Citrus Leaf Disease Detection By Image Processing

by

Mahir Faisal Chowdhury
19301026
Amit Nondi
19101247
Fardin Zaman
20301473
Sium Ibn Akhter
20101566
Tanjina Bilma Pathan
19101617

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
School of Data and Sciences
Brac University
January 2024

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Amit Nondi
19101247

Mahir Faisal Chowdhury
19301026

Fardin Zaman
20301473

Tanjina Bilma Pathan
19101617

Sium Ibn Akhter
20101566

Approval

The thesis titled "Citrus Leaf Disease Detection By Image Processing" submitted by

1. Mahir Faisal Chowdhury (19301026)
2. Amit Nondi (19101247)
3. Fardin Zaman (20301473)
4. Tanjina Bilma Pathan (19101617)
5. Sium Ibn Akhter (20101566)

Of Fall, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 09, 2024.

Examining Committee:

Supervisor:
(Member)

Mr. Dewan Ziaul Karim
Lecturer
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Md. Golam Rabiul Alam, PhD
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Ethics Statement

The research conducted here is original and accurate.All the resources has been cited properly. We have collected dataset from Kaggle.

Abstract

Citrus leaf diseases bring a danger to the earnings of citrus estates. When it comes to recovering from illness, early detection and accurate diagnosis are very necessary. In the last several decades, there have been advancements made in the diagnosis and classification of citrus leaf diseases via the use of deep learning techniques in image processing. When it comes to automating the detection of citrus leaf diseases, we recommend making use of pre-trained convolutional neural networks (CNNs) like ResNet-50, VGG16, MobileNet-V2, InceptionV3, InceptionResNet-V2, DenseNet-201, and DenseNet-121.To accomplish this goal, a comprehensive data collection consisting of images of citrus leaves that have been identified will be gathered and pre-processed. Citrus canker, greening, and black spot leaves will be included in the databases, along with healthy and diseased citrus leaves.For the purpose of extracting useful characteristics from leaf images, we shall make use of deep learning models. For the purpose of picture classification, the models that were discussed before are useful and often used.In this research, we propose a CNN model that is both effective and efficient. The model was originally trained on 596 pictures, and then it was augmented with 2800 images that were divided into three categories: training, validation, and testing.70% of the data goes towards training, 15% goes towards validation, and 15% goes towards testing.A few pieces of public data will be served this model.Following that, we will evaluate the findings in relation to the prebuilt models.Last but not least, we get a training accuracy of 95.95% and a validation accuracy of 97.84%. Furthermore, our suggested model has a lower number of training parameters compared to all other pretrained models, which enables our model to categorise illnesses more quickly.This provides us with a decent level of accuracy.

Keywords: CNN,ResNet-50, VGG16, MobileNet-V2,Inception-V3, InceptionResNet-V2, DenseNet-201,DenseNet-121,Pre-processing,Augmentation, Deep Learning.

Acknowledgement

Firstly, all praise to The Almighty for whom our thesis have been completed without any major interruption.

Secondly, to our supervisor Mr. Dewan Ziaul Karim sir for his kind support and advice in our work. He helped us whenever we needed help.

And finally to our parents without their throughout support it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

ANN Artificial Neural Networks

CNN Convolutional Neural Network

HLB Huanglongbin

HSI Hue,Saturation,Intensity

$MSVM$ Multi-Support Vector Machine

PCA Principal Component Analysis

RPN Region Proposal Network

$SGDM$ Stochastic Gradient Descent with Momentum

Chapter 1

Introduction

The collective term "leaf disease" is used to describe the presence of one or more ailments that negatively impact the health and productivity of leaves. This is due to the occurrence of a leaf disease, which may adversely affect the quality and functionality of an organism's leaves. The occurrence of similar illnesses in others may have been caused by infectious agents such as fungus, bacteria, or viruses, as well as environmental factors. The most common leaf diseases are often the result of fungal infections (such as powdery mildew), bacterial infections (such as bacterial leaf spot), viral infections (such as mosaic viruses), and physiological abnormalities (such as leaf chlorosis). Within the agricultural sector, a major obstacle leading to decreased productivity is the challenge of accurately identifying diseases that show in the foliage of plants. Plant leaf diseases are a major cause of reduced productivity. [1]-[2]

Citrus trees are one of the most economically crucial crop kinds growing all over the world and give a large amount to both the manufacture and the marketing of agricultural goods. Citrus is a highly sought-after commodity that is extensively traded on a global scale[3]. The citrus business faces many obstacles, with the most significant being the threat posed by illnesses. Infections may result in significant crop reductions, impacting both the amount and quality of citrus fruits. They are especially vulnerable to the adverse consequences of disease. Detecting illnesses promptly and administering appropriate treatment is essential to avoid the spread of infections and mitigate the damage they do to citrus plants.

Currently, citrus trees are at a heightened risk of acquiring many diseases.Leaf diseases have a particularly negative impact on the general well-being of the plant and its productivity. Citrus leaf diseases are caused by a range of infectious organisms, including as fungus, bacteria, and viruses. These diseases attack the leaves of the citrus plant and disturb its regular physiological functions. Citrus trees are prone to several diseases, such as citrus canker, citrus greening, citrus scab, and citrus black spot, among others. Specific symptoms, characterized by lesions on the leaves, serve as diagnostic indicators for particular illnesses. Necrosis, lesions, discoloration, and anomalies are representative manifestations of these illnesses[4].Previously, the identification and diagnosis of citrus leaf diseases relied on the expertise of trained specialists via the method of visual examination, a procedure that is both laborious and subjective. When evaluating the reliability of visual examination results, the

proficiency and experience of the examiner are crucial factors to consider. Scientists have developed automated methods to detect citrus leaf illnesses by using advancements in image processing and machine learning. Their objective was to overcome the obstacles posed by these limitations by creating a system capable of detecting citrus leaf illnesses.

Utilizing visual analysis and other automated learning techniques enables the creation of citrus leaf disease detection systems that are both efficient and precise. These devices may be used for monitoring the well-being of citrus plants. These systems use image processing methods to retrieve essential data from digital photographs of leaf surfaces. The obtained data is further classified based on the existence or nonexistence of illnesses in the corresponding images. Given a enough quantity of annotated leaf pictures during the training phase, machine learning models may be trained to identify patterns and provide precise predictions. Machine learning models are capable of acquiring information from instances, thereby making this feasible.

Convolutional neural networks, often referred to as CNNs, are a kind of deep learning algorithm particularly developed for analyzing and interpreting visual input, such as images [5].Convolutional neural networks (CNNs) are extensively used in many computer vision tasks, such as picture classification, object detection, and image segmentation. Neural networks consist of several layers, including convolutional layers, pooling layers, and fully connected layers. These layers work together to produce hierarchical representations of the input data. For instance, a pooling layer has the potential to evolve into a fully linked layer[6].

ResNet-50 is a convolutional neural network with 50 layers, with 48 being convolutional layers, one MaxPool layer, and one average pool layer. Remaining neural network structures are a subset of artificial neural networks (ANNs) that are created by sequentially stacking remained blocks throughout the network construction process[7]. VGG16, a CNN, is one of the finest computer vision models. The model's creators enhanced network depth using a 3x3 convolution filter design, significantly outperforming previous setups. They added 16–19 weighted layers for 138 parameters to be trained.

EfficientNet makes use of a technique known as compound coefficient in order to enhance the efficiency of scaling up models in an easy manner. Unlike the practice of arbitrarily increasing the size of settlement, width, or depths, compound scaling makes adjustments to each dimension in an equal manner by using a certain set of scaling parameters[8]. The developers used the scaling technique and AutoML to generate seven separate models of different sizes, each designed to be efficient. Each of them surpassed the accuracy achieved by most convolutional neural networks now available, and they achieved this with substantially higher efficiency.

MobileNet is a computer vision model built by Google and published into the public domain. It is a pre-trained model specifically designed for training classifiers[9].This is achieved by the use of depthwise convolutions, which, when compared to other networks, leads to a significant reduction in the number of parameters. The final

outcome is a neural network that is both lightweight and capable of deep learning. The first mobile artificial intelligence model created by Tensorflow is named MobileNet. The deep convolutional architecture of Inception was first presented under the titles GoogLeNet and Inception-v1[10].MobileNetV2 is a compact convolutional neural network model with 53 layers, designed to be lighter. It has a smaller number of parameters and accepts input images of size 224×224. The MobileNetV1 design included depth-wise separable convolutions, which involve applying a single filter to each input channel. Additionally, point-wise convolutions (1×11×1) were used to combine the output of the depth-wise convolutions.

Inception-v3 is a convolutional neural network architecture that is part of the Inception family. The model incorporates many improvements, such as the implementation of Label Smoothing, the employment of Factorized 7 x 7 convolutions, and the inclusion of an extra classifier to propagate label information further into the network.Again, InceptionResNet-v2 is a convolutional neural network design that extends the Inception structure by including connections that remain to replace the filtering assembly step.DenseNet-201 is a convolutional neural network with a depth of 201 layers. The system has the capability to load a preexisting version of the network that has been trained on over one million photos from the ImageNet database. The preexisting network has the ability to categorize photos into 1000 distinct item types. Consequently, the network has acquired comprehensive and complex visual patterns for a diverse set of pictures. The network's image input resolution is 224-by-224 pixels.

Recent research has focused on establishing algorithms and models to identify citrus leaf surface diseases.Smartphone image classification and detection of citrus leaf diseases is more accurate and efficient using the SSCNN. Barman et al.'s 2020 research confirmed similar. Furthermore, as compared to MobileNet, the SSCNN algorithm demonstrates a reduced computational time, making it a more economically efficient approach for diagnosing citrus illnesses[11]. Elaraby et al. (2022) proposed a technique including the implementation of two separate convolutional neural network (CNN) models. The models in question are AlexNet and VGG19.The system performed at a level that was 94 percent of its total capability.Compared to any other approach, it is far more efficient[12].

The use of computerized techniques for the detection of diseases on citrus leaves has the potential to revolutionize the disease management systems now employed in the citrus sector. Timely and precise identification of a disease may facilitate prompt actions, such as tailored therapies and the elimination of affected trees. These proactive strategies have the capacity to help reduce total financial losses and contain the transmission of diseases inside plantations.

This research uses a sophisticated automated method to enhance citrus leaf disease diagnosis. The system will leverage cutting-edge computer vision methods including neural networks and computational imaging. These methods seek to accurately identify and classify citrus leaf diseases. Using a huge dataset of annotated leaf ailment photographs to train and test machine learning models would provide excellent disease detection accuracy. Training models using the dataset will accomplish this.

Chapter 2

Research Problems and Objectives

2.1 Research Problems

The human intelligence plays a crucial part in the construction of complex plans that involve several phases. The development of plants for crops is negatively impacted when they have been harmed or afflicted by disease. Diseases can also infect crops that have been damaged. If there is no core or foundational basis, the readily apparent evidence will remain challenging and fragmented into several components. The failure of this attack is attributed to the synergy between the rising prevalence of global computer access and the continuous advancements in neuroscience facilitated by reflective learning. Both of these variables collaborate synergistically to attain this triumph. Given this symbiotic relationship, it is now imperative to be well equipped for the identification and recommendation of illnesses facilitated by systems.

Currently, the primary approach for determining the presence of citrus leaf diseases is to do visual inspections. Implementing this strategy necessitates a substantial allocation of time and is challenging to achieve flawlessness. Citrus trees can be affected by many diseases, such as black spots, cankers, citrus early blight, citrus late blight, and numerous more. Canker is a highly contagious disease that mostly affects the foliage and fruits of citrus plants. Reports indicate varying crop reductions, with Kinnow seeing a 22 percent fall, sweet oranges seeing a 25 percent decline, grease witnessing a 10 percent drop, and lemons showing a 2 percent loss. The percentages fluctuate depending on the type of fruit. It has been reported that the losses have happened in each of these several types of crops. Citrus fruit diseases are the primary cause of significant yearly losses in an otherwise promising export crop. As a result, early detection of citrus illnesses increases the likelihood of timely recognition, leading to reduced losses and expenses, as well as enhanced product quality.

Conversely, it is plausible that deep learning will autonomously discern the hierarchical organization of disorders.Therefore, there is a need for a reliable, efficient, and automated technique of disease diagnosis. The aim is to create a deep learning system that can reliably and effectively detect diseases in citrus leaf samples [13].In order to effectively detect and classify particular conditions, it's a must to have particular techniques and information.The challenges that may be encountered include

the following:

1. Effectiveness of Automated Disease Diagnosis.

2. Robustness to Diverse Disease Types.

3. Maximizing the efficiency of resource distribution.

4. Generalization Across Citrus Varieties.

5. Model Bias and Class Imbalance.

2.2 Research Objectives

For the purpose of developing a deep network of convolutional neural networks that is capable of classifying images of citrus fruit and leaves as either well or unhealthy, and if the condition is determined to be diseased, it will display the name of the ailment if it is discovered to be present in the image. One of the goals of our research will be to do the above. Images of citrus leaves that are put into it will quickly show its qualities. The system will learn attributes immediately. In consideration of this, the system has the intention of accomplishing the following objectives:

1. Develop a deep convolutional neural network that is capable of identifying and classifying diseases that may affect citrus leaves.

2. Perform an evaluation of the effectiveness of several pre-trained deep learning models, such as ResNet-50, VGG16, MobileNet-V2, Inception-V3, InceptionResNet-V2, DenseNet-201 and DenseNet-121 and after that, identify the most optimum model.

3. Develop a network that is capable of learning from data while simultaneously improving its ability to identify illnesses in citrus fruit or leaves.

4. Devise an effective technique for the training of a deep learning model using a balanced dataset.

5. Carrying out a validation examination on the system that has been proposed.

Our research aims to identify ailments that impact citrus leaves. The recommended approach utilizes the CNN model to classify citrus leaf diseases, including black spot, canker, early blight, and late blight. These diseases impact the foliage of citrus trees in various manners. The suggested deep learning model has an optimal number of layers and parameters. The primary goal of the offered technique is to diagnose citrus illnesses.A novel strategy is being implemented and should improve precision.

Chapter 3

Literature Review and Related Work

3.1 Related Works

Fruits are an excellent source of vital nutrients such as fiber, vitamins, and minerals, in addition to plant chemicals with a high concentration. If our bodies do not have the appropriate quantities of these components, we are more likely to suffer from a range of ailments. These components are essential for our bodies to work well, and these components are necessary for our bodies to function properly. There is also vitamin C, flavonoids, and fiber in citrus fruit, all of which help to the development of our immune system, enhance the working of our digestive tract, and play a significant role in the prevention of diseases such as cancer and diabetes. Citrus fruit is a great source of all of these nutrients[14].

Canker, Greening, Phyllocnistis citrella, Anthracnose, scale insect disease, and element deficiency are only few of the illnesses that citrus trees are prone to. Citrus trees are also subject to a broad number of other diseases. When it comes to the underlying causes of these ailments, it is conceivable that bacteria, a poor diet, or fungus are to blame. In order to guarantee that citrus fruit will continue to be in good condition, it is critical that diseases that affect the leaves of citrus trees be diagnosed and managed as early in the growing season as feasible. By observing the yellowing and browning of certain places on the plant's leaves, one may identify the canker disease that is affecting the plant. Because of the canker, trees lose their foliage, which results in a decrease in the amount of fruit they produce[4].Due to the fact that it causes citrus trees to stop producing fruit, the citrus greening disease is the most damaging disease that may impact citrus trees. Citrus diseases, such as a change in the color of the leaves from green to yellow, are frequently brought on by deficiencies in the essential nutrients that the plant needs[15].The black sores on the leaves are called anthracnose, and they are caused by fungi [16].If we want to avoid citrus leaf diseases, we need to make sure that the plants are adequately fertilized and that they are watered every day. In addition, we are able to identify trees and leaves that have been afflicted by the disease by performing field scouting on a weekly basis. This is essential in order to forestall the growth of the illness, which would lead to considerable financial losses for the farmers if the disease were to continue to spread[17].

The author of the paper [18] used AlexNet and ResNet, which are both different forms of CNN models.AlexNet's accuracy after the augmentation of its data is 97.92 percent, while ResNet's accuracy is 95.83 percent. Phyllocnistis citrella leaves, element-free leaves, scale insect leaves, and healthy leaves are some of the types of leaves that are included in the dataset. Using data augmentation allowed for an increase in the total number of photographs used for training purposes. To begin the process of classifying the illnesses, they began with a dataset that had 180 photographs and utilized two different CNN models. When we look at the confusion matrix of the AlexNet model with data augmentation, we can see that the type of diseases that are produced by scale insects on leaves is only identified 83.7 percent of the time. On the other hand, all of the other types of diseases are detected one hundred percent of the time. Moreover, in the absence of data enhancement, neither scale insects nor a shortage in elements are identified to their full degree. On the other hand, different types of ailments are brought to light. On the basis of the confusion matrix, we are able to make the observation that the ResNet model with data augmentation does not totally identify scale insects, the kind of leaves, or the lack of components; correspondingly, it identifies 83.7 percent and 71.4 percent of these items. On the other hand, the remaining illnesses have been recognized in complete detail. In the absence of data enhancement, indications of health and deficits in components are not totally detected, although signs of other types of illnesses are. We can see from the validation and test accuracy table that the validation and test accuracy are better with the inclusion of data augmentation for both models (AlexNet and ResNet) as compared to when they were done without the addition of data augmentation.

In paper [19], the model has four steps and those are image acquisition, image pre-processing, image segmentation, and lastly, feature extraction.Their approach involved employing a concealed Markov model for the purpose of categorization. Initially, for the process of obtaining the image, an image of a citrus leaf was captured and subsequently adjusted to a resolution of 250x250 pixels. Additionally, contrast adjustment was employed for picture preprocessing in order to distinguish the leaf from the backdrop. Additionally, normalization has been employed to scale the pixel values within the range of zero to one, hence preventing abrupt changes in color. Additionally, in the context of picture segmentation, they employed color transformation into YCbCr and bi-level thresholding. Finally, they employed feature extraction with GLCM. Following the use of image processing techniques, the Markov model was employed. The model accurately predicted the occurrence of anthracnose with an accuracy rate of 84.21 percent, canker with an accuracy rate of 85.71 percent, citrus greening with an accuracy rate of 78 percent, and overwatering with an accuracy rate of 82.50 percent.

In another paper [20],The study utilized k-means segmentation and top-hat enhancement image processing techniques in order to preprocess healthy leaves and three forms of citrus leaf illnesses. The classification of leaf diseases was performed using the multi-support vector machine (MSVM), achieving an accuracy rate of 93.18 percent.

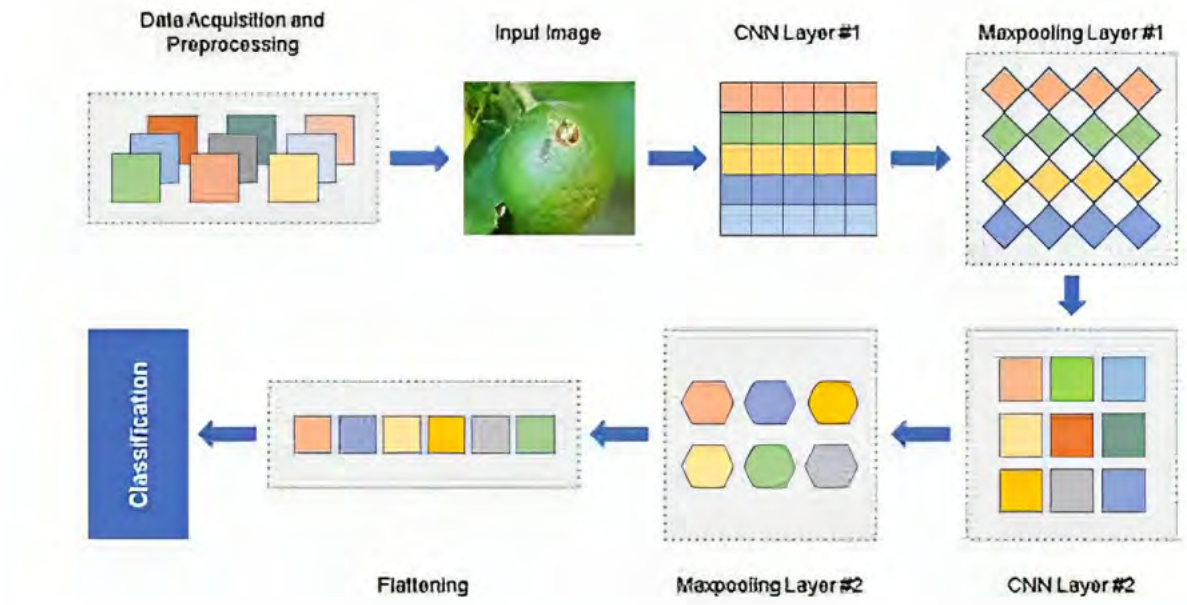

Figure 3.1: A brief description of the proposed system.

A dataset consisting of 140 photos was utilized, with 80 images allocated for training purposes and the remaining images reserved for testing. Four procedures were employed to categorize the citrus leaf disease: picture enhancement using a top-hat filter, image segmentation, k-means clustering, and finally, the GLCM function. In order to determine the affected region, the RGB pictures are transformed into HSV format, and then, the healthy and diseased regions are distinguished by pixel counting. Based on the accuracy graph of both clusters, it is evident that the second cluster approach outperforms the first cluster method significantly. The accuracy of the second cluster is 100 percent for healthy leaves and Phyllocnistis citrella, and 81.8 percent and 90.9 percent for leaves without an element and leaves affected by scale insects.

In paper [21],The collection has 11,800 photos depicting 7 distinct types of sick leaves and 1 type representing a healthy leaf. A deep convolutional neural network (CNN) model named TLNet (tomato leaf Net) was developed. The dataset was utilized by dividing it into three distinct portions: training (80percent of the dataset), validation (20percent of the dataset), and testing (50 photos from each category). The accuracy of the specified model for this dataset was 98.77 percent. The dataset preparation involved the removal of outliers, scaling of images, normalization of data, augmentation of data, and encoding of class labels. The CNN model (TLNet) consists of 5 convolutional layers and 3 dense layers. The Adam optimizer was employed to determine the ideal parameters for the model, initially set with a learning rate of 0.01 and a minimal threshold of 0.000001.

In paper [12],Data augmentation techniques such as Affine Transformation and Perspective Transformation can be employed. GANs are employed to produce very realistic counterfeit pictures when the training photographs of poor quality and traditional image editing methods are ineffective. The Stochastic Gradient Descent with Momentum (SGDM) optimization method has been utilized. Anthracnose, black spot, canker, scab, greening, and melanosis are among the citrus diseases that were identified and classified using databases.

In paper [20],At each phase of an image analysis inquiry, datasets are necessary, starting with the initial training of algorithms to the ultimate evaluation of the efficacy of these algorithms. A total of 2293 photographs were captured from the PlantVillage collection and the citrus dataset for this research. The primary objective of developing the Plant Village benchmark database was to furnish academics with further insights into the well-being of plants. Four unique collections of infected pictures were created using images that had been contaminated with one of four distinct diseases that can appear in citrus fruits and leaves. The illnesses examined in the data sets included greening, black spots, cankers, and scabs. Another condition was melanosis.The dataset may be divided into three separate sections: training data, test data, and validation data.

In paper [21],Proposed a patch-based classification network for accurate disease detection in citrus plants. The network comprises an embedding module, a cluster prototype module, and a simple neural network classifier.The approach exhibits a detection accuracy of 95.04 percent, requires less than 2.3 million tuning parameters, and enables the diagnosis of citrus ailments using the trained model in less than 10 milliseconds. These results surpass the performance of the currently employed state-of-the-art methods.

In study [22],Automated identification of yellow rust and septoria was suggested to be performed via watershed segmentation.The method achieved a 95.48 percent efficiency rate when applied to a set of 185 training photos.The sugar leaf spot infections were recognized more rapidly using a quicker R-CNN architecture compared to a convolutional neural network-based area framework.The technique of visible and near-infrared reflectance spectroscopy was employed to detect the presence of Penicillium digitatum-induced decay in citrus fruits.The efficacy of this strategy was evaluated using the manifold learning algorithms.It is crucial to identify citrus huanglongbing (HLB) in order to prevent the spread of illness.Low-altitude satellite imagery was employed to conduct area-wide surveillance of citrus huanglongbing.Citrus exports were impeded by fruit diseases such as citrus canker, black spot, and scab.A refined convolutional neural network (CNN) method was used to detect and evaluate imperfections in citrus fruits.The Sobel gradients partitioned the images into areas with defects, and the color and texture attributes were extracted.The machine vision system was utilized to identify and evaluate imperfections in citrus peels.Hyperspectral imaging was employed to detect citrus fruit illnesses, particularly fungal infections.The accuracy of multiple classifications using receiver operating characteristic curves was 89.00 percent.

In study[23],There are several methods available for diagnosing plant diseases, particularly those affecting citrus plants. These include many techniques for dividing and extracting information, such as Gabor, edge, regional, threshold, principal component analysis (PCA), and wavelet transform. Various techniques have been employed for citrus disease detection, including as machine learning (ML), color segmentation, lightweight CNN-based systems, texture-based hue, saturation, and intensity (HSI) color characteristics, and statistical classification methods.Scientists have devised techniques employing CNN, SVM, VGG16, Mask R-CNN, and region proposal network (RPN) to detect and classify plant illnesses. The accuracy of these procedures varies between 80percent and 100percent. Multiple studies have utilized hierarchical SVM algorithms and feature distribution analysis to investigate citrus leaf diseases, including canker, greening, and sooty moorland leaf miner. The objective of the proposed CNN-based system in this work is to enhance the accuracy of detecting citrus leaf diseases and contribute to the existing knowledge in the field of agriculture.

The study [24] primarily focuses on the utilization of two conventional neural network models, namely AlexNet and ResNet, for the purpose of illness categorization and detection. The main goal of this study is to utilize deep learning methods for diagnosing citrus leaf diseases. The authors emphasize the need of swiftly detecting plant illnesses, particularly in citrus plants, in order to guarantee food security. Their focus is on demonstrating the potential applications of deep learning, namely deep convolutional neural networks, for very accurate disease detection. Data augmentation techniques may significantly improve the effectiveness of small datasets by increasing the number of training samples without introducing new ones. The paper references the utilization of AlexNet for image classification in many tests. The authors also discuss the several afflictions that may damage citrus leaves, including infestation by insect scale, nutrient deficiency, and infection by Phyllocnistis citrella.Prior studies have explored the application of weakly dense connected convolutional networks and other CNN models to accurately identify citrus illnesses. In summary, the work provides insights into the application of deep learning techniques, specifically AlexNet and ResNet models, for diagnosing citrus leaf diseases. It highlights the importance of early detection in ensuring food security. Additionally, it discusses several ailments that might potentially damage citrus leaves and references previous studies on the efficacy of CNN models in detecting citrus infections.

The study[25] utilizes symptomatic data and employs advanced analytical approaches such as deep learning, machine learning, and image processing to assess nearly all of the published articles on citrus illnesses and fruit grading throughout this timeframe. This text provides a concise overview of pre-processing, segmentation, feature extraction, and classification approaches. It also includes a tabulated comparison of the benefits and drawbacks of the latest techniques. The work is on employing a hybrid methodology that integrates deep learning and machine learning to automatically diagnose and assess citrus diseases. Machine vision is predominantly employed in citrus farming to primarily recognize different types of citrus fruits, classify diseases, detect flaws, and evaluate the size and quality of the fruit.

This study[26] utilizes analytical approaches such as deep learning, machine learning, and image processing to analyze and evaluate a comprehensive range of publications on citrus diseases and fruit grading throughout the specified period, providing valuable symptomatic data. This text briefly discusses techniques for pre-processing, segmentation, feature extraction, and classification. It also provides an overview of the advantages and disadvantages of the most recent approaches. The work centers on the autonomous identification and assessment of citrus illnesses with a hybrid approach that combines machine learning and deep learning. Machine vision is extensively employed in citrus farming for disease classification, identification of various citrus fruit kinds, detection of defects, and assessment of fruit size and quality. The CNN model proposed in this paper achieves training, validating, and testing accuracies of 99.43percent, 99.48percent, and 99.58percent, respectively. It surpasses previous CNN models in accurately identifying and categorizing citrus plant illnesses. The proposed design employs a range of convolutional neural network (CNN) models that have been pre-trained to diagnose diseases in citrus plants via transfer learning, with the aim of enhancing classification accuracy rates.

The author of the paper[27] focuses on the identification and categorization of illnesses affecting citrus plants using image processing and computer vision techniques. The dataset included in the research study comprises of photos depicting both disease-free and diseased citrus oranges. The researchers utilized the dataset to discern and classify citrus diseases using a systematic four-step procedure, including dataset refinement, lesion segmentation, feature extraction, and classification. The scientists utilized a range of methodologies, including the Saliency map, Gaussian function, weighted segmentation, and Top-hat process, to improve and divide the photos. Ultimately, each instance of a picture was categorized utilizing the classification technique according to the illnesses class.

3.2 Background Study

A multitude of illnesses can manifest in Citrus Plants and create issues. Canker, Black Spot, and Greening are often occurring illnesses. Traditionally, the detection of citrus leaf diseases has mostly relied on visual examinations conducted by agricultural experts. The existing inspection approach is distinguished by its substantial labor demands, protracted duration, and susceptibility to human errors.Computer vision technology have made identification simpler in recent times.The primary objective is to identify any kind of problem at the earliest feasible stage.A comprehensive understanding of the diseases and the associated risk factors is essential for accurate detection and diagnosis of diseases.

3.2.1 Diseases

i) Canker

Citrus plants are susceptible to a condition called citrus canker, which causes the shedding of leaves and premature fruit falling to the ground. The ailment impacts several types of citrus fruits. Citrus canker has its most severe manifestations in regions characterized by high temperatures and high humidity. The disease can be transmitted between trees by irrigation or rain splashing, as the afflicted regions of the plants release sap. Citrus canker can rapidly and extensively spread when sick citrus fruits come into touch with other citrus fruits, flora, humans, and equipment. Given the absence of any known remedy for the disease, it is imperative to eliminate all affected trees and proceed with a thorough replanting of the orchards. Bacterial infections are the primary cause of citrus tree canker diseases. The most prevalent offenders are Xanthomonas bacteria, particularly Xanthomonas axonopodis. The bacteria infiltrate plant tissues via wounds or natural holes, inducing diseases. Canker infections have the potential to harm citrus trees, resulting in damage to both the fruit and foliage[28]. Citrus canker frequently presents as little, elevated lesions on the leaves. Initially, these lesions are immersed in water, but as they develop, they acquire a firm texture and have a characteristic ring-shaped arrangement. The tree's foliage may be shed due to the presence of lesions, thereby compromising its general well-being.

Citrus trees facilitate the entry of bacteria through the stomata, which are tiny openings in the leaves and stems, as well as through wounds caused by activities like as pruning and insect feeding. Upon entering the plant, the bacteria rapidly proliferate and disseminate, resulting in leaf infection. One of the signs is the presence of circular water-soaked lesions that may increase in size over time. The lesions may have a wet periphery and seem raised and spiky. Cankers may develop when the bacteria invades twigs and stems. Cankers are areas of necrotic or decayed tissue that might stimulate the regrowth of dead branches. Citrus canker can also cause damage to fruit. Lesions can develop on the surface of a fruit when germs enter through wounds or naturally occurring openings. The lesions may have a profound, cork-like look and appear soaked with water. Precipitation, specifically rain carried by the wind, as well as contaminated equipment or accessories, might facilitate the dissemination of the bacteria from infected trees to uninfected ones. Citrus canker is prone to

Figure 3.2: Citrus Canker

thrive and propagate in climates characterized by high temperatures and humidity. The bacteria flourish in these environments, and periods of intense rainfall or irrigation heighten the likelihood of infection.Treating bacterial illnesses such as citrus canker becomes challenging after a plant becomes infected, as the bacteria deeply infiltrate the plant tissues and may exhibit resistance to chemical treatments.Citrus canker has no treatment, however care can slow its growth. These methods lessen illness and prevent infection.Common management strategies include:

1. Pruning: Removing fruit, leaves, and other disease-causing plant detritus reduces germs and prevents sickness.
2. Copper-Based Sprays: These plant-surface bactericides reduce microbial populations. These sprays are best used early in an illness or as preventative measures.
3. Strict Quarantine regulations: Quarantine regulations can help stop unhealthy citrus plants and fruit from spreading. This is crucial to stopping the disease.

ii) Black Spot

Phyllosticta citricarpa is the main cause of citrus black spot, a fungal infection that reduces both the number and quality of citrus fruit on plants. The fungus causes unattractive skin blisters on the fruit, hence diminishing its marketability. The fruits that are considered to be the most delicate are grapefruit, lemons, navel and Valencia oranges, sour oranges, and their respective varieties. Citrus black spot is not seen on lemons with a rough texture or high acidity. Black spots manifest in several forms, including pathogenic, firm, cracked, and pseudo-melanose blemishes. Every location possesses a unique set of symptoms. The hard spot is the most prevalent and practical indicator. One may observe small, melancholic, spherical sores on the outer layer, characterized by gray centers and brick-red borders.

The disease primarily targets delicate and underdeveloped foliage, and its detrimen-

Figure 3.3: Citrus BlackSpot

tal effects can have adverse consequences on the overall well-being and productivity of citrus plants. The appearance of small, water-filled blisters on the surface of young leaves indicates the initial manifestation of black spot disease. Initially, it may be difficult to discern these lesions.As the condition advances, the lesions become increasingly conspicuous and acquire a unique raised, cork-like appearance. The central area of the lesion undergoes a color change from dark brown to black.Lesions on citrus leaves can impair chlorophyll, the pigment essential for photosynthesis. Enlargement of lesions on the leaf hinders the leaf's ability to carry out photosynthesis and produce energy for the plant.Black spot is a challenging fungal disease to cure because once it takes hold, the fungus typically infiltrates the plant tissues, making chemical treatments useless.There is no cure for citrus black spot disease, however some management methods may help: fungicides Chemical treatment may be used to prevent or treat black spot sickness. Fungicides prevent fungal growth and tissue contamination. The success of these therapies depends on the setting and time. The orchard's fungal bacteria can be reduced by removing damaged fruit and vegetation. Trimming also improves airflow, reducing disease transmission. Cultural methods that promote orchard cleanliness and suppress fungus may be beneficial. Controlling soil moisture, avoiding overhead watering, and spacing trees are among these methods.

iii) Greening

Citrus greening is a prevalent citrus plant disease worldwide. Once a tree gets diseased, it is incurable. The infection has caused extensive damage to citrus crops, spanning millions of acres, both locally and internationally. However, there is no risk to either humans or animals. The deformed, acrid, and unripe fruits from diseased trees are unsuitable for commercial distribution as either fresh produce or juice. A significant number of infected trees perish within a short span of a few years.

Figure 3.4: Citrus Greening

This disease has a detrimental impact on citrus trees, leading to significant harm to the leaves and other parts of the plant. An unmistakable indication of citrus greening is the discoloration of the leaves, resulting in a yellow hue. The occurrence of uneven yellowing, often confined to one side of the leaf, results in the leaf acquiring a mottled appearance. Citrus greening causes the leaf to exhibit variable yellowing. Instead, it typically leads to an asymmetrical pattern characterized by one side of the leaf being more yellow than the other. The disease is mostly diagnosed based on its asymmetry.Citrus greening hampers a tree's capacity to generate chlorophyll, the pigment responsible for the green color of leaves. Consequently, the leaves that are impacted exhibit a reduction in their green hue, resulting in the visible yellowing. Citrus greening is subtle and has the potential to impact several components of the citrus tree, hence presenting difficulties in its treatment. Traditional antibiotics are ineffective in reaching the bacteria due to its location within the phloem tissue. Scientists and researchers are now developing a range of management and mitigation measures for citrus greening. Several methodologies encompass:

Controlling the population of Asian citrus psyllids is of utmost importance due to their role in spreading the bacterium responsible for citrus greening. This entails employing pesticides, implementing biological control techniques, and employing other ways to diminish the population of psyllids. Enhancing the tolerance or resistance of citrus cultivars to citrus greening is a prominent subject of scientific inquiry. While type resistance can help lessen the impact, it should be noted that it is not a kind of treatment.

Chapter 4

Methodology

The research presented here gives a lot of information about how to use image processing to find diseases on citrus leaves. It explains in detail how to get useful information from computer pictures so that diseases can be correctly identified. As computer vision and machine learning have improved, the suggested method automates the finding process. This makes it more useful than looking at things manually. Getting the data, pre-processing it, achieving features, and using cutting edge image processing techniques are all parts of the method. For making models that can correctly group diseases, it is important to use the power of deep learning, especially convolutional neural networks (CNNs). There is a full breakdown of each step in the sections that follow. These sections show what was done to improve the usefulness, accuracy, and efficiency of the method for tracking citrus leaf diseases.

4.1 Overview about Proposed Model

Initially, in order to illustrate our suggested model, we need to gather the data that includes photos of citrus leaves which are effective in detecting illnesses.In order to increase the total amount of data contained in the dataset that we have achieved, we will need to employ data augmentation.In order for the models to be able to differentiate between the many changes, the next step is to preprocess the photos of the citrus leaves.In the following step, we will divide the data into three distinct parts: the first portion is for training, the second part is for validation, and the third part is for testing, which will be divided 70:15:15.Following that, the normalization, rotation, zooming, shearing, shifting, shuffling, and flipping operations will be used to resize all three components (the Training, Testing, and Validation data separately).Following this, the resized photos will be split into two parts: the first will be used by pretrained models, and the second will be used by our own custom model.The performance of the models will be improved via the use of hyper parameter tuning in order to achieve the best possible result.In the following stage, we will analyze both our unique model and all of the CNN models that have been pretrained.Following that, we will determine the accuracy of the training, the accuracy of the validation, and the accuracy of the testing.Last but not least, we presented all of the findings using a variety of graphical techniques.The actions that we took are depicted in a diagram in figure 4.1.

We employed the Adam optimizer to alter its learning rate for hyperparameter tuning. To enhance the precision of our model, we employed matrix accuracy and Binary Crossentropy as the loss function.The learning rate was maintained at 0.0001 for all pre-trained models. Every model underwent 20 epochs of training.

4.2 DataFlow Diagram

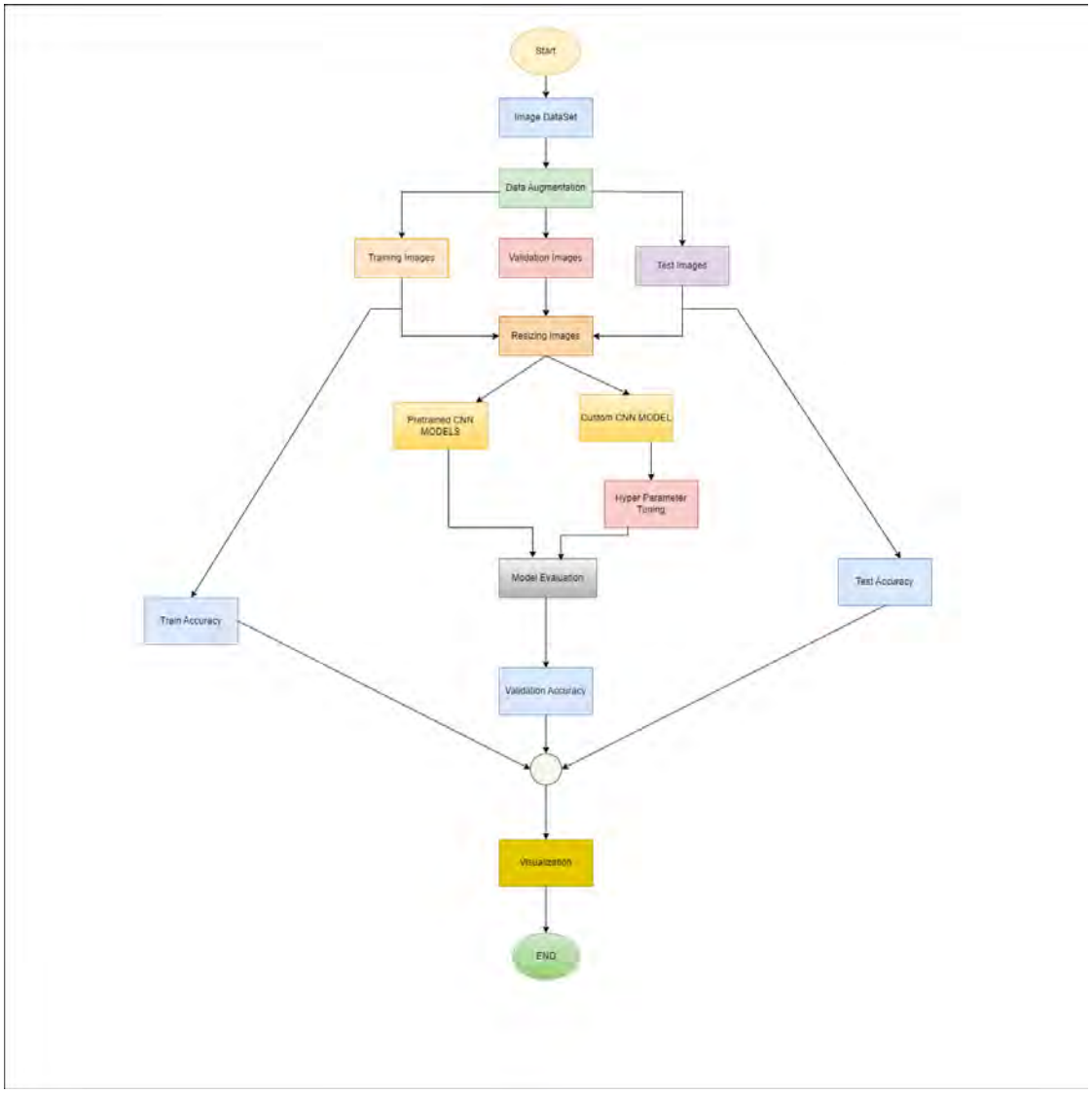

Figure 4.1: DataFlow Diagram

4.3 Dataset

4.3.1 Data Acquisition

As a direct consequence of diseases that attack citrus trees, the quantity of the citrus that can be harvested for use in a broad variety of different businesses has greatly diminished. This is because citrus plants are susceptible to a variety of illnesses. These diseases have contributed to an increase in the number of citrus diseases that have been reported. Researchers and industry experts working in this area came to the realization that the illnesses known as greening, black spot, and canker were the ones that happened the most frequently. Through the use of computer vision and many other forms of deep learning, illnesses that have the potential to affect citrus may be identified upon the basis of the symptoms that they exhibit. This procedure of recognition is capable of being computerized. In this research, a set of data is presented, which consists of 596 photographs that exhibit citrus leaves in both healthy and ill stages. Studies are able to put a broad variety of computer vision and image processing algorithms through their paces by using this collection of images as a testing ground for disease identification in citrus plant life. This collection of photographs acts as a proving ground.

There were three independent occurrences of citrus leaf infections, and each of these diseases were allowed to continue existing on its own. The photos that were impacted were split into these three distinct instances. As illnesses that were included in the data sets, we concentrated our attention on Greening, Canker, and Black Spot as the key topics of inquiry that we were looking at.

Description of data set against each of its disease class (Citrus Leaf)

Diseases	Number of Images
Greening	204
Canker	163
Black Spot	171
Healthy	58
Total	596

Table 4.1: Without Augmentation Dataset

Figure 4.2: Blackspot

Figure 4.3: Canker

Figure 4.4: Greening

Figure 4.5: Healthy

4.3.2 Data Preprocessing

A number of transformations, including rotation, zoom, skew, random distortion, shear, random cropping, and random flipping, were utilized by us during the process of augmenting the image.We have used Augmentator Library to augment our dataset. Our goal is to have 700 photographs shown on that page that are categorized according to each of the categories that it covers. This improvement will directly result in this achieving our goal. Through a process known as "Data Split," 70% of the data was distributed among the Training sets, 15% the Validation sets, and 15% the Testing sets.Therefore, this is the best splitting ratio to use in order to obtain more accurate results from the dataset. Since we needed each of the three sets (Training, Validation, and Testing) to have four sub-classes of folders, this was done in order to achieve our goal. The training sets provide a total of 1956 photos, the validation sets produce 420 images, and the testing sets produce 424 images. The total number of images obtained from these sets is 464. There were a total of 2800 images that were sent to us.Before training the CNN models, all three categories of all the images were rescaled from 0 to 1 pixels using Normalization,roatation,zoom,sheare,shifting, shuffling and flipping.As we are using multiclass classification thats why we have used classmode as catagorical, batch size was kept 32 and target size was (224,224). That's how all the data have been stored using ImageDatagenerator Library.

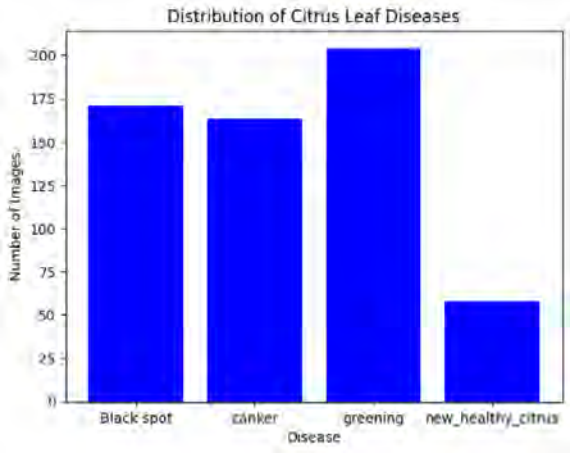

Figure 4.6: DataSet

4.4 Model Description

4.4.1 Convolutional Neural Network(CNN)

A Deep Convolutional Neural Network (CNN) is a specialized kind of Neural Networks that has demonstrated outstanding results in many contests focused on Computer Vision and Image Analysis[29].Convolutional Neural Networks (CNNs) are very advantageous in detecting visual patterns inside pictures for the purpose of object recognition. Additionally, they may be very efficient at categorizing non-image data, such as audio, time series, and signal data.When it comes to a wide range of computer vision tasks, CNN artificial neural networks have emerged as the most successful technique. People's interest in a wide range of subjects has been raised as a result of it. There are several layers that make up a convolutional neural network.

These layers include convolution layers, pooling layers, and fully connected layers. In order to gain knowledge organizational structures of input in an autonomous and adaptable manner, the network employs a reverse propagation technique.In CNN, the kernels are nothing but a way of extracting features from the provided images.Convolutional Layers are the the foundation of the CNN architechture and they are the ones to execute the convolutional operations.While the kernel is waiting for the entire picture to be scanned, it will make modifications in the vertical as well as horizontal directions based on the striding quantity. In comparison to an image, the kernel is smaller, but it has a greater degree of depth.This indicates that the kernel height and width will be very little in terms of space if the image has three RGB paths, yet the depth of the kernel will contain all three channels individually[30].In this layer, the calculation of convolution is carried out between the input image and a filter of a certain size MxM. This filter operates on the input the image. It is possible to calculate the dot product between the filter and the regions of the input picture that are proportional to the size of the filter (MxM) by moving the filter across the image on which it is being applied.In general, the most recent CNN model suggests using an input picture that is 224 pixels by 224 pixels.

In addition to convolution, the Non-linear activation function is an additional significant component of convolutional layers. This component functions in addition

to convolution. A non-linear activation function is applied to the outputs of linear operations such as convolution once they have been processed.

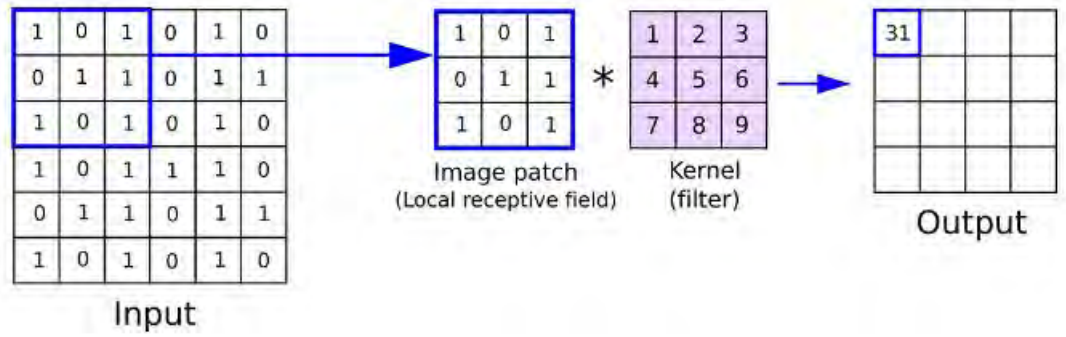

Figure 4.7: Convolution layer

In the past, smooth nonlinear functions like the sigmoid or hyperbolic tangent (tanh) function were employed since they are statistical illustrations of the operations that are carried out by biological neurons. The rectified linear unit, often known as the ReLU, is currently the non-linear activation function that is utilized the most frequently. f(x) = max(0, x/x).

A Convolutional Layer is often followed by a Pooling Layer in the majority of instances. It is the major objective of this layer to lower the size of the convolved feature map in order to cut down on the amount of computing resources required. Specifically, this is accomplished by reducing the connections that exist across layers and performing separate operations on each specific feature map. There are several sorts of Pooling procedures, each of which is determined by the method that is utilized. Additionally, it provides a concise summary of the characteristics that are produced by a convolution layer.One can distinguish between two distinct forms of pooling, namely maximum pooling and average pooling. The largest value that can be extracted from the region of the picture that is covered by the kernel is what is returned by max pooling. Average pooling is a technique that returns the average of all the values that are contained inside the portion of the picture that is covered by the kernel.

The final layer is called Fully Connected Layer.It is the Fully Connected layer that is responsible for connecting the neurons that are located between two separate layers. This layer is made up of the weights and biases in addition to the neurons[31]. Since the fully connected layer (FC) operates with a flattened input, this suggests each input is tied to each and every neuron in the network. Following that, the vector that has been flattened is sent across a few further FC levels, which are typically the locations where mathematical functional operations are carried out. It is at this moment that the categorization process is initiated. In the event that they are present, Fully Connected layers are often located not far from the conclusion of CNN designs.

There are some inner functions that are also used in CNN.One is Dropout and other one is Activation Function.Dropout is a method of regularization employed in neural networks, such as Convolutional Neural Networks (CNNs), to reduce excessive fitting. Overfitting arises if a model excessively captures every detail of the training data, covering its variance and outliers, hence resulting in inadequate generalization on novel, unknown information.Dropout is commonly used in the context of fully connected layers. During the process of prediction, the utilization of dropout is often deactivated.Activation functions are utilized in neural network models to handle non-linear behavior which enables them to acquire knowledge about subtle connections and trends within the information. Activation functions are carried out individually to the output of convolutional and fully connected layers in CNNs.ReLU, Softmax, Tanh, and Sigmoid functions are all examples of activation functions of various kinds.ReLU, Softmax, Tanh, and Sigmoid functions are all examples of activation functions of various kinds.

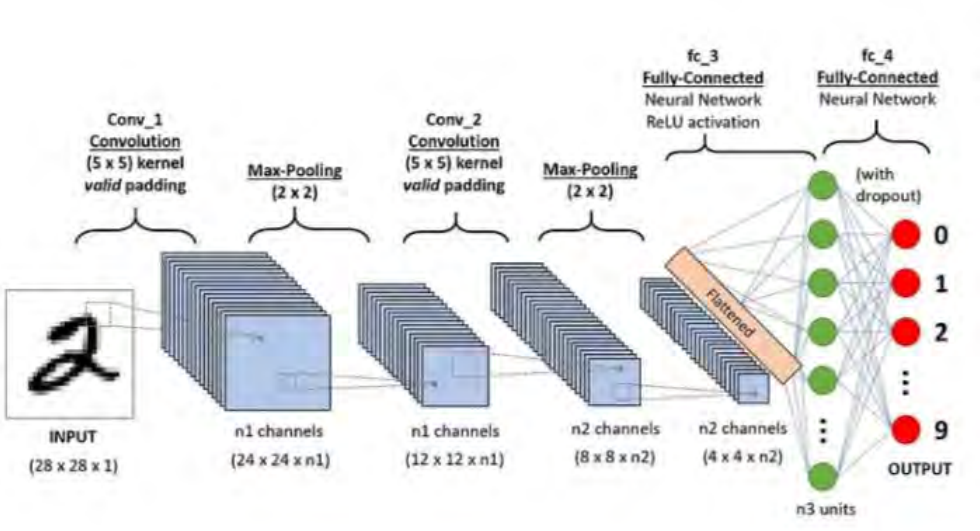

Figure 4.8: Basic Convolutional Neural Network Architecture

4.4.2 Inception-V3

Inception-v3, a member of the Inception family of convolutional neural network architectures, has several enhancements such as Factorized 7 x 7 convolutions, Label Smoothing, and an additional classifier to effectively propagate label information deeper into the network. The Inception V3 deep learning model utilizes convolutional neural networks to do picture categorization. The Inception V3 model is an upgraded version of the original Inception V1 model, representing an improvement. The Inception V3 model is an enhanced and polished version of the Inception V1 idea. The Inception V3 model employed many strategies to improve network optimization and model adaption. The model comprises convolutions, max pooling, average pooling, concatenations, dropouts, and fully linked layers, along with additional symmetric and asymmetric building pieces. The primary reliance of the

model is on batch normalization, which is applied to the inputs of the activation function. The SoftMax function is utilized for loss computation[32].

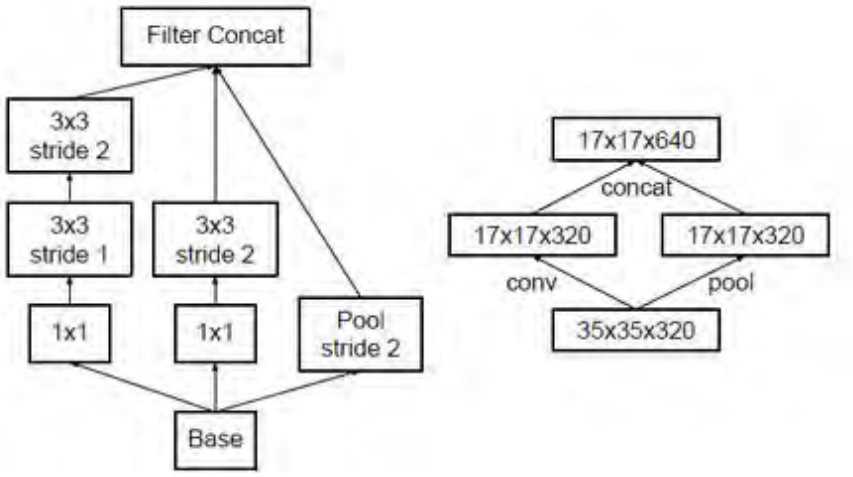

Figure 4.9: Gride Size

The model runs with more efficiency and lower computing intensity compared to the Inception V1 and V2 models. It utilizes auxiliary classifiers as regularizers and exhibits quicker performance, independent of the network's depth. In comparison to previous versions, it consists of a total of 42 layers and has a reduced number of errors overall. There are several improvements that strengthen the Inception V3 model. The Inception V3 model has had the subsequent notable modifications: Partitioning into smaller convolutions, utilizing asymmetric convolutions for spatial combination, implementing additional classifiers, and decreasing the actual grid size.In the past, the grid sizes of the feature maps were reduced by the implementation of max pooling and average pooling techniques.
Following all adjustments, the final Inception V3 model looks like this:

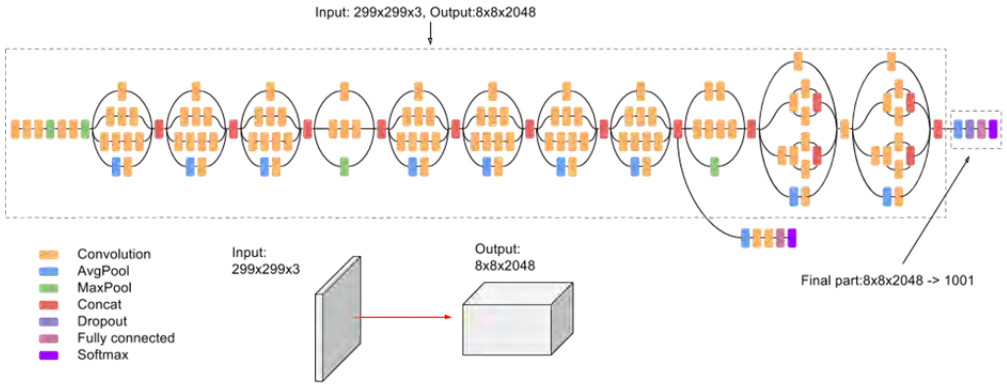

Figure 4.10: Inception V3 Architecture

The Inception V3 model increases the activation dimension of the network filters

to effectively reduce the grid size. Reduction transforms a grid of size d×d with k filters into a smaller grid of size d/2 × d/2 with double the number of filters, such as 2k filters. To do this, convolution and pooling are performed in two separate blocks, which are subsequently combined by concatenation.

4.4.3 Mobile Net-V2

For image classification, agile models with fast analysis speeds without compromising accuracy are essential. MobileNetV2, a well-known competitor, has received a lot of attention. This article examines MobileNetV2's architecture, training program, performance assessment, and practical implementation. The primary goal of MobileNetV2, a lightweight convolutional neural network (CNN) architecture, is to facilitate applications related to mobile and embedded vision. It was developed by Google researchers to build upon the original MobileNet concept. Notably, this kind strikes an excellent balance between accuracy and size, which makes it ideal for low-resource devices.

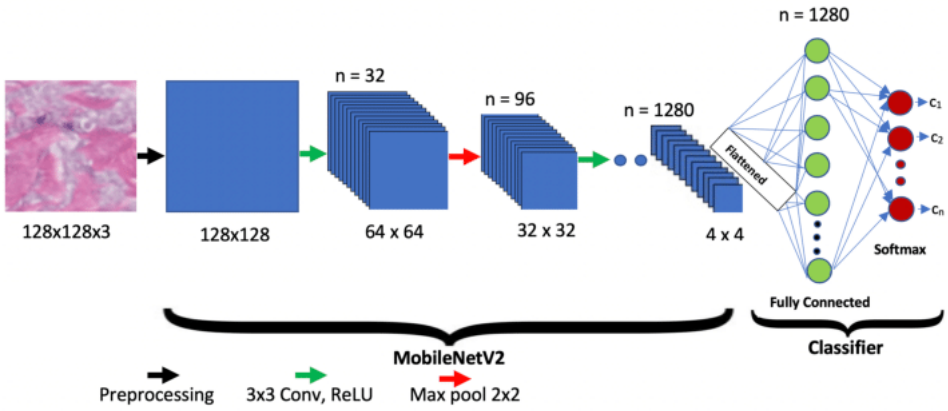

Figure 4.11: MobileNetV2 Architecture

The effectiveness and efficiency of MobileNetV2 in photo classification tasks are influenced by several significant parameters. Among these features are depthwise separable convolution, bottleneck design, inverted residuals, squeeze-and-excitation (SE) blocks, and linear bottlenecks. It takes all of these qualities to maintain the model's high accuracy while reducing its computational complexity. The use of MobileNetV2 for photo classification has several advantages. First off, because of its compact architecture, it may be used efficiently on embedded and mobile devices with little processing power. Second,MobileNetV2 achieves comparable accuracy to larger and computationally more costly models. Lastly, the model's small size, which enables quicker inference times, makes it suitable for real-time applications. The architecture of MobileNetV2 consists of squeeze-and-excitation blocks, bottleneck design, linear bottlenecks, inverted residuals, depthwise separable convolutions, and many convolutional layers. When combined, these components help the model capture complex features using fewer computations and inputs.To reduce the computational cost of convolutions, MobileNetV2 employs a technique known as depthwise

separable convolution. Pointwise and depthwise convolution are the two separate processes that result from this division of the traditional convolution. The model becomes more effective when the data are divided since it has to do fewer computations overall.Inverted residuals are a key component of MobileNetV2 that improves the model's accuracy. To increase the number of channels, a bottleneck structure is built prior to applying depthwise separable convolutions. This addition increases the representational capability of the model and enables it to capture more complex elements.

4.4.4 Inception-Resnet V2

The Inception Resnet V2 architecture is a kind of convolutional neural network (CNN) designed specifically for image recognition tasks. The technology was devised by Google researchers and is renowned for its exceptional precision and effectiveness[33].A combination of the Inception structure and the Residualconnection is the basis for the formulation of the Inception-Resnet-v2 algorithm. Multiple sized convolutional filters are merged via residual connections within the Inception-Resnet phase of the neural network pipeline.

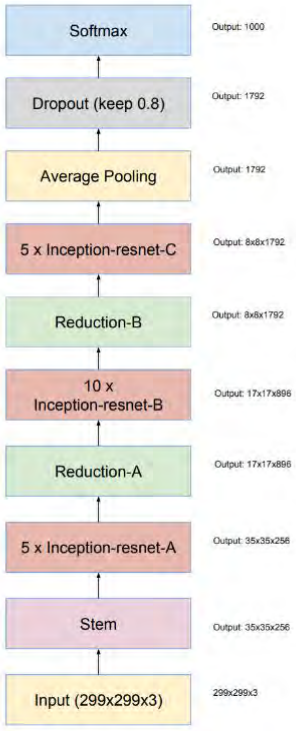

Figure 4.12: Inception-Resnet V2 Architecture Layers

Convolutional layers serve as the fundamental components of Convolutional Neural Networks (CNNs) and are tasked with extracting information from the input picture. The Inceptionv3 architecture consists of many convolutional layers organized

into modules known as "inception modules."In order to extract important characteristics from the pictures, the Inception-Resnet-v2 employs a complex design[34]. Inception modules are the fundamental innovation of the Inceptionv2 architecture. The network employs parallel convolutional filters of varying sizes (1x1, 3x3, and 5x5) to extract information at numerous scales concurrently. This enhances the precision and effectiveness of the model. Max pooling and average pooling layers: These layers decrease the dimensions of the feature maps by selecting the highest or average value from a small group of pixels. This aids in managing the intricacy of the network and mitigating overfitting. Fully linked layers are utilized to generate predictions by using the retrieved information. The Inceptionv2 architecture consists of two fully connected layers, which are then followed by a softmax layer. This softmax layer is responsible for producing the probability of each class in the ImageNet dataset, which has a total of 1000 classes. The figure additionally has supplementary components, including:
Dropout layers are utilized in neural networks during training to mitigate overfitting by randomly deactivating a portion of the units. Batch normalization is a method that enhances the stability of the training process and increases the accuracy of the model.

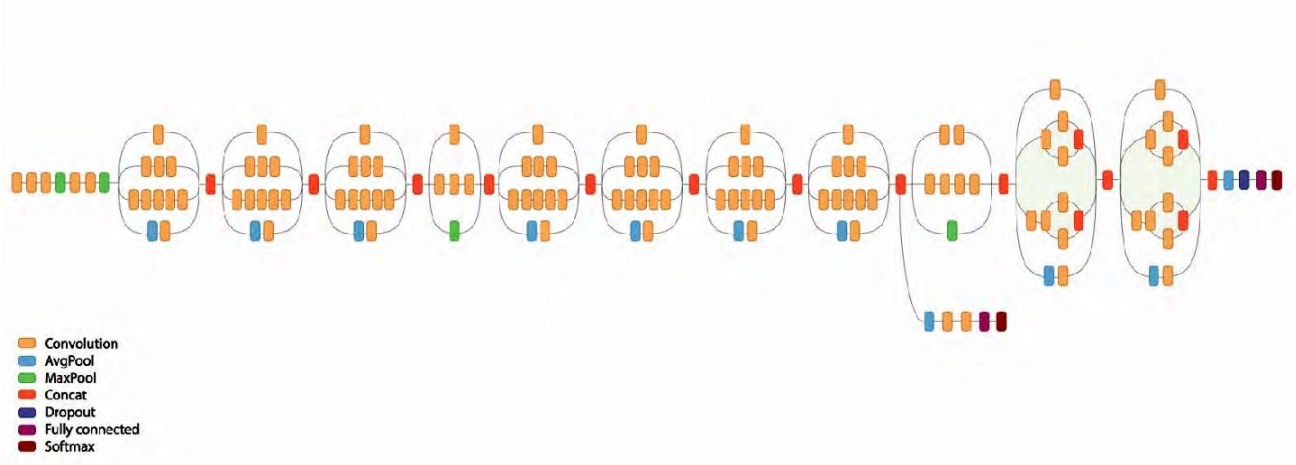

Figure 4.13: Inception-Resnet V2 Architecture

Auxiliary classification branches refer to supplementary branches of classification that are incorporated into some conception modules. These branches contribute to enhancing the precision of the ultimate forecasts. In summary, the Inceptionv2 architecture is a robust and effective convolutional neural network (CNN) that has attained cutting-edge performance on image identification tasks. The architecture is intricate, but the diagram you gave offers a comprehensive picture of its essential elements.

4.4.5 DenseNet 121

Each convolutional layer (apart from the first) in a conventional feed-forward convolutional neural network (CNN) receives input, analyzes the output of the preceding convolutional layer, and generates an output feature map that is then passed on to the convolutional layer that comes after it. As a result, for 'L' layers, every layer and the one that comes after it have 'L' direct connections.

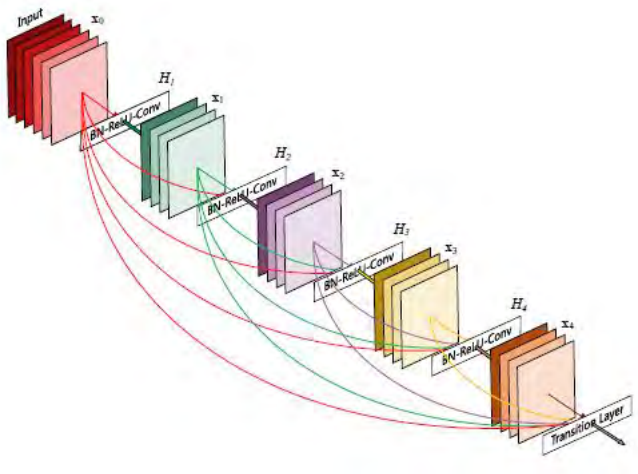

Figure 4.14: DenseNet121 Architecture

However, as the CNN goes deeper—that is, as the number of layers increases—the "vanishing gradient" problem becomes apparent. This suggests that as the channel that carries information from the input to the output layers gets longer, some information can "vanish" or be lost, which would reduce the network's ability to train efficiently.DenseNet-121 fixes this problem by modifying the traditional CNN architecture and simplifying the layer-to-layer link architecture. Convolutional networks with dense linking resemble the DenseNet design, in which all layers are connected to all other layers directly. For 'L' layers, there are L(L+1)/2 direct connections.

Using the DenseNet-121 architecture, we can see that each dense block has two convolutions per layer, which helps us understand the table better. A 1x1 kernel occupies the bottleneck layer and a 3x3 kernel performs the convolution process.DenseNet-121 comprises 120 convolutions and four average pools.Because all layers distribute their weights over several inputs, even those within the same dense block and transition layers, deeper levels can use characteristics that were recovered early on.Because they generate a lot of duplicated data, the layers in the second and third dense blocks give the transition layers' output the lowest weights[35].

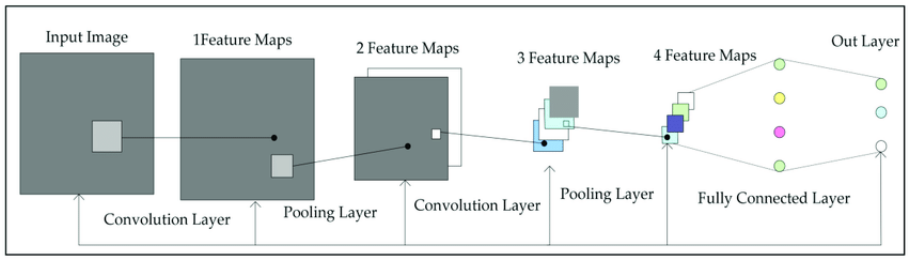

Figure 4.15: DenseNet121 Architecture Layers

4.4.6 DenseNet 201

Gao Huang, Zhuang Liu, and Laurens van der Maaten created DenseNet-201, which is an enhanced version of DenseNet-121, a very effective architecture for convolutional neural networks. The design of image classification addresses issues such as feature reuse, vanishing gradient difficulties, and model compactness[36]. DenseNet-201's intricate architecture comprises of the input layer, beginning convolutional layer, dense blocks, bottleneck layers, transition blocks, global average pooling (GAP), and fully connected final classification head. The gateway, also known as the input layer, is designed to receive 224x224-pixel images for various image classification tasks. The primary convolutional layer performs convolutions on the input data to collect essential features.

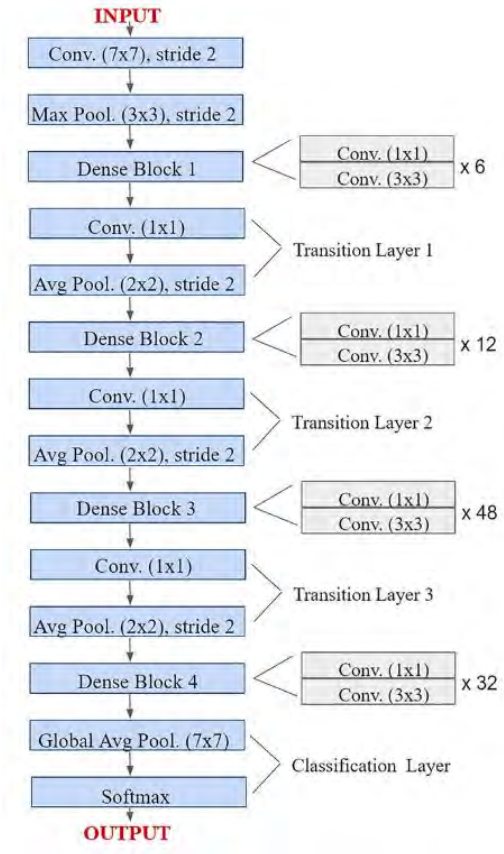

Figure 4.16: Layered Architecture of DenseNet201

DenseNet-201 utilizes thick blocks to accommodate a large number of dense layers. Bottleneck layers deliberately reduce the number of input feature maps in each dense layer before applying 3x3 convolutions, with the aim of enhancing computational efficiency. Strategically positioned transition blocks can decrease the spatial dimensions of feature maps. Feature map compression and down sampling are achieved by the utilization of 1x1 convolutions and average pooling. Once the dense blocks have been finished, the global average pooling operation calculates the average value of each feature map, resulting in a reduction of the spatial dimensions to 1x1. The final classification head utilizes a thick layer with softmax activation to produce the probability distribution for the target class.

4.4.7 ResNet50

The ResNet begins with ResNet-50, a pioneering design that uses smooth shortcut connections to turn a basic network into its residual counterpart. ResNet-50 distinguishes itself from other neural networks by using 3x3 convolutional networks and pulling inspiration from VGG neural networks, such as VGG-16 and VGG-19. ResNets differ from VGGNets in that they reduce filter count and complexity. Because of its fifty weighted layers, ResNet-50 can achieve 3.6 billion FLOPs. This is far better than the 1.8 billion FLOPs of its smaller 18-layer competitors. Layers follow simple design rules to maintain filter counts for identical output feature map sizes. With the feature map size decreased in half, the number of filters doubles, but each layer's temporal complexity is maintained.Due to ResNet-50 shortcut connection improvements, input and output dimension decision-making is improved. Identity shortcuts provide a direct path for stable dimensions, making them an appealing choice for expanding dimensions. A important choice is whether to continue identity mapping while padding extra zero entries for dimension augmentation or employ the projection shortcut to align dimensions harmonically.

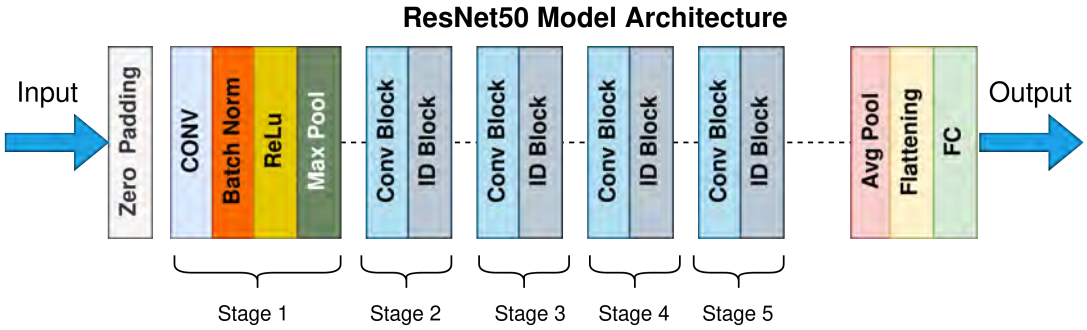

Figure 4.17: Resnet50 Architecture

Deep Residual Learning, which is part of the ResNet paradigm, may tackle challenging issues when training incredibly deep neural networks. ResNet's utilization of residual blocks and "skip connections." is its most crucial feature. These links reduce the hazards of disappearing gradients by providing an alternate gradient flow channel and promoting identity function acquisition.The mutually advantageous link

between skip connections and residual blocks allows ResNet to enhance top layers, guaranteeing that they operate similarly to lower ones. This unique trait streamlines identity function learning, improving deep neural network performance. The novel inclusion of skip connections combines outputs from previous levels with those of stacked layers. This enables deeper network training than ever before.ResNet has several variations, including ResNet-50, which has fifty neural network layers. This numerical distinction shows how ResNet can handle a broad range of model complexity while adhering to deep residual learning principles.

4.4.8 VGG16

VGG16, a Convolutional Neural Network (CNN) created by Karen Simonyan and Andrew Zisserman at the University of Oxford, is widely regarded as a significant advancement in the domain of computer vision. While it did not win the ImageNet Challenge in 2014, its impact on the field and future progress cannot be ignored.The key components of a convolution neural network are included in the VGGNet design[37]. Before delving into VGG16, it is crucial to acknowledge notable advancements in the realm of computer vision. Commencing with Kunihiko Fukushima's "neocognitron" in 1980, subsequently accompanied by Yann LeCun's innovation of backpropagation in 1998, and the establishment of ImageNet in 2011, a sequence of noteworthy breakthroughs has been vital in propelling the area forward. AlexNet, GoogLeNet, and ResNet have had a profound impact on the advancement of computer vision. The VGG16 model, introduced in 2013 and participated in the ImageNet Challenge in 2014, provided a unique approach. VGG16 distinguishes itself from earlier models by constantly utilizing small 3×3 receptive fields throughout the whole network, with a stride of 1 pixel. The use of several 3×3 filters offered flexibility, enabling the formation of receptive fields with larger dimensions, resulting in more effective outcomes[38].

The "D" configuration of VGG16 demonstrated outstanding performance on the ImageNet dataset. The network's architecture was streamlined by the regular utilization of 3 x 3 convolutions. The configurations involved the arrangement of convolution layers with 3 x 3 filters, followed by max-pooling layers, and concluding with completely connected layers.
The VGG16 design has five convolutional blocks, each including many layers, and is then followed by three fully connected layers. The input comprises an image with dimensions of 224×224 x 3, and the network progressively reduces the image size through the utilization of max-pooling. The subsequent layers employ a softmax activation function to facilitate the classification of categories. In order to construct a VGG16 model using Keras, it is necessary to specify a sequential model and include convolutional and pooling layers, followed by fully connected layers. Dropout layers are employed for regularization. While it is possible to utilize a pre-trained model using ImageNet weights, commencing from scratch allows for customization of the model to meet unique requirements.
VGG16 is characterized by a significant parameter count, exceeding 138 million. The complexity of this activity requires significant computer resources for training.

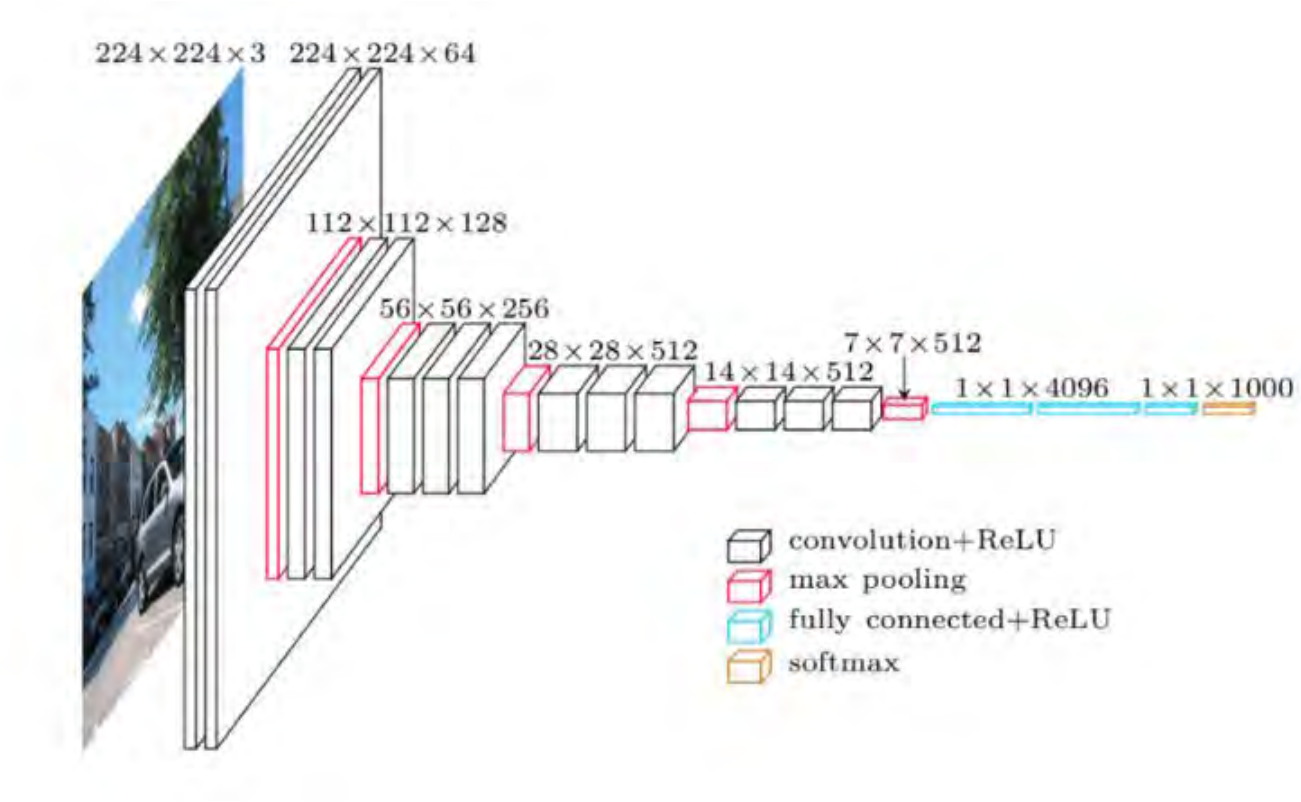

Figure 4.18: VGG16 Architecture

Despite not winning the ImageNet Challenge, VGG16's simplicity, improvement, and capacity to transfer information have solidified its significance in the field of computer vision.
Although VGG16 did not secure the top position in the ImageNet Challenge, its impact on the field of computer vision is undeniable. The utilization of 3 x 3 convolutions and combinations such as "D" lays the groundwork for future advancements. VGG16 is a notable achievement in the ongoing effort to enhance computers' capacity to understand visual data.

4.5 Custom Model

At the first level, there exists a convolutional layer. The layer consists of 32 filters with a kernel size of (3,3). The activation procedure has incorporated the utilization of a function called ReLU. The output shape was determined to be (222, 222, 32), whereas the input shape was found to be (224, 224, 3). The architecture of this layer consists of a total of 896 parameters.

The second layer's input form is (224,224,32), and its pool size is (2, 2). This is done in order to down sample the MaxPooling layer. The output shape with dimensions (111,111, 32) is obtained by doing so. Because there are no parameters that can be trained of this sort, it has parameters rather than none at all.A total of 64 filters with a kernel size of (3,3) are included in the convolutional layer, which is the third layer. A function known as ReLU has been applied as an activation function. The dimensions of the input shape are measured as (111,111,32), while the dimensions of

the output form are measured as (109,109,64). The total number of parameters in this layer is 18,496. In order to get the output shape with dimensions of (54,54,64), the input form for the fourth layer, which is referred to as the MaxPooling layer, is (109,109,64). The pool size comes in at (2, 2). It does not have any parameters because we do not have any trainable parameters of this kind.

The convolutional layer of the fifth layer consists of 128 filters, each with a kernel size of (3,3). We utilized an activation function known as Rectified Linear Unit (ReLU). The input shape has dimensions of (54,54,64), whereas the output form has dimensions of (52,52,128). This layer has a total of 73,856 parameters. The MaxPooling layer, positioned as the sixth layer, takes an input shape of (52,52,128) and does a pooling operation with a pool size of (2, 2). Next, determine the measurements of the resulting form, which are (26,26,128). Since there are no trainable parameters, it is devoid of any parameters.

The seventh layer, known as the convolutional layer, consists of 512 filters with a kernel size of (3,3). We utilized an activation function known as Rectified Linear Unit (ReLU). The dimensions of both the input shape and the output form are (24,24,512). The input shape is a three-dimensional tensor with dimensions of 26x26x128. This layer has a grand total of 590,336 parameters.

The input for the MaxPooling layer, which is the eighth layer, is in the form of (24,24,512). The layer has a pool size of 2x2.Furthermore, the dimensions of the output form are acquired as (12,12,512). Given the absence of any trainable parameters, it may be concluded that it lacks any parameters.

The Ninth layer, known as the convolutional layer, consists of 256 filters with a kernel size of (3,3). We utilized an activation function known as Rectified Linear Unit (ReLU). The input shape is (12,12,512), whereas the output shape is (10,10,256). The totality of this layer consists of a total of 1,179,904 parameters.The pool size is (2, 2) and the input shape for the MaxPooling layer, which is the Tenth layer, is (10, 10, 256).Furthermore, the dimensions of the output form are acquired as (5,5,256). Given the absence of trainable parameters, it may be concluded that there are no parameters associated with it.

The GlobalAveragePooling layer, positioned as the eleventh layer, takes in an input shape of (5,5,256) and utilizes a pool size of (2, 2).In addition, it is possible to obtain the dimensions of the output form, which are (256). Given the absence of trainable parameters, it may be concluded that there are no parameters associated with it.

The layer undergoing flattening is the twelfth layer. The resulting shape's dimension is obtained by this technique. As it does not employ backpropagation learning, this model lacks any trainable parameters. Instead, it just calculates an integer that is given as input.
The Thirteenth layer in the structure is a Dense layer with 512 units. The dimension of the output shape is determined by this method. We utilized an activation function known as Rectified Linear Unit (ReLU).This layer has a grand total of 131,584 parameters.

The Fourteenth layer consists of four densely packed units. By doing this, it obtains the (4,) dimension of the resulting shape.We utilized an activation function known as Rectified Linear Unit (ReLU).This layer has a total of 2,052 parameters.The Softmax function is utilized in the last layer of a Convolutional Neural Network (CNN) to provide a probability distribution that contains all possible classes.

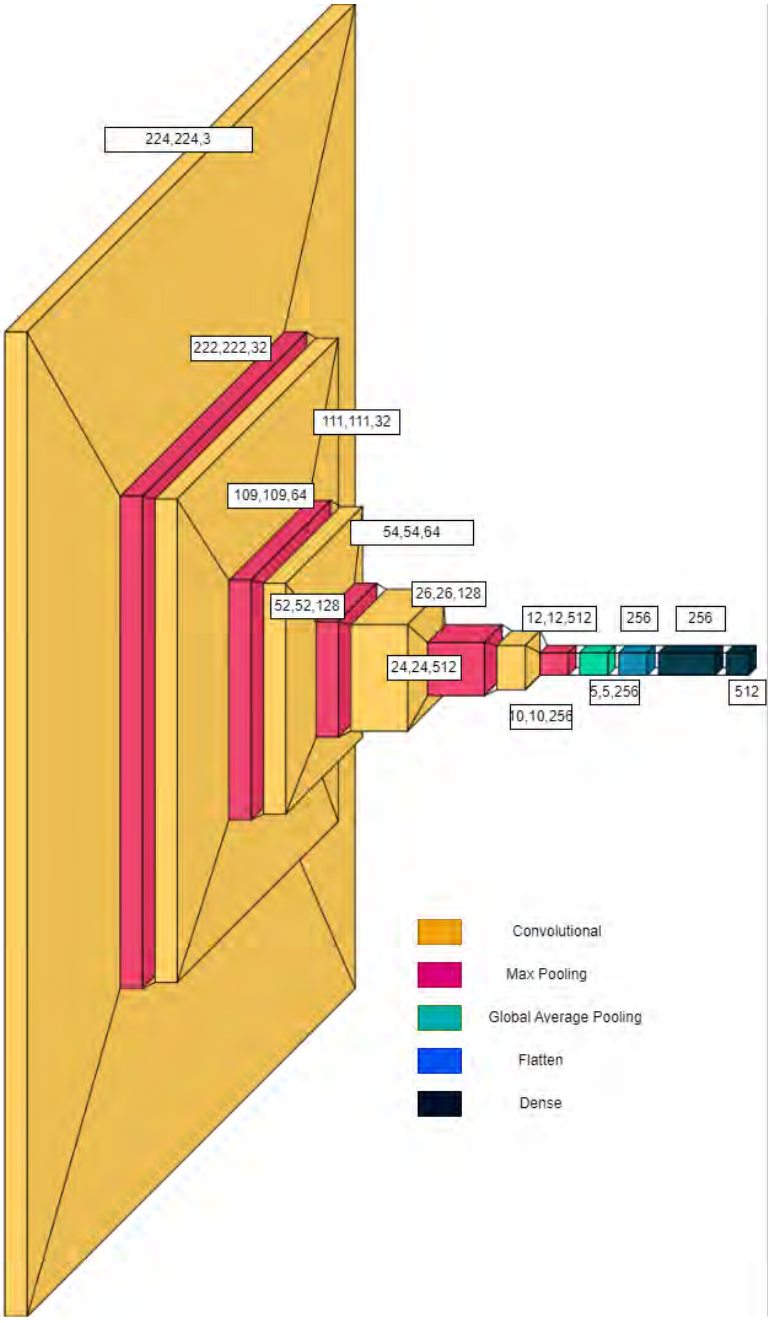

Figure 4.19: Custom Architecture

4.6 Interpreting LIME

Local Interpretable Model-agnostic Explanations, abbreviated as LIME. LIME prioritizes describing the model's prediction for specific occurrences rather than offering a comprehensive comprehension of the model across the whole dataset[39].The primary goal of LIME is to provide explanations for individual predictions that are both understandable and precise within the relevant local context. LIME utilizes surrogate models to accurately comprehend and replicate the behavior of complex models inside a specific instance's domain.

To begin the method, it is necessary to choose a particular occurrence for which an explanation is being sought. In the subsequent stage, LIME will generate a dataset comprising of altered samples around this particular occurrence by introducing controlled and minuscule alterations to the input characteristics. The complicated model is used to generate predictions for these modified samples, resulting in the generation of a labeled dataset that may be used to train a surrogate model.Lime maintains model-agnosticism by refraining from accessing the internal workings of the model. To determine the specific elements of the interpretable input that influence the estimate, we manipulate the input within the immediate area and observe the behavior of the model's predictions.

Subsequently, a simple and readily understandable model, such as a linear model, is trained on the dataset that was generated. From a local standpoint, this surrogate model offers an estimation of the behavior shown by the intricate model. Within this interpretable model, the coefficients function as a rationale for the forecast provided by the original model at the specific moment being examined.

This is particularly crucial in scenarios where the comprehensibility of the model is paramount. The objective of LIME is to provide insights into the decision-making process of complex models, hence enhancing their transparency and comprehensibility. Interpretability is highly valuable in building confidence in machine learning models, especially when decisions impact individuals or organizations.

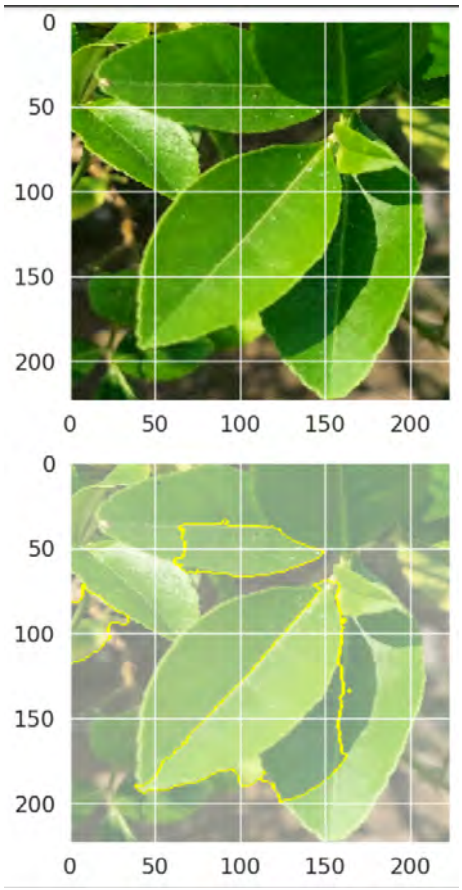

Figure 4.20: Healthy Leaf

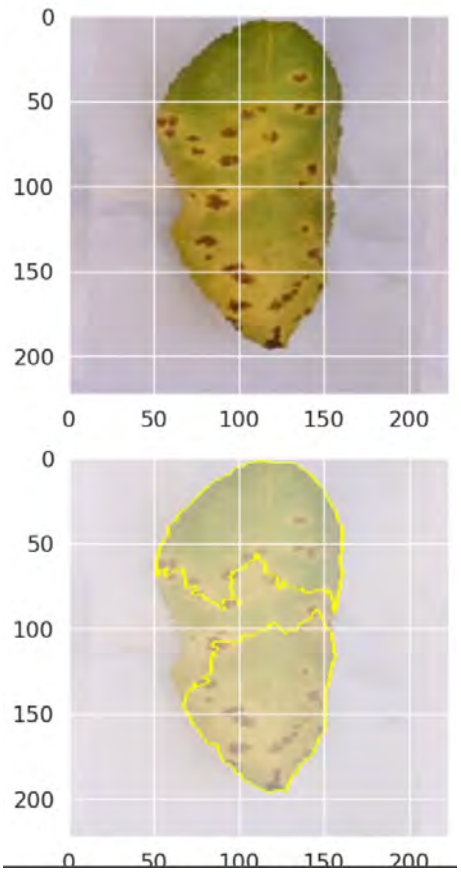

Figure 4.21: Canker

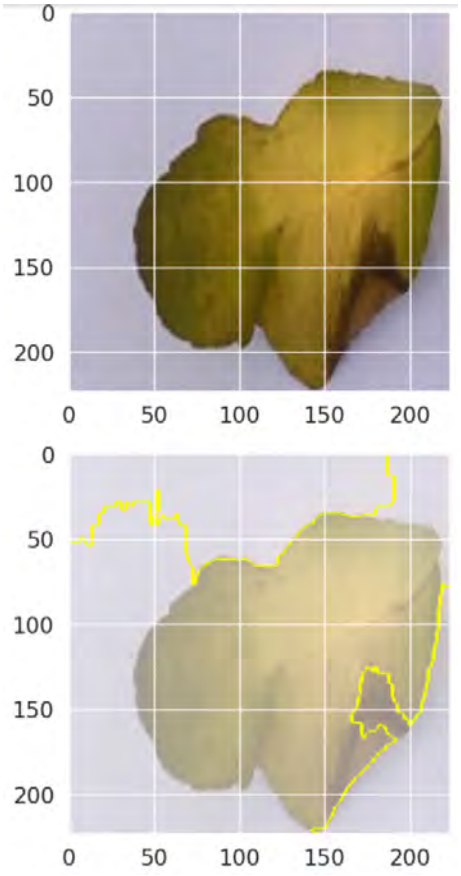

Figure 4.22: BlackSpot

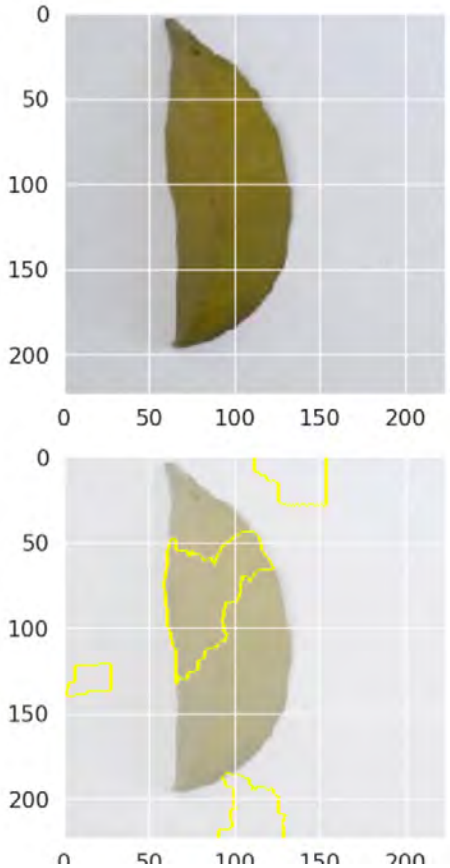

Figure 4.23: Greening

Chapter 5

Result Analysis and Discussion

5.1 Resnet50 Implementation Result

In terms of accuracy and validation, Resnet-50 exhibits a remarkable accuracy of 99.70%, with enhanced accuracy coming in slightly lower at 99.49%. The findings of the validation show that the accuracy of the unaugmented version dropped significantly to 28.74%, but that the augmented version showed a strong rebound to 98.57%. When compared to the F1 Score without augmentation, which is particularly low at 11.00%, the F1 Score with augmentation is exceptionally high at 99.00%Å similar trend can be seen in recollection, with un-enhanced recall coming in at 25.00% and augmented recall coming in at 99.00% total. Over the course of the augmentation process, precision demonstrates a significant improvement, increasing from 7.00% to 99.00%. During the process of augmentation, validation loss receives a dramatic reduction, falling from 2.97% to 3.66% percent.

Resnet50	UnAugmented	Augmented
Accuracy	99.70%	99.49%
Validation Accuracy	28.74%	98.57%
Test Accuracy	28.70%	98.60%
F1-score	11.00%	99.00%
Recall	25.00%	99.00%
Precision	7.00%	99.00%
Validation Loss	2.97%	3.66%

Table 5.1: Results Resnet50

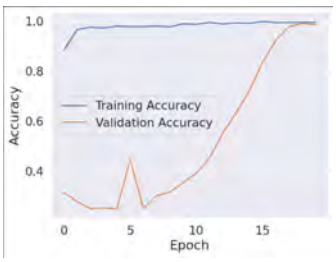

Figure 5.1: AugmentedResnet50

Figure 5.2: UnAugmentedResnet50

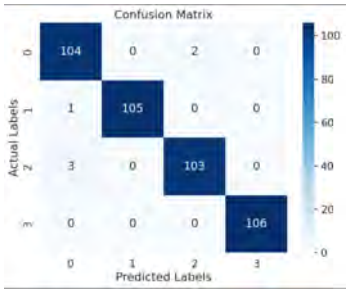

Figure 5.3: AugMented Resnet50 ConMatrix

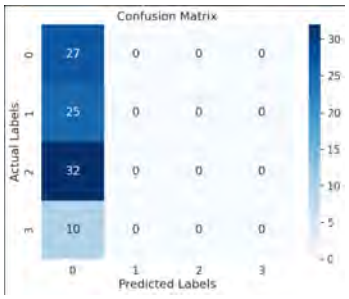

Figure 5.4: UnAugMented Resnet50 ConMatrix

5.2 MobileNetV2 Implementation Result

MobileNet-V2 succeeds in achieving an un-augmented accuracy of 97.35% and an augmented accuracy of 99.03% without any enhancements. The findings of the validation reveal a tremendous rise, beginning at 35.63% and ending at 84.52% with augmentation. "F1" The scores of 88.00%, 88.00%, and 91.00%, respectively, indicate that the score, recall, and accuracy all see significant gains as a result of the augmentation.
Validation loss lowers dramatically from 45.38% to 51.60% with augmentation.

MobileNetV2	UnAugmented	Augmented
Accuracy	97.35%	99.03%
Validation Accuracy	35.63%	84.52%
Test Accuracy	34.00%	88.00%
F1-score	13.00%	88.00%
Recall	25.00%	88.00%
Precision	9.00%	91.00%
Validation Loss	45.38%	51.60%

Table 5.2: Results MobileNetV2

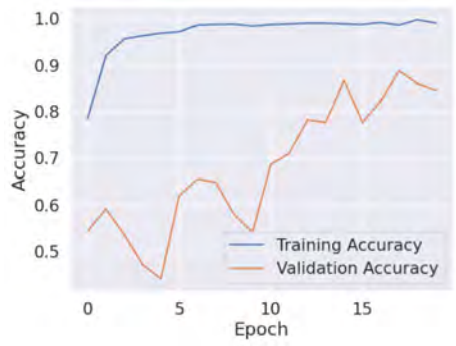

Figure 5.5: AugmentedMobilenetV2

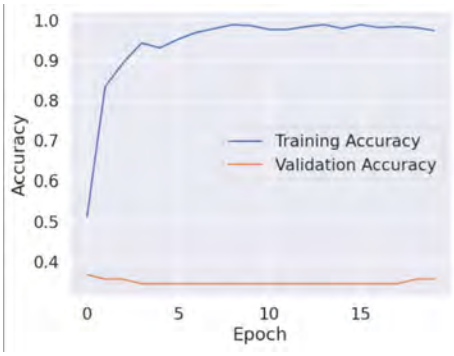

Figure 5.6: UnAugmentedMobilnetV2

Figure 5.7: Augmented-MobilenetV2 ConMatrix

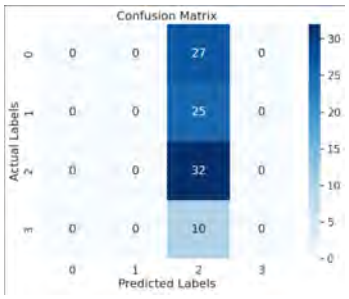

Figure 5.8: UnAugmentedMobilnetV2 ConMatrix

5.3 VGG16 Implementation Result

In terms of accuracy and validation, the VGG-16 displays a high level of accuracy, with un-enhanced and augmented accuracies coming in at 94.22% and 98.80%, respectively. Using augmentation results in a marginal improvement in validation accuracy, which goes from 90.80% to 98.33%. The F1 Score, recall, and accuracy all show values about 98.00% with augmentation, indicating that there is consistency in the performance parameters that are being examined. With augmentation, the validation loss goes from 34.84% to 6.75%, a significant drop.

VGG16	UnAugmented	Augmented
Accuracy	94.22%	99.00%
Validation Accuracy	90.80%	98.33%
Test Accuracy	82.00%	98.30%
F1-score	82.00%	98.00%
Recall	82.00%	98.00%
Precision	86.00	98.00%
Validation Loss	34.84%	6.75%

Table 5.3: Results VGG16

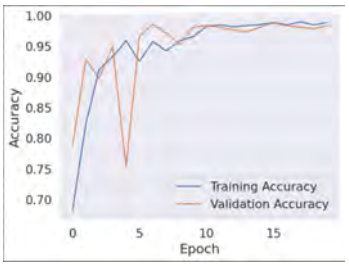

Figure 5.9: Augmented-VGG16

Figure 5.10: UnAugmentedVGG16

Figure 5.11: AugMented VGG16 ConMatrix

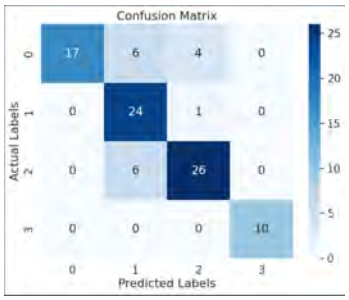

Figure 5.12: UnAugMented VGG16 ConMatrix

5.4 InceptionV3 Implementation Result

Inception-V3 is capable of achieving high accuracies of 99.28% (un-augmented) and 99.85% (augmented). This pattern is reflected in the validation findings, which increased from 98.25% to 98.81% as a result of the augmentation. During the process of augmentation, performance measurements experience slight enhancements, with the F1 Score, recall, and precision all hovering around 99.00%. With augmentation, the validation loss goes from 12.43% to 4.15%, a significant decrease.

InceptionV3	UnAugmented	Augmented
Accuracy	99.28%	99.85%
Validation Accuracy	94.25%	98.81%
Test Accuracy	90.40%	99.00%
F1-score	90.00%	99.00%
Recall	90.00%	99.00%
Precision	91.00%	99.00%
Validation Loss	12.43%	4.15%

Table 5.4: Results InceptionV3

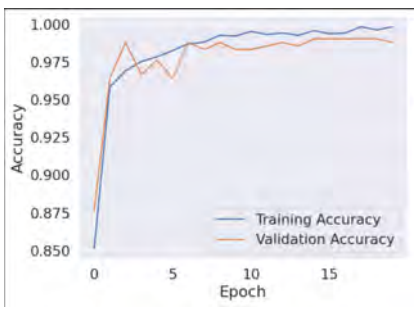

Figure 5.13: AugmentedInceptionV3

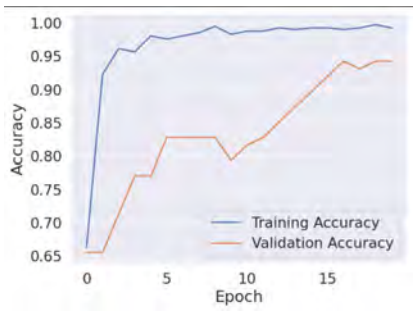

Figure 5.14: UnAugmentedInceptionV3

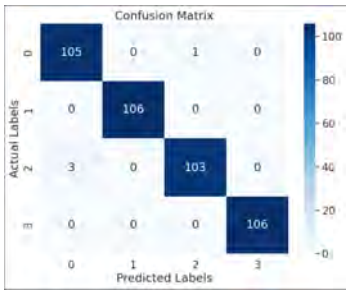

Figure 5.15: AugMented InceptionV3 ConMatrix

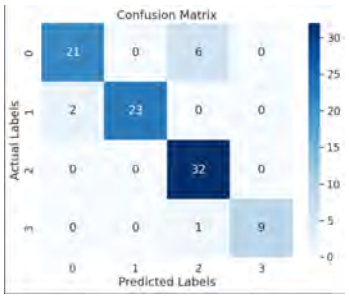

Figure 5.16: UnAugMented InceptionV3 ConMatrix

5.5 DenseNet-121 Implementation Result

Densenet-121 has a validation rate of 96.01% and an accuracy rate of 98.55% when it is not an enhanced version. The findings of the validation reveal that the augmentation led to an increase from 83.91% to 94.05%. The F1 Score, recall, and accuracy are all around 92.00%, which indicates that performance indicators have shown significant gains as a result of augmentation. With augmentation, the validation loss goes from 38.06% to 14.84%, a significant drop.

DenseNet-121	UnAugmented	Augmented
Accuracy	98.55%	96.01%
Validation Accuracy	83.91%	94.05%
Test Accuracy	78.00%	92.00%
F1-score	73.00%	92.00%
Recall	78.00%	92.00%
Precision	81.00%	94.00%
Validation Loss	38.06%	14.84%

Table 5.5: Results DenseNet-121

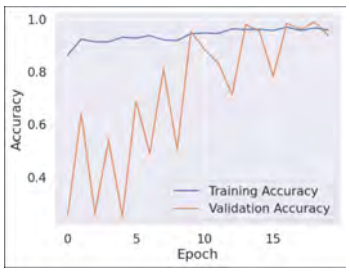

Figure 5.17: AugmentedDenseNet-121

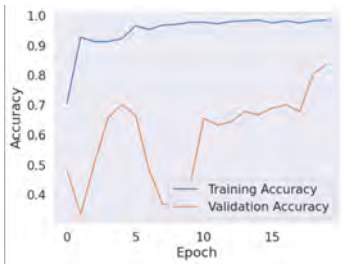

Figure 5.18: UnAugmentedDenseNet-121

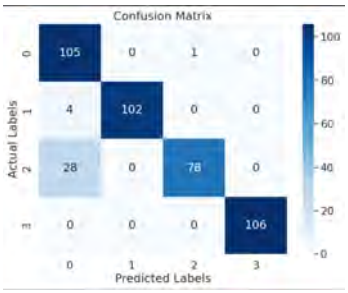

Figure 5.19: AugMented DenseNet-121 ConMatrix

Figure 5.20: UnAugMented DenseNet-121 ConMatrix

5.6 DenseNet-201 Implementation Result

The accuracy of Densenet-201 is 98.31% when it is not enhanced and 99.34% when it is equipped with augmentation. Only a slight improvement can be seen in the validation findings, which go from 96.65% to 98.81% with augmentation. The F1 Score, recall, and accuracy are all around 99.00% with augmentation, which indicates that performance measures continue to be at a consistently good level. With augmentation, validation loss drops from 9.96% to an astounding 5.02%, which is a significant improvement.

DenseNet-201	UnAugmented	Augmented
Accuracy	98.31%	99.34%
Validation Accuracy	96.65%	98.81%
Test Accuracy	91.50%	99.10%
F1-score	91.00%	99.00%
Recall	91.00%	99.00%
Precision	93.00%	99.00%
Validation Loss	9.96%	5.02%

Table 5.6: Results DenseNet-201

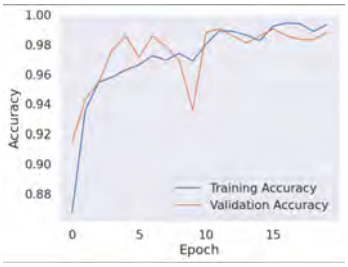

Figure 5.21: AugmentedDenseNet-201

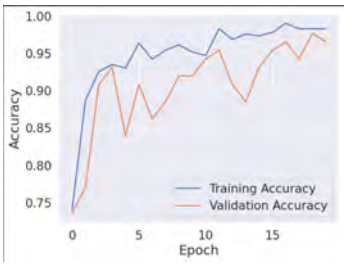

Figure 5.22: UnAugmentedDenseNet-201

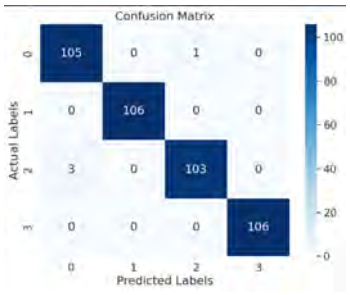

Figure 5.23: AugMented DenseNet-201 ConMatrix

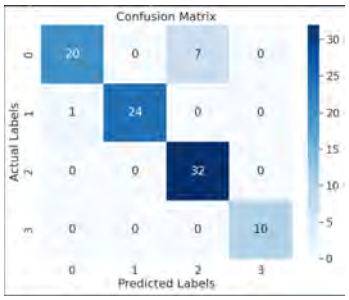

Figure 5.24: UnAugMented DenseNet-201 ConMatrix

5.7 Inception-Resnet-V2 Implementation Result

The accuracy of Inception-Resnet-V2 is 99.28% when it is not enhanced and 99.69% when it is equipped with augmentation. An increase from 94.25% to 98.81% is shown in the validation findings during the augmentation process. The F1 Score, recall, and accuracy are all around 99.00%, which indicates that the performance indicators showed minor gains with the addition of the augmentation. Through the use of augmentation, the validation loss is reduced from 15.31% to 5.85%.

Inception-Resnet-V2	UnAugmented	Augmented
Accuracy	99.28%	99.69%
Validation Accuracy	94.25%	98.81%
Test Accuracy	90.00%	99.10%
F1-score	90.00%	99.00%
Recall	90.00%	99.00%
Precision	92.00%	99.00%
Validation Loss	15.31%	5.85%

Table 5.7: Results Inception-Resnet-V2

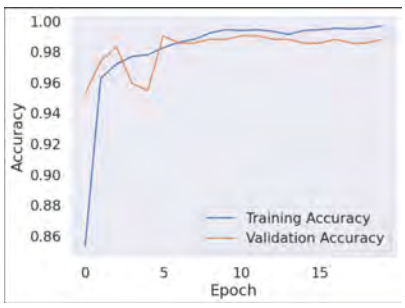

Figure 5.25: AugmentedInception-Resnet-V2

Figure 5.26: UnAugmentedInception-Resnet-V2

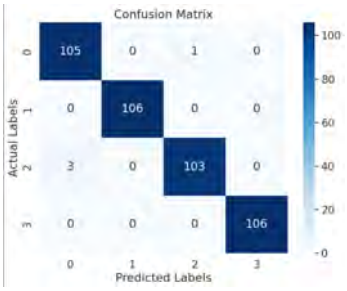

Figure 5.27: AugMented Inception-Resnet-V2 ConMatrix

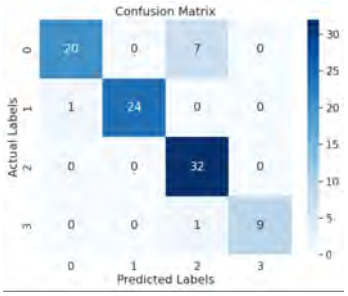

Figure 5.28: UnAugMented Inception-Resnet-V2 ConMatrix

5.8 Custom Model Implementation Result

The Custom CNN model achieves an un-enhanced accuracy of 78.80% and an augmented accuracy of 95.95%. Validation involves determining whether or not the model is accurate. The accuracy of validation is superior to both, with 86.00% of the data being unaugmented and an astonishing 97.84% of the data being enhanced. In terms of F1 Score, recall, and accuracy, there is a consistent relationship between the results obtained with and without augmentation. The F1 Score hovers around 83.00%, recall hovers around 97.00%, and precision hovers around 83.00%. There is a discernible reduction in validation loss as a result of augmentation, which goes from 42.33% to 0.081%.The augmented dataset eliminates the overfitting problem in the unaugmented dataset.

Custom Model	UnAugmented	Augmented
Accuracy	78.80%	95.95%
Validation Accuracy	86.00%	97.84%
Test Accuracy	83.00%	96.93%
F1-score	83.00%	97.00%
Recall	83.00%	97.00%
Precision	83.00%	97.00%
Validation Loss	42.33%	0.081%

Table 5.8: Results Custom Model

Figure 5.29: Augmented-Custom Model

Figure 5.30: UnAugmentedCustom Model

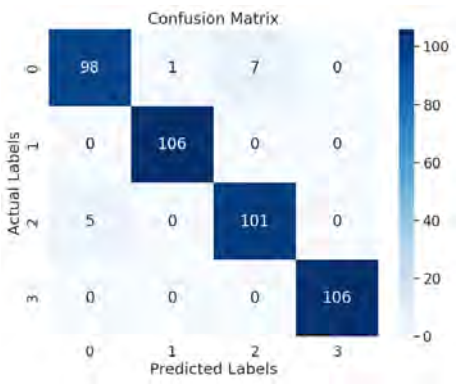

Figure 5.31: AugMented Custom Model ConMatrix

Figure 5.32: UnAugMented Custom Model ConMatrix

5.9 Observation of Models and Parameters

5.9.1 Observation

Below we can see the observations of Accuracy,F1 and ValidationLoss for Augmented and unaugmented Dataset :

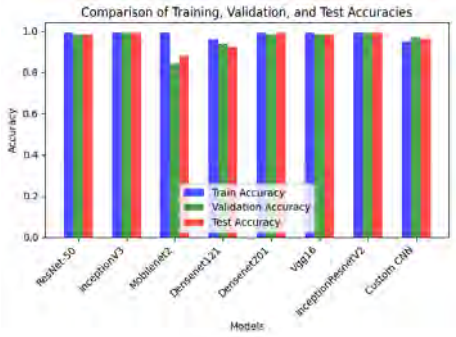

Figure 5.33: Augmented

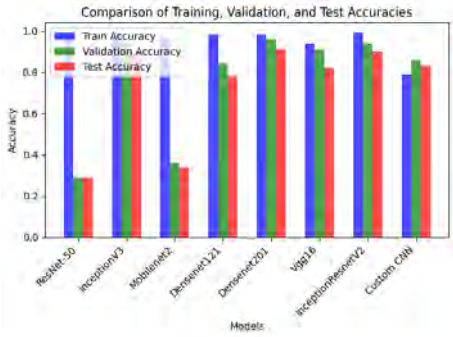

Figure 5.34: Unaugmented

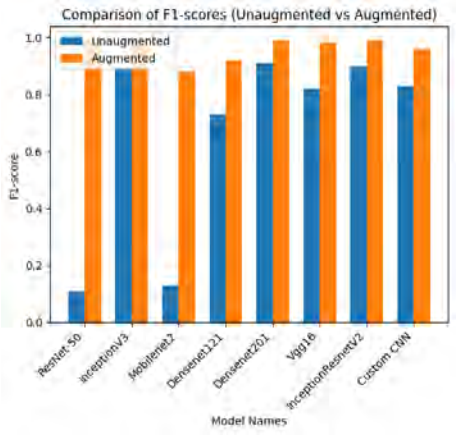

Figure 5.35: F1-score

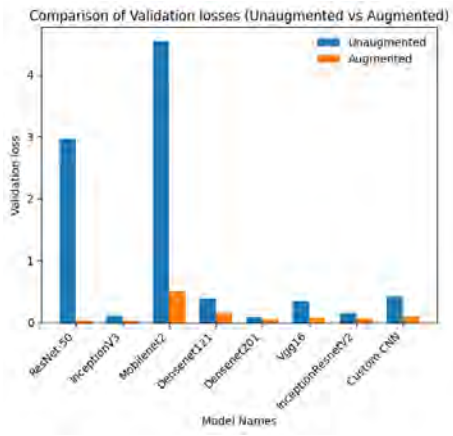

Figure 5.36: Validation Loss

Through the use of figures 5.33 and 5.34, we are able to see the accuracy of the training, testing, and validation accuracy for both the un-augmented and enhanced

versions of the models that were utilized.It is clear that the accuracy was not up to par before to the application of augmentation, but after the application of augmentation, there was a significant improvement in the accuracy.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$\text{F1 Score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}$$

Again,When it comes to classification jobs, the F1 score is a really essential indicator. The class imbalance and performance comparison are both easier to manage as a result.It is helpful for classification in both binary and multi class systems.Its sometimes more important than the accuracy for class identification.After augmentation we can see in figure 5.35, our F1 score has changed drastically.

Validation Loss is a very crucial metric for evaluating CNN models. It helps us to detect the over fitting or under fitting issues.To overcome over fitting we used augmentation and In figure 5.36 we see the differences between the augmented and un-augmented data set.

5.9.2 Model Parameters

Model Name	Total Parameters
Custom	1,997,124
VGG16	14,979,396
ResNet50	24,638,852
InceptionNetV3	22,853,924
InceptionresNetV2	55,125,732
MobileNetV2	2,915,908
DenseNet121	7,564,356
DenseNet201	19,307,588

Table 5.9: Parameters

In 5.9 figure, those models who have less accuracy than our custom CNN model, we have trained those models but we can observe much more parameters on those models. Also, on the other side, those who have just 1 percentage of better accuracy than our custom CNN model, we have also trained those models and get more parameters than our custom CNN model. So, the time complexity of our custom model is better than other models.

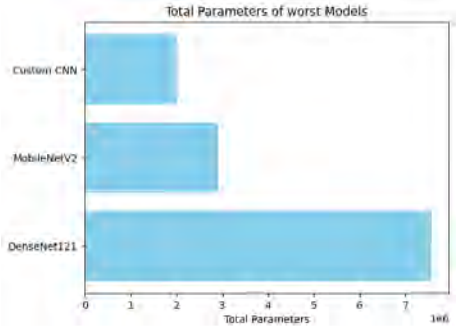

Figure 5.37: Comparison of Worst Models

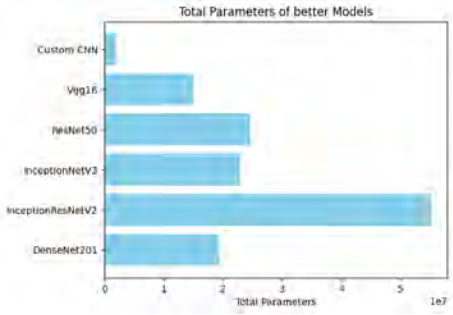

Figure 5.38: Comparison of Better Models

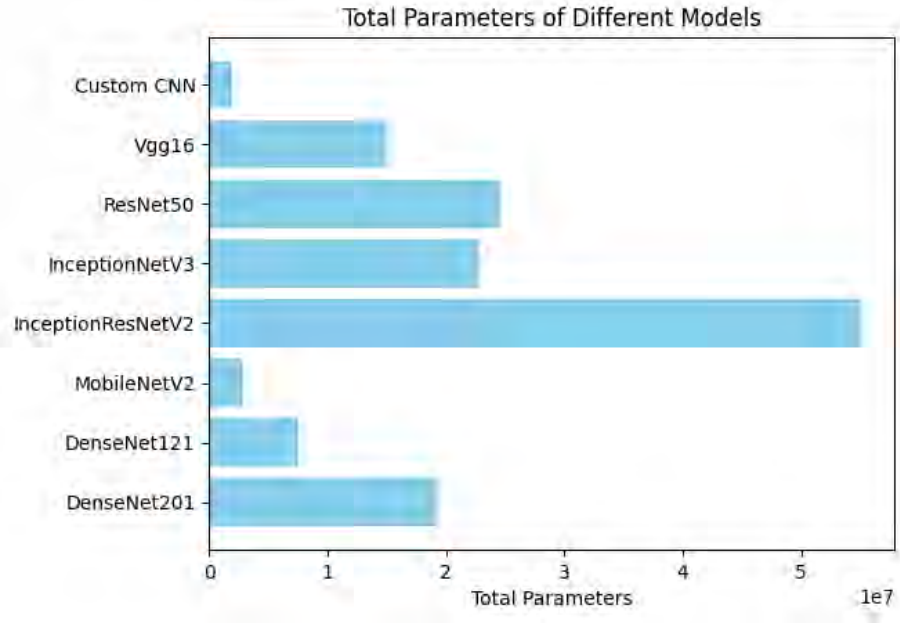

Figure 5.39: Comparison of Different Models

Chapter 6

Conclusion

We have built an unique CNN with an architecture that consists of five convolutional layers and two dense layers. This CNN will be able to identify three different types of illnesses with healthy leaves.We trained several different pre-trained CNN models, including ResNet-50, VGG16, MobileNet-V2,Inception-V3, InceptionResNet-V2, DenseNet-201 and DenseNet-121 through its paces by utilizing a dataset of photographs that included both healthy and damaged citrus leaves. The suggested model, which is based on a CNN, is able to differentiate between citrus leaves that are healthy and unhealthy. We have expanded our dataset and trained with our custom model,because we were facing overfitting issue before augmentation which was resolved afterwards.When we apply the supplemented dataset to the pre-trained models, with the exception of InceptionResnet-V2, which has high accuracy of 99.00% for training, testing, and validation, the other models show high accuracy for training but lower accuracy for testing and validation.Our Custom Model gave validation accuracy of 97.84% using 20 epochs.

6.1 Limitation

The level of accuracy that the model that we attempted to construct is capable of achieving is superior to the one that it presently achieves.We were unable to solve this issue because there was insufficient data supplied.It is possible that the outcome will be greater if we include additional data in the dataset.It is possible to find solutions to the challenges that we are now facing if we are able to make use of additional data.Then we will be able to train and test with greater precision.

6.2 Future Work

In the future, we will be working on this research with the goal of enhancing the performance of our model beyond what it already possesses.We aim to construct the model in such a manner that it is capable of identifying a large number of illnesses that are present on the leaves of citrus fruits.In addition, we want to work on enhancing the dataset by manually collecting photographs by visiting farms and nurseries, which provides us with the opportunity to get better results with our model.In addition, we intend to develop a website that will allow individuals to upload pictures of their plants in order to determine whether or not they are affected.

Bibliography

[1] *Leaf Diseases Caused by Fungi and Bacteria — uky.edu*, https://www.uky.edu/Ag/PAT/cat1/leafdis.htm, [Accessed 24-May-2023].

[2] L. S. P. Annabel, T. Annapoorani, and P. Deepalakshmi, "Machine learning for plant leaf disease detection and classification–a review," in *2019 international conference on communication and signal processing (ICCSP)*, IEEE, 2019, pp. 0538–0542.

[3] T. H. Spreen, "The world citrus industry," *Soils, Plant Growth and Crop Production*, vol. 3, pp. 249–269, 2010.

[4] *USDA APHIS — Citrus Diseases — aphis.usda.gov*, https://www.aphis.usda.gov/aphis/ourfocus/planthealth/plant-pest-and-disease-programs/pests-and-diseases/citrus/citrus-landing, [Accessed 24-May-2023].

[5] S. Saha, "A comprehensive guide to convolutional neural networks—the eli5 way," *Towards data science*, vol. 15, p. 15, 2018.

[6] *What are Convolutional Neural Networks? — IBM — ibm.com*, https://www.ibm.com/topics/convolutional-neural-networks, [Accessed 24-May-2023].

[7] *ResNet-50: The Basics and a Quick Tutorial — datagen.tech*, https://datagen.tech/guides/computer-vision/resnet-50/#, [Accessed 18-09-2023].

[8] A. Sarkar, *Understanding EfficientNet—The most powerful CNN architecture — medium.com*, https://medium.com/mlearning-ai/understanding-efficientnet-the-most-powerful-cnn-architecture-eaeb40386fad, [Accessed 18-09-2023].

[9] *Image Classification With MobileNet — builtin.com*, https://builtin.com/machine-learning/mobilenet, [Accessed 18-09-2023].

[10] *Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning — arxiv.org*, https://arxiv.org/abs/1602.07261, [Accessed 18-09-2023].

[11] 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," *Computers and Electronics in Agriculture*, vol. 177, p. 105 661, 2020.

[12] A. Elaraby, W. Hamdy, and S. Alanazi, "Classification of citrus diseases using optimization deep learning approach," *Computational Intelligence and Neuroscience*, vol. 2022, 2022.

[13] *Automatic Detection of Citrus Fruit and Leaves Diseases Using Deep Neural Network — ijraset.com*, https://www.ijraset.com/research-paper/automatic-detection-of-citrus-fruit-and-leaves-diseases, [Accessed 24-May-2023].

[14] *The Powerful Health Benefits of Citrus Fruits — thewholeu.uw.edu*, https://thewholeu.uw.edu/2022/05/23/citrus/#:~:text=Citrus%20fruits%20are%20rich%20in,diabetes%2C%20cancer%2C%20neurological%2, [Accessed 24-May-2023].

[15] S. Futch and D. Tucker, "Guide to citrus nutritional deficiency and toxicity identification," *Citrus and Vegetable Magazine*, pp. 26–29, 2003.

[16] M. Zekri and R. E. Rouse, *Citrus problems in the home landscape*. University of Florida Cooperative Extension Service, Institute of Food and ..., 2002.

[17] *Citrus Diseases — Citrus Tree Diseases Treatment — Citrus.com — yarden.com*, https://www.yarden.com/citrus-tree-care/pests-diseases/, [Accessed 24-May-2023].

[18] A. R. Luaibi, T. M. Salman, and A. H. Miry, "Detection of citrus leaf diseases using a deep learning technique," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 2, p. 1719, 2021.

[19] R. P. Shaikh and S. Dhole, "Citrus leaf unhealthy region detection by using image processing technique," in *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)*, IEEE, vol. 1, 2017, pp. 420–423.

[20] A. Khattak, M. U. Asghar, U. Batool, *et al.*, "Automatic detection of citrus fruit and leaves diseases using deep neural network model," *IEEE Access*, vol. 9, pp. 112 942–112 954, 2021.

[21] S. Janarthan, S. Thuseethan, S. Rajasegarar, Q. Lyu, Y. Zheng, and J. Yearwood, "Deep metric learning based citrus disease classification with sparse data," *IEEE Access*, vol. 8, pp. 162 588–162 600, 2020.

[22] *A Survey on Deep Learning and Its Impact on Agriculture: Challenges and Opportunities — mdpi.com*, https://www.mdpi.com/2077-0472/13/3/540, [Accessed 16-09-2023].

[23] https://www.researchgate.net/publication/374004915_CNN_and_transfer_learning_methods_with_augmentation_for_citrus_leaf_diseases_detection_using_PaaS_cloud_on_mobile, [Accessed 07-01-2024].

[24] https://www.researchgate.net/publication/347889486_Detection_of_citrus_leaf_diseases_using_a_deep_learning_technique, [Accessed 07-01-2024].

[25] *A Systematic Review of Citrus Disease Perceptions and Fruit Grading Using Machine Vision — sciencedirect.com*, https://www.sciencedirect.com/science/article/pii/S1877050923002259, [Accessed 07-01-2024].

[26] https://cdn.techscience.cn/files/cmc/2023/TSP_CMC-76-1/TSP_CMC_39781/TSP_CMC_39781.pdf, [Accessed 07-01-2024].

[27] *A citrus fruits and leaves dataset for detection and classification of citrus diseases through machine learning — sciencedirect.com*, https://www.sciencedirect.com/science/article/pii/S2352340919306948, [Accessed 07-01-2024].

[28] https://www.researchgate.net/publication/248617801_Mechanisms_of_infection_used_by_Xanthomonas_axonopodis_pv_citri_in_citrus_canker_disease, [Accessed 08-01-2024].

[29] 2. ·. U. Z. ·. A. S. Q. Asifullah Khan1 2 · Anabia Sohail1, https://www.researchgate.net/publication/330511306_A_Survey_of_the_Recent_Architectures_of_Deep_Convolutional_Neural_Networks, [Accessed 07-01-2024].

[30] Dharmaraj, *Convolutional Neural Networks (CNN)—Architectures Explained — draj0718*, https://medium.com/@draj0718/convolutional-neural-networks-cnn-architectures-explained-716fb197b243, [Accessed 07-01-2024].

[31] *Basic CNN Architecture: Explaining 5 Layers of Convolutional Neural Network — upGrad blog — upgrad.com*, https://www.upgrad.com/blog/basic-cnn-architecture/, [Accessed 07-01-2024].

[32] *Inception V3 Model Architecture — iq.opengenus.org*, https://iq.opengenus.org/inception-v3-model-architecture/#google_vignette, [Accessed 07-01-2024].

[33] https://www.researchgate.net/publication/324961229_Deep_CNNs_for_microscopic_image_classification_by_exploiting_transfer_learning_and_feature_concatenation, [Accessed 07-01-2024].

[34] *Image Captioning using Google's Inception-resnet-v2 and Recurrent Neural Network — ieeexplore.ieee.org*, https://ieeexplore.ieee.org/document/8844921, [Accessed 07-01-2024].

[35] *Architecture of DenseNet-121 — iq.opengenus.org*, https://iq.opengenus.org/architecture-of-densenet121/, [Accessed 07-01-2024].

[36] *Densely Connected Convolutional Networks — arxiv.org*, https://arxiv.org/abs/1608.06993, [Accessed 07-01-2024].

[37] *Understanding VGG16: Concepts, Architecture, and Performance — datagen.tech*, https://datagen.tech/guides/computer-vision/vgg16/, [Accessed 07-01-2024].

[38] G. L. Team, *Introduction to VGG16 — What is VGG16? — mygreatlearning.com*, https://www.mygreatlearning.com/blog/introduction-to-vgg16/, [Accessed 07-01-2024].

[39] *Explainable AI, LIME & SHAP for Model Interpretability — Unlocking AI's Decision-Making — datacamp.com*, https://www.datacamp.com/tutorial/explainable-ai-understanding-and-trusting-machine-learning-models, [Accessed 07-01-2024].