splitting
def get_dataset_partitions_tf(ds, train_split=0.6, val_split=0.2, test_split=0.2, shuffle=True, shuffle_size=10000):
    assert (train_split + test_split + val_split) == 1
    ds_size = len(ds)
    if shuffle:
        ds = ds.shuffle(shuffle_size, seed=12)
    train_size = int(train_split * ds_size)
    val_size = int(val_split * ds_size)
    train_ds = ds.take(train_size)
    val_ds = ds.skip(train_size).take(val_size)
    test_ds = ds.skip(train_size).skip(val_size)
    return train_ds, val_ds, test_ds
train_ds, val_ds, test_ds = get_dataset_partitions_tf(dataset)
print("Total Length of dataset: ",len(dataset))
print("After dataset partitions")
print("Length of Training dataset: ",len(train_ds))
print("Length of Validation dataset: ",len(val_ds))
print("Length of Testing dataset: ",len(test_ds))

train_ds = train_ds.cache().shuffle(1000).prefetch(buffer_size=tf.data.AUTOTUNE)
val_ds = val_ds.cache().shuffle(1000).prefetch(buffer_size=tf.data.AUTOTUNE)
test_ds = test_ds.cache().shuffle(1000).prefetch(buffer_size=tf.data.AUTOTUNE)

resize_and_rescale = tf.keras.Sequential([
layers.experimental.preprocessing.Resizing(IMAGE_SHAPE, IMAGE_SHAPE),
layers.experimental.preprocessing.Rescaling(1./255),])
```

## APPENDIX IV

```
#Loading Pre-trained model and change classifier and add dense layers
num_classes = 5
EffNet_model = tf.keras.Sequential([
  resize_and_rescale,
  hub.KerasLayer("https://tfhub.dev/tensorflow/efficientnet/b3/feature-vector/1",
         trainable=False),  # Can be True, see below.
  layers.Dense(64, activation='relu'),
  layers.Dense(32, activation='relu'),
  layers.Dropout(rate=0.1),
  tf.keras.layers.Dense(num_classes, activation='softmax')
])
EffNet_model.build([None, 256, 256, 3])
EffNet_model.summary()
EffNet_model.compile(
  optimizer='adam',
  loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=False),
  metrics=['acc']
)

#Re-Trained the Model
history = EffNet_model.fit(train_ds,
  validation_data=val_ds,
  batch_size=BATCH_SIZE,
  verbose=1,
  epochs=8)
scores = EffNet_model.evaluate(test_ds)
```

## APPENDIX V

```
#Plotting Accuracy and Loss graph
acc = history.history['acc']
val_acc = history.history['val_acc']

loss = history.history['loss']
val_loss = history.history['val_loss']
EPOCHS = 8
plt.figure(figsize=(20, 8))
plt.subplot(1, 2, 1)
plt.plot(range(EPOCHS), acc, label='Training Accuracy')
plt.plot(range(EPOCHS), val_acc, label='Validation Accuracy')
plt.legend(loc='lower right')
plt.ylim([min(plt.ylim()),1])
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.title('Training and Validation Accuracy')

plt.subplot(1, 2, 2)
plt.plot(range(EPOCHS), loss, label='Training Loss')
plt.plot(range(EPOCHS), val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.ylim([0,1.0])
plt.title('Training and Validation Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.show()
```

## APPENDIX VI

```
#Testing Result
def predict(model, img):
    img_array = tf.keras.preprocessing.image.img_to_array(images[i].numpy())
    img_array = tf.expand_dims(img_array, 0)
    predictions = EffNet_model.predict(img_array)
    predicted_class = class_names[np.argmax(predictions[0])]
    confidence = round(100 * (np.max(predictions[0])), 2)
    return predicted_class, confidence
plt.figure(figsize=(15, 15))
for images, labels in test_ds.take(1):
    for i in range(9):
        ax = plt.subplot(3, 3, i + 1)
        plt.imshow(images[i].numpy().astype("uint8"))

        predicted_class, confidence = predict(EffNet_model, images[i].numpy())
        actual_class = class_names[labels[i]]

        plt.title(f'Actual: {actual_class},\n Predicted: {predicted_class}.\n Confidence:
{confidence}%")

        plt.axis("off")
```

## APPENDIX VII

```
#Getting Confusion Metrix and Classification Report
from mlxtend.plotting import plot_confusion_matrix
from sklearn.metrics import confusion_matrix , classification_report

for images, labels in test_ds:

    predictions = EffNet_model.predict(images)
    predicted_class = np.argmax(predictions, axis = 1)
    actual_class = labels.numpy()
    math = tf.math.confusion_matrix(labels=actual_class,predictions= predicted_class)
    mat = math.numpy()
print()
print(classification_report(actual_class,predicted_class))

plot_confusion_matrix(conf_mat=mat, figsize=(10,7), class_names = class_names, show_normed
= True)
```

## REFERENCES


[1] M. Sharif, M. A. Khan, Z. Iqbal, M. F. Azam, M. I. U. Lali, and M. Y. Javed, “Detection and classification of citrus diseases in agriculture based on optimized weighted segmentation and feature selection,” *Comput. Electron. Agric.*, vol. 150, pp. 220–234, Jul. 2018, doi: 10.1016/j.compag.2018.04.023.

[2] R. Manavalan, “Automatic identification of diseases in grains crops through computational approaches: A review,” *Comput. Electron. Agric.*, vol. 178, p. 105802, Nov. 2020, doi: 10.1016/j.compag.2020.105802.

[3] W. Pan, J. Qin, X. Xiang, Y. Wu, Y. Tan, and L. Xiang, “A Smart Mobile Diagnosis System for Citrus Diseases Based on Densely Connected Convolutional Networks,” *IEEE Access*, vol. 7, pp. 87534–87542, 2019, doi: 10.1109/ACCESS.2019.2924973.

[4] G. Wang, Y. Sun, and J. Wang, “Automatic Image-Based Plant Disease Severity Estimation Using Deep Learning,” *Comput. Intell. Neurosci.*, vol. 2017, pp. 1–8, 2017, doi: 10.1155/2017/2917536.

[5] U. P. Singh, S. S. Chouhan, S. Jain, and S. Jain, “Multilayer Convolution Neural Network for the Classification of Mango Leaves Infected by Anthracnose Disease,” *IEEE Access*, vol. 7, pp. 43721–43729, 2019, doi: 10.1109/ACCESS.2019.2907383.

[6] Z. Iqbal, M. A. Khan, M. Sharif, J. H. Shah, M. H. ur Rehman, and K. Javed, “An automated detection and classification of citrus plant diseases using image processing techniques: A review,” *Comput. Electron. Agric.*, vol. 153, pp. 12–32, Oct. 2018, doi: 10.1016/j.compag.2018.07.032.

[7] A. R. Luaibi, T. M. Salman, and A. H. Miry, “Detection of citrus leaf diseases using a deep learning technique,” *Int. J. Electr. Comput. Eng.*, vol. 11, no. 2, p. 1719, Apr. 2021, doi: 10.11591/ijece.v11i2.pp1719-1727.

[8] M. Z. Asghar *et al.*, “Performance Evaluation of Supervised Machine Learning Techniques for Efficient Detection of Emotions from Online Content.” Preprints.org, 2019. doi: 10.20944/preprints201908.0019.v1.

[9] Y. Wang, F. Subhan, S. Shamshirband, M. Zubair Asghar, I. Ullah, and A. Habib, “Fuzzy-based Sentiment Analysis System for Analyzing Student Feedback and Satisfaction,” *Comput. Mater. Contin.*, vol. 62, no. 2, pp. 631–655, 2020, doi: 10.32604/cmc.2020.07920.

[10] M. Ali, M. Z. Asghar, M. Shah, and T. Mahmood, “A simple and effective sub-image separation method,” *Multimed. Tools Appl.*, vol. 81, no. 11, pp. 14893–14910, May 2022, doi: 10.1007/s11042-021-11680-8.

[11] A. Belhadi, Y. Djenouri, G. Srivastava, D. Djenouri, J. C.-W. Lin, and G. Fortino, “Deep learning for pedestrian collective behavior analysis in smart cities: A model of group trajectory outlier detection,” *Inf. Fusion*, vol. 65, pp. 13–20, Jan. 2021, doi: 10.1016/j.inffus.2020.08.003.

[12] L. Yang, Q. Song, and Y. Wu, “Attacks on state-of-the-art face recognition using attentional adversarial attack generative network,” *Multimed. Tools Appl.*, vol. 80, no. 1, pp. 855–875, Jan. 2021, doi: 10.1007/s11042-020-09604-z.

[13] J. Fang, B. Qu, and Y. Yuan, “Distribution equalization learning mechanism for road crack detection,” *Neurocomputing*, vol. 424, pp. 193–204, Feb. 2021, doi:

10.1016/j.neucom.2019.12.057.
[14] H. Ben Yedder, B. Cardoen, and G. Hamarneh, “Deep learning for biomedical image reconstruction: a survey,” *Artif. Intell. Rev.*, vol. 54, no. 1, pp. 215–251, Jan. 2021, doi: 10.1007/s10462-020-09861-2.
[15] M. Ji, L. Zhang, and Q. Wu, “Automatic grape leaf diseases identification via UnitedModel based on multiple convolutional neural networks,” *Inf. Process. Agric.*, vol. 7, no. 3, pp. 418–426, Sep. 2020, doi: 10.1016/j.inpa.2019.10.003.
[16] P. Lottes, J. Behley, A. Milioto, and C. Stachniss, “Fully Convolutional Networks With Sequential Information for Robust Crop and Weed Detection in Precision Farming,” *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 2870–2877, Oct. 2018, doi: 10.1109/LRA.2018.2846289.
[17] W. Xinshao and C. Cheng, “Weed seeds classification based on PCANet deep learning baseline,” in *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, Dec. 2015, pp. 408–415. doi: 10.1109/APSIPA.2015.7415304.
[18] X. Cheng, Y. Zhang, Y. Chen, Y. Wu, and Y. Yue, “Pest identification via deep residual learning in complex background,” *Comput. Electron. Agric.*, vol. 141, pp. 351–356, Sep. 2017, doi: 10.1016/j.compag.2017.08.005.
[19] A. Ahmad, D. Saraswat, and A. El Gamal, “A survey on using deep learning techniques for plant disease diagnosis and recommendations for development of appropriate tools,” *Smart Agric. Technol.*, vol. 3, p. 100083, Feb. 2023, doi: 10.1016/j.atech.2022.100083.
[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.
[21] C. Szegedy *et al.*, “Going deeper with convolutions,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 1–9. doi: 10.1109/CVPR.2015.7298594.
[22] Alireza Pourreza, Won Suk (Daniel) Lee, Eran Raveh, Youngki Hong, and Hyuck-Joo Kim, “Identification of citrus greening disease using a visible band image analysis,” 2013. doi: 10.13031/aim.20131591910.
[23] U. Barman and R. D. Choudhury, “Smartphone Assist Deep Neural Network to Detect the Citrus Diseases in Agri-Informatics,” *Glob. Transitions Proc.*, Oct. 2021, doi: 10.1016/j.gltp.2021.10.004.
[24] H. Xiaoling, D., Lan, Y., Xiaqiong, X., Huilan, M., Jiakai, L., & Tian-sheng, “Detection of citrus Huanglongbing based on image feature extraction and two-stage BPNN modeling,” *Int. J. Agric. Biol. Eng.*, vol. 9, pp. 20–26, 2016.
[25] M. Zia Ur Rehman *et al.*, “Classification of Citrus Plant Diseases Using Deep Transfer Learning,” *Comput. Mater. Contin.*, vol. 70, no. 1, pp. 1401–1417, 2022, doi: 10.32604/cmc.2022.019046.
[26] K. Kuwata and R. Shibasaki, “Estimating crop yields with deep learning and remotely sensed data,” in *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Jul. 2015, pp. 858–861. doi: 10.1109/IGARSS.2015.7325900.
[27] B. Zhao, X. Li, X. Lu, and Z. Wang, “A CNN–RNN architecture for multi-label weather recognition,” *Neurocomputing*, vol. 322, pp. 47–57, Dec. 2018, doi:

10.1016/j.neucom.2018.09.048.

[28] X. Zhu, Z. He, J. Du, L. Chen, P. Lin, and Q. Tian, “Soil moisture temporal stability and spatio-temporal variability about a typical subalpine ecosystem in northwestern China,” *Hydrol. Process.*, p. hyp.13737, Mar. 2020, doi: 10.1002/hyp.13737.

[29] Z. Lin *et al.*, “A Unified Matrix-Based Convolutional Neural Network for Fine-Grained Image Classification of Wheat Leaf Diseases,” *IEEE Access*, vol. 7, pp. 11570–11590, 2019, doi: 10.1109/ACCESS.2019.2891739.

[30] Y. Lu, S. Yi, N. Zeng, Y. Liu, and Y. Zhang, “Identification of rice diseases using deep convolutional neural networks,” *Neurocomputing*, vol. 267, pp. 378–384, Dec. 2017, doi: 10.1016/j.neucom.2017.06.023.

[31] B. Richey, S. Majumder, M. V. Shirvaikar, and N. Kehtarnavaz, “Real-time detection of maize crop disease via a deep learning-based smartphone app,” in *Real-Time Image Processing and Deep Learning 2020*, Apr. 2020, p. 10. doi: 10.1117/12.2557317.

[32] S. Mishra, R. Sachan, and D. Rajpal, “Deep Convolutional Neural Network based Detection System for Real-time Corn Plant Disease Recognition,” *Procedia Comput. Sci.*, vol. 167, pp. 2003–2010, 2020, doi: 10.1016/j.procs.2020.03.236.

[33] K. Golhani, S. K. Balasundram, G. Vadamalai, and B. Pradhan, “A review of neural networks in plant disease detection using hyperspectral data,” *Inf. Process. Agric.*, vol. 5, no. 3, pp. 354–371, Sep. 2018, doi: 10.1016/j.inpa.2018.05.002.

[34] C. B. Wetterich, R. Felipe de Oliveira Neves, J. Belasque, and L. G. Marcassa, “Detection of citrus canker and Huanglongbing using fluorescence imaging spectroscopy and support vector machine technique,” *Appl. Opt.*, vol. 55, no. 2, p. 400, Jan. 2016, doi: 10.1364/AO.55.000400.

[35] H. Patel, R. Prajapati, and M. Patel, “Detection of Quality in Orange Fruit Image using SVM Classifier,” in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, Apr. 2019, pp. 74–78. doi: 10.1109/ICOEI.2019.8862758.

[36] H. Singh, R. Rani, and S. Mahajan, “Detection and Classification of Citrus Leaf Disease Using Hybrid Features,” 2020, pp. 737–745. doi: 10.1007/978-981-15-0751-9_67.

[37] S. Xing, M. Lee, and K. Lee, “Citrus Pests and Diseases Recognition Model Using Weakly Dense Connected Convolution Network,” *Sensors*, vol. 19, no. 14, p. 3195, Jul. 2019, doi: 10.3390/s19143195.

[38] P. Dhiman *et al.*, “A Novel Deep Learning Model for Detection of Severity Level of the Disease in Citrus Fruits,” *Electronics*, vol. 11, no. 3, p. 495, Feb. 2022, doi: 10.3390/electronics11030495.

[39] M. Khanramaki, E. Askari Asli-Ardeh, and E. Kozegar, “Citrus pests classification using an ensemble of deep learning models,” *Comput. Electron. Agric.*, vol. 186, p. 106192, Jul. 2021, doi: 10.1016/j.compag.2021.106192.

[40] S. S. Hari, M. Sivakumar, P. Renuga, S. Karthikeyan, and S. Suriya, “Detection of Plant Disease by Leaf Image Using Convolutional Neural Network,” in *2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)*, Mar. 2019, pp. 1–5. doi: 10.1109/ViTECoN.2019.8899748.

[41] A. Khattak *et al.*, “Automatic Detection of Citrus Fruit and Leaves Diseases Using Deep Neural Network Model,” *IEEE Access*, vol. 9, pp. 112942–112954, 2021, doi: 10.1109/ACCESS.2021.3096895.

[42] Z. Liu, X. Xiang, J. Qin, YunTan, Q. Zhang, and N. N. Xiong, “Image Recognition of Citrus Diseases Based on Deep Learning,” *Comput. Mater. Contin.*, vol. 66, no. 1, pp. 457–466, 2020, doi: 10.32604/cmc.2020.012165.

[43] U. Barman, R. D. Choudhury, D. Sahu, and G. G. Barman, “Comparison of convolution neural networks for smartphone image based real time classification of citrus leaf disease,” *Comput. Electron. Agric.*, vol. 177, p. 105661, Oct. 2020, doi: 10.1016/j.compag.2020.105661.

[44] M. Zhang, S. Liu, F. Yang, and J. Liu, “Classification of Canker on Small Datasets Using Improved Deep Convolutional Generative Adversarial Networks,” *IEEE Access*, vol. 7, pp. 49680–49690, 2019, doi: 10.1109/ACCESS.2019.2900327.

[45] C. T. Soini, S. Fellah, and M. R. Abid, “Citrus Greening Infection Detection (CiGID) by Computer Vision and Deep Learning,” in *Proceedings of the 2019 3rd International Conference on Information System and Data Mining - ICISDM 2019*, 2019, pp. 21–26. doi: 10.1145/3325917.3325936.

[46] V. Kukreja and P. Dhiman, “A Deep Neural Network based disease detection scheme for Citrus fruits,” *Proc. - Int. Conf. Smart Electron. Commun. ICOSEC 2020*, no. Icosec, pp. 97–101, 2020, doi: 10.1109/ICOSEC49089.2020.9215359.

[47] A. Elaraby, W. Hamdy, and S. Alanazi, “Classification of Citrus Diseases Using Optimization Deep Learning Approach,” *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–10, Feb. 2022, doi: 10.1155/2022/9153207.

[48] H. T. Rauf, B. A. Saleem, M. I. U. Lali, M. A. Khan, M. Sharif, and S. A. C. Bukhari, “A citrus fruits and leaves dataset for detection and classification of citrus diseases through machine learning,” *Data Br.*, vol. 26, p. 104340, Oct. 2019, doi: 10.1016/j.dib.2019.104340.

[49] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 2818–2826. doi: 10.1109/CVPR.2016.308.

[50] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” Dec. 2015, doi: https://arxiv.org/abs/1512.03385v1.

[51] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” Jan. 2018, [Online]. Available: http://arxiv.org/abs/1801.04381

[52] M. Tan and Q. V. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” May 2019, [Online]. Available: http://arxiv.org/abs/1905.11946

[1] https://www.freshplaza.com/article/9216529/citrus-greening-threatens-pakistan-s-citrus-production-export/