Plant Disease Diagnosis Using Deep Transfer Learning Architectures- VGG19, MobileNetV2 and Inception-V3

by

Khadija-Tul-Kobra
17301018
Rahmatul Rashid Suham
18101681
Maisha Fairooz
18101067

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
Brac University
May 2022

Declaration

It is hereby declared that

1. The thesis submitted is my/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:

Khadija-Tul-Kobra

Khadija-Tul-Kobra
17301018

Rahmatul Rashid Suham

Rahmatul Rashid Suham
18101681

Maisha Fairooz

Maisha Fairooz
18101067

Approval

The thesis/project titled "Plant Disease Diagnosis Using Deep Transfer Learning Architectures- VGG19, MobileNetV2 and Inception-V3" submitted by

1. Khadija-Tul-Kobra (17301018)
2. Rahmatul Rashid Suham (18101681)
3. Maisha Fairooz (18101067)

Of Spring, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on May 26, 2022.

Examining Committee:

Supervisor:
(Member)

Dr. Jia Uddin
Associate Professor
Department of Computer Science and Engineering
BRAC University (On Leave)
Assistant Professor(Research Track)
Technology Studies Department
Woosong University, Daejeon, South Korea

Co Supervisor:
(Member)

Ahanaf Hassan Rodosh
Lecturer
Department of Computer Science and Engineering
BRAC University

Thesis Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Associate Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi
Chairperson
Department of Computer Science and Engineering
BRAC University

Ethics Statement

We researched a range of papers and used websites, journals, and publications to support our proposed method. Our data was sourced from Kaggle and Github.

Abstract

The importance of a tree's involvement in human life and the environment cannot be overstated. Plants, like humans and animals, are susceptible to disease. Many vegetation diseases might impair a plant's healthy development. For accurate identification and treatment of plant pathogens, precise detection of those chronic conditions is essential. This paper represents 3 deep learning approaches to distinguish and classify plant diseases by analyzing the leaf of a given plant. We worked on late leaf curl, leaf spot, mosaic virus, black rot, powdery mildew, common rust, bacterial spot, leaf scorch, syndromes of late and early blight of crops similar to corn, potato, tomato, squash, pepper, cherry, grape, orange, strawberry, apple etc. The proposed strategy improves disease identification and classification of deformed collected leaves. The model performs its function by categorizing images into two groups, diseased and healthy. Moreover, deep learning architectures are made up of several processing layers that learn the data visualizations with discrete levels of abstraction. Collecting data sets is one of the most crucial steps to creating any recognition system. Labeling an image means pinpointing the subject we will be trying to find, Training the algorithms through those images to detect the subjects is critical in detecting diseases. In this research paper, to detect diseases from images, firstly, we collected data sets containing more than 87,123 images of plant diseases. After that, we labeled those images in 38 labels and we used VGG19 model on those images to train the model to detect diseases from the images given to it, afterwards two deep learning models MobileNetV2 and Inception-v3 was used to detect diseases which provided us with 94.21%, 97.93% and 98.52% accuracy respectively. In short, we're using three deep learning models and comparing the accuracy rate on a huge data set with 38 classes which will help the masses to detect abnormalities in plants. It will also help the harvesters related to the agricultural works find the contagion in their cultivated crops further to develop our horticulture sector and our farmers' situation.

Keywords: Deep Learning, Plant diseases, VGG19, MobileNetV2, Inception-v3, Crop.

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.
Secondly, to our supervisor Dr. Jia Uddin for his kind support and advice in our work. He helped us whenever we needed help.
And finally to our parents without their throughout sup-port 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

Chapter 1

Introduction

1.1 Overview

Deformed leaves have been a significant sign of infected plants and crop damage, tainted leaves detection by hand is a time consuming and labor-intensive task. A variety of techniques have been employed to detect deformed leaves and many other plant diseases. Each of the strategies, however, has its own set of limitations. This is why this research paper aims to see if the present algorithms for identifying plant diseases can be used to assist farmers to detect plant blights. The key to future reconstruction decision making is the evaluation of damage based on gathered data. Both internal and external factors cause plant diseases. Internal factors include the deterioration and responsiveness or durability of the pavement material to climate change factors such as heavy rainfall and fertilization of soil, insects in the soil and temperature. External factors include taking care of the plants and constructive management. Keeping in line with the development of Bangladesh, the amount of plant diseases is increasing rapidly. Deep learning, a part of machine learning, is an advanced type of machine learning that employs neural networks that function similarly to the human brain. Semantic characteristics are used as the categorization strategy in traditional methods. According to LeChun et al, deep learning is a neural network learning process and one of its features is that it can automatically retrieve features from visual patterns [1]. With the ability to analyze multimedia information such as photographs collected by various computing systems, computers have evolved to be a critical technology in a variety of applications such as defense, medical, agriculture, and engineering. Plant diseases have been identified as a growing hazard to global food security. As a result, detecting plant diseases is the most critical stage in producing high-quality crops. Because of the diversity and similarity of plants in nature, classifying them as “with and without illness” is a challenging task [2]. The Investigation of machine learning and neural network’s methodology for agricultural applications has accelerated in research, opening the door to novel apps and continuous improvement over current methods. Our algorithm is used to identify the extraction of characteristics in a synthetic manner and to provide a clear detection on a specified data set. In general, “CNN” is the best solution for any prediction issue that involves input picture data and requires little pre-processing. It’s designed to categorize large-scale photos. Because the choice of architecture is so difficult, it’s critical to research and explain effective architectures to help us with our research. On the “AgrilPlant” dataset, GoogleNet produced the best results, with accuracies of 98.33 percent and 97.66 percent, respectively "LeafSnap" is a dataset [2]. Instead of CNN based research, this paper is focused on deep learning based models and the comparison in their accuracy rate. This proposed model of ours will predict maladies (of plants) from a given leaf picture hence resulting in applying proper medicine and steps for the affected crop.

1.2 Research Problem

A plant's blight detection system is designed to find out which disease is causing the unexpected stain, spots, vapidity, dullness on the given leaf of a certain plant/crop. We study the various ways that the system's goals can be achieved. Our's is a land of agriculture, where a good portion of our economy is based on food export. Looking at our economy we understand how crops play a vital role to secure the poor farmers' fortune. Now, crop damage becomes an alarming issue for a developing country with impoverished farmers. The majority of vegetables are cultivated in the winter, from September until February, while only a handful are produced during summer, between March and September. Vegetable agriculture wraps 4,98,073 acres and gives abundant opportunities for men and women from under served communities. However, the productivity per unit area is minimal since insect pests cause 30–40% losses in general and even 100% damages in the case of a crisis if no control measures are taken. management measures are taken. Annual yield losses in vegetables are estimated to be around 25% owing to insect pests alone, according to a conservative estimate [3]. Deep learning's image-based detection has become more popular due to the constant increase in the complexity of diseases. The view behind ML and DL(Deep Learning) is to create a set of algorithms which are able to learn by itself and recognize between typical and abnormal images of a plant. The method of instructing a machine takes distinctive shapes; supervised, unsupervised and reinforcement learning. Image-based plant illness location is among the fundamental exercises in accuracy horticulture for watching frequency and measuring the seriousness of inconstancy in crops. 70% to 80% of the variability are credited to illnesses caused by pathogens, and 60% to 70% show up on the clears out in comparison to the stem and natural products. This work gives a comparative investigation through the demonstrated execution of the two eminent models of machine learning, the support vector machine (SVM) and profound deep learning (DL), for plant malady discovery utilizing leaf picture information. Until as of late, most of these picture handling methods had been, and a few still are, abusing what a few considered as "shallow" machine learning structures [4] . The robotized distinguishing proof of plant infections based on plant clears may be a significant landmark within the field of horticulture. Additionally, the early and convenient recognizable proof of plant diseases emphatically impacts trim abdicate and quality. Due to the development of a large number of small items, indeed, an agriculturist and pathologist may frequently come up short to identify the disorders in plants by visualizing disease-affected takes off. Deep-learning-based methods, especially CNNs, are the most promising approach for naturally learning absolute and discriminating highlights. Deep learning (DL) comprises distinctive convolutional layers that extract important features from the data set and filter out less important features. Plant-disease location can be localized by employing a deep-learning model. Insects are important players in agro-ecosystems, wreaking havoc on crops. Agricultural producers use a variety of strategies to control insect pests, including synthetic pesticides, but these treatments can disturb beneficial insect activity. DL models can learn relevant highlights from input photographs at unprecedented convolutional levels, analogous to the work of the human brain, DL is the most well-known engineering. With good segmentation precision and a DL can solve problems with a minor error rate. complicated situations quickly and adequately. Convolutional, pooling, and fully linked layers and enactment functions are among the components of the DL model. Furthermore, research shows that Lu et al [5]. utilized distinctive pooling operations, channel sizes, and calculations to distinguish 10 common rice infections. The proposed convolutional neural networks (CNNs) based solution shows an accomplished precision of 95.48%. Dechant et al [6]. prepared CNNs to naturally identify northern leaf scourge of maize.

1.3 Research Objective

Our target in this research paper is to build a system where we'll detect a plant's unhealthiness using deep learning approaches or models-VGG19, MobileNetV2 and Inception-v3. However, this system is built to collect images of plant leaves through any devices and detect if the plant is infected with any malady or is it healthy based on the deep learning algorithms.

1. Build an image data set to train our algorithm for detecting the abnormality of a given plant's leaf so that the algorithm gives a result immediately.

2. Ensure that distinct 38 classes of data are represented as images in order to construct the Transfer learning-based system.

3. To dodge a huge amount of crop damage because of pests, bacteria or fungal infection.

1.4 Thesis Orientation

1. Our collected image data set was taken from online sources since we could not go out due to Covid-19.

2. We have used many pre-trained layers in the trained model for deep learning implementation.

3. We implemented three methods which are VGG19, MobileNet and Inception-v3 to detect crops' diseases in images. After implementing VGG19, MobileNetV2 and Inception-v3 we get the model accuracy of 94.21% in VGG19 ,97.93% in MobileNetV2 and 98.52% in Inception-v3.

Chapter 2

Literature Review

There can be many reasons in the world for plant diseases. Such as fungus, bacterium, mycoplasma, virus, viroid, nematode, or parasitic flowering plants. To avoid such diseases, a technological approach can be used to detect plant diseases. The process basically indicates diseases and notify the farmers/ cultivators. Various methods are there to show plant diseases. Several researchers have developed strategies for image processing and pattern identification in smart agriculture for detection of disease in a field, sorting of crops, fruits and vegetables etc. An imperative research topic is automated identification of plant diseases as it could be useful for scanning large fields of crops and identifying disease signals as soon as they appear on plant leaves. Real-time PCR and ELISA are standard approaches for plant disease detection; however they are highly invasive. This mini-review will not cover them. Thermography, RGB imaging, fluorescence imaging, and hyperspectral imaging are all examples of imaging techniques. Hence there is no standard for detecting plant damage. Although data augmentation is suitable for training generic plant detectors, it's performance can be limited by poor image quality. Despite the fact that simple ML and DL algorithms have been demonstrated to be effective and widely utilized in crop disease prediction, most previous publications had difficulties boosting classification accuracy rates to some amount. Furthermore, there is some performance deterioration in the neural network model due to poor parameter and layer selection. In the suggested CNN model for diagnosing citrus illnesses in both fruit and leaf pictures, we use a distinct number of layers and parameter settings. Furthermore, we examined various CNN model versions and compared the outcomes to the baseline investigations. We present a CNN model with numerous layers for accurately diagnosing citrus illnesses from fruit and leaf pictures [7]. Several studies have found that photograph-based absolute evaluation procedures generate more exact than visual processing tests. Ehler investigated the illness indications and Zymoseptoria tritici's early signs on afflicted wheat leaves using an automatic picture analysis approach. This method allowed for the analysis of pycnidia duration and solidity, as well as other developments and their linkage, which resulted in higher accuracy and precision than human visual estimations of infection [4]. Barbedo created a segmentation of pictures' approach to grade intensity of the infection in a backdrop color: white/black, eliminating the prospect of human mistake and shortening the time required to grade sickness difficulty. Atouma et al. developed a one-of-a-kind computer imaginative and prescient gadget, Cercospora Leaf Spot (CLS) Rater, to examine plant images in the field according 5 to the United States

Department of Agriculture (USDA) scale. The CLS Rater outperformed human experts in terms of consistency when judging the most recent deviance. Several of these photo- built evaluation tools for leaves' fungal diseases work in the same basic way [4].To begin, pre-processing procedures are used to remove the old and lesion tissue of sick flowers. Following that, discriminating abilities are removed for further study. Finally, unsupervised cluster algorithms or supervised kind algorithms utilized to group abilities based on the specific action. Many interactive devices have advanced in tandem with improvements in computer technology. Barbedo has designed and modelled multiple sets of algorithms with image-based segments to improve the crop or foliage disease and its functional properties. Pigment occurrences associated with opened or closed cases result in fewer fault and a shortest response time on the leaves or crop. Yousef Atouma et al. offered a novel situation of the layout picture associated with process computers employing real objects of datasets on improving agriculture efficiency and wider user preferences modeling. These image-modeling and its incidence in databases have higher grading and design criteria range [4]. Everingham et al. is the most widely used and widely utilized utility for estimating sickness intensity. The Leaf Doctor software, developed by Francisco et al as an engaging smartphone application, could be utilized on color images to differentiate scared sections from healthy tissues and determine the proportion of problem severity. The software outperformed the Assess in terms of accuracy. Fenfang Lin, Dongyan Zhang, Yanbo Huang, Xiu Wang, and Xinfu Chen are the most widely used and well-known software to evaluate illness intensity. The Leaf Doctor program, which began as just an interactive phone software, may be used on color images to distinguish diseased areas from healthy cells and determine the percentage of the complexity of the malady. The software outperformed the Assess in terms of accuracy. Gittaly Dhingra, Vinay Kumar, and Hem Dutt Josh have improved a typical guide's remarkable scrutiny, but this method is unreliable and imprecise, thus it cannot be stated systematically. Furthermore, it entails a massive volume of information in the field of plant disease pathologies (phytopathol- ogy), as well as improper processing duration's. As a result, photo processing has been used to identify plant diseases [4]. The rapid advancement of AI in recent years has improved the quality of life, and AI is currently a well-known technology. AlphaGo, for instance, defeated the world Go champion, Siri and Alexa, Apple and Amazon's voice assistants, are all representations of artificial intelligence technology presented by deep learning in many domains. Image recognition has advanced significantly in recent years as a key research topic in computer vision and artificial intelligence. Image recognition's objective in agricultural applications is to distinguish and categorize various types of pictures, as well as to analyze crop types, disorder types, difficulty, and so on. It will be possible for us devise appropriate remedy to address discrete issues in agriculture supply that is reliable and cheap [8]. The surrounding interference concerns are, however, stronger because of the rocky topography of the alpine area. As a result, image acquisition is more challenging than in a normal setting. Furthermore, the transmission via camera and network required for picture handling and identification will have an effect. As a result, doing intelligent recognition in mountainous terrain is more difficult. This study aims to develop an algorithm in the Neural Network platform in the complicated environment of mountainous locations, as well as conduct research on crop disease and insect pest diagnosis models [8]. Finally, when 6 it comes to motion tracking sensors, cattle management, such as accelerometers, gyroscopes, and pedometers, were the most frequent devices for gathering data on animals' daily activities. This type of sensor was previously only employed in animal welfare research. However, pictures, sound, and motion were the first to debut in both the animal welfare and cattle production subcategories. Physical and development features appeared in the livestock production sub-category with a slightly lower frequency. Weight, gender, age, metabolites, bio metric characteristics, back fat and muscle thickness, and heat stress were all taken into account. The last feature could have negative effects for animal health and product quality, but estimates of carcass lean yield can be established using back fat and muscle thickness measurements [9].

Chapter 3

Work Plan and Methodology

3.1 Work Plan

First and foremost, we must collect plant images from crop firms or cultivators in the field. In this regard, our research efforts have concentrated on predicting diseases and pest crops through the use Machine Learning techniques such as supervised deep learning algorithms. As a result, in this paper,we provide an overview of supervised learning algorithms commonly used for image detection. The model (figure-3.1) for detecting diseases was developed using a data set and image processing techniques in

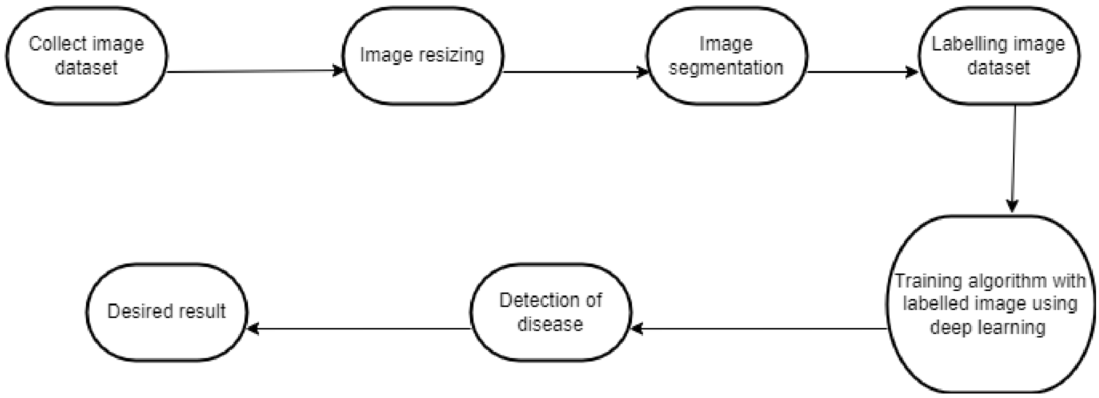

Figure 3.1: Flowchart for plant disease detection.

an algorithmic manner. Image labeling is the process of identifying and emphasizing different aspects of an image. When creating metadata in mage labeling, it is detrimental to automate the process. To teach the algorithm how to identify diseases, a very well deep learning method will be used. To begin labeling the images, upload all of the image data to the system. Following that, image labeling software will be installed to annotate such photos using specific techniques based on the customized criteria.

3.2 Methodology

The proposed plant disease detection model aims to identify diseases by incorporating VGG19, Inception-v3, and MobileNetV2. To do so, the model requires planning a process that takes information from an image data set as an input, efficiently processes input data, and delivers predictions of two types: diseases or the purpose of this study was to determine the accuracy of diseases. As a result, it was a straightforward multi class classification. We have used 38 classes in our model. The flowchart(figure-3.2) shows the steps of our proposed model's methods. The

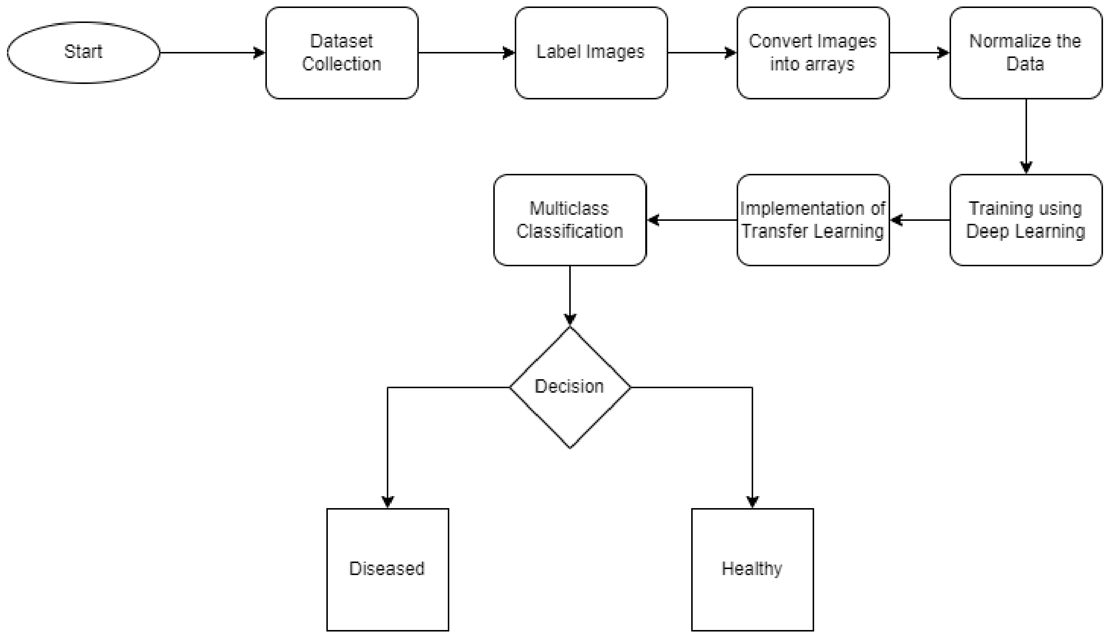

Figure 3.2: Flowchart of methodology for plant disease detection.

disease detection method is comprised of three major steps:

1. Pre-processing of images: The study started with data collection and labeling of disease and non-disease incidences. This period is concerned with organizing the input data, which is the entire image data set, so that the method (disease detection) can process it easily.

2. Training: This phase is concerned with converting the image data-set into arrays, normalizing the data set, and training using deep learning methods (VGG19, Inception-v3, MobileNetv2), as well as the incorporation of transfer learning and multi class classification to provide prediction.

3. Testing: This stage is concerned with taking random images to test to see if there are any diseases in that particular images, after normalizing the data, we have split the 100% data-set into three parts that are one pant (70%) is for training, one part (20%) for testing and other part (10%) for validation.

Chapter 4

Model Description

4.1 VGG19

VGG19 is a 19 layers deep convolutional neural network. The network is capable of categorizing photographs into 1000 different object categories, including keyboards, mouse, pencils, and other animals. The VGG19 model is a variation of the VGG model with 19 layers (16 convolution layers, 3 Fully connected layer, 5 MaxPool layers and 1 SoftMax layer). Other VGG variations include VGG11, VGG16, and more. There are 19.6 billion FLOPs in VGG19. In layman's terms, VGG is a deep CNN that is used to classify images. The VGG19 model has the following layers: Conv3x3 (64), Conv3x3 (64), MaxPool, Conv3x3 (128), Conv3x3 (128), MaxPool, Conv3x3 (256), Conv3x3 (256), Conv3x3 (256), Conv3x3 (256), MaxPool, Conv3x3 (512), Conv3x3 (512), Conv3x3 (512), Conv3x3 (512), MaxPool, Conv3x3 (512), Conv3x3 (512), Conv3x3 (512), Conv3x3 (512), MaxPool, Fully Connected (4096), Fully Connected (4096), Fully Connected (1000), SoftMax[10] The following picture(figure-4.1) shows the network architecture of VGG19-

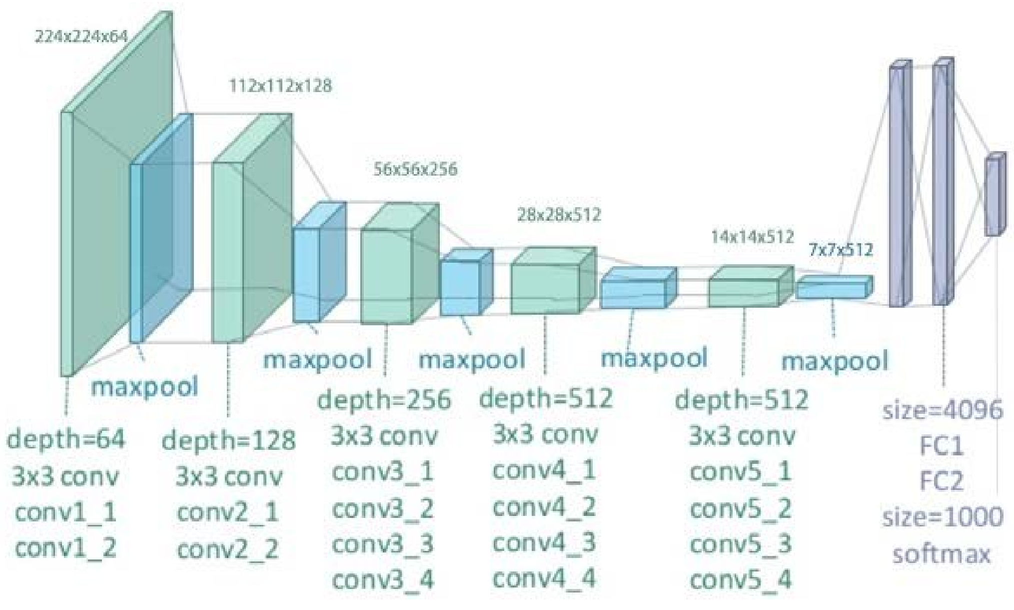

Figure 4.1: The network architecture of VGG19 model [11]

Architecture of VGG19:

1. As input, this network was given a specific size (224 * 224) RGB image, implying that the matrix was of configuration (224,224,3).
2. The only pre-processing was eliminated from the mean RGB value from each pixel, which was estimated for the entire training set.
3. To cover the complete visual notion, they employed kernels with a size of (3 * 3) and strides of 1 pixel.
4. Spatial padding was used to maintain the image's spatial resolution.
5. Optimal pooling was obtained with sride 2 over a 2 × 2-pixel window.
6. The Rectified linear unit (ReLu) was introduced to incorporate non-linearity into the model in order to enhance categorization and reduce processing time, whereas previous models used tanh or sigmoid functions.
7. The initial two fully linked layers were 4096 in size, followed by a layer of 1000 channels for 1000-way ILSVRC classification, and finally a softmax function [10].

4.2 MobileNetV2

4.2.1 MobileNet

TensorFlow's first mobile computer vision model, the MobileNet model, is intended for use in mobile applications, as the name implies. MobileNet employs depthwise separable convolutions.It drastically reduces the number of parameters when contrasted to a network with conventional convolutions of the same depth in the nets.Lightweight deep neural networks are constructed as a result. Two operations are used to create a depthwise separable convolution.

1. **Depthwise Seperable Convolution:** Using depth-wise convolutions, a single filter is applied to each input channel. The filters are applied to all of the input channels in a conventional convolution, but not in this case.

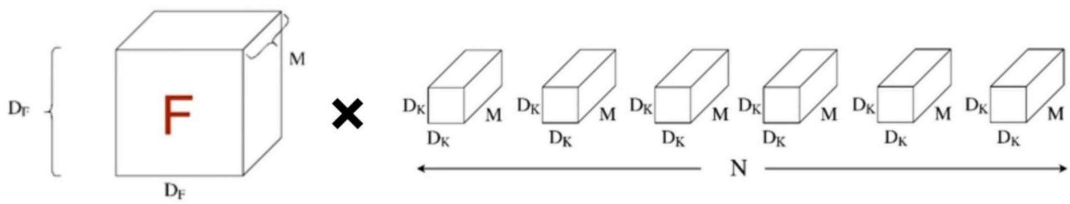

Figure 4.2: Standard Convolution [12]

From this image(figure-4.2), the computational cost calculation can be done:

$$D_{\mathrm{k}}.D_{\mathrm{k}}.M.N.D_{\mathrm{F}}.D_{\mathrm{F}} \tag{4.1}$$

Where DF denotes the input feature map's special dimensions, and DK denotes the convolution kernel's size.The amount of input and output channels, respectively, are M and N. The computing cost of a conventional convolution is multiplicatively based on the number of input/output channels, as well as the spatial dimensions of the input feature map and convolution kernel. In the case of depthwise convolution, an input feature map of dimension DF*DF and M number of kernels of channel size 1 are used, as shown in the graphic below. As seen in the diagram below(figure-4.3), the total computational cost

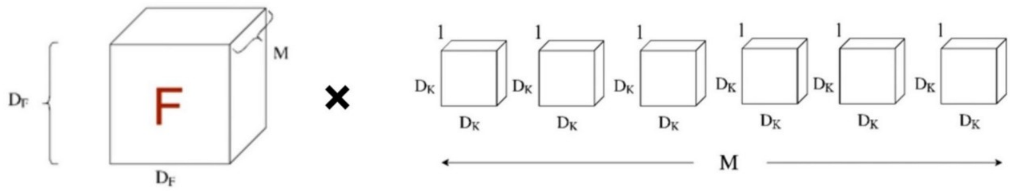

Figure 4.3: Depthwise convolution [12]

can be determined as follows:

$$D_{\mathrm{k}}.D_{\mathrm{k}}.M.D_{\mathrm{F}}.D_{\mathrm{F}} \tag{4.2}$$

2. **Pointwise Convolution:** The depthwise convolution only filters the input channel and does not integrate it with other channels to produce additional features. As a result, a pointwise convolution layer is created, which computes a linear combination of the depthwise convolution output using a 1x1 convolution.

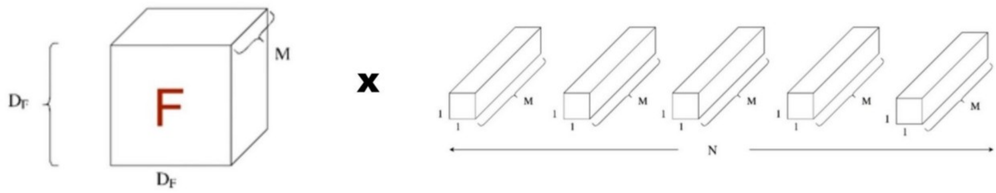

Figure 4.4: Pointwise convolution[12]

Let's calculate the computing cost of pointwise convolution again, as shown in the image(figure-4.4):

$$M.N.D_{\mathrm{F}}.D_{\mathrm{F}} \tag{4.3}$$

So, the overall computational cost of Depthwise separable convolutions is:

$$D_{\mathrm{k}}.D_{\mathrm{k}}.M.D_{\mathrm{F}}.D_{\mathrm{F}} + M.N.D_{\mathrm{F}}.D_{\mathrm{F}} \tag{4.4}$$

When we compare it to the computational cost of ordinary convolution, we get a computation cost reduction of depthwise separable convolutions cost along with standard convolution cost:

$$\frac{D_{\mathrm{k}}.D_{\mathrm{k}}.M.D_{\mathrm{F}}.D_{\mathrm{F}} + M.N.D_{\mathrm{F}}.D_{\mathrm{F}}}{D_{\mathrm{k}}.D_{\mathrm{k}}.M.N.D_{\mathrm{F}}.D_{\mathrm{F}}} \tag{4.5}$$

$$\equiv \frac{1}{N} + \frac{1}{DK^2} \tag{4.6}$$

To put this in context, this depthwise separable convolution was tested for effectiveness. Consider the following scenario. Let's use the values N=1024 and Dk=3 to solve the equation. We get 0.112, or 9 times more multiplications in regular convolution than in depthwise convolution.

4.2.2 MobileNetV2

MobileNetV2 is a convolutional neural network architecture designed to operate distinctively on mobile devices.It employs an inverted residual topology with residual connections between the bottleneck levels. This module takes as input a low-dimensional compressed representation and extends it to a high dimension before filtering it with a compact depthwise convolution. A linear convolution is then used to project the features back to a low-dimensional representation. Architecture of MobileNetV2-

1. MobileNetV2's architecture includes an initial fully convolution layer with 32 filters, followed by 19 residual bottleneck layers.

2. Because of its stability when employed with low-precision computations, we use ReLU6 as the non-linearity.

3. We always use kernel size 3x3 during training, as is prevalent in modern networks, and we use dropout and batch normalization.

Furthermore, since big intermediate tensors are not fully materialized, this convolutional module is particularly suited to mobile designs, allowing for a significant reduction in the memory footprint required during inference. This reduces the load for main memory access in several embedded hardware systems that provide small amounts of highly responsive software-controlled cache memory.[13] The following figure-4.5 shows the comparison between basic layers of MobileNet and MobileNetV2.

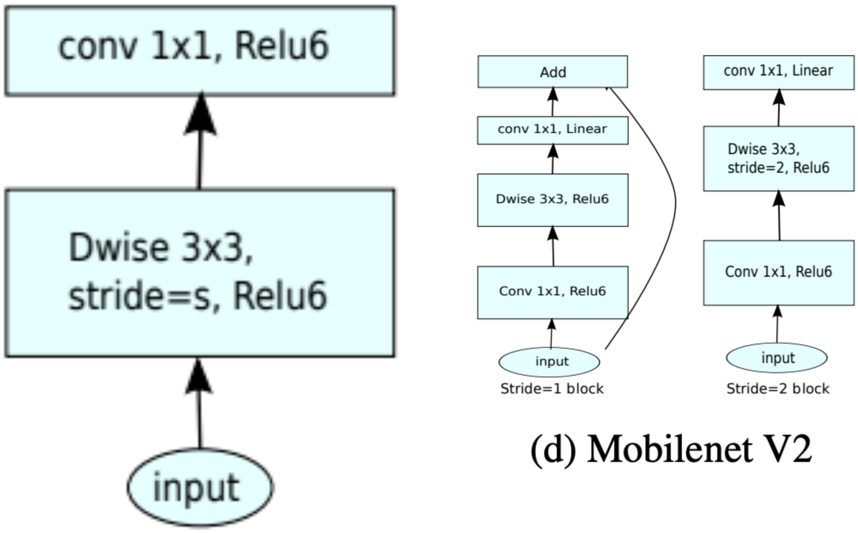

Figure 4.5: Basic layers of MobileNet and MobileNtV2 [13]

4.3 Inception-v3

On the ImageNet dataset, Inception v3 is an image processing model that has demonstrated greater than 78.1 percent accuracy. Several ideas explored throughout time by numerous researchers resulted in the model. These are based on the first paper on Inception v3 : "Rethinking the Inception Architecture for Computer Vision" by Szegedy, et. al.[14] The Inception v3 model, released in 2015, is merely an upgraded version of the Inception V1 model.The Inception V3 model employs numerous strategies to optimize the network. This image recognition model possesses 42 layers, a reduced error rate than its predecessors, convolutions, average pooling, max pooling, concatenations, dropouts, and fully linked layers are among the symmetric and asymmetric building components in the model. Batch normalization is performed on activation inputs and is widely used throughout the model. Loss is calculated using Softmax.Figure-4.6 shows the complete model of Inception v3. Let's have a look at the kind of optimizations that have been made to the Inception V3 model.

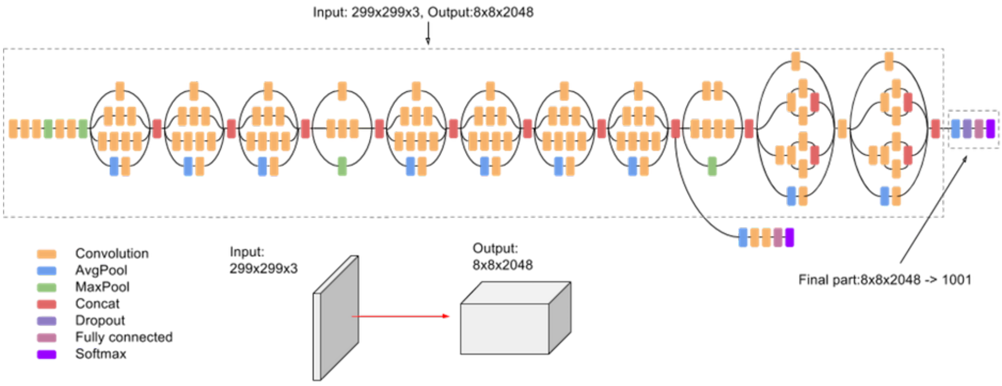

Figure 4.6: Complete Inception v3 model[15]

1. **Factorization into Smaller Convolutions:** The extensive dimension reduction was one of the main advantages of the Inception V1 model. The model's larger Convolutions were factorized into smaller Convolutions to make it even better. Inception v1 had 5x5 convolutional layer which was computationally expensive. To save time and money, 5x5 convolutional layer was replaced with 3x3 convolutional layers.

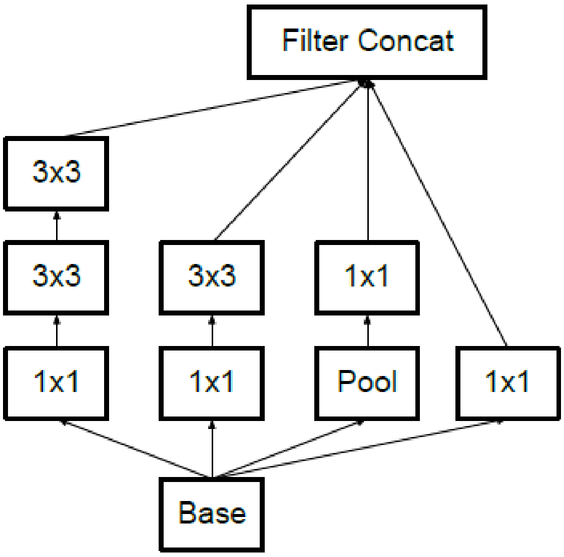

Figure 4.7: Convolutional layers of Inception v3 [14]

The figure-4.7 shows how 5x5 convolution is taken over by 3x3 convolution. Moreover, computational costs are decreased as a result of the reduced number of parameters. A relative gain of 28% was achieved by factoring bigger convolutions into smaller convolutions

2. **Spatial Factorization into Asymmetric Convolutions:** Although the larger convolutions are divided into smaller ones, What occurs if we factorize any further, say to a 2x2 convolution. Asymmetric convolutions, on either hand, were a better choice for optimizing the model's efficiency. The shape of asymmetric convolutions is nx1. So, instead of 3x3 convolutions, they used a

1x3 convolution followed by a 3x1 convolution. It's the equivalent of sliding a two-layer network that has the same receptive field as a 3x3 convolution(figure-4.8). When the number of input and output filters is equal, the two-layer technique costs 33% less for the same number of output filters.

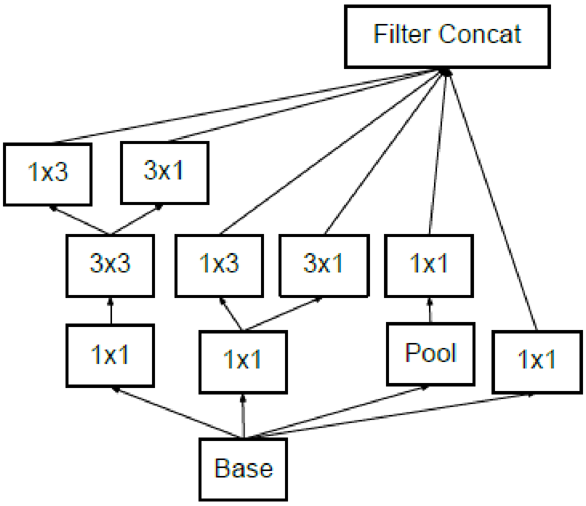

Figure 4.8: Asymmetric Convolutions' Structure [14]

3. **Utility of Auxiliary Classifiers:** The idea is to use an auxiliary classifier to assist the convergence of very deep neural networks.Figure-4.8 shows filter bank outputs on Inception modules have been expanded. In very deep networks, The auxiliary classifier is mostly used to deal with the vanishing gradient problem. The auxiliary classifiers made little difference in the early rounds of training. However, in terms of accuracy, the network with auxiliary classifiers outperformed the network without them. As a result, the auxiliary classifiers in the Inception V3 model architecture operate as a regularizer.

4. **Efficient Grid Size Reduction:** For reducing the grid size of feature maps, max pooling and average pooling were traditionally utilized. The activation dimension of the network filters is enhanced in the inception V3 model to lower the grid size well. If we have a nxn grid with m filters, after reduction we get a n/2 n/2 grid with 2m filters. The following figure-4.9 shows the grid size reduction of Inception module that shrinks the grid while increasing the number of filter banks. It is also inexpensive.

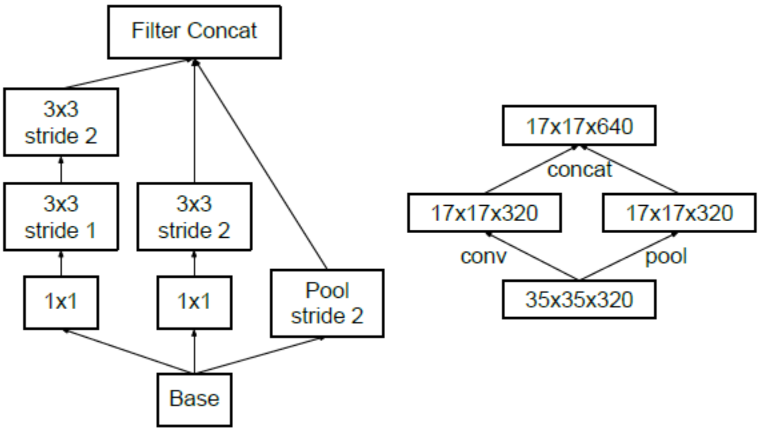

Figure 4.9: Grid size reduction of inception v3 [14]

4.4 Data Preliminary Analysis

1. **Collecting Data set:** We have used a dataset containing 87,123 images of healthy and diseased crops. This collected dataset will be used to identify if our model can identify an infected plant or not.

2. **Labeling Image:** The process of recognizing and labelling distinct aspects in an image is known as image labeling. It is also known as image annotation, is the process of defining the boundaries of an image and labeling the elements within it. Images can be labeled by humans, AI algorithms, or a combination of both. In image labeling, there are numerous annotation methods. The first is a straightforward bounding box. Because only two (diagonal) vertices must be defined, this method is easier for labelers and AI models to use than the others. We have labelled our dataset into 38 labels which are apple scab, black rot, cedar rust, apple healthy, blueberry healthy, powdery mildew in cherry, cherry healthy, grey leaf spot in corn, corn common rust, corn northern leaf blight, corn healthy, black rot(grape), grape black measles, orange haunlobing citrus, bacterial spot in peach, potato and tomatoes' early and late blight, potato healthy, raspberry and soybean healthy, powdery mildew in squash, strawberry leaf scorch, bacterial spot in tomato, spider mitesand leaf mold of tomato, tomato septora leaf spot, target spot in tomatoes, yellow leaf curl virus of tomato, tomato mosaic virus.

3. **Converting Images into Arrays:** Images are a more convenient way to represent the working model. Python uses image data in the Height, Width, Channel format for Machine Learning. Specifically, pictures are converted into Numpy Arrays in the length, Width, and Channel formats. When using the library to read in digital images, they are represented as Numpy arrays. The array's rectangular shape corresponds to the image's shape. We can see how a chair is structured by reading in an image and using image.shape, which returns a tuple (height, width, channels). If the image is colored, the image properties will be a numerous number of rows, columns, and channels. Image.shape only returns the number of rows and columns if the picture is grayscale. When working with OpenCV images, the y coordinate is specified

first, followed by the x coordinate. Colors are saved as BGR values, with blue in layer zero, green in layer one, and red in layer two.

4. **Normalizing the data:** For our convenience, we reshape and resize the entire dataset's images into a single size.

5. **Training using deep learning (VGG 19, MobileNet, Inception V3):** We imported the required Deep Learning libraries, downloaded pre-trained weights, and printed out our VGG 19, MobileNet, Inception V3 model. We started with VGG19 then MobileNet and later moved on to Inception-v3.

6. **Implementation of Transfer learning:** The main advantage of transfer learning is that it shortens training time. To implement transfer learning base input and output have been used. The trained model contains several pre-trained layers, and for base input we used our model's input layer from the pre-trained model, and for base output we used the dense layer with 38 neuron from the pre-trained model. We tweaked the model to improve its accuracy. We don't want to train them again because it would take a long time.

7. **Multiclass classification:** A classification problem that involves more than two classes, such as classifying a batch of fruit photos that could be oranges, apples, or pears. Each sample is allocated to one and only one label in multi-class classification, a fruit can be either an apple or a pear, but not both at the same time. However, since we have used a big data set, we have 38 classes here in our model. We have implemented batch size, loading 132 images at a time. We have used epochs as they produce valid results. We ran 50 distinct epochs and they showed valid accuracy.

8. **Train and Test:** For training and testing, we used three models and they were VGG 19, MobileNet, Inception v3. We split the data set into 3 parts, 70.% of data is used for training, 19.576% for testing, and 9.846% for validation. The following fgure-4.10 shows the tested pictures of our used data set.

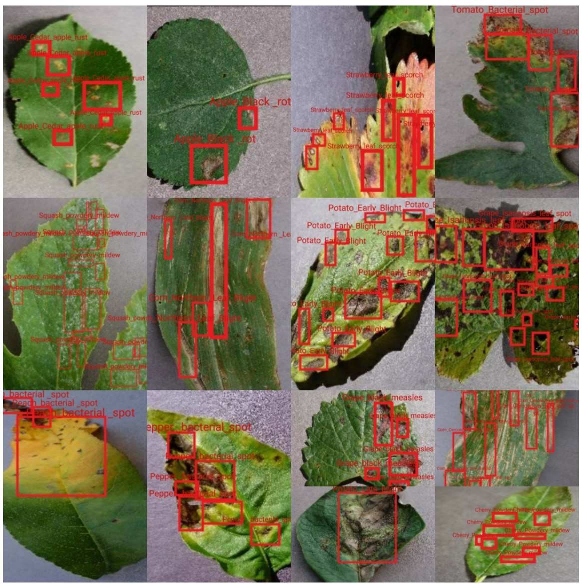

Figure 4.10: Trained and Tested Data set

4.5 Result Analysis

The following figure-4.11 is the classification report of VGG19 model on 38 classes, where we got the accuracy 0.94, macro average 0.92, weighted average 0.94.

	precision	recall	f1-score	support
0	1.00	0.87	0.93	252
1	0.98	0.99	0.98	248
2	0.99	0.97	0.98	220
3	0.92	0.95	0.94	251
4	0.96	0.98	0.97	227
5	1.00	0.95	0.97	210
6	1.00	0.99	0.99	282
7	0.80	0.98	0.88	205
8	0.98	1.00	0.99	238
9	1.00	0.79	0.88	236
10	1.00	1.00	1.00	232
11	0.99	0.76	0.86	236
12	0.81	1.00	0.90	240
13	0.99	1.00	0.99	215
14	0.99	1.00	0.99	211
15	1.00	0.98	0.99	251
16	0.98	0.98	0.98	229
17	1.00	0.95	0.98	216
18	0.95	0.97	0.96	239
19	0.00	0.00	0.00	0
20	0.88	1.00	0.93	242
21	0.99	0.82	0.90	242
22	0.95	1.00	0.97	228
23	0.98	1.00	0.99	222
24	0.96	0.99	0.97	252
25	0.97	1.00	0.98	217
26	0.99	0.98	0.98	221
27	0.99	0.99	0.99	228
28	0.96	0.88	0.92	212
29	0.74	0.97	0.84	240
30	0.97	0.75	0.84	231
31	0.97	0.89	0.93	235
32	0.73	0.97	0.83	218
33	0.94	0.81	0.87	217
34	0.91	0.79	0.84	228
35	1.00	0.96	0.98	245
36	1.00	0.98	0.99	223
37	0.92	1.00	0.95	240
accuracy			0.94	8579
macro avg	0.92	0.92	0.92	8579
weighted avg	0.95	0.94	0.94	8579

Figure 4.11: F1 score of VGG19

Following figure-4.12 is the classification report of MobileNetV2 on 38 classes, where we got the accuracy 0.98, macro average 0.95 and weighted average 0.98.

	precision	recall	f1-score	support
0	0.97	0.97	0.97	252
1	1.00	1.00	1.00	248
2	0.97	0.99	0.98	220
3	0.98	0.97	0.97	251
4	0.98	1.00	0.99	227
5	1.00	0.99	0.99	210
6	1.00	0.99	0.99	282
7	0.89	0.98	0.93	205
8	1.00	1.00	1.00	238
9	0.99	0.91	0.95	236
10	1.00	1.00	1.00	232
11	0.99	0.98	0.99	236
12	0.98	1.00	0.99	240
13	1.00	0.99	0.99	215
14	1.00	1.00	1.00	211
15	1.00	1.00	1.00	251
16	0.99	0.98	0.98	229
17	0.98	1.00	0.99	216
18	0.99	0.99	0.99	239
19	0.00	0.00	0.00	0
20	0.99	1.00	0.99	242
21	0.96	0.98	0.97	242
22	1.00	1.00	1.00	228
23	0.99	1.00	0.99	222
24	0.99	0.99	0.99	252
25	1.00	1.00	1.00	217
26	0.99	1.00	1.00	221
27	1.00	1.00	1.00	228
28	0.98	0.96	0.97	212
29	0.93	0.95	0.94	240
30	0.97	0.94	0.95	231
31	0.97	0.98	0.98	235
32	0.94	0.93	0.94	218
33	0.94	0.96	0.95	217
34	0.94	0.87	0.90	228
35	1.00	0.98	0.99	245
36	0.97	1.00	0.99	223
37	0.98	0.99	0.99	240
accuracy			0.98	8579
macro avg	0.95	0.95	0.95	8579
weighted avg	0.98	0.98	0.98	8579

Figure 4.12: F1 score of MobileNetV2

Following figure-4.13 is the classification report of inception-v3, where we got accuracy as 0.99, macro average 0.96 and weighted average 0.99.

	precision	recall	f1-score	support
0	0.99	0.98	0.99	252
1	0.99	1.00	1.00	248
2	0.99	1.00	0.99	220
3	0.99	0.98	0.99	251
4	0.99	0.99	0.99	227
5	1.00	0.99	1.00	210
6	1.00	0.99	1.00	282
7	0.93	0.99	0.96	205
8	0.99	1.00	1.00	238
9	1.00	0.94	0.97	236
10	1.00	1.00	1.00	232
11	0.99	0.98	0.99	236
12	0.98	1.00	0.99	240
13	1.00	1.00	1.00	215
14	1.00	1.00	1.00	211
15	0.98	1.00	0.99	251
16	1.00	0.98	0.99	229
17	1.00	1.00	1.00	216
18	1.00	0.98	0.99	239
19	0.00	0.00	0.00	0
20	1.00	1.00	1.00	242
21	0.97	0.98	0.98	242
22	0.98	1.00	0.99	228
23	1.00	1.00	1.00	222
24	1.00	0.99	0.99	252
25	1.00	1.00	1.00	217
26	0.99	1.00	0.99	221
27	1.00	1.00	1.00	228
28	0.98	0.98	0.98	212
29	0.94	0.96	0.95	240
30	0.97	0.91	0.94	231
31	0.99	0.97	0.98	235
32	0.97	0.97	0.97	218
33	0.95	0.97	0.96	217
34	0.96	0.94	0.95	228
35	1.00	0.99	0.99	245
36	1.00	1.00	1.00	223
37	0.98	1.00	0.99	240
accuracy			0.99	8579
macro avg	0.96	0.96	0.96	8579
weighted avg	0.99	0.99	0.99	8579

Figure 4.13: F1 score of Inception v3

The given picture figure-4.14 is the confusion matrix of VGG19. To map the confusion matrix, we mapped the 38 classes from our data set.

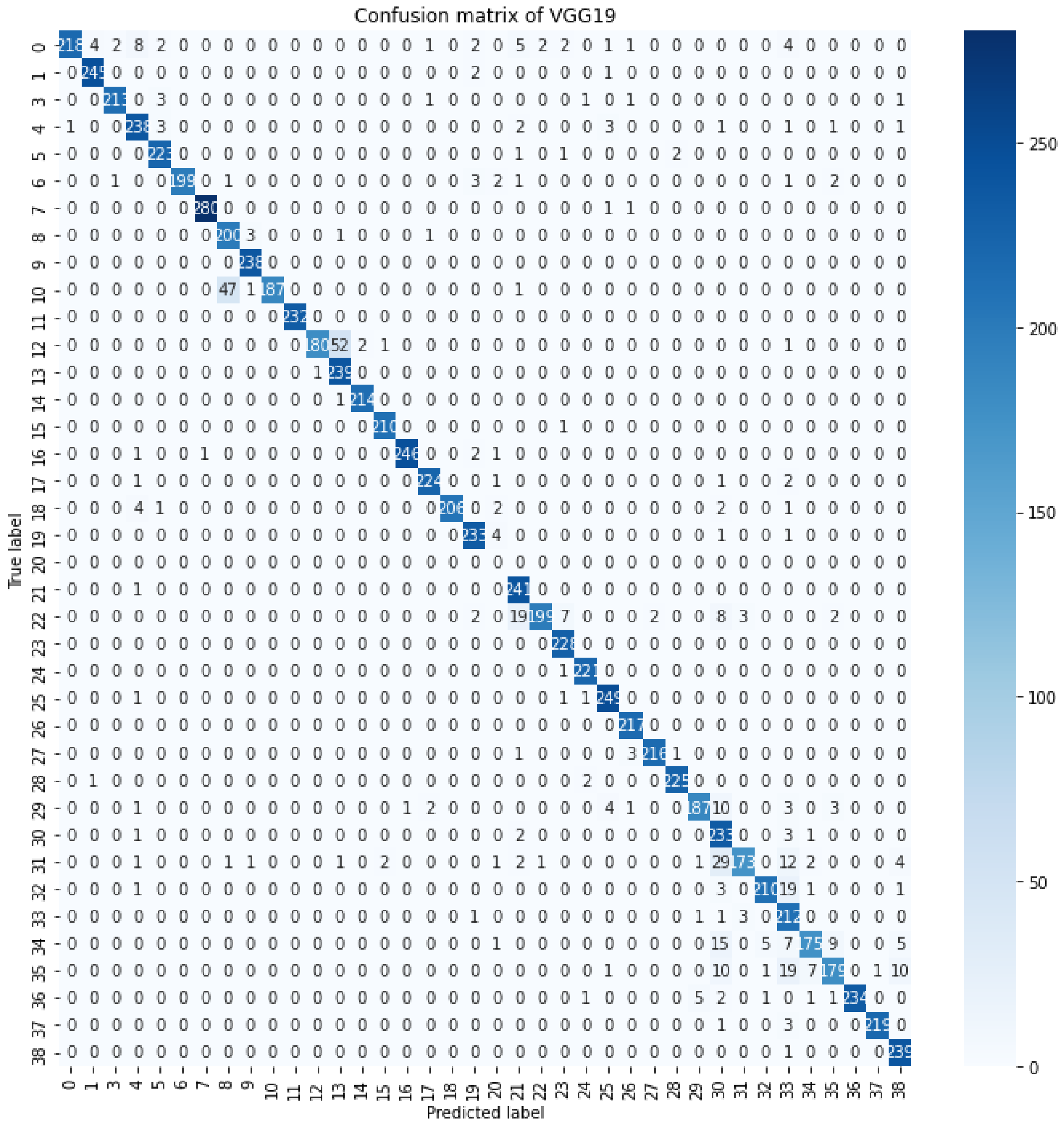

Figure 4.14: Confusion matrix of VGG19

The following figure-4.15 is the confusion matrix of MobileNetV2, where 0-38 shows the classes of our data set.

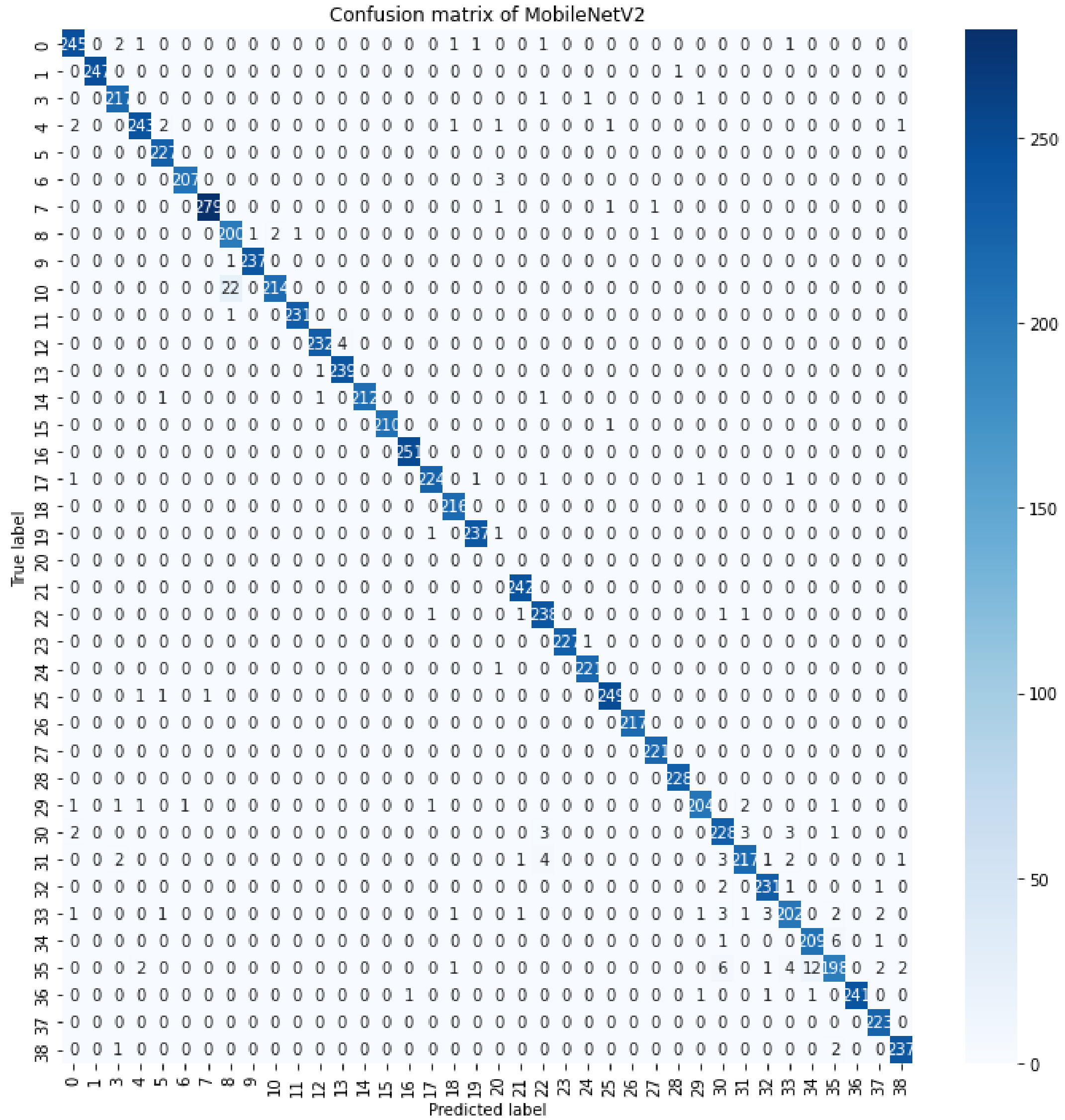

Figure 4.15: Confusion matrix of MobileNetV2

The following figure-4.16 is the confusion matrix of model Inception-v3 where we mapped 38 classes to give it a proper look.

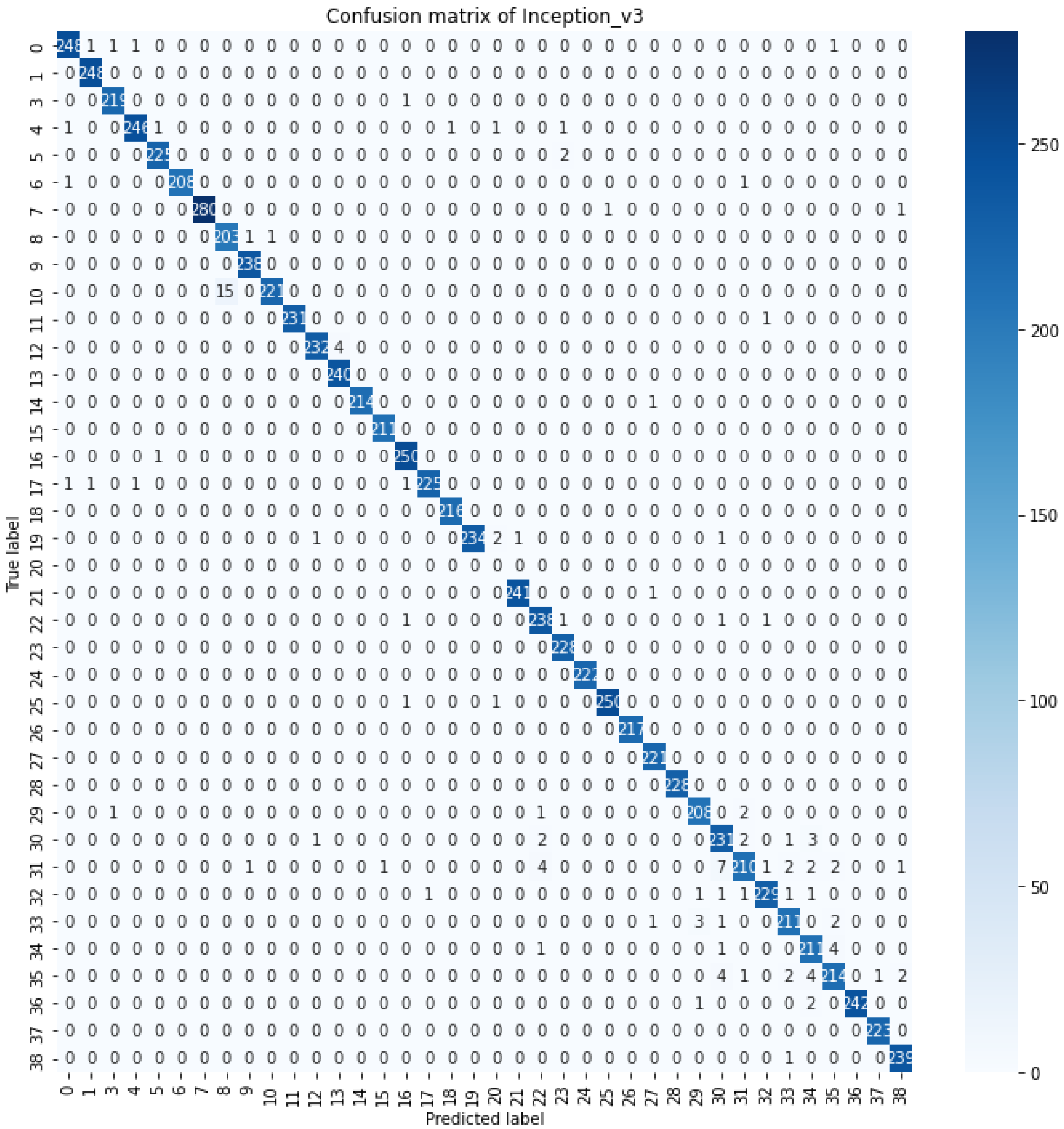

Figure 4.16: Confusion matrix of Inception-v3

After running VGG19, MiobileNetV2 and Inception-v3 on our data set with 38 classes on different crops, we got results for model accuracy and loss. We set our epoch as 40, it ran till 13 and gave us accuracy of 94.21% and loss 0.178365424. The left sided picture shows the accuracy graph of VGG19 and right sided graph shows the model loss. The following figure-4.17 is the accuracy and loss graph for VGG19 model. For MobileNetV2 we set our epoch as 80, it ran till 49 and gave us

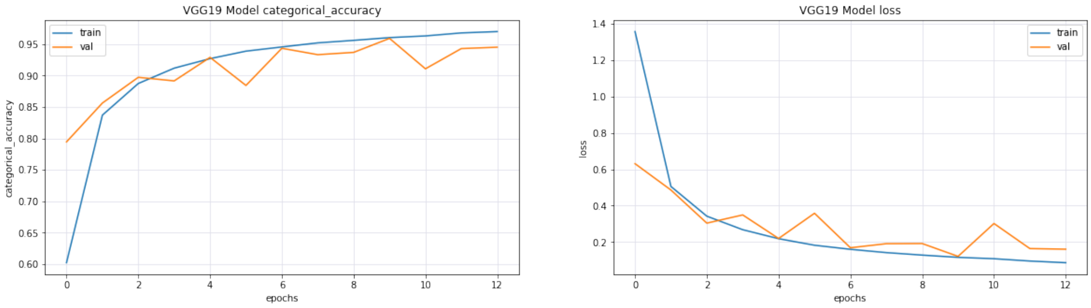

Figure 4.17: Accuracy and loss graph of VGG19 model on the given data set.

the accuracy of 97.93% and loss 0.96%.
The following figure(Figure-4.18)left sided part is the accuracy graph and right sided part is the loss graph for MobileNetV2. For Inception-v3 we set our epoch as 100,

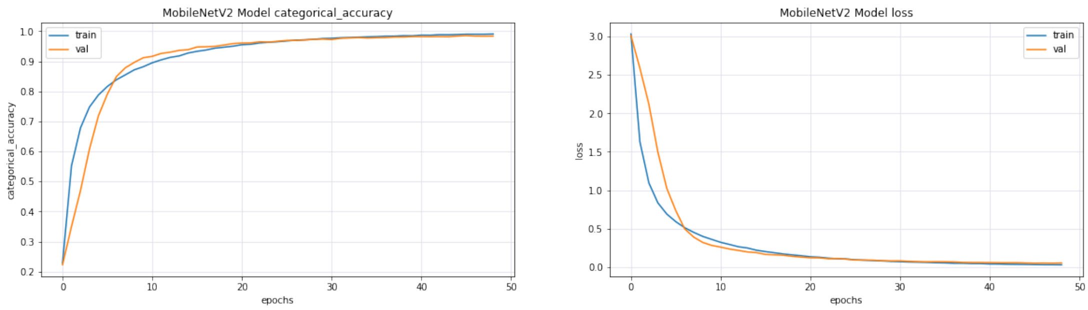

Figure 4.18: Accuracy and loss graph of MobileNetV2 model on the given data set.

it ran till 50 and gave us the accuracy of 98.52% and loss 4.77%. The following figure(Figure-4.19) left sided part is the accuracy graph and right sided part is the loss graph for Inception-v3.

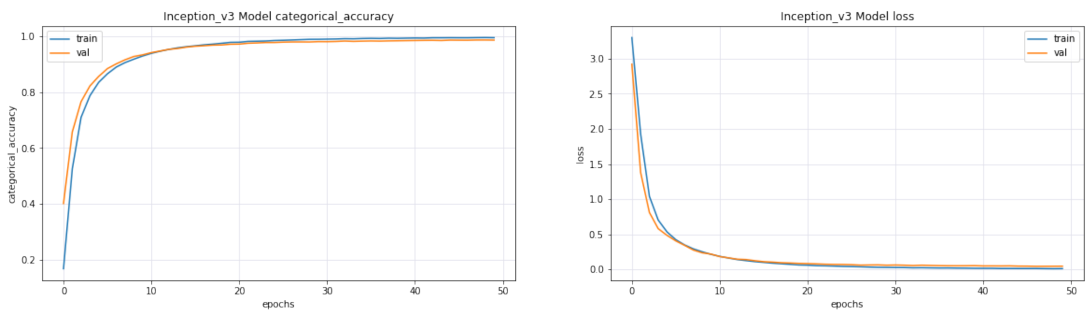

Figure 4.19: Accuracy and loss graph of Inception model on the given data set.

We ran deep transfer learning's 3 model VGG19, MobileNetV2 and Inception-v3 on our dataset consisted with 38 classes. After running VGG19 we received the accuracy of 94.215% and loss 0.178%, MobileNetV2 gave 97.93% accuracy and loss 6.96% and at last Inception-v3 gave the accuracy of 98.52% and loss 4.77%. The table-1 shows the accuracy and loss of the three deep learning model.

Table 4.1: Output Results of VGG19, MobileNet and Inception-v3

Model	**Peak Accuracy**	**Loss**
VGG19	94.21%	0.178%
MobileNetV2	97.93%	6.96%
Inception-v3	98.52%	4.77%

Chapter 5

Conclusion and Future Scope

5.1 Conclusion

Crops or plant products that are grown to provide food, fuel, clothing, and other benefits are an essential part of our horticulture. Plant protection is the monitoring of pests, viruses and diseases that harm or impede the growth of natural products, vegetables, and other crops. So, detecting the unwellness of the crops and plants can flourish the growth of agriculture and ease the stress of the cultivators. As mentioned previously, various methods have been implemented but our paper exceptionally focuses on ML's deep learning methods to detect a diseased and healthy plant. And for this study, we used three models: VGG19, MobileNetV2 and Inception-V3. To improve accuracy, the algorithm in this system was trained with a vast number of pictures of tainted and healthy plants' leaves. We are hopeful, with this study we will be able to help the horticulture sector and lessen the worry of our farmers.

5.2 Future Scope

For this study we used available online resources to collect our data set for Covid-19 restrictions. When everything gets normal, we will make our very own data set by taking pictures of the crops from crop fields or farms. Sometimes distinct diseases can have similar symptoms, again rats, mice, spider, birds can cause harm to the crops, we are working to implement this model in drones and hoping to build a more modified and powerful model to detect even more complex infections accurately.

Bibliography

[1] S. V. Militante, B. D. Gerardo, and N. V. Dionisio, "Plant leaf detection and disease recognition using deep learning," *2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE)*, pp. 579–582, 2019.

[2] F. Marzougui and M. Elleuch, "A deep cnn approach for plant disease detection," Nov. 2020, pp. 1–6. DOI: 10.1109/ACIT50332.2020.9300072.

[3] M. Rahman, "Vegetable ipm in bangladesh, in e. b. radcliffe and w. d. hutchison [eds.], radcliffe's ipm world textbook, url," in Aug. 2020, Not metioned, ISBN: http://www.ipmworld.umn.edu/chapters/rahman.htm. DOI: 10.1007/s11356-.

[4] A. Abdu, M. Mokji, and U. Sheikh, "Machine learning for plant disease detection: An investigative comparison between support vector machine and deep learning," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 9, p. 670, Dec. 2020. DOI: 10.11591/ijai.v9.i4.pp670-683.

[5] Y. Lu, S. Yi, N. Zeng, Y. Liu, and Y. Zhang, "Identification of rice diseases using deep convolutional neural networks," *Neurocomputing*, vol. 267, Jul. 2017. DOI: 10.1016/j.neucom.2017.06.023.

[6] C. DeChant, T. Wiesner-Hanks, S. Chen, *et al.*, "Automated identification of northern leaf blight-infected maize plants from field imagery using deep learning," *Phytopathology*, vol. 107, pp. 1426–1432, Jun. 2017. DOI: 10.1094/PHYTO-11-16-0417-R.

[7] 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. DOI: 10.1109/ACCESS.2021.3096895.

[8] Y. Ai, C. Sun, J. Tie, and X. Cai, "Research on recognition model of crop diseases and insect pests based on deep learning in harsh environments," *IEEE Access*, vol. 8, pp. 171 686–171 693, Jan. 2020. DOI: 10.1109/ACCESS.2020.3025325.

[9] L. Benos, A. Tagarakis, G. Dolias, B. Remigio, D. Kateris, and D. Bochtis, "Machine learning in agriculture: A comprehensive updated review," *Sensors*, vol. 21, May 2021. DOI: 10.3390/s21113758.

[10] *Understanding the vgg19 architecture.* [Online]. Available: https://iq.opengenus.org/vgg19-architecture/#:~:text=VGG19%20is%20a%20variant%20of,VGG19%20has%2019.6%20billion%20FLOPs.

[11] Y. Zheng, C. Yang, and A. Merkulov, "Breast cancer screening using convolutional neural network and follow-up digital mammography," in *Computational Imaging III*, International Society for Optics and Photonics, vol. 10669, 2018, p. 1 066 905.

[12] A. G. Howard, M. Zhu, B. Chen, *et al.*, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[13] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[14] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[15] H. Mela Karim Barznji, "Transfer learning as new field in machine learning," Nov. 2020.