Leaf Classification by Feature Extraction Using CNN

by

Md. Mazharul Islam Bhuiyan
15201042
Jakia Nowshin
15201021
Atkiya Jaheen
15301118

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
December 2019

Declaration

It is hereby declared that

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

Student's Full Name & Signature:

Md. Mazharul Islam Bhuiyan
15201042

Jakia Nowshin
15201021

Atkiya Jaheen
15301118

Approval

The thesis/project titled "Leaf Classification by Feature Extraction Using CNN" submitted by

1. Md. Mazharul Islam Bhuiyan (15201042)
2. Jakia Nowshin (15201021)
3. Atkiya Jaheen (15301118)

Of Fall, 2019 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on December 26, 2019.

Examining Committee:

Supervisor:
(Member)

Amitabha Chakrabarty,PhD
Associate Professor
Computer Science and Engineering
BRAC Univeristy

Program Coordinator:
(Member)

Md. Golam Rabiul Alam,PhD
Associate Professor
Computer Science and Engineering
BRAC Univeristy

Head of Department:
(Chair)

Mahbubul Alam Majumdar,PhD
Chairperson
Computer Science and Engineering
BRAC Univeristy

Abstract

Plants are an integral part of our nature. The identification and classification of plant leaves has always been a matter of interest for the botanists as well as the laymen. Classification of plant leaves will enable us to know the heritage and details of plants at a glance avoiding the duplication of popular names. This recognition system will be beneficial to different sectors of our society including botanic research, medical field, the study of plant taxonomy etc. As leaves carry a lot of information about plant species, extraction of feature is a better way to classify the leaves. In this paper, we have proposed Convolutional Neural Network (CNN) and analyzed plant leaves with different models. We have collected the dataset from Kaggle. By pre-processing the images and extracting the features we have trained our pre-trained model. In our research, we have chosen three models of CNN which are InceptionV3, VGG16 and MobileNet. MobileNet achieved the highest accuracy of 69.47% with a mean absolute error of 30.26, while VGG16 achieved the lowest accuracy of 57.05% with a mean absolute error of 42.95 and 66.13% accuracy for Inception V3.

Keywords: Convolutional neural network; Pre-processing; MobileNet; VGG16; Inception V3

Acknowledgement

First and foremost, we would like to thank God Almighty for giving us the ability and opportunity to undertake the thesis and to complete it. This achievement would not have been impossible without his blessings. After that, we want to thank and express our gratitude toward our supervisor Dr. Amitabha Chakrabarty for his constant guidance and support as well as providing us necessary information regarding our work. His valuable feedback helped us to improve ourselves and complete the thesis with our hard work. We also want to thank BRAC University IT Department for allocating us a laboratory where we have done all of our works. We are also very grateful to our family and friends who supported us directly and indirectly in our entire thesis period. Sometimes, we used to feel less confident because of facing difficulties in our work. Then, our family members and friends gave the mental support and encouraged us a lot to stick with the work and complete it in time. We would like to acknowledge the assistance that we found a huge number of online resources specially the work of our fellow researches. Finally, we are grateful to BRAC University to give us the opportunity to do such research that helped us to enhance our knowledge.

Table of Contents

List of Figures

List of Tables

Nomenclature

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

CNN	Convolutional Neural Network
CSV	Comma Separated Values
$ILSVRC$	ImageNet Large Scale Visual Recognition Competition
MAE	Mean Absolute Error
NN	Neural Network
$ReLu$	Rectified Linear Units
SF	Sigmoid Function
$Tanh$	Hyperbolic Tangent Function
VGG	Visual Geometry Group

Chapter 1

Introduction

Plants exists everywhere around us and places where human has not stepped yet. Most of the trees around us can be easily identified by their flowers and fruits but identifying trees by their leaves is a complex process. A huge number of leaves from different species are very similar in shape, pattern and arrangement. So, recognizing the diversity of leaf types and knowing the terminology for different features are important to classify the leaves. Plants are the main source that produces all the oxygen to the living organs. As the natural environment is reduced, there is constant extinction of different plant species. There are about 400,000 species of plants all over the world of which Botanists have identified and named 270,000 species of plants[1]. The further research on plant is very complicated because it is not easy for the botanists or non researcher to find out than a little fraction from the overall number of named species. Leaf recognition is generally the specialization of plant taxonomists. There are some methods like Cytotaxonomy, Chemotaxonomy, Serotaxonomy etc which are carried out only by the botanists because of its complexity and time consuming attributes. Leaves are now classified widely by its morphological image [2]. It is a long discussed topic on how to extract and measure leaf features that will help to make the application of pattern recognition easier.

The advancement of computer technology would be the alternative to classify plants for the non-specialists. Scientists are continuously trying to get the leaf identity in different ways for many years. There are many mathematical models have been introduced to extract the morphological characteristics of leaves. As today is the era of machine learning, scientists have also introduced some processes to sort and identify leaves by using deep learning which is a sub-field of machine learning. Deep learning methods are based on the algorithms inspired by the functionality of the brain called artificial neural network. For classifying leaves, CNN is very efficient which is a class of deep neural network and is widely used for image recognition and classification, object detection, recognition faces etc [3]. CNN image classifications take an input image, process it and learn the features from it. It focuses on pattern recognition among the image of its own. As a result, if we give any image as an input, it can recognize it accurately. There are many CNN models which have been invented in previous years. In our research, we have used some pre-trained models in various types of leaf image dataset to predict the class of the leaves and also compared the accuracy rate among the models that we used.

1.1 Motivation

In the natural ecosystem, there are hundred kinds of trees those leaves are difficult to distinguish. There are a lot of plants which leaves have same shape, texture and venation. Leaves those carry same types of features and characteristics, have been sorted in same class. This classification will be beneficial to botanists in their research, plant taxonomy and pharmaceutical field to use more plant leaves effectively in their work. This leaf recognition and extraction method can be established by using of Convolutional Neural Network (CNN). CNN helps to find patterns in an image by its various layers. After convoluting over an image, some patterns are found. These patterns are passed down through the neural net and then more complex feature can be recognized. This property of CNN is really good at identifying objects in image [4]. In this thesis, we are using some existing CNN models which will help us to make a comparative analysis among all the information of the leaves that have been obtained from the leaf images stored in our dataset.

1.2 Aims and Objectives

The main objectives of the thesis are:

- Creating a dataset from the images and using them to have training and pre-training models.
- Predicting the class of leaves by applying different pre-trained models and comparing the accuracy rate of the models.
- Finding out the reasons of having performance variation between the models when applying to the same dataset.

1.3 Thesis Overview

1. **Chapter 1** Refers to the introduction part where we have discussed about our motivation to do this thesis as well as the aims and objectives that have been followed during our work.
2. **Chapter 2** shows our background study related to our work in the literature review part.
3. **Chapter 3** consists of proposed model, pre-processing and compilation of our pre-trained models.
4. **Chapter 4** contains the pre-trained models and implementation of our research.
5. **Chapter 5** consists of conclusion and our future work.

Chapter 2

Literature Review

2.1 Convolutional Neural Network

Over the last decades deep neural network has been known as the most powerful tool and has become highly popular in the literature as it can handle huge amounts of data. The Deep Neural Network refers to Artificial Neural Networks (ANN) having multi layers and hidden layers which used for pattern recognition. Convolutional Neural Network (CNN) is one of the most famous Deep Neural Networks. CNN is an effective processing of images, a deep learning (AI), which uses both reproductive and expressive tasks [5]. A convolution is a mathematical process that is defined by a sign that moves through the device.

The Convolutional Neural Network deals with this concept of convolution. It is nearly like popular neural networks. Some neurons are present which have learning weights. In Neural Network, the input layer is in left side and the neurons are considered as the input neurons because they are inside the layer. Output layer is in the right side. Moreover, in this situation, a neuron inside the output layer that is called single output neuron. Since the neurons are not inputs or outputs in this layer, the middle layer is considered as hidden layer. There are certain neurons in each hidden layer and every neuron in each single layer is completely associated to all neurons of the previous layer. Neuron function is totally independent and has no connections[6]. Neural Networks actually scale the whole image which is considered to be large. If the image size is 224×224 then neurons will have $224 \times 224 \times 3$=150528 weights (3 is the color of RGB). In that case, CNN operates better and efficiently. CNN removes the image's function without modifying or losing its characteristics and converts it into a lower dimension. CNN layers are structured in 3 dimensions and they are width, height and depth as opposed to neural network.

2.2 How CNN works

CNN contains multiple layers [7]. They are:

1. Layer for Input
2. Layer for Convolution
3. Layer for ReLu
4. Layer for Pooling
5. Fully Connected Layer
6. Layer for output

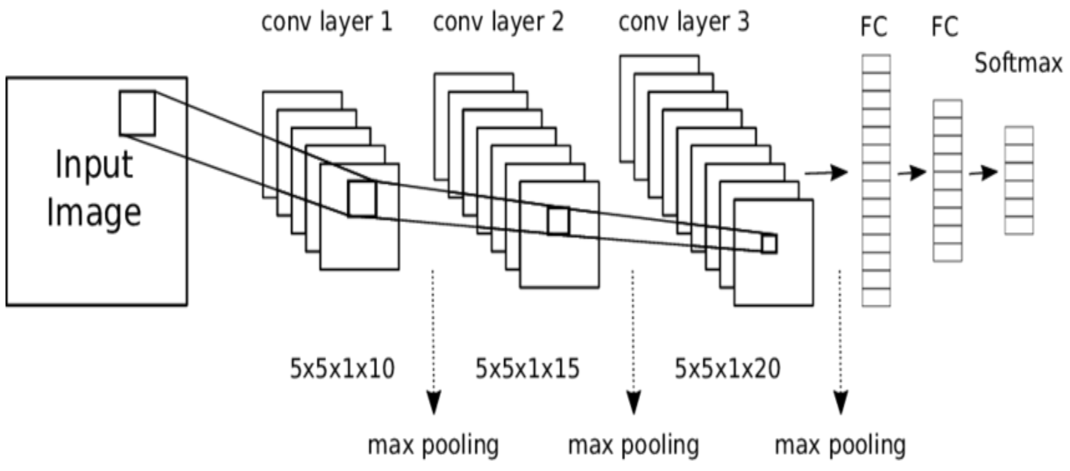


Figure 2.1: CNN architecture [8]

2.2.1 Input layer

The input layer is an image with the resulting measurements of width, height, and depth. As for an example, the input is $64 \times 64 \times 3$ where the width=64, height=64 and depth=3, the depth represents RGB channels here. If the image is 224×224, it needs to be converted to 50176×1. If the input is X, all the samples looking like X should be observed and categorized by CNN.

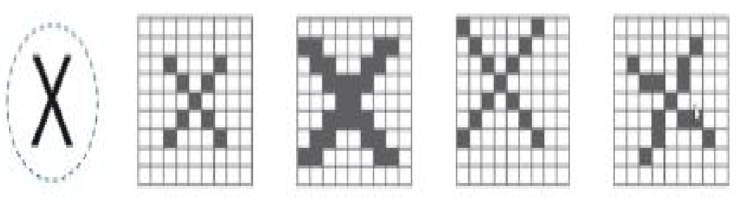


Figure 2.2: Input layer [9]

2.2.2 Convolution Layer

The first layer to remove features from an input image is Convolution. Convolution preserves the relationship between pixels through the use of limited input data squares to learn object functions. Each layer compares the pieces of objects and the pieces found are called characteristics. Convolution maintains the relationship between pixels through the use of limited input squares for learning object features. The operation takes two inputs, for example the picture matrix and a kernel. The output neurons connected to local regions by computation. By choosing 1 or several features of an object and constructing one or several matrix and dot product by using matrix of images, and finally it will give the result that is the convolution layer after the entire procedure. If the size of the image is $M \times M$ and the filter size is $E \times E$ after convolution ,the equation :

$(\mathrm{M}\times M) \times (E \times E) = (M - E + 1) \times (M - E + 1)$ (2.1)

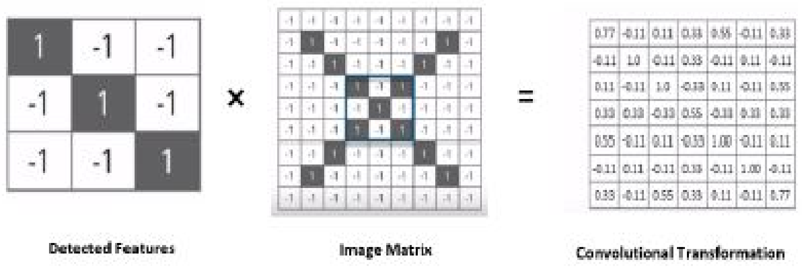
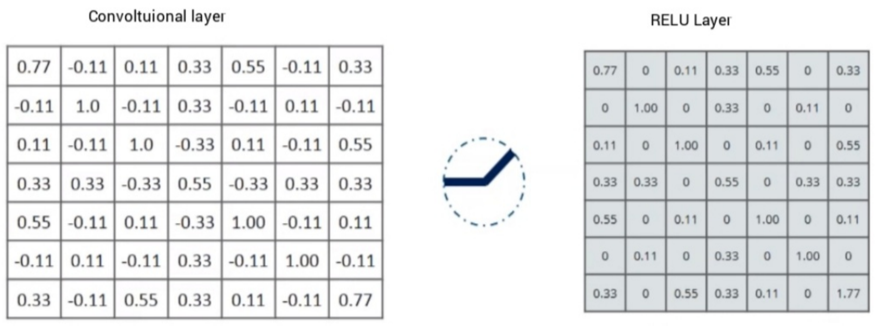


Figure 2.3: Convolutional layer [9]

2.2.3 ReLu Layer

The ReLU is a type of activation function and stands for rectified linear unit. It is used mostly in neural network and efficiently in CNN. The performance of the neural network is calculated like yes or no. The resulting values are fixed between 0 to1 or -1 to 1 etc which depends on functions.



0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.0	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.0	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

0.77	0	0.11	0.33	0.55	0	0.33
0	1.00	0	0.33	0	0.11	0
0.11	0	1.00	0	0.11	0	0.55
0.33	0.33	0	0.55	0	0.33	0.33
0.55	0	0.11	0	1.00	0	0.11
0	0.11	0	0.33	0	1.00	0
0.33	0	0.55	0.33	0.11	0	1.77

Figure 2.4: ReLu layer [9]

2.2.4 Pooling Layer

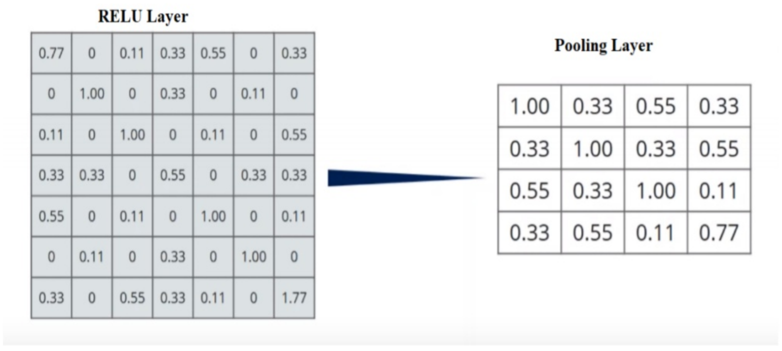
The block of a CNN is a pooling layer. It aims to gradually lower the representation's spatial size, to lessen the network parameters and calculation. Each feature map has its own pooling layer. To pooling, max pooling is the most common approach. There are no parameters but there are further hyper parameters. Filter (A) and Stride (B) are two hyper parameters. If we have input dimension $J_1 \times K_1 \times L_1$ then ,

$$J_2 = (J_1 - A)/(B + 1) \tag{2.2}$$

$$K_2 = (K_1 - A)/(B + 1) \tag{2.3}$$

$$L_2 = L_1 \tag{2.4}$$

Where J_2, K_2 and L_2 are the width, height and depth of output. At first the RELU layer is converted into 4×4 matrix which is shown below ,



RELU Layer

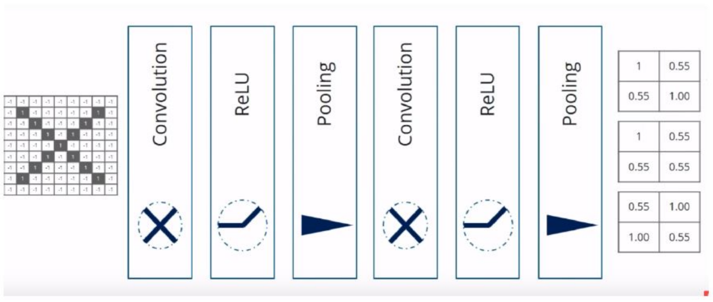
0.77	0	0.11	0.33	0.55	0	0.33
0	1.00	0	0.33	0	0.11	0
0.11	0	1.00	0	0.11	0	0.55
0.33	0.33	0	0.55	0	0.33	0.33
0.55	0	0.11	0	1.00	0	0.11
0	0.11	0	0.33	0	1.00	0
0.33	0	0.55	0.33	0.11	0	1.77

Pooling Layer

1.00	0.33	0.55	0.33
0.33	1.00	0.33	0.55
0.55	0.33	1.00	0.11
0.33	0.55	0.11	0.77

Figure 2.5: Pooling layer [9]

The entire procedure occurs repeatedly and reduces the image data to 2×2 matrix that is shown below in the image.



1	0.55
0.55	1.00

1	0.55
0.55	0.55

0.55	1.00
1.00	0.55

Figure 2.6: Converted matrix [9]

2.2.5 Fully Connected Layer

Fully connected layers are an essential part of the CNN, which has proved to be very successful in the recognition and classification of computer vision images. The CNN procedure begins with convolution and grouping, the image is divided into functions and independently analyzed. It takes the previous layer output, flattens it and transforms it into a single input vector for the next stage. Fully Connected Layer calculates class scores for the column $1 \times 1 \times 12$ for the following picture because there are 3 functions selected and a matrix was generated in the pooling layer for each function. When the number of functions selected was 2, the matrix $1 \times 1 \times 8$ for the same image would have been created.

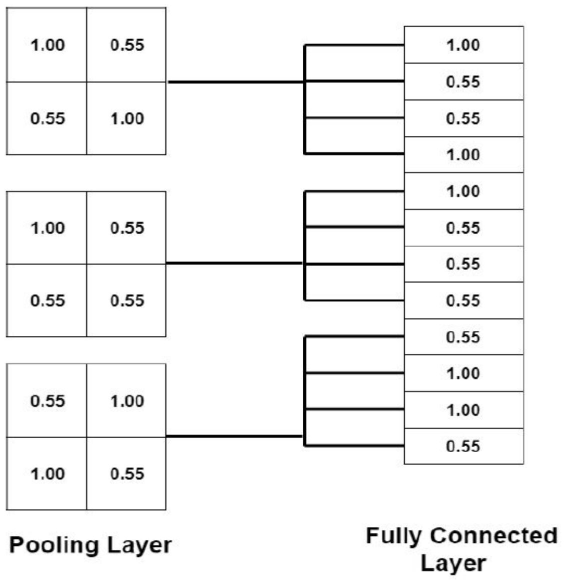


Figure 2.7: Fully connected layer [9]

2.2.6 Output layer

The convolution and pooling layer extract features and from the original images ,they decrease the number of parameters .Although , to determine the final output, we need to apply a fully connected layer to generate an output which is equal to the classes. The layer output contains the 1 dot programmed label. All the data is saved and it is marked as X. it checks how many similarities when another new image is given and then detects whether or not the image is X by providing the data that is saved in its memory. Likewise, CNN then transforms the original pixel image to the ultimate class scores from the original pixel values.

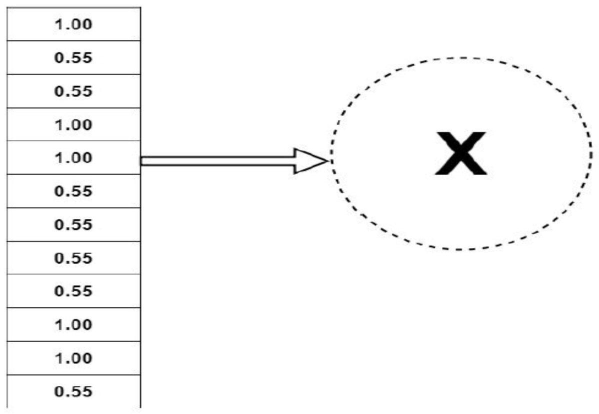


Figure 2.8: Output layer [9]

2.3 Previous works

In this competitive era, many researchers have worked hard for making the classification of leaf more efficient. Efficiency through works makes the best result where we are expected to implement our accuracy rate better. Recognition in leaf attracts many researchers so there are lot of papers which are associated to figure out the classification of leaf through convolutional neural networks and other processing models.

Wang Su Jeon and Sang –Yong Rhee approached a new process for classifying of leaf by using CNN in their paper where they introduced two models for the modification of the depth of network with the use of GoogleNet. Performance of evaluation by each model's techniques has been estimated by them. Although 30% of the leaf was damaged, the classification rate was shown higher than 94% [10].Xiang He, Gang Wang, Li Shang and Xiao Ping proposed Single Connected Layer (SCL) structure that added with the CNN model in their paper. With the use of CNN model they classify the leaf classification and made some improvement of the result. Moreover, they suggested that the efficient accuracy of CNN can be improved by advanced Single Connected Layer [11]. Krizhevsky, Sutskever and Hington had introduced a modified deep convolutionary neural network where they applied resolution images which is 1.2 million considered as high, in the LSVRC-2010 ImageNet competition. They attained 37.5% and 17% accuracy by testing the data which is much better than previous result. They collected the image where the CNN has 5 convolutionary layers, each have and 650,000 neurons and 60 million parameters. They collected some of them that max-pooling layers are gone together by them, and final 1000-way softmax having three fully-connected layers. They introduced neurons which is non-saturating and actual operative GPU application to make training faster while operating the convolution. Moreover, they used an advanced method in recent times and works for regularization to lessen over fitting in the layers of fully connected which is called dropout and this is proved that it is very successful in their research [12].

In the significant classification of image situation K. Simonyan and A. Zisserman examined the convolutional network depth based on its accuracy rate and the effects. Their foremost involvement is a systematic assessment of accumulating the networks of depth by generating very small (3x3) convolution filters structure in their research. They also determined that their performances to be generalized well to other datasets which will made the result accomplished compare to state-of - the-art results. For further facilitation, they have made publicly accessible their two best performing CNN models [13]. Zhang, Zou, He and Sun generated an operative solution where there is no need of Stochastic Gradient Descent (SGD) in CNN model for the result of nonlinear optimization problem. For the generally used VGG-16 model, achieving a full-model speed-up of 4 ranges and with an ordinary 0.3% increase in the classification in ImageNet has a big approach for them where there is just top-5 error. Their VGG-16 accelerated model also exhibits a smooth descent of accuracy [14].

Sabri, N., Abdul Aziz, Z., Ibrahim, Z., Akmal Rasydan Bin acknowledged that classification of leaf by using CNN is useful and more applicable for the appropriate accuracy rate.Although some determined that mobile net model of CNN is capable of precisely classifying leaf species through images .In this paper, the contrast between the simple CNN, AlexNet and GoogleNet was made in connection with accuracy and the time elapsed. The results of the research show that GoogLeNet succeeds a higher accuracy rate than the other two models, which is 100% accurate through testing the dataset. Nevertheless GoogLeNet has the highest accuracy because of the number of layers and more in-depth learning process [15].

Xu.Y, Wang.Y, Zhang Xiang implemented a Probalistic Neural Network (PNN) for applying a general customized algorithm for recognition of leaf with the dataset for classifying the images. The PNN is trained by 1800 leaves to classify 32 plant sort with an isolate accuracy beyond 90% [16]. Lee, S. H., Chan, C. S., Wilkin proposed including species and organ characteristics into classification of plant training the CNN model. They defined their architecture's methodology and evaluated the results which is based on a set of authentications and tests. Comparison of the VGG network with CNN model, the results of the proposed combination architecture are favorable but still partial [17].

Champ, Loreul, Sevajean and Joly applied Neural Network to their model and experiment two classification model such as Convolutional neural network and on the other hand biased model based on fisheries vector. Moreover they conclude that deep neural networks outperforms fisheries vectors for classification in any task especially with a significant number of classes and when you have extensive training data for your model [18].

Jamil, Hossain, Nordin and Awang examined that texture is the feature that really influenced the classification of leaf while shape is supposed as the key feature of leaf classification. The identification of a single texture feature also completed the highest identification rate which compared to color or shape identification. After completing the model it is shown that a single texture has better color or shape feature that succeeds an identification rate of 92% [19]. Yalcin and Razavi propose a construction to categorize the variety from the image categorization of plant in where CNN helps to collect it from smart agro-stations so they can generate their process for the research. First they introduced the change of illumination and deblurring and then they are removed with some preprocessing steps. Then feature of images are extracted by Convolutional Neural Network (CNN). They determined that the construction and depth of CNN are critical points as the capability of recognition of the architecture of neural networks are affected. Furthermore they generated the results of CNN and then compared the result with SVM classifier of different kernels. Their approach is based on the dataset which is collected through project named TARBIL of a government support for which in Turkey there are 1200 agro-stations are positioned [20].

Caglayan, Guclu , Can and Petrosino identified the images of leaves with the help of plant. They extracted the shape and color features of images by applying KNN, SVM , Random Forest and Naïve Bayes that are used for image classification. They test 1897 leaf images and 32 kinds of leave for their research. They found the best result from Random Forest algorithm of 96% by applying both shape and color of leaf.

After experiment they confirmed that there is lack of good result as shape features is classified by similar shaped leaves. By combining shape and color features, the accuracy of classification can be improved. However they also added that reduction of accuracy of the classification happens for seasonal changes of leaf [21].Ruberto, C.D. and Putzu proposed a method for the identification of the leaf by using a set of new features which integrates the features shape, color and texture. They generated the number of 138 features that to be extracted by Support Vector Model for training the research. The approach was tested by using Flavia data set (Wu et al., 2007) to demonstrate excellent performance in terms of 100% accuracy and speed of processing and then removing features from an image by less than a second [22].

Qing-feng, Kun-hui, Chang-le, and Li approached for identifying plant leaf using artificial neural network by extracting features based on shape, margin, dent ,vein etc. First they extracted the leaf of the plant from a background image of different objects and then constructed the hierarchical structure of image classification of leaf. In addition, all visual features of leaf objects have been identified. They have compiled six sorts of pictures of plants, and in every class there are 30 images. The images were distributed into the test set and the training set with a proportion of 6:4 respectively [23]. Tuan, L proposed a model of plant image identification by applying Neural Network. He used background propagation for dividing into two separate phases. Firstly, the derivative function shall be determined by weight. For this, the error is spread back into the network at this point. Additionally, the derivatives are used for measuring change of weight. The easiest method here is the technique of gradient descent [24].

Chapter 3

Workflow and Feature Extraction

3.1 Workflow

In this chapter, there is a description about our proposed model in details as well as the steps of our work. Figure 3.1 shows how we divide our work step by step.

First of all, we have collected the data set then the next step is to divide our data for training and testing. That is why, we split our data into train data and test data. Our train data contains 792 image data of 99 species and the data contains 8 samples of each species. Here, our data set will be trained. Validation data contains total 198 image data where every species has 2 samples for testing. After training and validating it will generate data from images and on the other side, the test data will also generate data from the test images.Finally, The generated data and the pre-trained models we are using will go to the prepared model where it will compare the trained data with the test data and will give the accuracy based on the analysis.

3.2 Data-set Description

We are using a secondary dataset for our work. We have collected the data set from Kaggle [25]. The dataset is known as "Leaf Classification". It is basically a competition to see how people get best accuracy classifying leaves. There are two parts of parts of these dataset, one is Image data and other is csv file. We took only the image data to classify the leaves. The image of the leaves were converted to binary black leaves against the white background. There are total 990 images of leaves in our dataset and they belong to 99 different species. Every species have 10 images of leaves. We took 792 images to from the dataset to train and 198 to test. We have used three models of CNN to test our dataset and the details will be given about models are described in details in chapter 4.

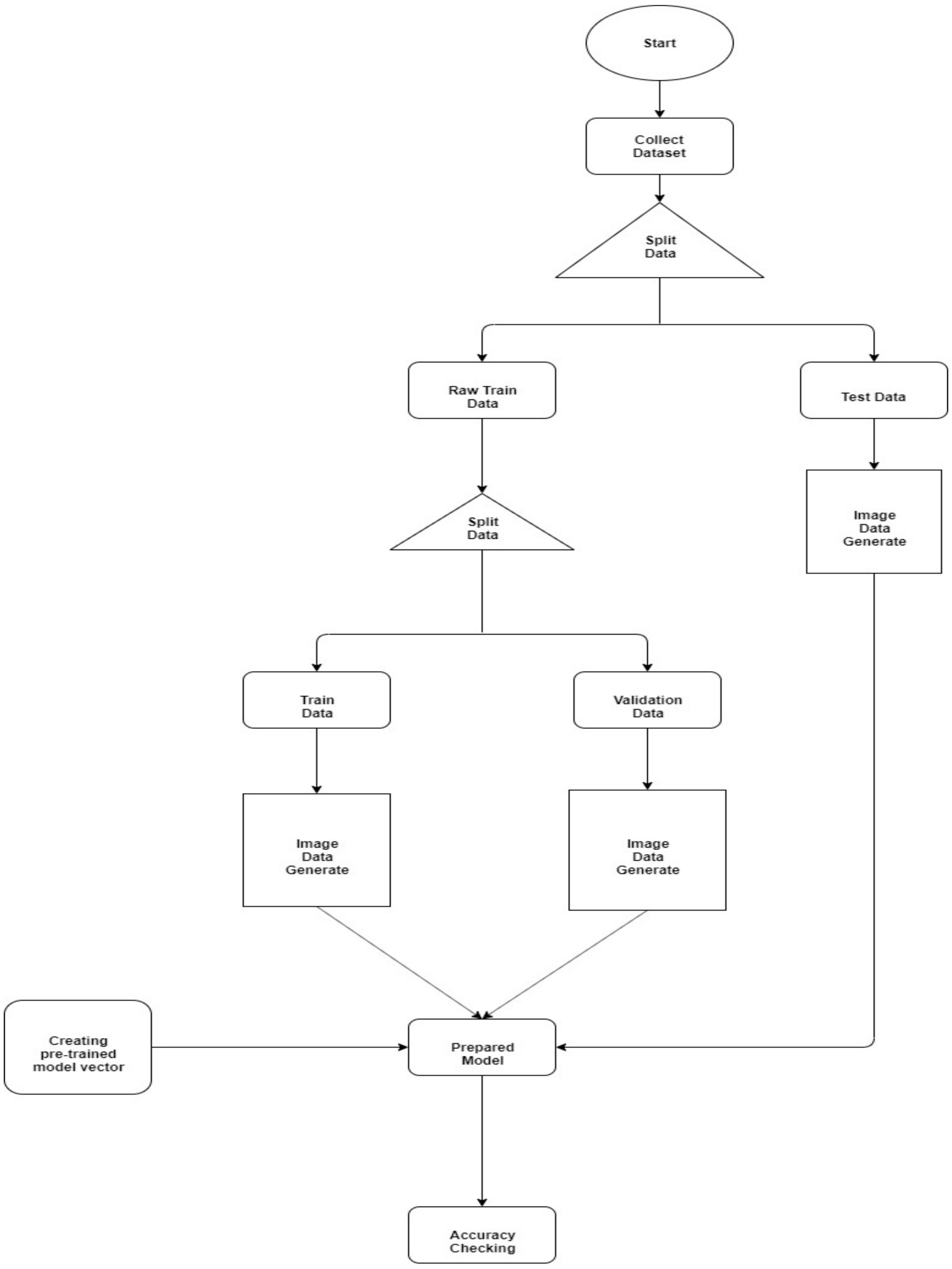


Figure 3.1: Proposed system

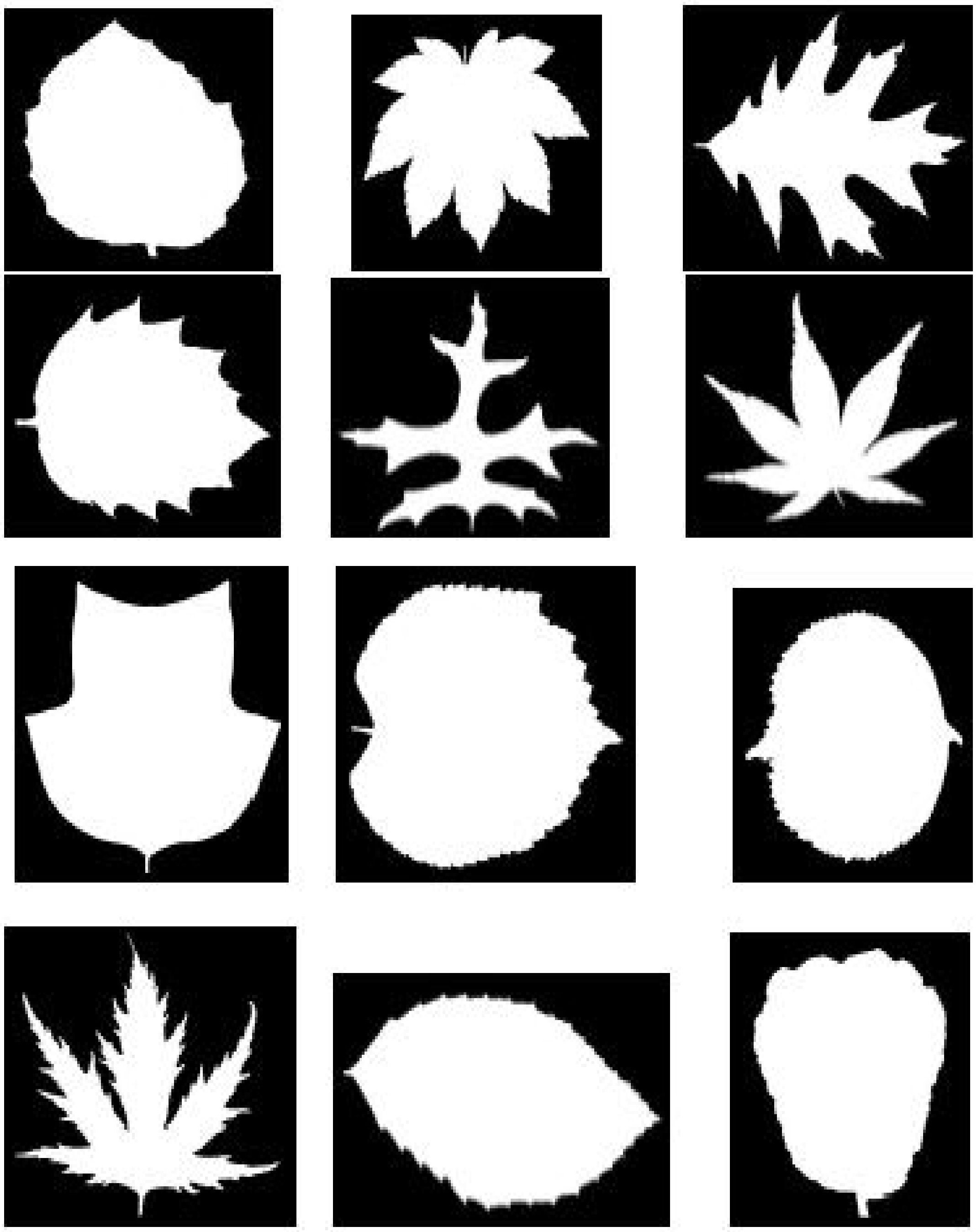


Figure 3.2: Image Dataset

3.3 Pre-processing

3.3.1 Image Processing

Image processing is a method of executing certain operation on an image to generate an optimized image or to extract information from the image. It is a signal processing type where an image is an input and an object or feature is an output which are associated with the image. Respectively, many image processing techniques include considering the image as either a signal or a matrix where typical signal processing or matrix processing techniques is applied.

Purpose of image processing

There are several reasons behind image processing. We divide it into five groups.

1. **Visualization:** Non-visible objects are observed in this state.
2. **Non-visible objects are observed in this state:** To make an efficient image.
3. **Image reclamation:** Determine the interesting picture.
4. **Dimension of pattern:** Various types of images are calculated.
5. **Image recognition:** Generate the object from an image by calculation.

Types of image processing
Image processing has two types and they are called analog image processing and digital image processing [26]. Digital image processing technique uses digital computers for evaluating image. It is used in research and can be defined as it is subjecting a numerical representation of an object. On the other hand, analog image processing defines two-dimensional image and examines hard copy related things like photographs, printout etc. For sharpening and restoration image, we use digital image processing techniques to provide the image to the pre-trained model.

3.3.2 Image pre-processing with Keras

Image classification and recognition is a rapidly growing field in the area of machine learning where object recognition plays as key feature of the image. Keras image processing evaluates and develops deep learning models when we use data preparation and data augmentation with our image datasets. This generator of images provides lots of image data with an increase of real data [27]. The images having different resolution pixels greyscale format is used in our image dataset. Moreover, we used the standard input of the pre-trained model of CNN in keras in where we convert it into resolution 224×224 pixels 8 bit RGB format. Our dataset consists of about 990 images of specimens of leaves (10 samples each of 99 species). These have been converted into binary black leaves against background of white We have used 80% of images as for training and others 20% are used for testing for our model. Our image data have images of different sizes. So, our first job is to convert the images of size 224 × 224 because of the CNN models requirement.First of all, we called our images from the directory to determine the actual size of every image. We, noticed that every images are of different. Figure 3.3 shows the size of a sample of our dataset.

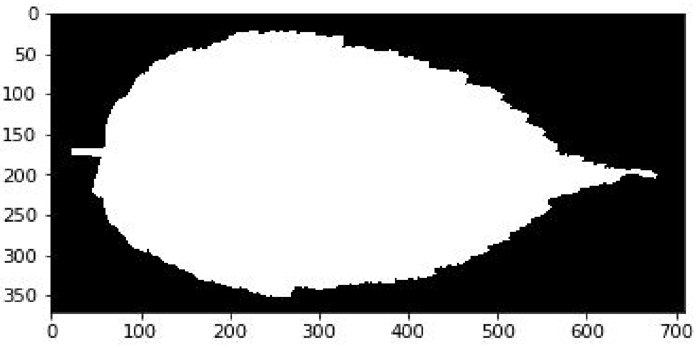


Figure 3.3: Default size of an image from our dataset

From the figure 3.3 we can see that the image size is around 700 × 350 and just like this image, all the images are of different shapes so we need to resize them and the models we are using requires 224 × 224 shape images so we need to convert the image shape.
The following figure shows the reshaped image.

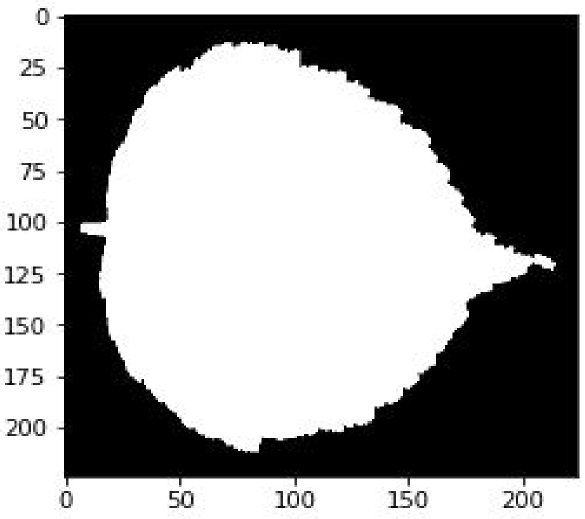


Figure 3.4: Reshaped image.

3.4 Compiling model

We have compiled our models when we completed the image pre-processing step. In our models, we took the image shape as the input which is $224 \times 224 \times 3$ for our data.

3.4.1 inactivating fully connected layer

The fully connected layer which is at the top of the network has been disabled after making the include top false. The fully connected data works for the input image but not ideal for images. It only takes the fixed size of input at the end. When it is the case of whole image, the fully connected network works well but in case of a cropped image, the actual object can't be classified easily. The individual results of the object should be identified such as the type of curves and lines in the image so that the object can be classified with the partial information. It gives a better result. If the fully connected layers are activated, the whole model will be overfitted. As a result, it will work well in the known instances but in case of unknown instances, it will perform badly. When the fully connected network is false, the specific size of images can be given. Otherwise, the size which is fixed by the models will be concentrated.

3.4.2 Selection of weight

In our research, we used imagenet as the weights. The other weights also can be used as null or the path where the weight file is loaded. Here,we are using pre-trained models for comparison. So, using imagenet will help to take the weight from the pre-trained models that have already found which works well.

3.4.3 Selection of pooling layer

We have used pooling 2D for our pooling layer average. We have not used any specific parameters for our average pooling. as a result, it will take the default pool size in keras models which is 2×2. We are using 2D because the model we are using is 2D. it will hamper the result if we use 1D or 3D. However, max pooling would have been used here but it only takes the maximum value from the matrix where in average pooling, it takes the average of the matrix. Max pooling rejects a big chunk of data and it retains at max 1/4th whereas average pooling uses all of it and more information can be retained from average pooling compared to max pooling. That's why we have used average pooling layer to get a better result.

3.4.4 Dropout selection

We have selected dropout 0.2 for our model. It is a form of regularization in CNN that is used for CNN mainly so that it can prevent the overfitting of data. Ignoring the randomly selected neurons during training is the main concept of dropout. Dropout leads a the neural network to learn more robust feature and training time is lessened here. The 0.2 dropout refers that it will ignore 20% of neurons while training. After some research, we found 0.2 is giving more accuracy on image dataset. That is why, we took this dropout rate in our thesis.

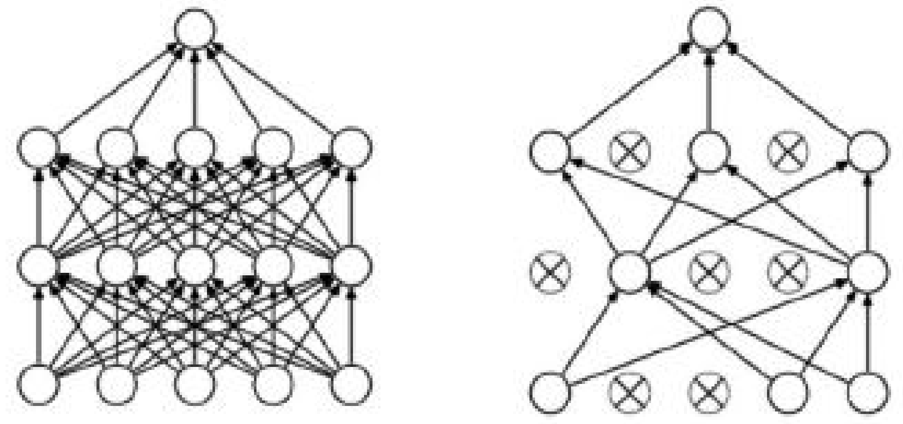


Figure 3.5: Dropout of CNN [28]

3.4.5 Activation function

We have used "ReLu" activation function. Purpose of this function is to convert an input signal as a node in a neural network output signal [29]. Next layers input signal can take the output signal as an input. There are different types of activation functions and the popular ones are:

1. Sigmoid Function (SF)
2. Tanh Function
3. ReLu Function

Sigmoid Function

Sigmoid function has s-shaped curve and it has the range between 0 to 1. It is easy to apply compared to other functions but it vanishes the gradient and its output is not zero centered. As a result, the optimization becomes very harder as the gradient updates go too far in different direction.

Tanh Function Though it also suffers from vanishing gradient problem, it gives a better result than sigmoid function and its output is zero centered.

ReLu Function

the mathematical function of ReLu is [30]

$$Z(x) = max(0, x),\ \{Z(x) = x, ifx \geq 0),\ Z(x) = 0, ifx < 0\} \quad (3.1)$$

It is very efficient and it does not vanish the gradient. Though it has a limitation and that is it can be applied only in the hidden layers yet, this activation function is less complex that is why we used it in our model.

3.5 Applying the prepared model for testing

Now we need test our dataset with the model we created to see the accuracy of different models of CNN. The next chapter will describe about this in details.

Chapter 4

Implementation and result

In this chapter we will give the detail description of the models we have used in our research work and the outcomes of the research as well.

4.1 Inception V3

4.1.1 Concept of Inception

The first inception model was introduced by Google, named Inception-v1 [31]. The first model has three individual size filters to perform convolution on a particular input. The filter sizes are 1×1, 3×3 and 5×5. First the model does max pooling and after that the outputs are concatenated and then it goes to the next step of inception. The major issue is that it needs a huge amount of operation and when the pooling layer is added to mix the issue gets more complexed. The idea to add a convolution with the size of 1×1 helps to perform faster compared to naïve inception model. Factorization was introduced to minimize the dimensionality, it helped to reduce the overfitting problem [32]. As an example, if we take a layer of 5×5 filter then we get a total 25 parameters but if we take two layers of 3×3 filter then we get $3 \times 3 + 3 \times 3 = 18$ parameters. We can clearly see the difference between the two parameters and also, we can see by changing filter size we can reduce the total number of parameters.

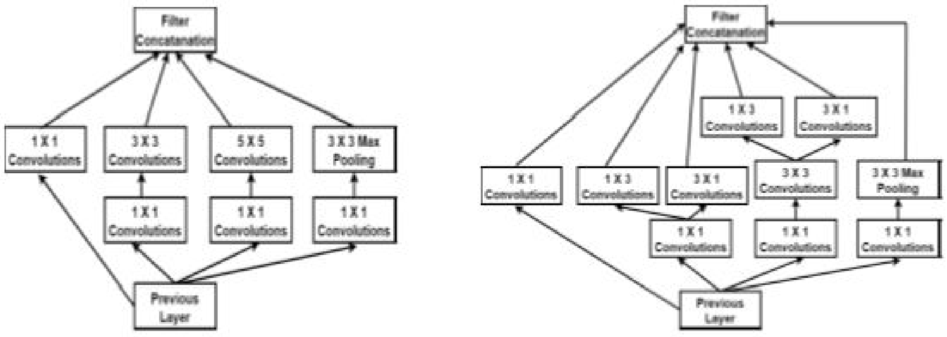


Figure 4.1: Inception V1 vs Inception V3 [33]

We can reduce the number of parameters even more. Here, for the 3×3 filter, if we take one 3×1 filter and one 1×3 filter then the total numbers of parameters will be reduced to $3 \times 1 + 1 \times 3 = 6$. In this way, the number of parameters of the whole factorization process can be reduced and the network can go deeper.

4.1.2 Training our dataset with Inception V3

First of all, we prepared our model then we have fit our train image data and validation data in our model. Here, we used 64 and 128 batch size per epoch with 20 epochs to observe the difference of the outputs. At each epoch the validation data evaluates the loss. We noticed that steps per epochs affects the accuracy rate. If we increase the batch size then the accuracy rate increases because more steps, we increase the more the model gets data to train at a particular epoch and this is why accuracy increases. As the models save only the best values increasing epoch may give better accuracy but in our case the difference was negligible.

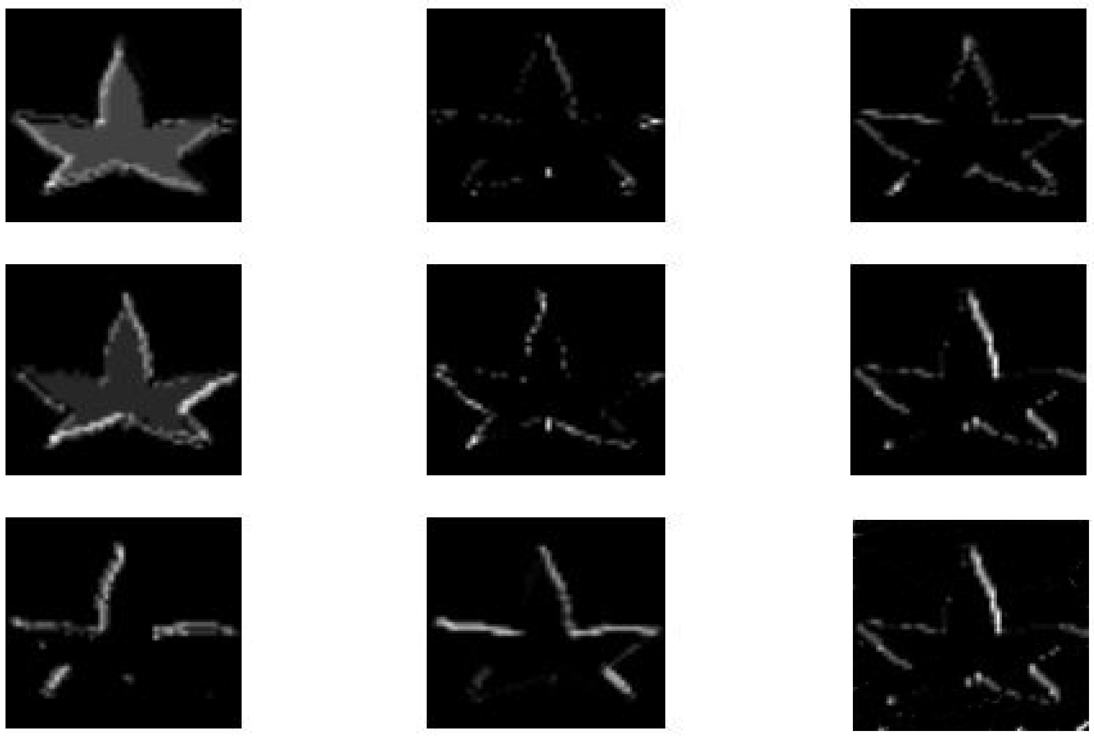


Figure 4.2: Inception v3 training at 20 epochs 64 steps

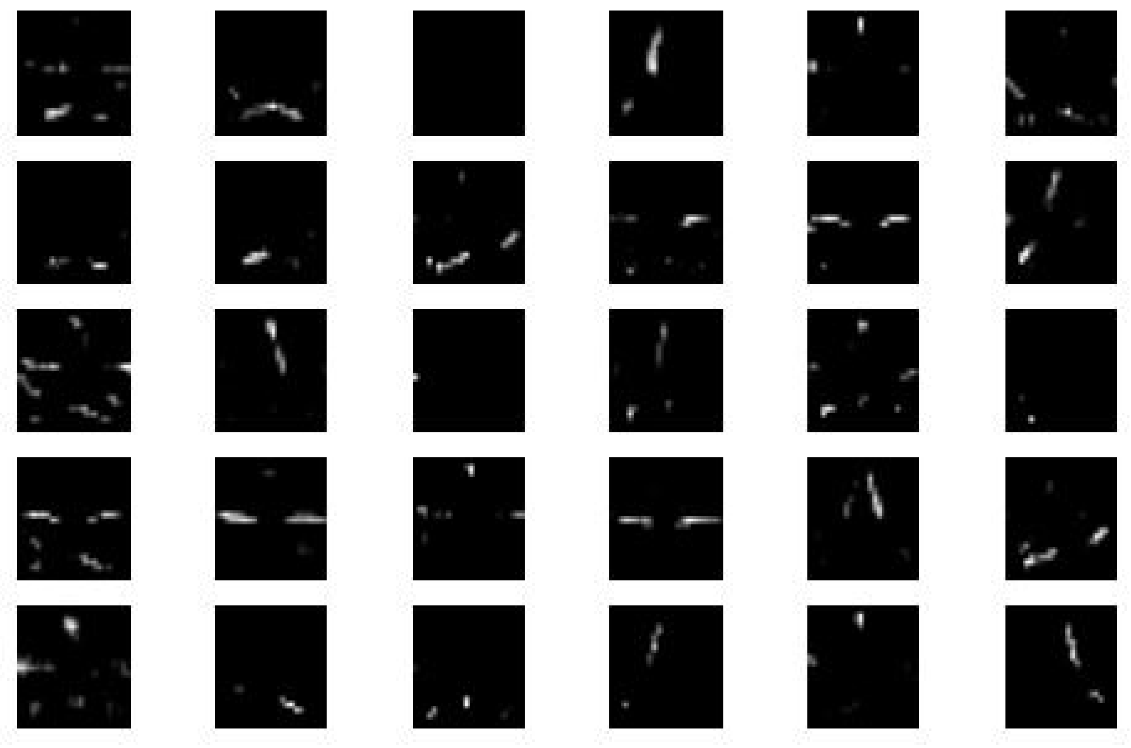


Figure 4.3: Inception v3 training at 20 epochs 128 steps

4.1.3 Result of Inception V3

After training our data we validate it with our test file. The following table shows the result of inception v3 at 20 epochs 64 steps followed by 128 steps.

Steps per epochs	Mean absolute error	Explained Variance Score	Value loss
64	35.24	0.647	0.352
128	33.87	0.661	0.338

Table 4.1: Inception V3 result with 20 epochs

from the table 4.1 we can see that, the accuracy rate of inception v3 at 20 epochs and 128 steps is 66.13% and value loss is 33.87% and at 64 steps the accuracy rate is 64.76% and value loss is 35.24%.

4.2 Mobile Net

4.2.1 Idea of MobileNet

Mobile Nets are based on depth wise separable convolutions and renowned for taking less time and short size which is not like other convolution models. Other convolution models take huge time for compilation where Mobile Net performs better because it's a class for low latency. Mobile nets are not as accurate compared to other large models because it has less parameters compared to them but it gives better accuracy in small datasets and as our dataset has only 990 images mobile net gave the best accuracy for us. Moreover, Mobile Net is famous for trading off reasonable amount of accuracy by reducing size and latency. So mobile nets are more applicable for deep learning models for mobile data. For Mobile Nets, a single filter is applied to each input channel by the depth wise Convolution. The point-sensitive convolution then applies a 1×1 convolution to combine the deep convolution outputs. Then, a standard convolution filters and combines the input for a new output in the next step. The depth wise separable convolution splits this into two separate layers, one is for filtering and another for combining [34].

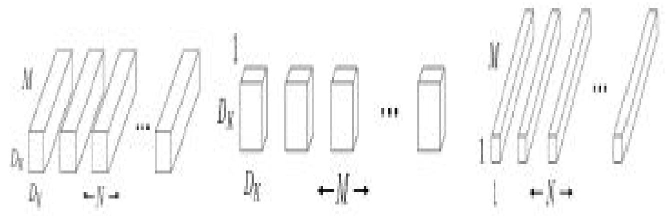


Figure 4.4: Standard Convolution Filter and MobileNet Convolution Filter [34]

A standard convolutional layer takes as input a $D_f \times D_f \times M$ feature map called f and generates a $D_f \times D_f \times N$ feature map called G where D_f is the spatial width and height of a square, M is the number of input depth, Dg is the spatial width and height of a square output feature map and N is the number of output depth. A convolution kernel k parameterizes the standard convolutional layer of size $D_k \times D_k \times M \times N$ where D_k is the spatial dimension of the kernel which is square. Standard convolutions have the computational cost that is given below.

$$\mathrm{D}_k \times D_k \times M \times N \times D_f \times D_f \quad (4.1)$$

The computational cost depends on the number of input depth M, the number of output depth N and the kernel size which is $D_k \times D_k$ and the feature map size which is $D_f \times D_f$ [11].

Mobile Net generates 3×3 depth wise separable convolutions which is used between 8 to 9 times less calculation than standard convolutions at only a small decrease in accuracy.

4.2.2 Training our dataset with MobileNet

Same as Inception v3, we prepared our model then we have fit our train image data and validation data in our model and we used 64 and 128 batch size per epoch with 20 epochs to observe the difference of the outputs.

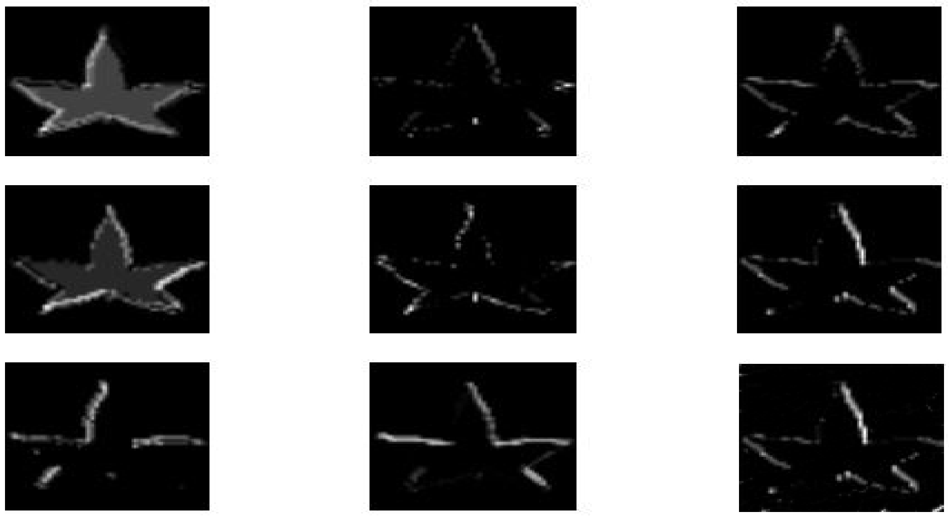


Figure 4.5: MobileNet training at 20 epochs 64 steps

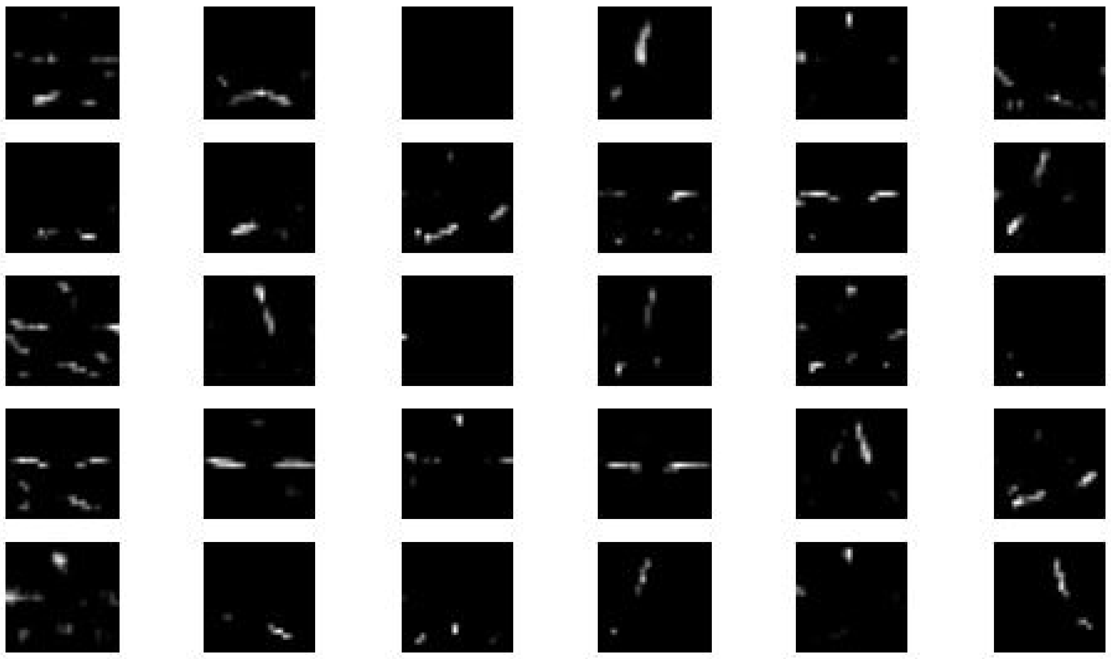


Figure 4.6: MobileNet training at 20 epochs 128 steps

4.2.3 Result of MobileNet

After training our data we validate it with our test file. Table 4.2 shows the result of MobileNet at 20 epochs 64 steps followed by 128 steps.

Steps per epochs	Mean absolute error	Explained Variance Score	Value loss
64	32.80	0.672	0.328
128	30.53	0.694	0.305

Table 4.2: MobileNet result with 20 epochs

from the table 4.2 we can see that, the accuracy rate of mobilenet at 20 epochs and 128 steps is 69.47% and value loss is 30.53% and at 64 steps the accuracy rate is 67.20% and value loss is 32.80%.

4.3 VGG16

4.3.1 VGG Concept

This model is from VGG group, Oxford in 2014 [35]. It is a simple architecture model since it does not use much hyper parameters.The input of this algorithm is fixed in size 224×224 RGB image in training time. The main processing is to subtract the mean RGB value and calculate the training set from each pixel. Then the image is transferred through some convolution layers of 3×3 filters with a stride 1 pixel. It is used as the least size to capture the notion. In the setup, we used 1×1 convolution channel that is the linear transformation of the input channels. Protecting the spatial resolution after the convolution is the end goal which depends on the spatial cushioning of convolution layer input. Five max-pooling layers carry the spatial pooling which follow some of the convolutional layers. In the max-pooling, 2×2 pixel with stride 2 is used. Three fully connected layers trail a stack of convolution layers. The initial two have 4096 channels each and the third performs 1000 –way ILSVRC arrangements that contains 1000 channels (one for each class). Soft max layer is the last layer. The fully connected layers setup is equivalent in all networks. Rectification (ReLu) non-linearity is used to furnish the shrouded layers.

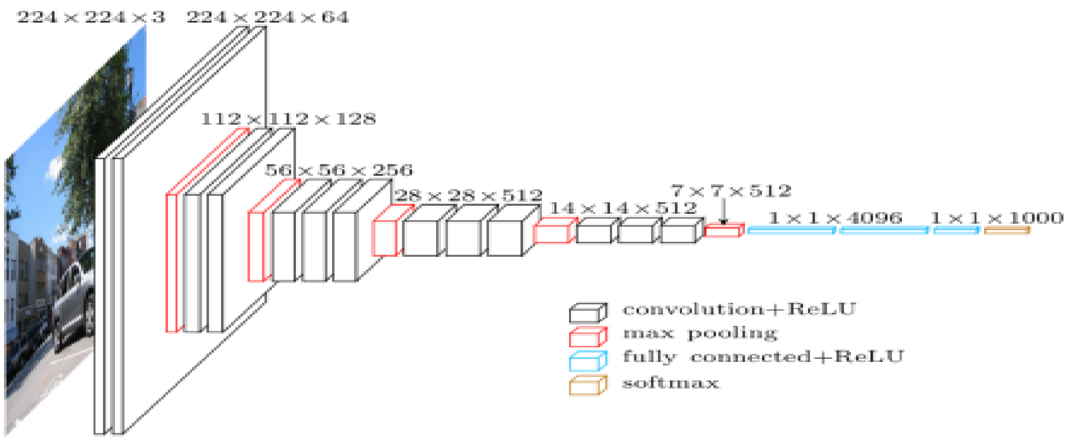


Figure 4.7: VGG16 model architecture [36]

4.3.2 Training our dataset with VGG16

Same as Inception v3 and MobileNet we fit our image data and did validation data in the model and we used batch size of 64 and 128 with 20 epochs to train the image data.

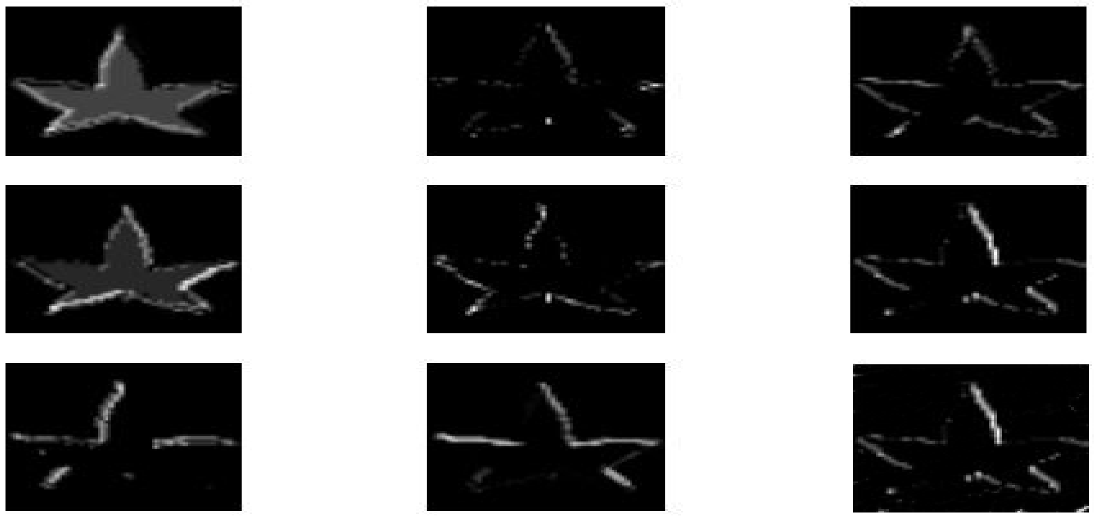


Figure 4.8: VGG16 training at 20 epochs 64 steps

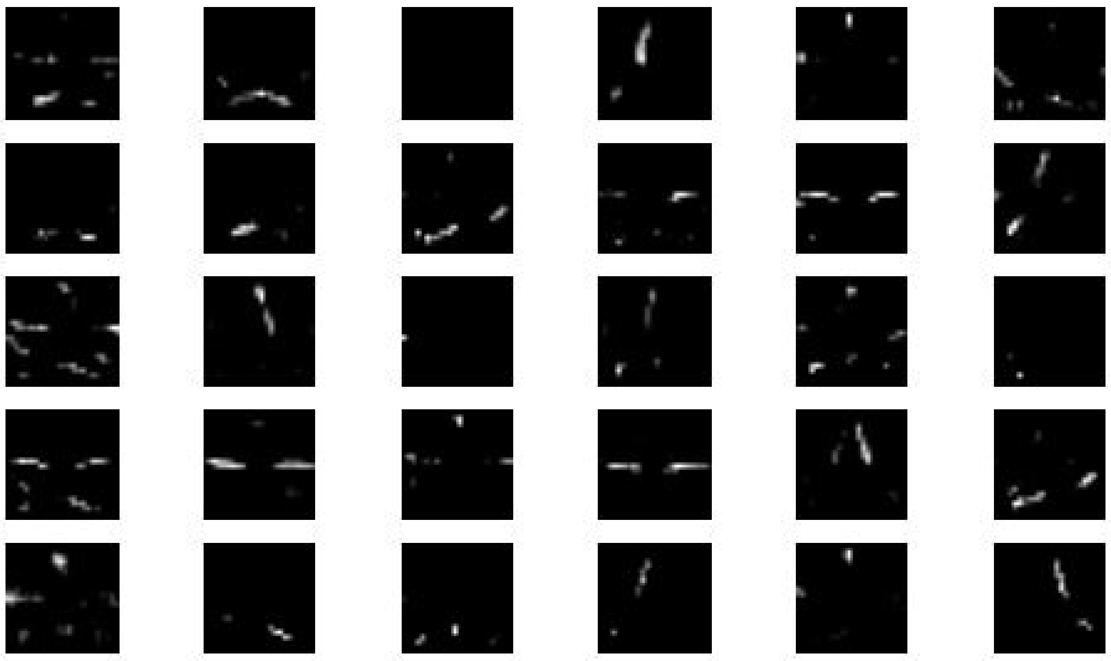


Figure 4.9: VGG16 training at 20 epochs 128 steps

4.3.3 Result of VGG16

After training our data we validate it with our test file. The following table shows the result of VGG16 at 20 epochs 64 steps followed by 128 steps.

Steps per epochs	Mean absolute error	Explained Variance Score	Value loss
64	44.36	0.556	0.443
128	42.95	0.570	0.429

Table 4.3: VGG16 result with 20 epochs

from the table 4.3 we can see that, the accuracy rate of vgg16 at 20 epochs and 128 steps is 57.05% and value loss is 42.95% and at 64 steps the accuracy rate is 55.64% and value loss is 44.36%.

4.4 Comparison of CNN Models

MobileNet gave the best accuracy among the three models. It has accuracy of 69.47% and value loss is 30.53%. It has performed very well than the other two models with its less parameter. The reason behind this is its depth wise convolution layers and also the pooling layer shape and the convolution shape are similar which has caused less data loss.

Inception V3 has the accuracy of 66.13% and the value loss is 33.87%. It has more parameters than the MobileNet model but gives very close accuracy. The reason of giving this accuracy is its high non trainable parameters.

VGG16 has given the less accuracy than the two other models which is 57.05%. This model has less parameters than others which results in huge value loss. The convolution parameters are lesser than the other two models which affects the accuracy rate. We know that the more accuracy is showed when the model has more parameters. Figure 4.38-4.39 shows the graphical representation of the comparison between MobileNet, Inception V3 and VGG16.

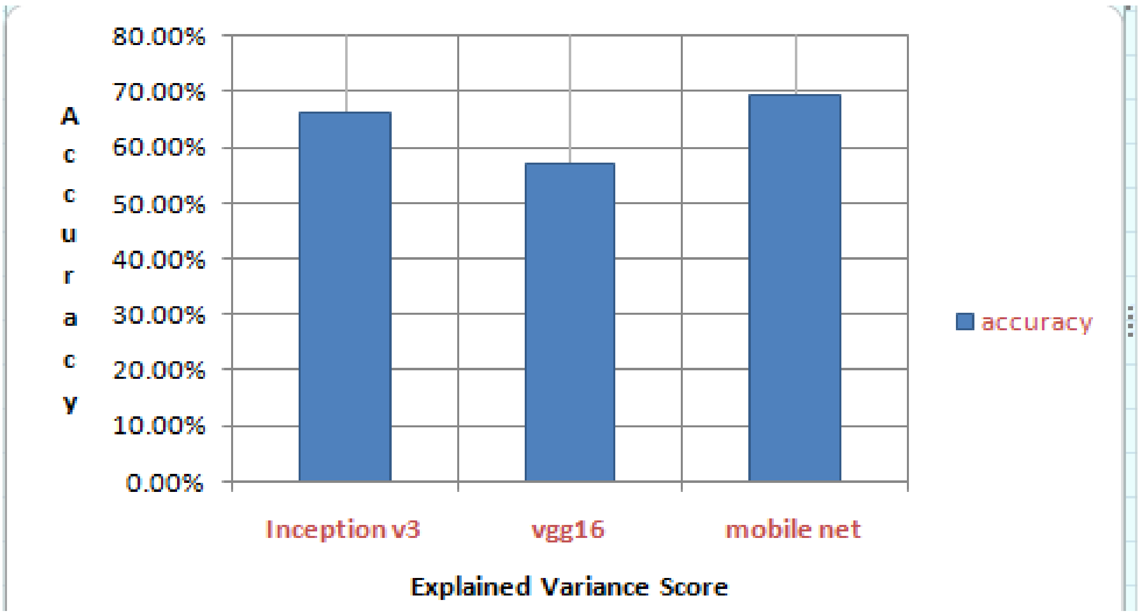


Figure 4.10: Accuracy rate of the pre-trained models

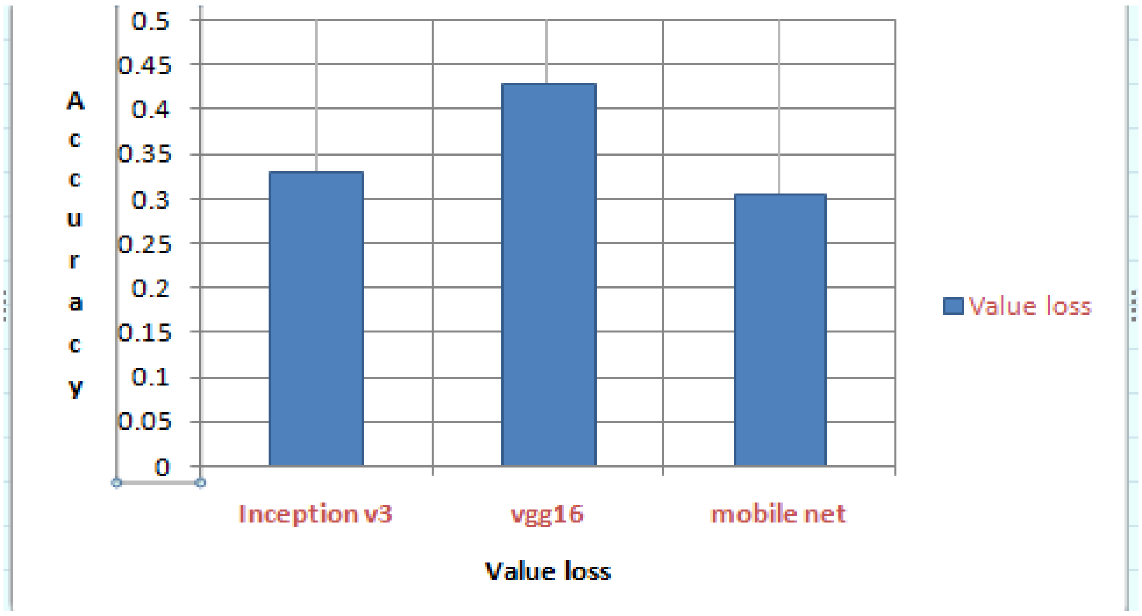


Figure 4.11: Value loss of the pre-trained models

Chapter 5

Conclusion

5.1 Conclusion

The purpose of our thesis is to determine the accuracy rate of leaf classification from CNN models. There are some CNN models which work better for detection. Our work is to find out the accuracy through the images. We get so many things which will help us working with image recognition and classification while working with CNN models. Our experimental results exhibit that learning the features by using CNNs which can provide better feature representations of leaf images. We defined our architecture approach and evaluated the results both validated and tested. At first we divide our works into two segments by testing and training then we use three CNN models for finding accuracy rate. We also computed the features that repre- sent the leaves for the purpose of species detection proficiently. Accuracy rate of MobileNet is 69.47% and value loss is 30.53% , accuracy rate of Inception V3 is 66.13% and the value loss is 33.87%. VGG16 has given less accuracy compared to the other two models which is 57.05%. Moreover, we find that calculation time of MobileNet is faster than VGG16.VGG16 has less amount of accuracy and having more value loss. The entire procedure was identified from the collection of images of dataset used for training and testing through image process- ing and the training of the deep CNN is finally followed. Several studies have been conducted to verify the performance of our models.

5.2 Future plan

CNN helps us efficiently for the classification of leaves but It is quite easy for machines to classify cats and dogs from images compared to leaf classification because cats and dogs have a significant number of dissimilarities between them whereas, leaves have very less dissimilarities and many things in common. So, it is difficult to classify leaves from images with greater accuracy.As we know, machine learning approaches a big dataset for training the model. But we have got 990 image data of leaves to train which is to be expected high. For this reason our accuracy rate is not so high as we expected.To improve the generalization of each model.the first and observable step is to collect more data. Moreover, in today's world, we see leaves are having different kind of diseases. Many leaf diseases can be identified by observing the leaf. Sometimes, we see some symptoms on leaves such as, having holes, faded color and also, seems like the leaf is dying or dead already. By taking those samples in Image data we will be able to predict diseases of leaves. In future works, we are planning to use local images and improve CNN models in order to increase the performance of classification. Furthermore, we will collect more images of different diseases in order to improve the database. We will take the pictures of leaves from real environment, so that the background of the image will be in RGB instead of grey.

In our research, we have used different models of CNN and CNN automatically extracts the feature. It is used to analyze data and image recognition and processing. We used one algorithm which is slower, time consuming and less accurate but in future we want to use more algorithm and models of CNN like ResNet, AlexNet, LeNet etc to check the better accuracy and performance.

Bibliography

[1] C. Sumathi and A. S. Kumar, "Plant leaf classification using soft computing techniques", *International Journal of Future Computer and Communication*, vol. 2, no. 3, p. 196, 2013.

[2] M. A. F. Azlah, L. S. Chua, F. R. Rahmad, F. I. Abdullah, and S. R. Wan Alwi, "Review on techniques for plant leaf classification and recognition", *Computers*, vol. 8, no. 4, p. 77, 2019.

[3] Prabhu, *Understanding of convolutional neural network (cnn) — deep learning*, https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148, [Accessed: 08-Nov-2019], Nov. 2019.

[4] J. Shuvam, *Understanding of convolut*, https://https://becominghuman.ai/what-exactly-does-cnn-see-4d436d8e6e52, [Accessed: 12-Nov-2019], Nov. 2019.

[5] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network", in *2017 International Conference on Engineering and Technology (ICET)*, IEEE, 2017, pp. 1–6.

[6] L. Deng, G. Hinton, and B. Kingsbury, "New types of deep neural network learning for speech recognition and related applications: An overview", in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2013, pp. 8599–8603.

[7] H. Greenspan, B. Van Ginneken, and R. M. Summers, "Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique", *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1153–1159, 2016.

[8] T. Hinz, *Cnn architecture*, https://https://www.researchgate.net/figure/CNN-architecture-our-CNN-consists-of-three-convolutional-layers-with-10-15-and-20_fig1_306081277, [Accessed: 13-Nov-2019], Nov. 2016.

[9] Edureka, *Ai deep learning with tensorflow*, https://https://www.edureka.co/ai-deep-learning-with-tensorflow/, [Accessed: 19-Nov-2019], Nov. 2019.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", *arXiv preprint arXiv:1409.1556*, 2014.

[12] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition”, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[13] N. Sabri, Z. A. Aziz, Z. Ibrahim, M. M. Rosni, and A. A. Ghapul, “Comparing convolution neural network modelsfor leaf recognition”, *International Journal of Engineering & Technology*, pp. 141–144, 2018.

[14] S. G. Wu, F. S. Bao, E. Y. Xu, Y.-X. Wang, Y.-F. Chang, and Q.-L. Xiang, “A leaf recognition algorithm for plant classification using probabilistic neural network”, in *2007 IEEE international symposium on signal processing and information technology*, IEEE, 2007, pp. 11–16.

[15] S. H. Lee, C. S. Chan, P. Wilkin, and P. Remagnino, “Deep-plant: Plant identification with convolutional neural networks”, in *2015 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2015, pp. 452–456.

[16] J. Champ, T. Lorieul, M. Servajean, and A. Joly, “A comparative study of fine-grained classification methods in the context of the lifeclef plant identification challenge 2015”, 2015.

[17] N. Jamil, N. A. C. Hussin, S. Nordin, and K. Awang, “Automatic plant identification: Is shape the key feature?”, *Procedia Computer Science*, vol. 76, pp. 436–442, 2015.

[18] K. Documentation, *Image preprocessing*, https://https://https://keras.io/preprocessing/image/, [Accessed: 5-Nov-2019], Jun. 2013.

[19] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications”, *arXiv preprint arXiv:1704.04861*, 2017.

[20] H. Yalcin and S. Razavi, “Plant classification using convolutional neural networks”, in *2016 Fifth International Conference on Agro-Geoinformatics (Agro-Geoinformatics)*, IEEE, 2016, pp. 1–5.

[21] A. Caglayan, O. Guclu, and A. B. Can, “A plant recognition approach using shape and color features in leaf images”, in *International Conference on Image Analysis and Processing*, Springer, 2013, pp. 161–170.

[22] C. Di Ruberto and L. Putzu, “A fast leaf recognition algorithm based on svm classifier and high dimensional feature vector”, in *2014 International Conference on Computer Vision Theory and Applications (VISAPP)*, IEEE, vol. 1, 2014, pp. 601–609.

[23] Q. Wu, C. Zhou, and C. Wang, “Feature extraction and automatic recognition of plant leaf using artificial neural network”, *Advances in Artificial Intelligence*, vol. 3, pp. 5–12, 2006.

[24] A. Begue, V. Kowlessur, F. Mahomoodally, U. Singh, and S. Pudaruth, “Automatic recognition of medicinal plants using machine learning techniques”, *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 4, pp. 166–175, 2017.

[25] Kaggle, *Leaf classification*, https://www.kaggle.com/c/leaf-classification/data, [Accessed: 12-Jun-2019], Nov. 2016.

[26] D. Gardner, A. W. Toga, G. A. Ascoli, J. Beatty, J. F. Brinkley, A. M. Dale, P. T. Fox, E. P. Gardner, J. S. George, N. Goddard, *et al.*, "Towards effective and rewarding data sharing", 2003.

[27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting", *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[28] J. Heaton, *Programming Neural Networks with Encog2 in C.* Heaton Research, Inc., 2010.

[29] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[30] 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.

[31] F. Nawaz, M. Akib, A. Imtiaz, S. U. Ahmad, *et al.*, "Bone age comparison using convolutional neural network", PhD thesis, Brac University, 2019.

[32] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", *arXiv preprint arXiv:1409.1556*, 2014.

[33] W. J. Schroeder, L. S. Avila, and W. Hoffman, "Visualizing with vtk: A tutorial", *IEEE Computer graphics and applications*, vol. 20, no. 5, pp. 20–27, 2000.

[34] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications", *arXiv preprint arXiv:1704.04861*, 2017.

[35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", *arXiv preprint arXiv:1409.1556*, 2014.

[36] R. Thakur, *Step by step vgg16 implementation in keras for beginners*, https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c, [Accessed: 14-Nov-2019], May 2015.