

## Declaration of Authorship

This is to certify that the work presented in this thesis paper is the outcome of research carried out by the candidates under the supervision of Dr. Md. Ashraf ul Hoque, Professor, Department of Electrical and Electronic Engineering (EEE), Islamic University of Technology (IUT). It is also declared that neither this thesis paper nor any part thereof has been submitted anywhere else for the reward of any degree or any judgment.

### Authors

*Ashekur*

---

**Md. Ashekur Rahman**

**ID-170021099**

*Samadul  
Amin*

---

**Samadul Amin**

**ID-170021141**

*Md. Mahbub - Ul - Huq Alvi*

---

**Md. Mahbub-Ul-Huq Alvi**

**ID-170021144**

# Certificate of Approval

The thesis titled “Transfer Learning-Based Approach for Citrus Disease Detection” submitted by Md. Ashekur Rahman (170021099), Samadul Amin (170021141), Md. Mahbub-Ul- Huq Alvi (170021144) has been found as satisfactory and accepted as partial fulfillment of the requirement for the degree of Bachelor of Science in Electrical and Electronic Engineering on 10th May, 2022.

Approved by:



(Signature of the Supervisor)

**Dr. Md. Ashraful Hoque**

Supervisor and Professor

Dean Faculty of Engineering

Department of Electrical and Electronic Engineering (EEE)

Islamic University of Technology (IUT)



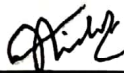
(Signature of the Co-Supervisor)

**Fahim Faisal**

Assistant Professor

Department of Electrical and Electronic Engineering

Islamic University of Technology



(Signature of the Co-Supervisor)

**Mirza Muntasir Nishat**

Assistant Professor

Department of Electrical and Electronic Engineering

Islamic University of Technology

# Transfer Learning-Based Approach for Citrus Disease Detection

by

Md. Ashekur Rahman, 170021099

Samadul Amin, 170021141

Md. Mahbub-Ul- Huq Alvi, 170021144

A Dissertation

Submitted to the Department of Electrical & Electronic Engineering

In Partial Fulfillment of the Requirements

for the Degree of

Bachelor of Science in Electrical & Electronic Engineering



Islamic University of Technology (IUT)

Gazipur, Bangladesh

May 2022

# Certificate of Approval

The thesis titled “Transfer Learning-Based Approach for Citrus Disease Detection” submitted by Md. Ashekur Rahman (170021099), Samadul Amin (170021141), Md. Mahbub-Ul- Huq Alvi (170021144) has been found as satisfactory and accepted as partial fulfillment of the requirement for the degree of Bachelor of Science in Electrical and Electronic Engineering on 10th May, 2022.

**Approved by:**

---

(Signature of the Supervisor)

**Dr. Md. Ashraful Hoque**

Supervisor and Professor

Dean Faculty of Engineering

Department of Electrical and Electronic Engineering (EEE)

Islamic University of Technology (IUT)

---

(Signature of the Co-Supervisor)

**Fahim Faisal**

Assistant Professor

Department of Electrical and Electronic Engineering

Islamic University of Technology

---

(Signature of the Co-Supervisor)

**Mirza Muntasir Nishat**

Assistant Professor

Department of Electrical and Electronic Engineering

Islamic University of Technology

## ***Dedicated to***

*Our beloved parents & teachers whose support made it all possible for us*

## **ACKNOWLEDGEMENTS**

Foremost, we would like to express our sincere gratitude and gratefulness to the Almighty Allah; without His graces and blessings, this study would not have been possible.

Acknowledging all who helped us complete this work, we wish to compliment the university's significant role and the department that has been very amiable to us during the entire period of our research.

We are indebted to our honorable supervisor, Dr. Md. Ashraful Hoque sir and co supervisors Fahim Faisal sir and Mirza Muntasir Nishat sir for their selfless support, motivation, patience, enthusiasm, and extensive knowledge of the relevant fields. Their continuous guidance and careful supervision kept us going even during the hardest of hours.

Lastly, our warmest tribute to our parents, family members, and friends, whose moral support and well wishes benefitted us spiritually in achieving our goals.

## **Abstract**

Fruit diseases have a devastating impact on the food economy, wreaking havoc on the global food business. Therefore, in agriculture and food security, disease detection and diagnosis systems must be efficient and dependable. Technology supported by computers can be used to detect illnesses. Convolutional neural network (CNN) approaches have recently demonstrated impressive performance in image categorization applications. CNN can operate relatively efficiently with high-volume datasets and extracts more circumstantial features. This paper offers a strategy that employs deep convolutional neural networks with transfer learning to identify and categorize citrus crop diseases. Useful features were identified from fine-tuned, pre-trained deep learning models that were retrained using an image dataset of citrus fruits with infections. On a large dataset (ImageNet), the InceptionResNetV2, DenseNet169, MobileNet, NasNetLarge, NASNetMobile, and VGG19 models have been pre-trained to increase prediction accuracy. The VGG19 model has a classification accuracy of 97.54 percent, which is superior to current methods.

# TABLE OF CONTENTS

<b>Acknowledgment</b> .....	iv
<b>Abstract</b> .....	v
<b>Table of Contents</b> .....	vi
<b>List of Tables</b> .....	viii
<b>List of Figures</b> .....	ix
<b>List of Acronyms</b> .....	x
<b>1. Introduction</b> .....	<b>01</b>
1.1 Introduction.....	01
1.2 Significance of Research.....	02
1.3 Objectives of this Research.....	03
1.4 Main Contribution.....	04
1.5 Thesis Outline.....	04
<b>2. Literature Review</b> .....	<b>06</b>
2.1 Relevant Research.....	06
2.1.1 Preprocessing techniques.....	06
2.1.2 Classification and Segmentation-based techniques.....	06
2.2 Comparative Analysis of Relevant Research.....	07
<b>3. Methodology</b> .....	<b>09</b>
3.1 Transfer Learning Models.....	09
3.1.1 MobileNet.....	09
3.1.2 DenseNet.....	09
3.1.3 Inception-ResnetV2.....	10
3.1.4 NasNet(NASNetLarge & NASNetMobile).....	10
3.1.5 VGG-19.....	10
3.2 Methodology.....	10
3.2.1 Image accretion.....	11
3.2.2 Image Pre-processing.....	12
3.2.3 Dataset Splitting.....	13
3.2.4 Implementation of Proposed model based on transfer learning.....	13



<b>4. Transfer Learning review.....</b>	<b>14</b>
4.1 What is Transfer Learning.....	14
4.2 Approaches to Transfer Learning .....	14
4.2.1 Training a Model to Reuse it.....	14
4.2.2 Pre-trained Model.....	14
4.2.3 Feature Extraction.....	16
<b>5. Introduction to Algorithms.....</b>	<b>17</b>
5.1 MobileNet.....	17
5.2 DenseNet.....	19
5.3 Inception-ResnetV2.....	21
5.4 NasNet.....	27
5.5 Vgg-19.....	29
<b>6. Evaluation Criteria.....</b>	<b>30</b>
6.1 True vs False and Positive vs Negative.....	30
6.2 Accuracy.....	30
6.3 Precision.....	30
6.4 Recall.....	30
6.5 F-1 score.....	31
6.6 AUC.....	31
6.7 Confusion Matrix.....	31
<b>7. RESULT &amp; ANALYSIS.....</b>	<b>32</b>
7.1 Transfer Learning Models.....	32
7.1.1 DenseNet169.....	32
7.1.2 Inception-ResnetV2 .....	34
7.1.3 MobileNet.....	35
7.1.4 NASNetLarge.....	37
7.1.5 NASNetMobile.....	38
7.1.6 Vgg19.....	40
7.2 Observation.....	41
<b>8. CONCLUSION &amp; FUTURE SCOPES.....</b>	<b>43</b>
8.1 Conclusion.....	43
8.2 Limitations.....	44
8.3 Future Scopes.....	44
<b>References.....</b>	<b>45</b>

## **LIST OF TABLES**

<b>Table 1.1</b>	Total production of citrus in Asia & Bangladesh.....	01
<b>Table 1.2</b>	Total import of citrus in Asia & Bangladesh .....	02
<b>Table 2.1</b>	Papers Reviewed for Pre-processing techniques .....	08
<b>Table 2.2</b>	Papers Reviewed for classification and segmentation-based techniques ...	08
<b>Table 3.1</b>	Specifics of the datasets.....	12
<b>Table 5.1</b>	MoblieNet Architecture .....	18
<b>Table 5.2</b>	DenseNet Architectures for ImageNet.....	21
<b>Table 6.1</b>	Confusion Matrix.....	31
<b>Table 7.1</b>	Classification report (DenseNet169).....	32
<b>Table 7.2</b>	Confusion Matrix (DenseNet169).....	34
<b>Table 7.3</b>	Classification report (Inception-ResnetV2).....	35
<b>Table 7.4</b>	Confusion Matrix (Inception-ResnetV2).....	35
<b>Table 7.5</b>	Classification report (MobileNet).....	35
<b>Table 7.6</b>	Confusion Matrix (MobileNet).....	37
<b>Table 7.7</b>	Classification report (NasNetLarge).....	37
<b>Table 7.8</b>	Confusion Matrix (NasNetLarge).....	38
<b>Table 7.9</b>	Classification report (NASNetMobile).....	40
<b>Table 7.10</b>	Confusion Matrix (NASNetMobile).....	41
<b>Table 7.11</b>	Classification report (Vgg19).....	41
<b>Table 7.12</b>	Confusion Matrix (Vgg19).....	42
<b>Table 7.13</b>	Comparison Table.....	43

## LIST OF FIGURES

<b>Figure 3.1:</b>	Basic Methodology Flowchart .....	11
<b>Figure 3.2:</b>	Gamma Correction.....	12
<b>Figure 4.1:</b>	Basic Framework.....	16
<b>Figure 5.1:</b>	Schematic for stem of the pure Inception-v4 and Inception-ResNet-v2 networks	23
<b>Figure 5.2:</b>	The overall schematic view of the Inception-v4 network .....	24
<b>Figure 5.3:</b>	“Reduction-B” 17×17 to 8×8 grid-reduction module .....	25
<b>Figure 5.4:</b>	Schematic view for 8×8 grid (Inception-ResNet-C) module.....	25
<b>Figure 5.5:</b>	Schematic view for 35 × 35 grid (Inception-ResNet-A) module.....	26
<b>Figure 5.6:</b>	The schematic view for 35 × 35 to 17 × 17 reduction module .....	26
<b>Figure 5.7:</b>	The schematic view for 17 × 17 grid (Inception-ResNet-B) module .....	27
<b>Figure 5.8:</b>	Overview of Neural Architecture Search .....	28
<b>Figure 5.9:</b>	Scalable architectures for image classification .....	28
<b>Figure 5.10:</b>	VGG-19 Network Architecture .....	29
<b>Figure 7.1:</b>	Training and Validation Accuracy(DenseNet169).....	33
<b>Figure 7.2:</b>	Training and Validation Loss (DenseNet169).....	33
<b>Figure 7.3:</b>	Training and Validation Accuracy(InceptionResNetV2).....	34
<b>Figure 7.4:</b>	Training and Validation Loss (InceptionResNetV2).....	35
<b>Figure 7.5:</b>	Training and Validation Accuracy(MobileNet).....	37
<b>Figure 7.6:</b>	Training and Validation Loss (MobileNet).....	37
<b>Figure 7.7:</b>	Training and Validation Accuracy(NasNetLarge).....	38
<b>Figure 7.8:</b>	Training and Validation Loss (NasNetLarge).....	39
<b>Figure 7.9:</b>	Training and Validation Accuracy(NasNetMobile).....	40
<b>Figure 7.10:</b>	Training and Validation Loss (NasNetMobile).....	40
<b>Figure 7.11:</b>	Training and Validation Accuracy(Vgg19).....	42
<b>Figure 7.12:</b>	Training and Validation Loss (Vgg19).....	42

## LIST OF ACRONYMS

<b>AI</b>	Artificial Intelligence
<b>ML</b>	Machine Learning
<b>DL</b>	Deep Learning
<b>CNN</b>	Convolutional Neural Network
<b>ANN</b>	Artificial Neural Network
<b>VGG</b>	Visual Geometry Group
<b>FAIR</b>	Facebook AI Research
<b>BN</b>	Batch Normalization
<b>ResNet</b>	Residual neural network
<b>DenseNet</b>	Densely Connected Convolutional Networks
<b>TP</b>	True Positive
<b>TN</b>	True Negative
<b>FP</b>	False positive
<b>FN</b>	False Negative
<b>CM</b>	Confusion Matrix
<b>SVM</b>	Support Vector Machine

# Chapter 1

## INTRODUCTION

### **1.1 Introduction**

One of the oldest ways for providing food still in use today is agriculture. Plant diseases, on the other hand, are the most serious danger to agricultural productivity, as they cause significant economic losses [1] [2]. Early diagnosis of plant diseases can help to reduce the severity of the consequences. Generally speaking, illnesses of citrus fruits cause a 30-50 percent decrease in output and a loss of approximately 80 million taka annually. [25] Citrus fruit production in Arizona is predicted to decline by 5 86,000 tons in 2019/20 [1], primarily as a result of plant diseases].

In this study, We suggested an efficient automated method for detecting two of the most common citrus illnesses, which were previously discussed. Because lemon is the most prevalent citrus fruit in Asia, we'll use it as an example in this section. Citrus has recently been discovered to have sedative and antispasmodic properties [2] and is being studied further. Citrus fruits are grown in more than 140 nations around the world, including the United States. [23] In 2019, a total of 165.3 thousand tonnes of citrus fruit were produced in Bangladesh, while a total of 104.8 thousand tonnes of citrus fruit were imported. [23] Citrus fruits are packed in fiber, vitamins, limonoids, carotenoids, and flavonoids, among other nutrients.

**Table 1.1 - Total production of citrus in Asia & Bangladesh (in thousand tonnes)**

	2011	2012	2013	2014	2015	2016	2017	2018	2019
<b>Asia</b>	53730.6	56445.4	62083.3	64182.3	66507.5	61751.3	63911.7	65335.0	71887.8
<b>Bangladesh</b>	136.7	147.3	150.4	154.6	156.9	155.9	158.7	162.0	165.3

**Table 1.2 - Total import of citrus in Asia & Bangladesh (in thousand tonnes)**

	2011	2012	2013	2014	2015	2016	2017	2018	2019
<b>Asia</b>	3642.63	564.9 3	646.8 4	090.44	385.94	529.94	270.34	260.64	236.3
<b>Bangladesh</b>	120.0	101.3	102.6	78.8	165.1	73.7	121.6	104.8	104.8

Citrus canker, on the other hand, can be stopped before it spreads if it is discovered early enough by the symptoms in the photographs [5]. Similarly, in the instance of black spot illness, hyperspectral image analysis can be used to detect illness at an early stage in the progression. Consequently, to reduce production losses, the use of computerized early detection image processing methods in citrus fruit production can be critical in reducing production losses in citrus fruit production. Scientists from all across the world have collaborated to create a variety of image processing methods that make use of a variety of machine learning algorithms and models in order to detect diseases early in their progression. Our research provided a model that can categorize and diagnose these two diseases more efficiently and effectively than the other models reviewed in our literature study, which we believe is the most accurate model available.

Since the beginning of the previous decade, For the identification and categorization of plant diseases, computer vision and image processing methods have become increasingly prominent [09]. Our method, on the other hand, comprises the following steps: image accretion, processing, and implementation of the proposed model Six criteria were used to compare and contrast the model with other models that had been offered before it (True vs False and Positive vs Negative, Accuracy, Precision, Recall, F-1 score, and AUC). Furthermore, we highlighted the limits of our suggested model as well as its potential future applications in the conclusion of the paper.

## ***1.2 Significance of Research***

It is the goal of our research to provide a comprehensive technique for evaluating a variety of Transfer Learning Models that employ deep convolutional neural networks to identify and

categorize diseases of citrus fruits (Canker & Black Spot). [26] reported that 14 of the 25 citrus diseases known worldwide were found in Bangladesh, with a moderate to high prevalence and severity [26]. Canker and black spot, on the other hand, are two of the most prevalent and destructive diseases that impact citrus trees, according to our findings.

### **A. Black spot**

Citrus black spot (CBS) is a fungal disease that affects citrus trees caused by the fungus *Phyllosticta citricarpa* that affects citrus fruit. When the plant is most sensitive and the climate is most favourable to disease growth, the citrus black spot, or CBS, is accessible. Citrus leaf and fruit symptoms include small, round spots under hazardous spots with gray squares, as well as small, round spots on the leaves and fruits. The hue of the black spot lesion is a dark brown, and the spot diameter ranges from 0.12 to 0.4 millimeters. [27]

### **B. Canker**

Citrus canker is caused by the bacteria *Xanthomonas axonopodis*, which is found in citrus fruits. Citrus canker spot is covered in a layer of moisture brought in by the wind. The size of the lesions on the citrus leaf ranges from 2 to 10mm in diameter, and they are arranged in rings around the leaf. A fruit lesion, on the other hand, might be anywhere from 1mm to 10mm in size, with each lesion being somewhat different in size from the others. The majority of the time, the lesions are surrounded by a yellow halo that has been soaked in water. Despite its dark brown and black appearance, this lesion is not painful. (27)

Cultivars of citrus fruits suffer from illnesses that have a negative impact on their production and quality. Direct canker management costs are anticipated to be US\$214.00 per hectare in Florida or 3.84 percent of total production expenses [3]. However, in South Africa, Kotzé (1963) discovered that CBS symptoms were found to be uniformly spread throughout the plant, with 52.8 percent occurring in the top canopy and 47.2 percent occurring in the lower canopy. For decades, these two diseases have had a severe impact on the production of citrus fruits. In Conclusion we can see that, among the reviewed papers the highest accuracy obtained was 97%. Whereas, VGG19 provided us with an accuracy of 97.54%. [09-10]

## ***1.3 Objectives of this Research***

Successful research collaboration begins with clearly defined study objectives. The underlying goal of investigation and analysis should be addressed in the research objectives. It should lay out the measures someone would take to reach their goals. Research objectives keep one focused and allow to change your expectations as they go. Here, are the objectives of our research:

- 1.Revolutionize the agricultural & food industry with the help of extreme computational power.
2. Early detection of fruit’s contagious disease for saving the entire farm from its havoc.
- 3.Saving human effort and time by engaging intelligence machine.
- 4.Benefit beverage industry and take part in Fourth Industrial Revolution.

## ***1.4 Main Contribution***

Our study is primarily focused on facilitating the farming business in order to make the lives of farmers and industry workers easier. We are developing an intelligent support system to detect the most prevalent diseases of citrus fruits, which will be used to detect these diseases. A significant role for learning algorithms can be played in the classification and detection of any disease translocation. In pursuit of this goal, we utilized six transfer learning algorithms in the hunt for a rapid, accurate, and long-lasting technique of detecting and categorizing diseases from an image dataset. However, in order to produce better outcomes with great accuracy, we have used hyper parameter tuning. In addition, the computation time was short.

This research also assisted us in learning about computational intelligence and algorithms, which we were able to examine and learn from. It spurred us on to pursue improvements in machine learning and transfer learning in order to achieve higher performance.

## ***1.5 Thesis Outline***

In chapter – 1, the significance of research on Citrus disease classification from our data is explained. The necessity of ML & DL algorithms for increasing the accuracy of diagnostics & the main contribution of our research work is also described.

In chapter – 2, relevant research works published recently on Citrus data classification using ML & DL algorithms are analyzed comparatively that in-line with our research topic of interest.

In chapter – 3, the transfer learning models & methodology of our entire work which we followed from the beginning to the end.

In chapter – 4, we analyzed and reviewed the transfer learning models and approached the models which had the possibility to give the best outcome

In chapter – 5, the 5 transfer learning algorithms & their working procedures are briefly described that we've implemented.



Chapter – 6 describes the parameters that we have chosen to show the comparison between our results.

In chapter – 7, we compared the outcomes of the various transfer learning models and presented the training and validation accuracy and loss, as well as each model's confusion matrices.

Finally, the discussion of our research work is concluded in chapter – 8 by mining out the Future-scopes to improve & better implement our study of interest.

## **Chapter 2**

### **LITERATURE REVIEW**

#### ***2.1 Relevant Research***

A collection of related works on this specific issue was compiled in order to gain a sense of the progress that has been done. According to our technique, the section on literature review was divided into two parts. Techniques for preprocessing and categorization are included in this category of techniques. A large number of researchers have already experimented with a variety of strategies. Here are some of the more recent research that have produced more conclusive results.

##### ***2.1.1 Preprocessing techniques***

In order to classify Black Spot, Canker, Melanose, Scab, and Greening disease in the year 2019, Hafiz et al. employed techniques such as the Top Hat Filter and the Principal Component Analysis. It was discovered that this research has improved upon the classification approaches used in previous publications at the time.

To increase the precision of the Canker classification in 2018, Shoby et colleagues used Gaussian and CLAHE approaches to improve the precision of Canker classification. These high-precision approaches also assisted in precisely identifying Canker disease images among all of the data.

To diagnose Canker disease in 2015, Thangadurai et al. used in combination with the Gaussian technique, low pass and high pass filters are used, which they developed themselves. At the time, this method of detecting Canker disease from photographs was successful in detecting the disease.

##### ***2.2.2 Classification and Segmentation-based techniques***

A method comprising lesion segmentation and optimal weighted segmentation techniques that can classify Anthracnose, Blackspot, Canker, Melanose, and Scab illnesses was presented by Muhammad et al. in 2018. It is based on lesion segmentation and optimized weighted segmentation techniques. The approach was trained and tested using a dataset comprising of photographs of

citrus fruits, which was created specifically for this purpose. The proposed method consisted of a confluence of four basic phases. Some of the objectives include improving picture contrast using the Gaussian technique, identifying infected regions using lesion segmentation and feature extraction, developing a codebook and classification, and developing a codebook and classification. The accuracy of this approach was 97 percent when applied to the image gallery dataset. However, when merged with local statistics, the percentages dropped to 89 percent and 90.4 percent, respectively.

In 2019, Benjamin et al. developed a method that used ANN (Artificial Neural Network) and SVM (Support Vector Machine) with Phenotypic Features to detect and classify Anthracnose, Blackspot, Canker, Melanose, and Scab in a variety of plant diseases. The purpose of this study was to conduct a comparative analysis between ANN and SVM. The SVM approach was found to be superior, with an accuracy of 93.12 percent, while the ANN method fared comparatively badly, with an accuracy of 88.53 percent.

This method, developed by Wenyan and colleagues, is much more sustainable and relevant for detecting and classifying HLB, Anthracnose, Canker, Blackspot, Sandpaper rust, and Scab in 2019 than previous methods. This work has not only demonstrated a method for detecting diseases, but it has also investigated its application in field situations. In order to combat these diseases, this strategy made use of technologies and algorithms such as the Nginx server, Dense Net, WeChat applet, Keras framework, and Stochastic Gradient Descent. However, this approach was only able to complete 88.53 percent of the texture evaluation tasks.

## ***2.2 Comparative Analysis of Relevant Research***

In our research, we sought to develop a more precise classification and identification system for citrus canker and black spot disease in plants. We looked at some papers that were connected to similar types of works and discovered that there are just a few works on the subject and that our paper is more accurate than the majority of them.

The reviewed papers are briefly summarized in the following tables:

- i. For pre-processing techniques these papers were reviewed:

**Table 2.1:** Papers Reviewed for Pre-processing techniques

References	Year	Diseases	Techniques	Results
Hafiz et.al [6]	2019	Blackspot, Canker, Melanose, Scab, Greening	Top hat filter, PCA	Classification
Shoby et.al [7]	2018	Canker	Gaussian, CLAHE	Precision
Thangadurai et.al [8]	2015	Canker	Gaussian, median, Low pass, High pass filters.	Detection

ii. For classification and segmentation-based techniques, these papers were reviewed:

**Table 2.2:** Papers Reviewed for classification and segmentation-based techniques.

References	Year	Diseases	Techniques	Results
Muhammad et.al.[9]	2018	Anthracnose, Blackspot, Canker, Melanose, Scab	Lesion segmentation, Optimized weighted segmentation	Accuracy Citrus disease image gallery dataset-97% combined dataset-89% local dataset 90.4%
Benjamin et.al.[10]	2019	Anthracnose, Blackspot, Canker, Melanose, Scab	Phenotypic Features, Support Vector Machine, Artificial Neural Network	Accuracy SVM-93.12% ANN- 88.96%
Wenyan et.al. [11]	2019	HLB, Anthracnose, Canker, Blackspot, Sandpaper rust, Scab	Nginx server, Dense Net, WeChat applet, Keras framework, Stochastic Gradient descent	Evaluation texture-88.53%

## **Chapter 3**

### **METHODOLOGY**

#### ***3.1 Transfer Learning Models:***

While there are several deep learning models to choose from, the convolutional neural network (CNN) is the most well-established algorithm among them [12-13]. CNNs are a class of artificial neural networks that have become widely used in computer vision tasks ever since the astonishing results of the ImageNet Large Scale Visual Recognition Competition (ILSVRC) in 2012 were made public. As a result, machine learning researchers frequently rely on field specialists to label such photographs, which is both expensive and time-consuming. As a result, it has been demonstrated that creating the vast amount of labels required to construct flourishing deep networks is not practical [14]. Transfer learning enables the transferability of characteristics between distinct train and target domains, which is advantageous. Many complex deep neural networks for image classification tasks have been designed by the deep learning community and are freely available in the python Keras toolkit. Some often used CNN architectures include: GoogLeNet (for example, Inception-v3, Xception), GoogLeNet (for example, Inception-v3, Xception), residual network (for example, ResNet50), and others. A brief discussion of the models employed in our investigation is provided in the next sections.

##### ***3.1.1 MobileNet:***

The fundamental concept behind the MobileNetV1 network is to replace the normal convolution operation with depth-wise separable convolution (DSC) in order to reduce the number of parameters in the model. The DSC algorithm will employ the 3\*3 convolution kernel with only one layer thickness, sliding the input tensor layer by layer, and construct an output channel after each convolution. A team of Google engineers presented their paper titled MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications [15] at CVPR 2017 and introduced the concept of convolutional neural networks.

##### ***3.1.2 DenseNet***

An output from the previous layer can be utilized as an input for the second layer using the composite function method. Convolution, pooling, batch normalization, and non-linear activation are among the layers that make up this composite procedure. As a result of these connections, the network has a  $L(L+1)/2$  number of direct connections. The number of levels in the architecture is denoted by the letter L. A collaborative effort between Cornwell University, Tsinghua University, and Facebook AI Research (FAIR) [16] resulted in its development.

### ***3.1.3 The inception-ResnetV2***

The network accepts an image with a resolution of 299 by 299 pixels as input, and it returns a list of estimated class probabilities as output. It is based on the Inception structure and the Residual connection to be formally formulated. With residual connections paired with multiple-sized convolutional filters, this block is known as the Inception-Resnet block. In addition to avoiding the deterioration problem caused by deep structures, the inclusion of residual connections minimizes the amount of time required for training [17].

### ***3.1.4 NasNet(NASNetLarge & NASNetMobile)***

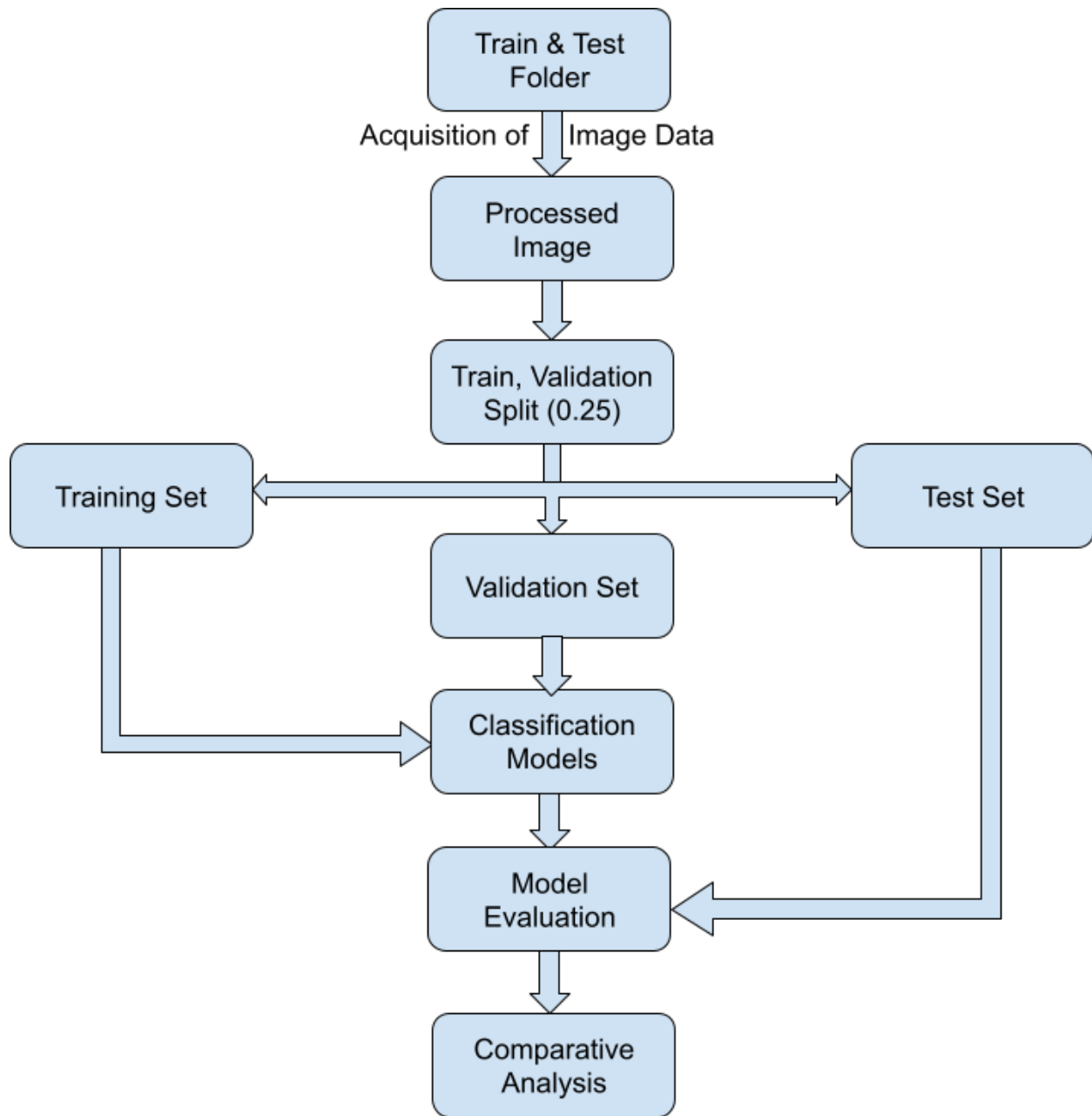
There are pre-trained on ImageNet and weighted models of Google Brain's Neural Architecture Search Network (NASNet) available. Input sizes of 331x331 pixels are used for the NASNetLarge model, while 224x224 pixels are used for the NASNetMobile model [18].

### ***3.1.5 VGG-19***

The vgg19 algorithm was employed in this experiment. It consists of 19 components (16 convolution layers, three fully connected layers, 5 MaxPool layers, and 1 SoftMax layer). [19] It was developed by the Visual Geometry Group at Oxford University in 2014, and it has demonstrated accurate classification performance on the ImageNet dataset.

## ***3.2 Methodology***

It will be discussed in this section how we came up with our work-frame and the convolutional neural network that we used for our research. On Kaggle, we discovered the dataset we used for our work, which was well-balanced and meticulously labeled. In order to cope with the brightness issue, we had to apply gamma correction to both the training and test sets of photos. Figure X depicts a condensed version of the entire procedure. Subsequently, in the following section, each procedure is explored in more detail.



**Figure 3.1:** Basic Methodology Flowchart

### ***3.2.1 Image accretion:***

Everything that was acquired from a single platform was divided into two folders, which were labeled train and test, to better organize the information. Following these folders were two subfolders labeled according to the image classifications (black-spot & citrus canker) that we categorized using the computer-aided classification approach described below. The following are the specifics of the datasets:

**Table 3.1:** Specifics of the datasets

<b>Class Name</b>	<b>Total Train Images</b>	<b>Total Test</b>
Black-Spot	1030	206
Citrus Canker	1002	201

Size: 800\*800\*3  
Format: JPG

### **3.2.2 Image Pre-processing:**

The color photographs we got had a low-light issue, which might have an influence on computer vision systems' overall performance. Gamma Correction is used as a result of this problem. The next paragraphs go about GC (gamma correction) in depth. The GC method maps the picture with low contrast in the input.

$$y = x^\gamma, \gamma < 1$$

The normalized input and output images, respectively, are represented by  $x$  and  $y$ , respectively, and 1 is a constant, which is denoted by 0.5. Figure 1 depicts an original low-light image that has been enhanced using GC. It can clearly be observed that the augmented image has a more pleasing aesthetic impression on the overall than the original image. However, the bright sections of the image are also magnified, making them appear so brilliant that the enhancement diminishes local contrast inside high-illuminance zones as a result of the enhancement. After that, we scaled the photographs to match the input size required by CNN models, and we were finished.



**Figure 3.2:** Gamma Correction



### ***3.2.3 Dataset Splitting:***

In order to achieve the lowest bias and variance possible for the machine learning and deep learning algorithms we deployed, training datasets were randomly mixed and then divided into 75:25 ratios for training and validation, respectively.

### ***3.2.4 Implementation of Proposed model based on transfer learning:***

According to the requirements of the feature extraction models, the input photos needed to be scaled. In order to take advantage of our enormous dataset, we used nearly all of its layers in the transfer learning architecture. It was necessary to experiment with different layers of each CNN design as well as different Tunable parameters before settling on the optimal overall performance. In some cases, a pre-trained model might be used when a fixed feature extraction technique is required. The trainable parameters of the basic model are maintained frozen to make use of previously learned features, but the fully connected layers are kept trainable to learn from our training data. Experiments using various combinations of optimizer function, batch size, learning rate, and loss function were carried out, with the best performing combination being chosen. As a result of the use of deep neural networks, overfitting difficulties have arisen. In which case, the following approaches are used:

- Regularization techniques L2 are used. It is also called Ridge Regression.

$$\text{Cost Function: } \sum_{i=1}^n (y_i - \sum_{j=1}^p x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^p \beta_j^2$$

- Early stopping is done with the patience value of 15. If the model starts to overfit in any epoch, it is terminated.

# *Chapter 4*

## *Transfer Learning Review*

### *4.1 What is Transfer Learning*

Machine learning has already been effective in a number of real-world applications. Machine learning models are typically trained using sets of labeled samples extracted from a data collection after it has been collected. Following the collection of training data, these models may be used to predict the label of a new dataset; this new dataset is referred to as "test data." A significant number of training instances must be gathered for traditional machine learning algorithms to be successful. Only if the training and testing data are from the same domain and have the same distribution can these approaches be used. In real-world applications, training and testing instances are drawn from a variety of distinct feature spaces or distributions, however this is not the case in all circumstances. As a result of the difficulty in collecting new data with comparable features, dimensions, and distributions, this occurs on a fairly regular basis in practice. The result is that any assumption made previously would fail to hold true when applied to test data, and the model must be retrained for the new data set from the beginning, if the previous assumption is violated. In other cases, costly and time-consuming model training from scratch or the purchase of new annotated training data is not an option due to practical considerations. For this reason alone, developing a learner who can be taught in a single domain using readily available data and then apply that knowledge to another domain with sparsely labeled data is a better option than the alternative. When people apply what they've learned from previous experiences to new situations, they can learn more quickly, easily, and efficiently than if they don't. That is the driving force behind the concept of "transfer learning." In this methodology, training and testing data are linked in some ways, but it is critical to highlight that the domains, tasks, and distributions of the train and test sets may vary. During the preceding decade, there has been a great deal of interest in the areas of transfer learning and domain adaptation. Adaptation to different domains [28], [29] is a subset of transfer learning, in which the tasks are similar across different domains. Transfer learning is a technique for improving the performance of a target model when there is little or no annotated data available. It involves using knowledge from another relevant source domain and labeled data to accomplish so. Models trained with a small amount of known target data are expected to perform better at predicting the label for a fresh, unknown target data set, according to the research.

### *4.2. Approaches to Transfer Learning*

#### ***4.2.1 TRAINING A MODEL TO REUSE IT***

Consider the case when you want to complete job A but don't have enough data to train a deep neural network. Finding a task B that is tied to the first and has a huge amount of data is one way around this. After training the deep neural network on task B, the model should be utilized as a starting point for task A. The problem you're trying to solve will define how much of the model you'll need to use and how many layers you'll need to use. If you have the same input in both tasks, you may be able to reuse the model and

make predictions for your new input if you have the same input in both tasks. As an alternative, modifying and retraining distinct task-specific layers and the output layer is an approach that should be investigated.

### ***4.2.2 Pre-trained Models:***

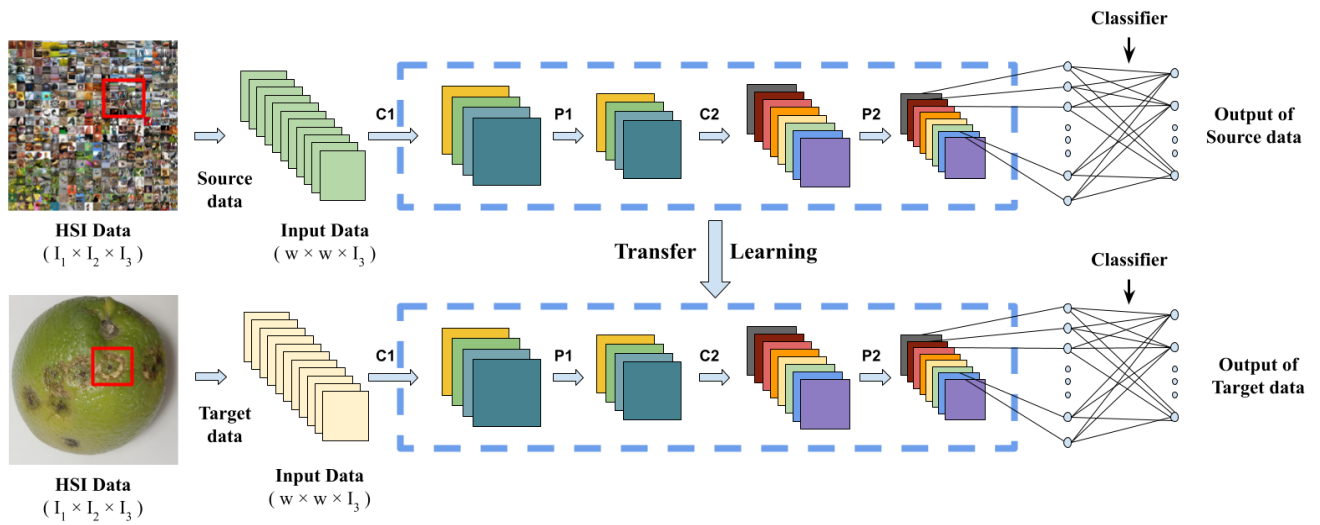
Before you can add a new classifier that is suited for your needs, you must first delete the old one. After that, you must fine-tune your model using one of three strategies: removing the original classifier, adding a new classifier that is appropriate for your purposes, and fine-tuning your model using one of three strategies: removing the original classifier, adding a new classifier that is appropriate for your purposes, and fine-tuning your model deleting the first

Enroll the whole model in the training. In this case, you utilize the pre-trained model's architecture to train it with the dataset you've given. Because you'll be developing the model from the ground up, you'll need a huge dataset (and a lot of computational power).

A layer or two is trained while the rest of the layer or layers is frozen. As you recall, lower levels refer to general characteristics (that are not dependent on the problem), and higher layers refer to specific characteristics (that are reliant on the problem) (problem dependent). By adjusting the weights of the network to various degrees of sensitivity, we can experiment with that dilemma in this way: (a frozen layer does not change during training). Most of the time, if you have a limited dataset and a big number of parameters, you'll leave more layers frozen to avoid overfitting the model. If, on the other hand, the dataset is huge and the number of parameters is modest, you can improve your model by training more layers to the new task because overfitting is not an issue in this situation.

The base of convolution should be frozen. This is the most severe example of the train versus freezing trade-off. The basic idea is to keep the convolutional base in its original form and then feed the classification algorithm with the outputs of the convolutional base. You're using a fixed feature extraction strategy with the pre-trained model, which might be useful if your computing capability is restricted, you have a tiny dataset, and/or the pre-trained model solves a problem that's very similar to the one you're attempting to solve in your study.

Figure 4.1, depicts a schematic representation of the fundamental framework with three techniques.



**Figure 4.1:** Basic Framework

### 4.2.3 FEATURE EXTRACTION

The use of deep learning to determine the best representation of your problem, which involves finding the most significant features, is another way. Informally termed as "representation learning," this strategy can frequently result in significantly higher performance than that which can be attained with manually designed representations.

## **Chapter 5**

### **INTRODUCTION TO ALGORITHMS**

In artificial neural networks, a convolutional neural network (CNN) is a type of neural network that is used to learn from data. It has proven to be a powerful method in computer vision problems, as seen by the spectacular results shared in the object identification competition known as the ImageNet Large Scale Visual Recognition Competition (ILSVRC) in 2012. The upshot is that machine learning researchers commonly recruit the assistance of field specialists to classify such photographs; nevertheless, this is an expensive and time-consuming endeavor. [31] As a result, it has been discovered that it is impossible to generate the large number of labels required to form healthy deep networks. Transfer learning makes it possible for features to be transferred between different trains and target domains. There have been various complex deep neural networks for picture categorization constructed by the deep learning community [32]. In the Python Keras library [33], they are easily accessible.

The basic framework has been sketched in fig (a).

#### ***5.1 MobileNet***

The fundamental concept behind the MobileNetV1 network is to replace the normal convolution operation with depth wise separable convolution (DSC) in order to reduce the number of model parameters. The DSC algorithm will employ the 3\*3 convolution kernel with only one layer thickness, sliding the input tensor layer by layer, and construct an output channel after each convolution. In their article titled MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications [6] presented at CVPR 2017, a team of Google engineers introduced the concept of convolutional neural networks. A new class of CNN dubbed MobileNet was proposed by Howard et al. (2017), who published their findings in 2017. Because it is designed to be a tiny network, it enables for faster and easier integration into mobile device apps. Furthermore, Togaçar, Egen, and Cömert (2020) describe MobileNet as a deep learning model that may be employed in low-cost devices with limited hardware resources. MobileNet has lowered the computational cost of its algorithms by employing a Depth wise Separable Convolution method. The architecture of this CNN is depicted in the following Figures.

**Table 5.1: MoblieNet Architecture**

Type/Stride	Filter Shape	Input Size
Conv / s2	3x 3 x 3 x32	224 x 224 x 3
Conv dw / s1	3 x 3 x 32 dw	112 x 112 x 32
Conv / s1	1 x 1 x 32 x 64	112 x 112 x 32
Conv dw / s2	3x3x64 dw	112 x 112 x 64
Conv / s1	1 x 1 x 64 x 12	56 x 56 x 64
Conv dw / s1	3 x 3 x128 dw	56 x 56 x 12
Conv / s1	1 x 1 x 128 x 128	56 x 56 x 128
Conv dw / s2	3 x 3 x128 dw	56 x 56 x 128
Conv / s1	1 x 1 x 128 x 256	28 x 28 x 128
Conv dw / s1	3 x 3x 256 dw	28 x 28 x 256
Conv / s1	1 x 1 x 256 x 256	28 x 28 x 256
Conv dw / s2	3 x3 x256 dw	28 x 28 x 256
Conv / s1	1 x 1 x 256 x 512	14 x 14 x 256
<i>5x Conv dw / s1</i>	<i>3 x 3 x 512 dw</i>	<i>14 x 14 x 512</i>
<i>1 Conv / s1</i>	<i>1 x 1 x 512 x 512</i>	<i>14 x 14 x 512</i>
Conv dw / s2	3 x 3 x 512 dw	14 x 14 x 512
Conv / s1	1 x 1 x 512 x 1024	7x 7x 512
Conv dw / s2	3 x 3 x 1024 dw	7 x 7 x 1024
Conv / s1	1 x 1 x 1024 x 1024	7 x 7 x 1024
Avg Pool / s1	Pool 7 x 7	7 x 7 x 1024
FC / s1	1024 x 1000	1 x 1 x 1024
Softmax / s1	Classifier	1 x 1 x 1000

## 5.2 DenseNet

A prior layer's output can be used as an input for a subsequent layer by using the composite function method. The convolution layer, pooling layer, batch normalization layer, and nonlinear activation layer make up this composite technique. These connections imply that the network has  $L(L+1)/2$  direct connections, the maximum allowable number. The layer count of a structure is represented by the letter "L." Researchers from Cornell University, Tsinghua University, and Facebook AI Research (FAIR) worked together on its development [7]. In some cases, the function  $H_l(\cdot)$  is a hybrid of approaches such as Batch Normalization (BN) [14], ReLU [6], and pooling [19]. Convolutional networks are used to process a single image,  $x_0$ . Nonlinear transformations  $H_l(\cdot)$  or convolution are implemented in each layer of the network, considering  $l$  is the index. (Conv). As you can see, the  $l$ -th layer's output is marked by  $x_l$ .

**ResNets.** A layer transition occurs when the output of the  $l$ -th layer is used as an input for the  $(l+1)$ -th layer [16] in a traditional convolutional feed-forward network. This layer transition is represented by the following equation:  $x_l = H_l(x_{l-1})$ . In ResNets [11], the identity function has a skip-connection that avoids non-linear transformations.:

$$x_l = H_l(x_{l-1}) + x_{l-1}. \quad (1)$$

The gradient can flow straight via the identity function in ResNets, from later layers to earlier layers, which has the benefit of being quicker. The information flow in the network may be hampered if the identity function and the output of  $H_l$  are jumbled by summing.

**Dense connectivity.** In order to dramatically improve the information flow between layers, we propose introducing direct connections from any layer to all subsequent levels as an alternate connectivity design. Because of this, the feature-maps of all preceding layers,  $x_0, \dots, x_{l-1}$ , are passed to the  $l$ -th layer, which receives the following data as input:

$$x_l = H_l([x_0, x_1, \dots, x_{l-1}]), \quad (2)$$

$[x_0, x_1, \dots, x_{l-1}]$  indicates the concatenation of the feature-maps produced in layers  $0 \dots l-1$ . This network architecture is referred to as a Dense Convolutional Network due to its dense connectedness (DenseNet). Because of the need for simplicity, we concatenate the numerous inputs of  $H_l(\cdot)$  in eq. (2) into one tensor in order to simplify the implementation.

**Composite function.** In accordance with [12], we define  $H_l(\cdot)$  as a composite function comprising three sequential operations: batch normalization (BN) [14], accompanied by a rectified linear unit (ReLU) [6], and a three-by-three convolution [7, 8]. (Conv).

**Pooling layers.**

It is not possible to use the concatenation procedure described in Eq. (2) when the size of the feature-maps changes. The down-sampling layers in convolutional networks, on the other hand, are critical since they alter the size of the feature maps. It's easier to downscale the network because of our design, which divides it into a large number of densely connected pieces. Transition layers are layers that reside between blocks and execute convolution and pooling operations. In our tests, we utilized an initial batch normalization layer and a 1x1 convolutional layer, followed by a 2x2 average pooling layer in the end experiment.

**Growth rate.** When each function  $H_l$  produces  $k$  feature-maps, it concludes that the  $l$ th layer has an input feature-map score equal to  $k_0 + k \times (l-1)$ , where  $k_0$  is the number of channels in the input layer and  $k \times l$  is the number of feature-maps in the output layer. Among the many differences between DenseNet and existing network topologies is that DenseNet can have extremely small layers, such as  $k = 12$ . Our terminology for this metric is the network's growth rate ( $k$ ). According to Section 4, a relatively low growth rate is adequate to achieve the desired result.



**Table 5.2: DenseNet Architectures for ImageNet**

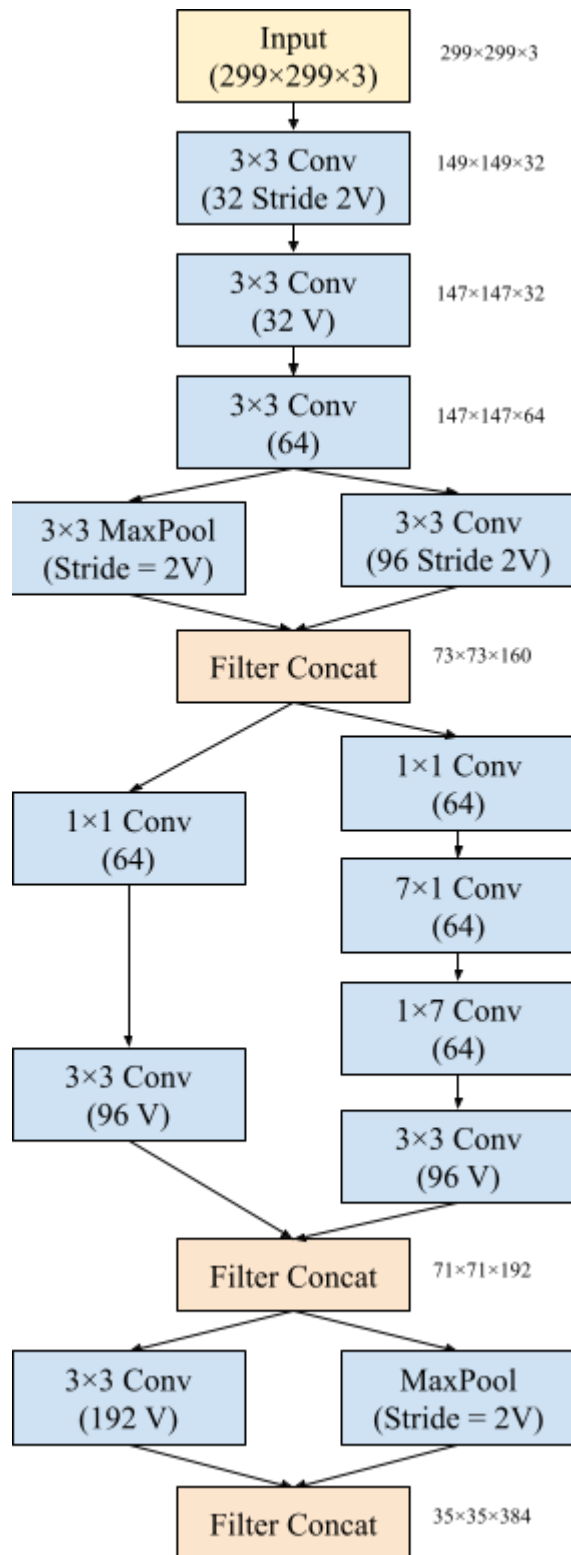
Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	$112 \times 112$	$7 \times 7$ conv, stride 2			
Pooling	$56 \times 56$	$3 \times 3$ max pool, stride 2			
Dense Block (1)	$56 \times 56$	$[1 \times 1 \text{ conv } 3 \times 3 \text{ conv}] \times 6$	$[1 \times 1 \text{ conv } 3 \times 3 \text{ conv}] \times 6$	$[1 \times 1 \text{ conv } 3 \times 3 \text{ conv}] \times 6$	$[1 \times 1 \text{ conv } 3 \times 3 \text{ conv}] \times 6$
Transition Layer (1)	$56 \times 56$	$1 \times 1$ conv			
	$28 \times 28$	$2 \times 2$ average pool, stride 2			
Dense Block (2)	$28 \times 28$	$[1 \times 1 \text{ conv } 3 \times 3 \text{ conv}] \times 12$	$[1 \times 1 \text{ conv } 3 \times 3 \text{ conv}] \times 12$	$[1 \times 1 \text{ conv } 3 \times 3 \text{ conv}] \times 12$	$[1 \times 1 \text{ conv } 3 \times 3 \text{ conv}] \times 12$
Transition Layer (2)	$28 \times 28$	$1 \times 1$ conv			
	$14 \times 14$	$2 \times 2$ average pool, stride 2			
Dense Block (3)	$14 \times 14$	$[1 \times 1 \text{ conv } 3 \times 3 \text{ conv}] \times 24$	$[1 \times 1 \text{ conv } 3 \times 3 \text{ conv}] \times 32$	$[1 \times 1 \text{ conv } 3 \times 3 \text{ conv}] \times 48$	$[1 \times 1 \text{ conv } 3 \times 3 \text{ conv}] \times 64$
Transition Layer (3)	$14 \times 14$	$1 \times 1$ conv			
	$7 \times 7$	$2 \times 2$ average pool, stride 2			
Dense Block (4)	$7 \times 7$	$[1 \times 1 \text{ conv } 3 \times 3 \text{ conv}] \times 16$	$[1 \times 1 \text{ conv } 3 \times 3 \text{ conv}] \times 32$	$[1 \times 1 \text{ conv } 3 \times 3 \text{ conv}] \times 32$	$[1 \times 1 \text{ conv } 3 \times 3 \text{ conv}] \times 48$
Classification Layer	$1 \times 1$	$7 \times 7$ global average pool			
		1000D fully-connected, softmax			

The Growth rate for all the networks is  $k = 32$ . Note that each ‘‘conv’’ layer shown in the table corresponds the sequence BN-ReLU-Conv.

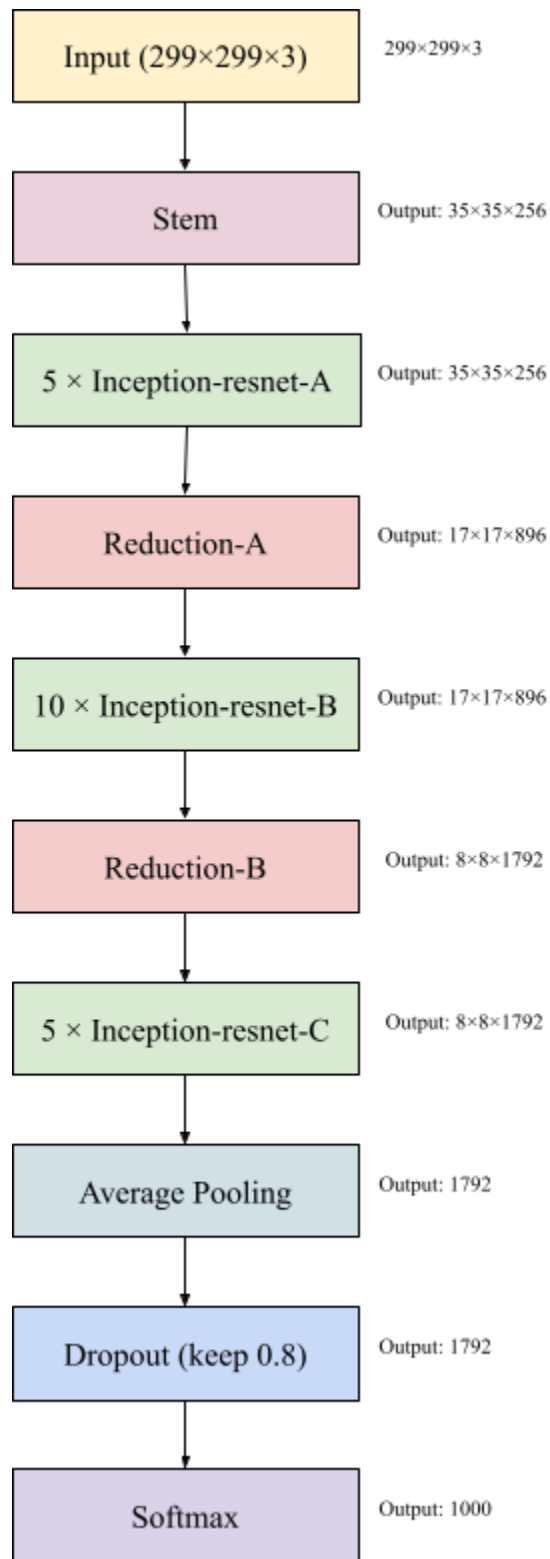
### 5.3 Inception-ResnetV2

The network's output is a list of predicted class probabilities, with image input size of  $299 \times 299$  pixels. It's made by combining the Inception structure with the Residual connection. In the Inception-Resnet block of code, convolutional filters of various sizes are combined with residual connections. Because of the utilization of residual connections, not only is the degradation problem caused by deep structures avoided, but the training time is also reduced [34]. Object identification, segmentation, human posture estimation, video classification, object tracking, and superresolution have all been successfully implemented using Krizhevsky et al's "AlexNet" network, which won the 2012 ImageNet competition [35]. Deep convolutional networks have proven to be effective in a wide range of applications in recent years, and these are just a few examples. There are a plethora of more examples. [36] Residual connections, presented by K. He et al. in [36] and the most recent improved form of the Inception architecture [37] are the two most recent theories that we will be

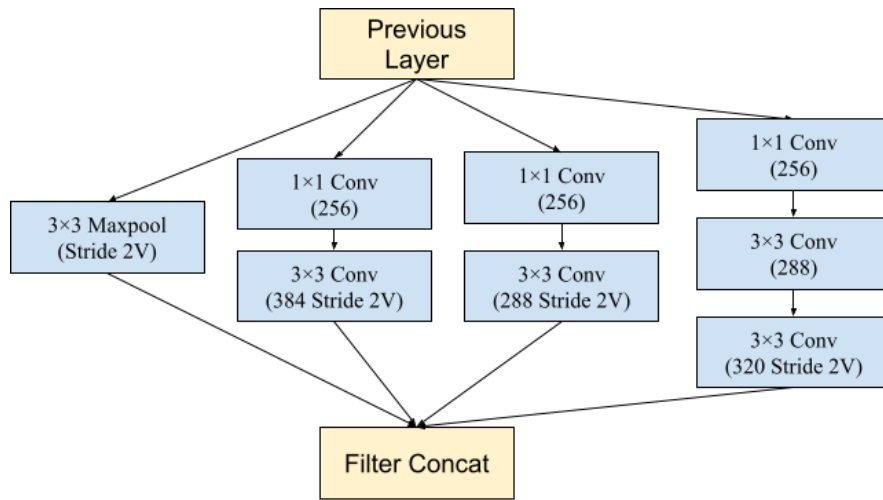
exploring in this paper. Remaining connections are of fundamental importance when training very deep architectures, according to the argument made in [36]. Using residual connections instead of the filter combining stage of the Inception architecture makes sense because Inception networks tend to be exceedingly thick. Thus, Inception would be able to profit from every advantage of the residual approach while still maintaining its processing economy. In addition to a simple integration, we looked at whether Inception itself could be made more effective by rendering it deeper and broader in nature. The outcome was a revised feature of Inception-v4 that has a more uniform, simplified architecture and contains more convolution layers than Inception-v3, which we named Inception-v4 to reflect its new architecture and increased inception module count. Historical records show that Inception-v3 had inherited a great deal from its predecessors. The necessity to split the model for distributed training using DistBelief [39] was the primary source of the technical limitations [38]. These constraints have been eliminated as a result of the migration of our training set up to TensorFlow [38], which has allowed us to drastically simplify our architecture. Section 3 describes the specifics of this streamlined architecture in more detail. On the basis of these findings, we will conduct a comparison between two pure Inception varieties, Inception-v3 and Inception-v4, with the comparably priced blended Inception-ResNet variants, which are both available at a similar price. That is not to say that the models were chosen arbitrarily; rather, the key constraint was that the parameters and computing complexity of the models had to be equivalent to the price of the non-residual models, which posed a significant challenge. As a matter of fact, we have tried Inception-ResNet variations that are larger and wider in scope, and they all fared fairly similarly on the ImageNet classification challenge [35].



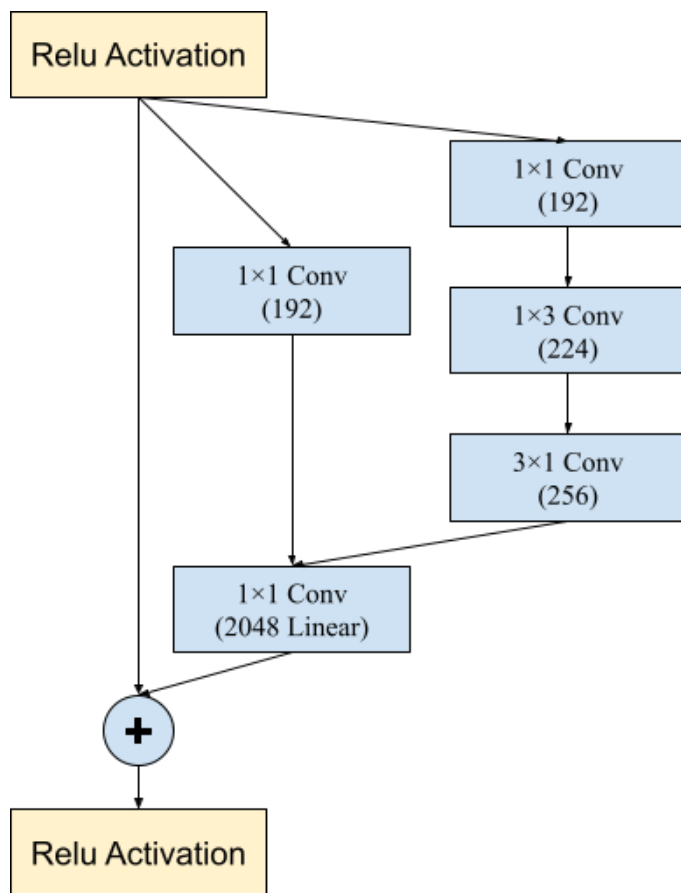
**Figure 5.1:** Schematic view for stem of the pure Inception-v4 and Inception-ResNet-v2 networks



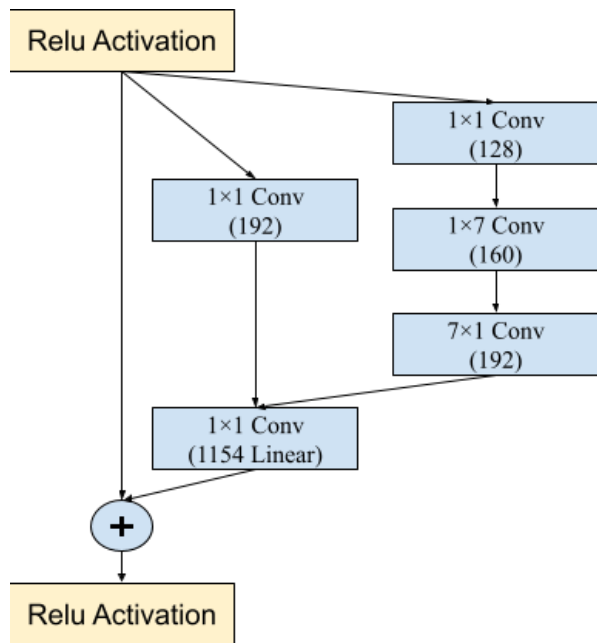
**Figure 5.2:** The overall schematic view of the Inception-v4 network



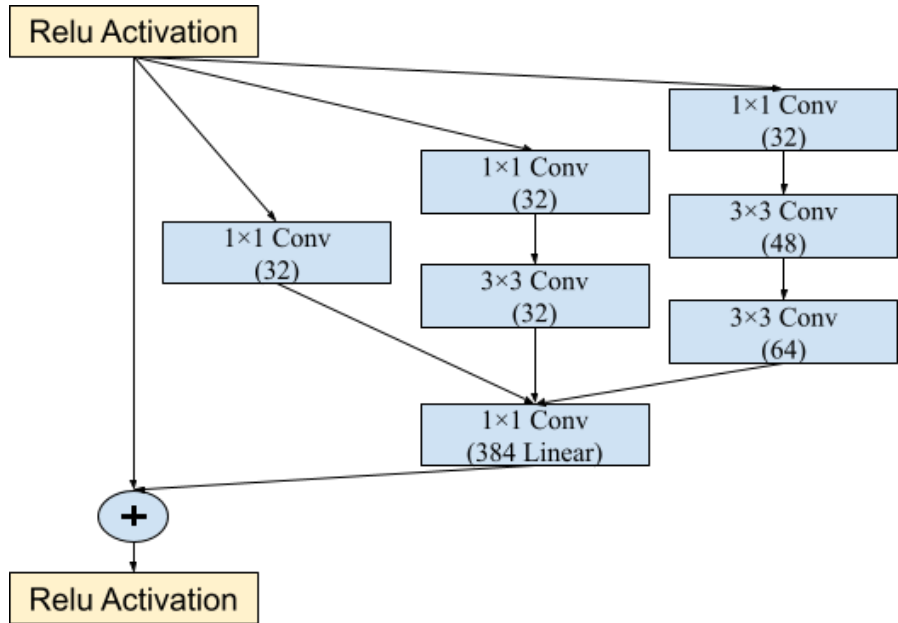
**Figure 5.3:** “Reduction-B” 17×17 to 8×8 grid-reduction module



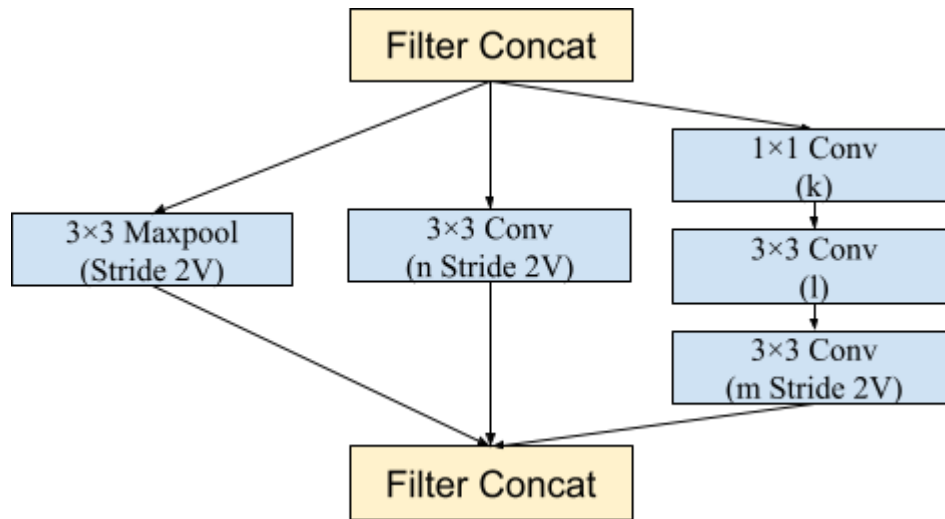
**Figure 5.4:** Schema for 8×8 grid (Inception-ResNet-C) module



**Figure 5.5:** Schematic view for  $35 \times 35$  grid (Inception-ResNet-A) module



**Figure 5.6:** The schematic view for  $35 \times 35$  to  $17 \times 17$  reduction module



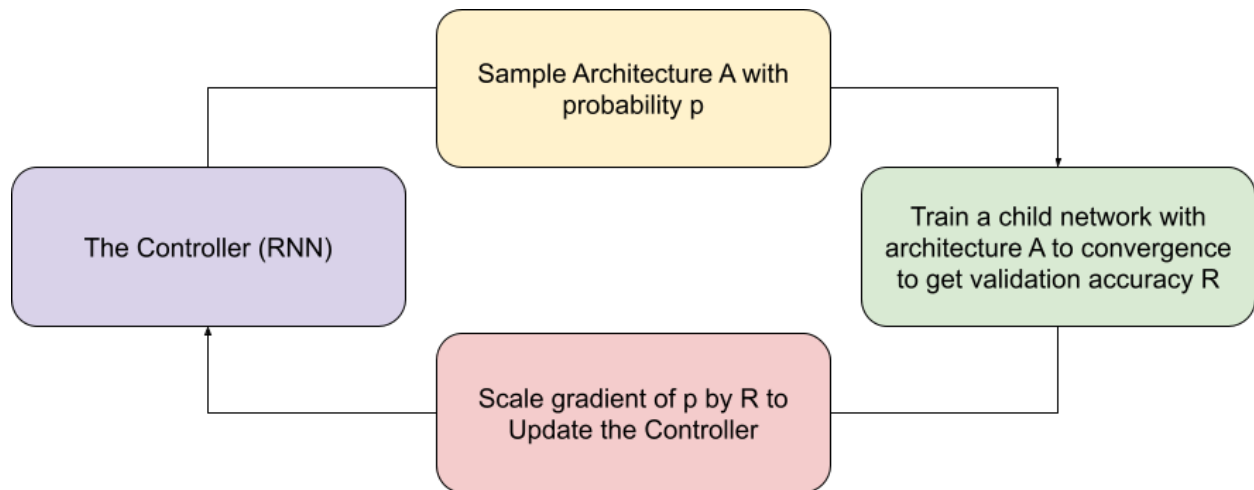
**Figure 5.7:** The schematic view for  $17 \times 17$  grid (Inception-ResNet-B) module

## 5.4 NasNet

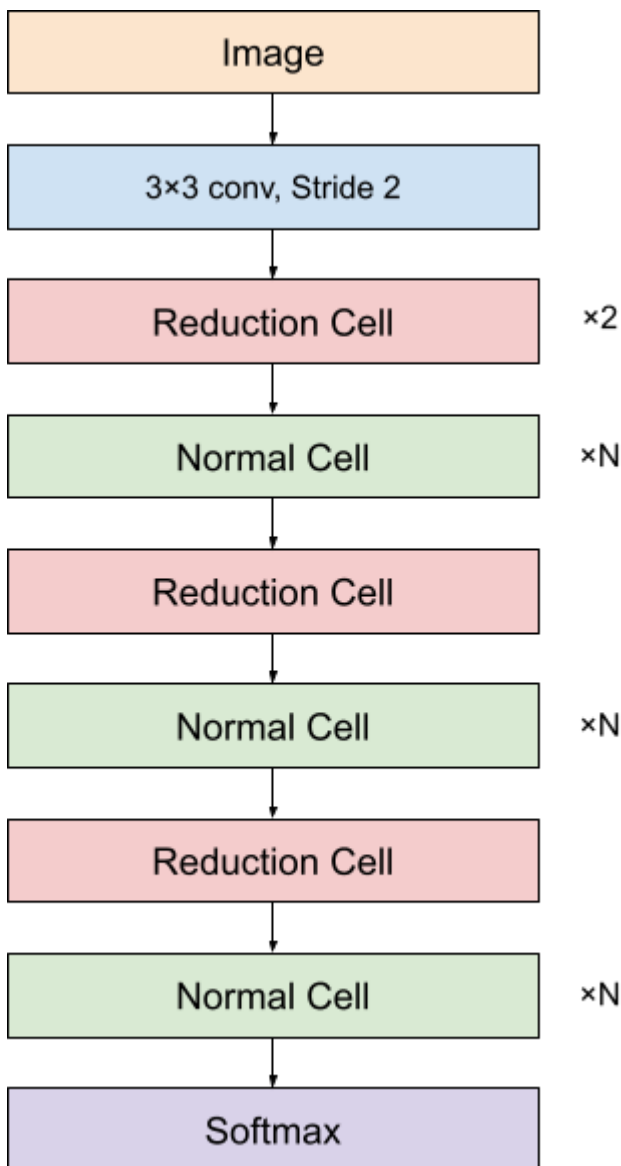
NASNet is a convolutional neural network that was discovered in the field of neural architecture search. The structure is constructed using normal cells and reduction cells as building components. We looked at two distinct sorts of NasNet models in our research.

### NasNet Large:

Neural Architecture Search Network (NASNet) models by Google Brain, with weights pre-trained on ImageNet. The default input size for the NASNetLarge model is  $331 \times 331$  and for the NASNetMobile model is  $224 \times 224$ [41].



**Figure 5.8:** Overview of Neural Architecture Search



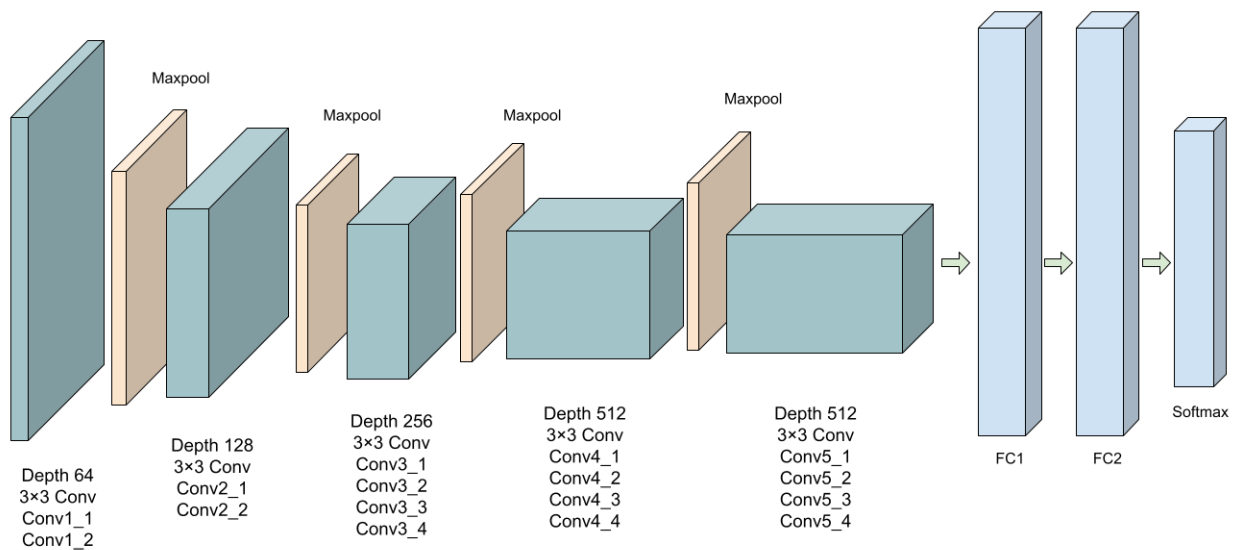
**Figure 5.9:** Scalable architectures for image classification



## 5.5 Vgg19

### Architecture

- This network was given an RGB picture of fixed size (224 by 224) as input, suggesting that the matrix was of shape (224 by 224). (224,224,3).
- The only preprocessing was removing each pixel's mean RGB value, which was determined over the whole training set.
- They were able to cover the complete visual notion using kernels of size (3 3) and a stride size of 1 pixel.
- Spatial padding was used to retain the image's spatial resolution.
- With side 2, the max-pooling process was performed over 2 \* 2 pixel windows.
- A Rectified linear unit (ReLU) was then used to incorporate non-linearity into the model, improving classification accuracy while reducing computing time. Because previous models used tanh or sigmoid functions, this model showed to be much superior than them.
- This project entailed the creation of three fully connected layers, the first two of which had 4096 channels each and the third of which had 1000 channels for 1000-way ILSVRC classification and the softmax function.



**Figure 5.10:** VGG-19 Network Architecture

# Chapter 6

## Evaluation Criteria

### **6.1 True vs False and Positive vs Negative**

This is referred to be a genuine positive when the model successfully predicts the positive class. A real negative, on the other hand, is when the model predicts the negative class properly. Whenever the model forecasts the positive class wrong, it is called a "false positive." Also known as "false negatives," these outcomes occur when a model predicts the negative class inaccurately.

### **6.2 Accuracy**

It measures how many positive and negative observations were correctly classified. It is calculated using the following formula:

$$Accuracy = \frac{tp + tn}{tp + fp + tn + fn}$$

### **6.3 Precision**

If you have a two-class imbalanced classification problem, precision is computed by dividing the number of true positives by the total number of both true positives and false positives. If you have a single-class imbalanced classification problem, precision is calculated as follows:

$$Precision = \frac{tp}{tp + fp}$$

The result is a value between 0.0 for no precision and 1.0 for complete or perfect precision.

### **6.4 Recall**

Recall is determined as the number of true positives divided by the number of true positives plus the number of false negatives in a two-class unbalanced classification issue with a single class.

$$Recall = \frac{tp}{tp + fn}$$

## 6.5 F-1 score

The traditional F measure is calculated as follows:

$$F-1 \text{ score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

This is the harmonic mean of the two fractions.

## 6.6 AUC

It's a summary of the Receiver Operating Characteristic (ROC) curve that's often utilized. The Area Under the Curve (AUC) measures how well a classifier can discriminate between classes. The area under the curve (AUC) measures how effectively the model differentiates between positive and negative categories. It can handle categories depending on the threshold or cut-off value settings.

## 6.7 Confusion Matrix

In this case, the confusion matrices are used to generate the performance matrices for ML and DL. A visual representation of the performance of ML/DL models is provided by confusion matrices. When the matrix is displayed, each row represents an instance of the real class, and each column represents an instance of the anticipated class, or vice versa. Below is an illustration of a confusion matrix.

**Table 6.1:** Confusion Matrix

Confusion Matrix		Predicted	
		Black Spot	Canker
Actual	Black Spot		
	Canker		

Here, we can see four terms like True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

Their description is given below -

- True Positive (TP): A class was predicted Positive, which is True actually.
- True Negative (TN): A class was predicted Negative, which is True actually.
- False Positive (FP): A class was predicted Positive, which is False actually.
- False Negative (FN): A class was predicted Negative, which is False actually.

# Chapter 7

## RESULT & ANALYSIS

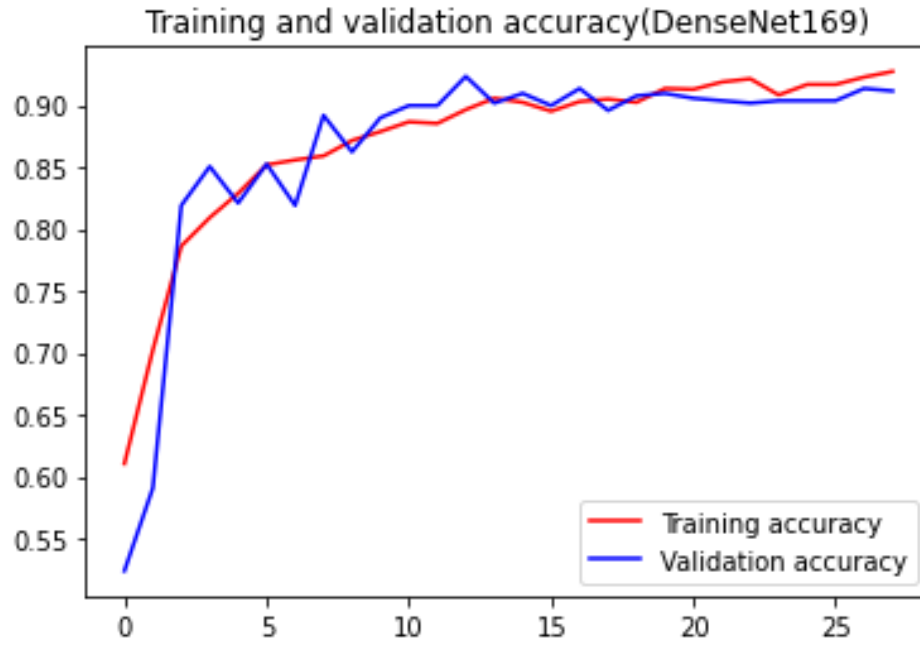
**7.1 Transfer Learning Models:** When it comes to diagnosing citrus fruit diseases, we assess the effectiveness of six well-known CNN structures (black spot & citrus-canker). The performance measure for pre-trained models is illustrated in Table X, which includes MobileNet, DenseNet169, Inception ResnetV2, NasNetLarge, NASNetMobile, and Vgg19. In different epochs spanning from 28 to 73, we looked at training accuracy, training loss, validation accuracy, validation loss, and ETA. However, after a certain amount (say, 100 epochs), accuracy does not improve, therefore increasing the number of epochs is a waste of time, as the training accuracy grows with the number of epochs. As indicated in the Methodology section, we took the appropriate steps to address this issue. When it came to getting to the optimal result, different CNN models took different amounts of time. To prevent overfitting, we looked at the loss and accuracy curves for both the Training and Validation datasets. This allowed us to validate our network on the validation set and evaluate it against the loss and accuracy curves.

### *7.1.1 DenseNet169*

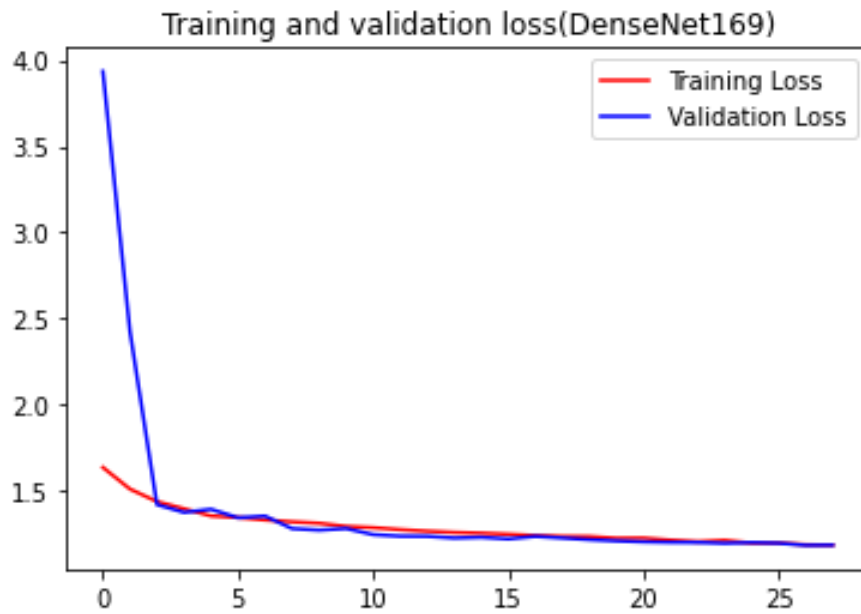
With the use of the DenseNet169 design, we are able to transfer the weights that were learned through Imagenet training to this transfer learning network. After 28 iterations, the optimal result was obtained. In order to maintain precision, the call-back function was active and terminated prematurely. Presented below is the output from the test dataset:

**Table 7.1:** Classification report (DenseNet169)

<b>Loss</b>	<b>Accuracy</b>	<b>Precision</b>	<b>Auc</b>	<b>Recall</b>	<b>F1</b>
1.1908	.9361	.9534	.9890	.9154	.9340



**Figure 7.1:** Training and Validation Accuracy(DenseNet169)



**Figure 7.2:** Training and Validation Loss (DenseNet169)

**Table 7.2:** Confusion Matrix (DenseNet169)

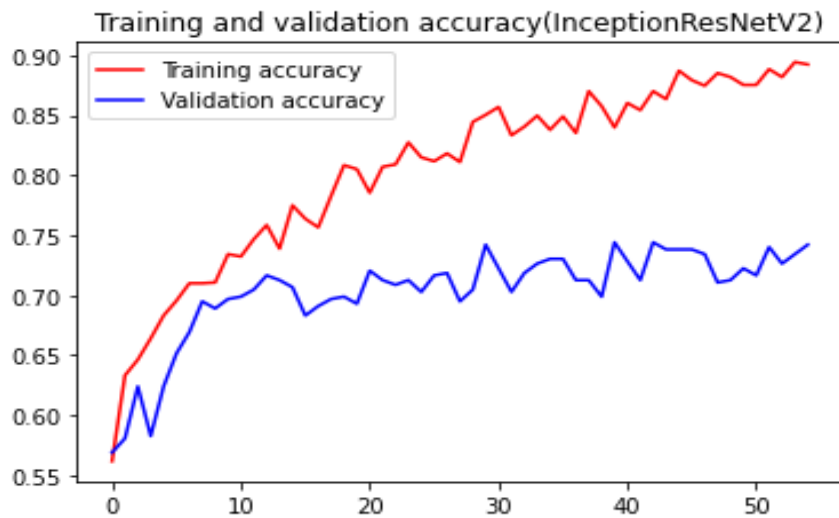
Confusion Matrix		Predicted	
		Black Spot	Canker
Actual	Black Spot	197	17
	Canker	9	184

### 7.1.2. Inception-ResnetV2

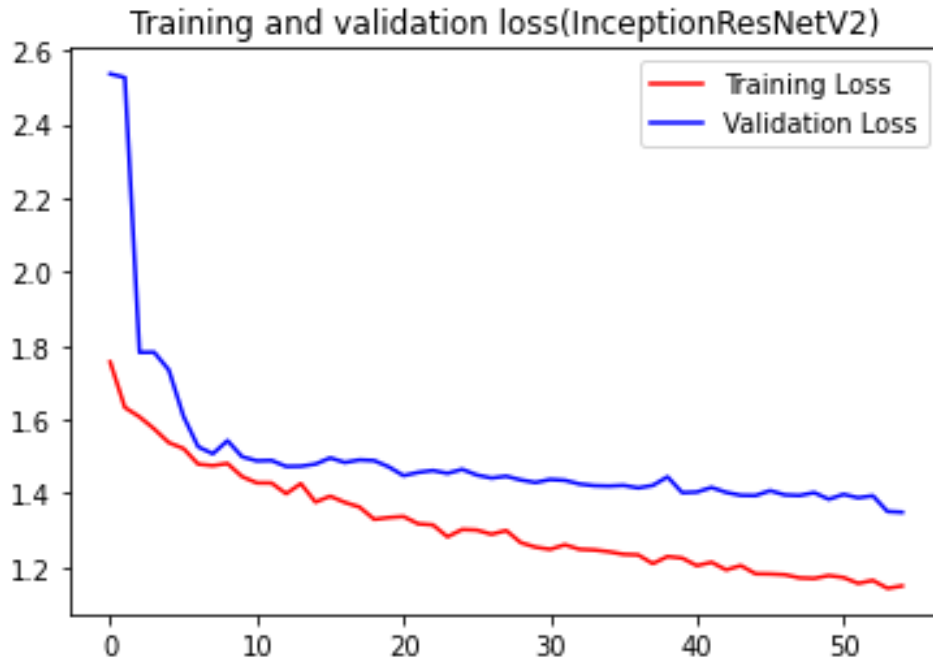
It took 28 epochs to reach the optimized result.

**Table 7.3:** Classification report (Inception-ResnetV2)

Loss	Accuracy	Precision	Auc	Recall	F1
1.2250	.8624	.8085	.9472	.9453	.875596



**Figure 7.3:** Training and Validation Accuracy(InceptionResNetV2)



**Figure 7.4:** Training and Validation Loss (InceptionResNetV2)

**Table 7.4:** Confusion Matrix (Inception-ResnetV2)

Confusion Matrix		Predicted	
		Black Spot	Canker
Actual	Black Spot	161	11
	Canker	45	190

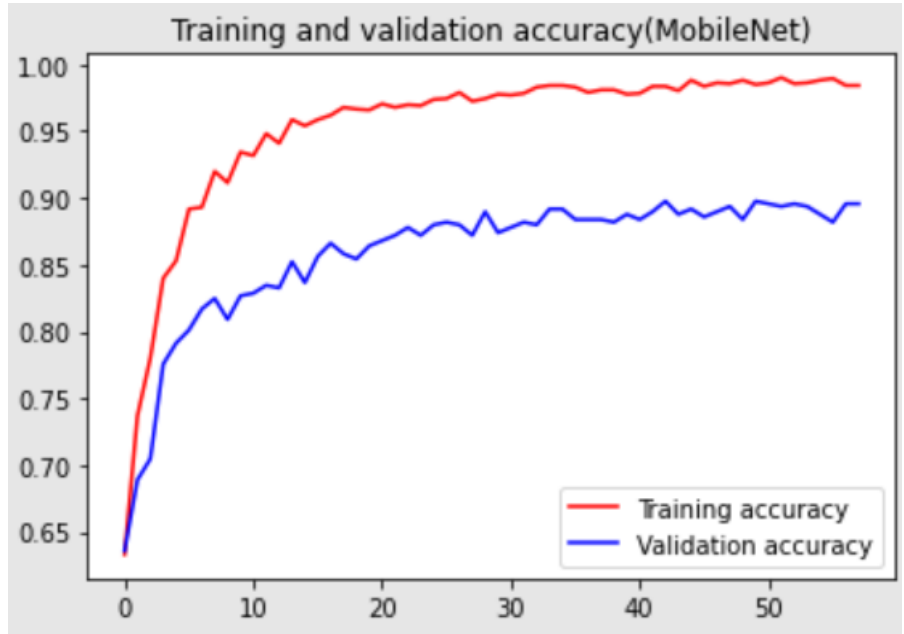
### 7.1.3. MobileNet

It took 58 epochs to reach the optimized result.

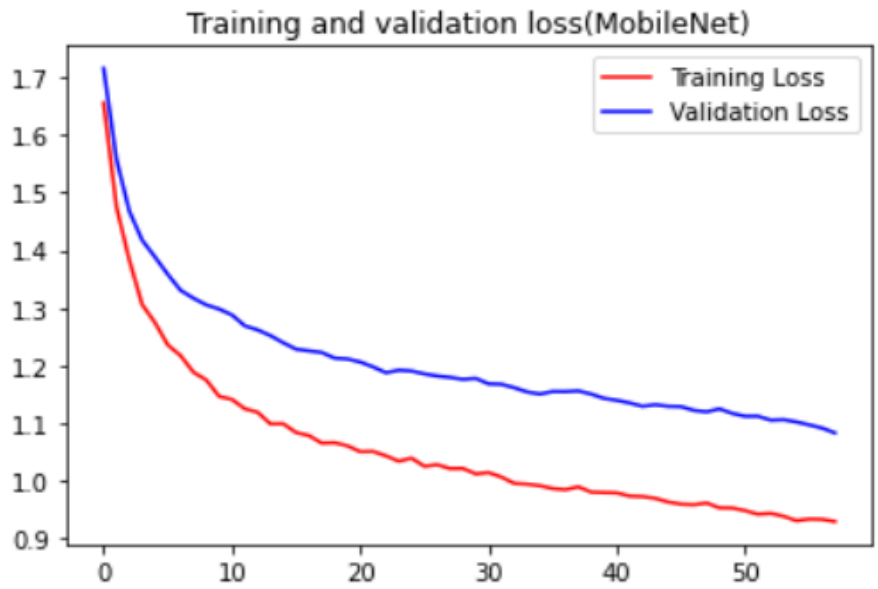
**Table 7.5:** Classification report (MobileNet)

Loss	Accuracy	Precision	Auc	Recall	F1

.9810	.9607	.9557	.9926	.9652	.9603960579437386
-------	-------	-------	-------	-------	-------------------



**Figure 7.5:** Training and Validation Accuracy(MobileNet)



**Figure 7.6:** Training and Validation Loss (MobileNet)



**Table 7.6:** Confusion Matrix (MobileNet)

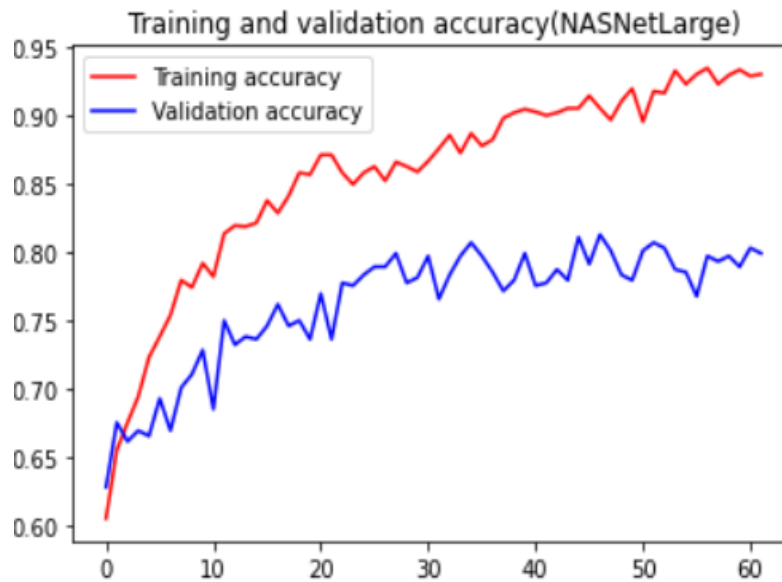
Confusion Matrix		Predicted	
		Black Spot	Canker
Actual	Black Spot	197	7
	Canker	9	194

### 7.1.4. NasNetLarge

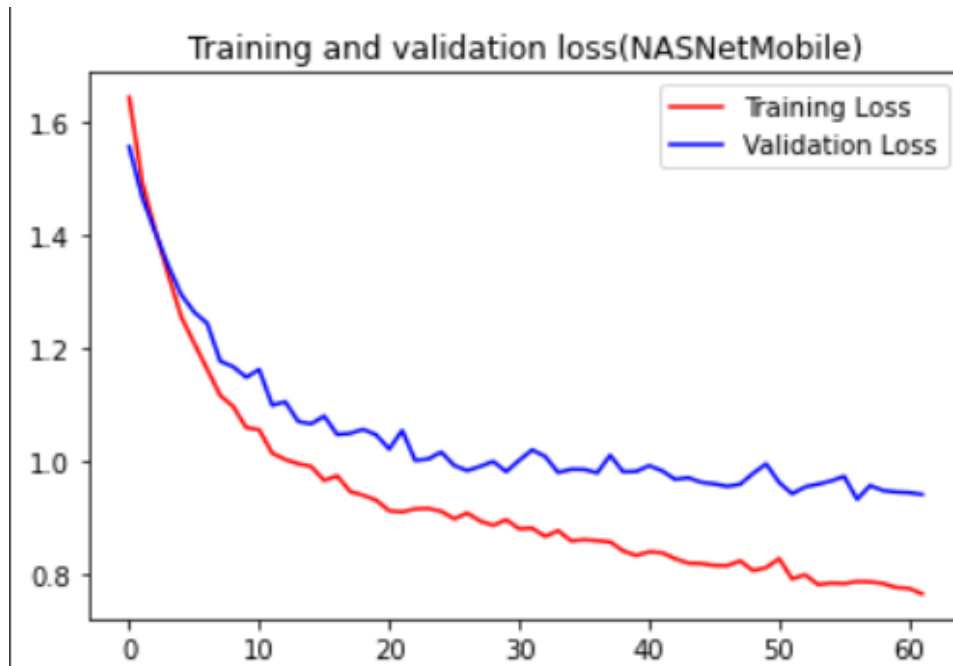
It took 62 epochs to reach the optimized result.

**Table 7.7:** Classification report (NasNetLarge)

Loss	Accuracy	Precision	Auc	Recall	F1
.7952	.9214	.9163	.9635	.9254	.9207920664438438



**Figure 7.7:** Training and Validation Accuracy(NasNetLarge)



**Figure 7.8:** Training and Validation Loss (NasNetLarge)

**Table 7.8:** Confusion Matrix (NasNetLarge)

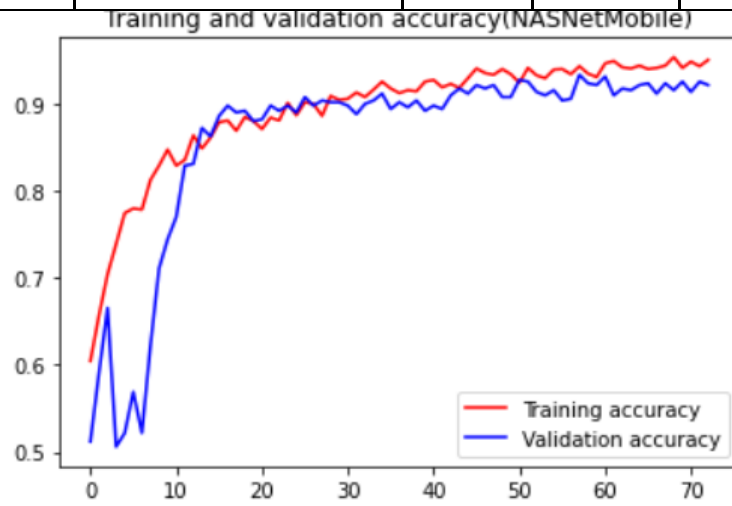
Confusion Matrix		Predicted	
		Black Spot	Canker
Actual	Black Spot	189	15
	Canker	17	186

### 7.1.5. NASNetMobile

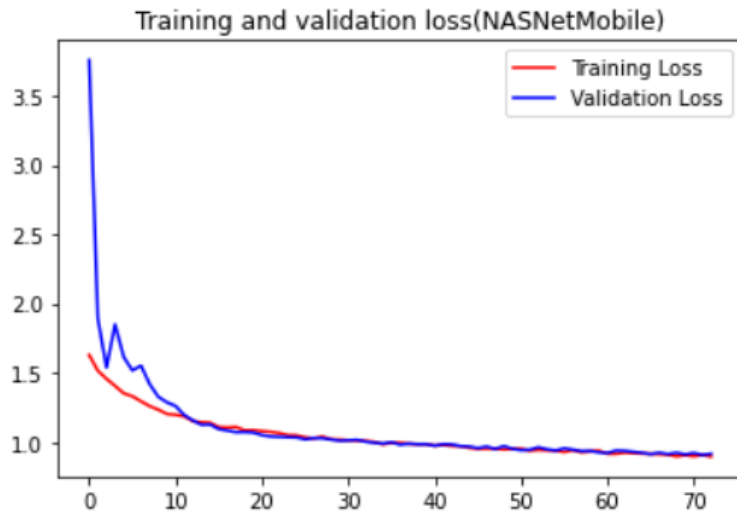
It took 73 epochs to reach the optimized result.

**Table 7.9:** Classification report (NASNetMobile)

Loss	Accuracy	Precision	Auc	Recall	F1
.8909	.9558	.9841	.9932	.9254	.9538461557083582



**Figure 7.9:** Training and Validation Accuracy(NasNetMobile)



**Figure 7.10:** Training and Validation Loss (NasNetMobile)

**Table 7.10:** Confusion Matrix (NASNetMobile)

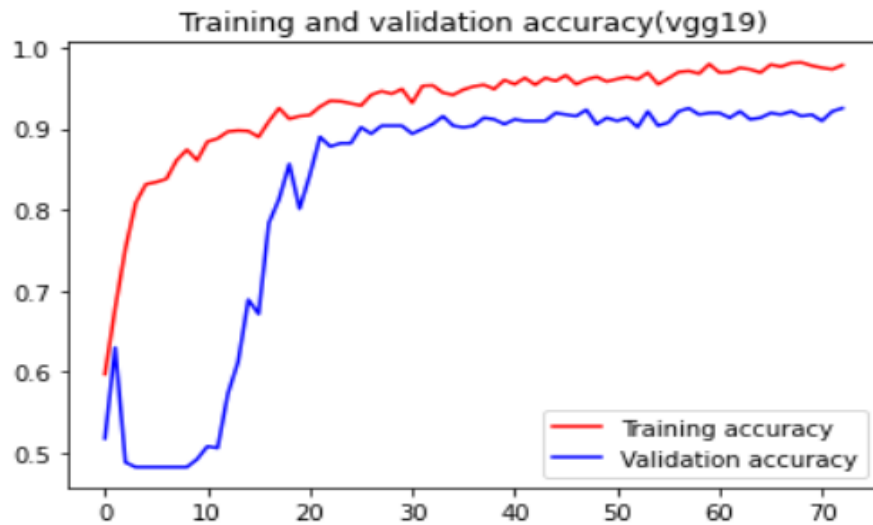
Confusion Matrix		Predicted	
		Black Spot	Canker
Actual	Black Spot	203	15
	Canker	3	186

### 7.1.6. Vgg 19

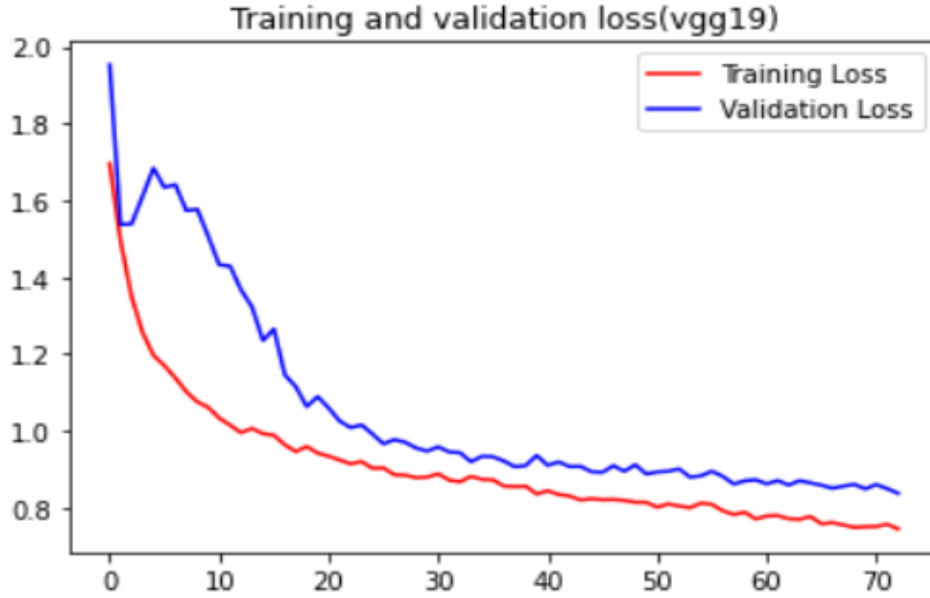
It also took 73 epochs to reach the optimized result.

**Table 7.11:** Classification report (Vgg19)

Loss	Accuracy	Precision	Auc	Recall	F1
.7538	.9754	.9799	.9933	.9701	.9750000265166163



**Figure 7.11:** Training and Validation Accuracy(Vgg19)



**Figure 7.12:** Training and Validation Loss (Vgg19)

**Table 7.12:** Confusion Matrix (Vgg19)

Confusion Matrix		Predicted	
		Black Spot	Canker
Actual	Black Spot	202	6
	Canker	4	195

## 7.2. Observation

The results show that the accuracy of individual models ranged from 0.8624 to 0.9754, depending on the model. The deepest model, VGG-19, achieved the highest accuracy of 9754 percent with 9799 percent precision, 9701 percent recall, and 9750 percent F-1 score, while the NASNetMobile model achieved the highest Precision of 9841 percent. The shallower models, on the other hand, were shown to have significantly worse performance.

**Table 7.13: Comparison Table**

<b>Network's Name</b>	<b>Accuracy</b>	<b>Precision</b>	<b>Auc</b>	<b>Recall</b>	<b>F1</b>
MobileNet	.9607	.9557	.9926	.9650	.9604
DenseNet169	.9361	.9534	.9890	.9150	.9340
Inception-ResnetV2	.8624	.8085	.9472	.9450	.8716
NasNetLarge	.9214	.9163	.9635	.9254	.9208
NASNetMobile	.9558	.9841	.9932	.9254	.9938
vgg19	.9754	.9799	.9933	.9701	.9750

## **Chapter 8**

### **CONCLUSION & FUTURE SCOPES**

#### ***8.1 Conclusion***

Computer-Aided Diagnosis (CAD) has extensive dominance in the pathological arena in order to precisely and reliably determine the underlying health problems of crops. To take advantage of the "Prevention is better than cure" philosophy, however, numerous AgriTech companies and research institutes have already begun working on the early detection of diseases by utilizing early-stage biomarkers.

It is the purpose of this paper to describe the detection of black-spot and citrus-canker-infected fruit. Transfer learning was used to try to create a model that focuses on recognizing citrus diseases by processing digital photographs of citrus fruit that had been acquired before. The modified VGG-19 demonstrated the best performance, achieving 97.54 percent accuracy, 97.5 percent F-1 score, and an AUC ROC of 99.33 percent, whereas DenseNet169, MobileNet, NasNetLarge, and NASNetMobile were all in the same ballpark in terms of overall performance. When it comes to precision, NASNetMobile produced the best results, with an accuracy of 98.41 percent. The positive results obtained from this study, which used transfer learning on a new dataset, open the door to a plethora of new research opportunities in the future. If enough time and resources are available, it is possible to work with larger datasets.

In our effort, we hope to lower the cost of food monitoring by computerizing the disease detection procedure, which was previously carried out by human eyes.

Because high-performance computing equipment is readily available to us, we must take advantage of them by using them in agricultural applications to ensure that not a single fruit is wasted as a result of these diseases.

This early diagnosis of disease may allow us to take the required precautions against the outbreak of Blackspot and Canker disease, and we would be able to continue to provide high-quality citrus fruit supplies.

## ***8.2 Limitations***

Our team experimented with a number of different convolutional architecture models and even constructed our own model, but none of them were able to deliver the results we were looking for.

Our research discovered that the deeper convolutional architecture was more successful in extracting the necessary features when compared to the shallower ones when it came to extracting the required features.

While initially configured to train our neural network model over 100 epochs, our callback function detected overfitting throughout the training period and ended the training process immediately.

Because of the poor resolution pictures, several of the black spots were so small that our model was unable to distinguish them from the image.

## ***8.3 Future Scopes***

Using more complex and updated architectures such as EfficientNetV2B3, EfficientNetV2S, and EfficientNetV2M in the future to achieve greater efficiency is something we are looking forward to.

As part of our performance testing, we will also run our model on larger and more realistic datasets.

The goal is to integrate our model in a hardware project that will use real-time images captured by a camera to achieve the goal of detecting the disease early.

As we improve our prediction, we will tweak the model and then apply it to other agricultural sectors that are linked to the one we're studying.



## References

1. Jahanbakhshi, Ahmad, et al. "Classification of Sour Lemons Based on Apparent Defects Using Stochastic Pooling Mechanism in Deep Convolutional Neural Networks." *Scientia Horticulturae*, Elsevier, 17 Dec. 2019, <https://www.sciencedirect.com/science/article/abs/pii/S0304423819310192>.
2. K.Lalitha, K.Muthulakshmi, A.Vinothini(2015) " Proficient acquaintance based system for citrus leaf disease recognition and categorization",
3. Bulanon, Duke M., et al. "Citrus Black Spot Detection Using Hyperspectral Image Analysis." *Agricultural Engineering International: CIGR Journal*, 2013, <https://cigrjournal.org/index.php/Ejournal/article/view/2382>.
4. Kotze, Johannes Marthinus. "Studies on the Black Spot Disease of Citrus Caused by *Guignardia Citricarpa* Kiely with Particular Reference to Its Epiphytology and Control at Letaba." *UPSpace Home*, University of Pretoria, 18 Jan. 2007, <https://repository.up.ac.za/handle/2263/23891>.
5. Lins, Emery C., et al. "Detection of Citrus Canker in Citrus Plants Using Laser-Induced Fluorescence Spectroscopy - Precision Agriculture." *SpringerLink*, Springer US, 17 June 2009, <https://link.springer.com/article/10.1007/s11119-009-9124-2>.
6. Rauf, Hafiz Tayyab, et al. "A Citrus Fruits and Leaves Dataset for Detection and Classification of Citrus Diseases through Machine Learning." *Data in Brief*, vol. 26, 2019, p. 104340., <https://doi.org/10.1016/j.dib.2019.104340>.
7. Sunny, Shoby and Dr. M. P. Indra Gandhi. "An Efficient Citrus Canker Detection Method based on Contrast Limited Adaptive Histogram Equalization Enhancement." (2018).
8. Thangadurai, K. & Padmavathi, K. (2019). Citrus Canker Disease Detection Using Genetic Algorithm in Citrus Plants.
9. Sharif, Muhammad, et al. "Detection and Classification of Citrus Diseases in Agriculture Based on Optimized Weighted Segmentation and Feature Selection." *Computers and Electronics in Agriculture*, vol. 150, 2018, pp. 220–234., <https://doi.org/10.1016/j.compag.2018.04.023>.
10. Doh, Benjamin, et al. "Automatic Citrus Fruit Disease Detection by Phenotyping Using Machine Learning." *2019 25th International Conference on Automation and Computing (ICAC)*, 2019, <https://doi.org/10.23919/iconac.2019.8895102>.

11. Pan, Wenyan, et al. "A Smart Mobile Diagnosis System for Citrus Diseases Based on Densely Connected Convolutional Networks." *IEEE Access*, vol. 7, 2019, pp. 87534–87542., <https://doi.org/10.1109/access.2019.2924973>.
12. Russakovsky, Olga, et al. "Imagenet Large Scale Visual Recognition Challenge." *International Journal of Computer Vision*, vol. 115, no. 3, 2015, pp. 211–252., <https://doi.org/10.1007/s11263-015-0816-y>.
13. Krizhevsky, Alex, et al. "ImageNet Classification with Deep Convolutional Neural Networks." *Communications of the ACM*, vol. 60, no. 6, 2017, pp. 84–90., <https://doi.org/10.1145/3065386>.
14. Alzubaidi, Laith, et al. "Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions." *Journal of Big Data*, vol. 8, no. 1, 2021, <https://doi.org/10.1186/s40537-021-00444-8>.
15. Howard, Andrew G., et al. "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications." *ArXiv.org*, 17 Apr. 2017, <https://arxiv.org/abs/1704.04861>.
16. Huang, Gao, et al. "Densely Connected Convolutional Networks." *ArXiv.org*, 28 Jan. 2018, <https://arxiv.org/abs/1608.06993>.
17. Szegedy, Christian, et al. "Inception-V4, Inception-Resnet and the Impact of Residual Connections on Learning." *ArXiv.org*, 23 Aug. 2016, <https://arxiv.org/abs/1602.07261>.
18. Zoph, Barret, et al. "Learning Transferable Architectures for Scalable Image Recognition." *ArXiv.org*, 11 Apr. 2018, <https://arxiv.org/abs/1707.07012>.
19. Simonyan, Karen, and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition." *ArXiv.org*, 10 Apr. 2015, <https://arxiv.org/abs/1409.1556>.
20. Czakon, Jakub. "24 Evaluation Metrics for Binary Classification (and When to Use Them)." *Neptune.ai*, 5 Jan. 2022, <https://neptune.ai/blog/evaluation-metrics-binary-classification>.
21. Brownlee, Jason. "How to Calculate Precision, Recall, and F-Measure for Imbalanced Classification." *Machine Learning Mastery*, 1 Aug. 2020, <https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced-classification/>.
22. Aniruddha. "AUC-Roc Curve in Machine Learning Clearly Explained." *Analytics Vidhya*, 20 July 2020, <https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/>.

23. Fao. "Citrus Fruit Fresh and Processed Statistical Bulletin 2020." *Policy Commons*, FAO, 13 Sept. 2021, <https://policycommons.net/artifacts/1813113/citrus-fruit-fresh-and-processed-statistical-bulletin-2020/2549212/>.
24. <https://www.sciencedirect.com/science/article/abs/pii/S0168169917306373>.
25. Prevalence and Severity of Different Citrus Diseases in Sylhet Region. [https://www.researchgate.net/publication/341319002\\_Prevalence\\_and\\_severity\\_of\\_differ ent\\_citrus\\_diseases\\_in\\_Sylhet\\_region](https://www.researchgate.net/publication/341319002_Prevalence_and_severity_of_differ ent_citrus_diseases_in_Sylhet_region).
26. Uddin, H., Latif, A., Huq, F., Mia, A. T., Latif, A. and Ahmed, S. (2014). Pest Risk Analysis (PRA) of Citrus under Strengthening Phytosanitary Capacity in Bangladesh Project (SPCB), DAE.
27. *Automatic Citrus Fruit Disease Detection by Phenotyping Using Machine ...* [https://www.researchgate.net/publication/337212837\\_Automatic\\_Citrus\\_Fruit\\_Disease\\_Detection\\_by\\_Phenotyping\\_Using\\_Machine\\_Learning](https://www.researchgate.net/publication/337212837_Automatic_Citrus_Fruit_Disease_Detection_by_Phenotyping_Using_Machine_Learning).
28. G. Csurka, "A comprehensive survey on domain adaptation for visual applications," in *Domain adaptation in computer vision applications*. Springer, 2017, pp. 1–35.
29. A. Farahani, S. Voghoei, K. Rasheed, and H. R. Arabnia, "A brief review of domain adaptation," arXiv preprint arXiv:2010.03978, 2020.
30. Farahani, Abolfazl, et al. "A Concise Review of Transfer Learning." *ArXiv.org*, 5 Apr. 2021, <https://arxiv.org/abs/2104.02144v1>.
31. BAR, Y. et al. Deep learning with non-medical training used for chest pathology identification. In: *SPIE medical imaging, Orlando*, v. 9414, p. 7, 2015.
32. BUETTI-DINHA, A. et al. Deep neural networks outperform human expert's capacity in characterizing bioleaching bacterial biofilm composition. *Biotechnology Reports*, v. 22, p. 108–119. 2019. Available in: <https://doi.org/10.1016/j.btre.2019.e00321>. Access: 22 Ago. 2020.

33. GAD, A. F. Practical computer vision applications using deep learning with CNNs: With Detailed Examples in Python Using TensorFlow and Kivy. New York: Apress, 2018. E-book. ISBN: 978-1-4842-4167-7 Available in: <https://doi.org/10.1007/978-1-4842-4167-7>. Access: 29 ago. 2020.
34. A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.
35. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. 2014.
36. K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385, 2015.
37. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. arXiv preprint arXiv:1512.00567, 2015.
38. M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Ward, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from [tensorflow.org](http://tensorflow.org).
39. J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, A. Senior, P. Tucker, K. Yang, Q. V. Le, et al. Large scale distributed deep networks. In Advances in Neural Information Processing Systems, pages 1223–1231, 2012.
40. Szegedy, Christian, et al. "Inception-V4, Inception-Resnet and the Impact of Residual Connections on Learning." *ArXiv.org*, 23 Aug. 2016, <https://arxiv.org/abs/1602.07261v2>.
41. F. Chollet. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.