

Flower Identification Using Machine Learning

BY

Md. Mizanur Rahman

ID: 151-15-4910

Akash Ahmed Khan

ID: 151-15-4883

Md. Baher Uddin Shameem

ID: 151-15-5299

This Report conferred to the Department of CSE of Daffodil International University in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Dr. Syed Akhter Hossain

Professor and Head

Department of Computer Science and Engineering

Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

DECEMBER 2018

APPROVAL

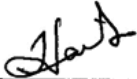
This Project/internship titled “**Flower Identification Using Machine Learning**”, submitted by Md. Mizanur Rahman, ID: 151-15-4910 ; Akash Ahmed Khan, ID: 151-15-4883 ; Md. Baher Uddin Shameem, ID: 151-15-5299 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 11 December 2018.

BOARD OF EXAMINERS

Dr. Syed Akhter Hossain
Professor and Head

Chairman

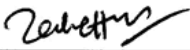
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University



Dr. Sheak Rashed Haider Noori

Internal Examiner

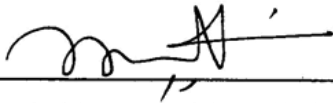
Associate Professor & Associate Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University



Md. Zahid Hasan

Internal Examiner

Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University



Dr. Mohammad Shorif Uddin

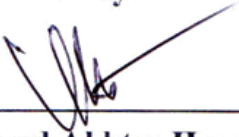
External Examiner

Professor
Department of Computer Science and Engineering
Jahangirnagar University
©Daffodil International University

DECLARATION

We hereby declare that, this research based project has been done by us under the supervision of **Dr. Syed Akhter Hossain** Professor and Head Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:



Dr. Syed Akhter Hossain
Professor and Head
Department of CSE
Daffodil International University

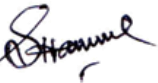
Submitted by:



(Md. Mizanur Rahman)
ID: 151-15-4910
Department of CSE
Daffodil International University



(Akash Ahmed Khan)
ID: 151-15-4883
Department of CSE
Daffodil International University



(Md. Baher Uddin Shameem)
ID: 151-15-5299
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

We take this opportunity to express our sincere gratitude to **Prof. Dr. Syed Akhter Hossain** Sir, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of “*Machine Learning*” to carry out this project. His endless patience, scholarly guidance, strong motivation, continual encouragement, constant and energetic supervision, constructive criticism , valuable advice, reading many inferior draft and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to the Almighty Allah and also to other faculty member and the staff of CSE department of Daffodil International University to finish our project.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

Beauty is incomplete without flower. Bangladesh is the land of flower. In our everyday life, On the way of walking, beside the rail line or in our garden we used to see a lot of flower. But in most case we have no knowledge about that flower. Even we don't know its name. In that case we choose this idea to research and develop our project. That will introduce people about that unknown flower which they see but don't know about that. Our developed application recognizes the flower in real time by using mobile camera. This project is an attempt at using the concepts of neural networks to create an image classifier by Tensorflow on Android platform. Convolutional neural networks are a popular realm of machine learning, and are widely used in image classification. So that we choose this topic to research about image classification by CNN and Tensorflow. Though we developed an application of our research project but we have a lot of plan to do more research to upgrade our system.

TABLE OF CONTENTS

CONTENTS	PAGE
Acknowledgement	iv
Abstract	v
CHAPTER	
CHAPTER 1: INTRODUCTION	1-4
1.1 Introduction	1
1.2 Motivation	2
1.3 Rationale of the Study	2
1.4 Research Questions	3
1.5 Expected Outcome	4
CHAPTER 2: BACKGROUND	5-9
2.1 Introduction	5
2.2 Related Works	5-6
2.3 Research Summary	7
2.4 Scope of the Project	8
2.5 Challenges	9
CHAPTER 3: RESEARCH METHODOLOGY AND IMPLEMENTATION	10-18
3.1 Introduction to Machine Learning	10
3.2 Convolutional Neural Network	11
3.3 Design of Dataset	12
3.4 Data Collection Procedure	13
3.4 Statistics of Data	14
3.5 Implementation Requirements	14
3.6 Implementation of Network Design	15-17
3.7 Design Development	18

CHAPTER 4: EXPERIMENTAL RESULTS AND DISCUSSION	19-27
4.1 Introduction	19
4.2 Experimental Results	20-21
4.3 Main Graph	22
4.4 Descriptive Analysis	23-27
4.4 Experiment Summary	27
CHAPTER 5: SUMMARY, CONCLUSION, RECOMMENDATION AND IMPLICATION FOR FUTURE RESEARCH	
5.1 Summary of the Study	28
5.2 Future work	29
5.3 Conclusions	29
References	30
Appendices: Accuracy Value For Some Training Steps	31-33

List OF FIGURES

FIGURES	PAGE
Figure 1.5 : Rose Identification with highest confidence value	4
Figure 2.2 : PictureThis application feature graphic image	6
Figure 2.3 : PlantSnap application feature graphic image	6
Figure 3.1: Learning System Structure	10
Figure 3.2: Typical CNN design	11
Figure 3.3: Design of Dataset	12
Figure 3.4: Dataset of Shapla	13
Figure 4.1: Image pre processing by tensorflow	19
Figure 4.2: Training Step	20
Figure 4.3: Experiment Result using Android Application	21
Figure 4.4: Main Graph of Tensorflow Computation	22
Figure 4.5: Train Accuracy Graph	23
Figure 4.6 : Cross Entropy Graph	24
Figure 4.7: Final Training Summaries Graph	24
Figure 4.8: Activation Graph	25
Figure 4.9: Final Training Summaries Histogram	25
Figure 4.10: Final Training Summaries Histogram	26
Figure 4.11 : Train Accuracy Value	27
Figure 4.12: Cross Entropy Value	27

LIST OF TABLE

TABLE	PAGE
Table 3.4 : Statistics of Data	14

CHAPTER 1

1.1 INTRODUCTION

Flower is a very important part of nature. Mostly we identify a plant through its flower. Experienced botanists do this identification of flower but a naive person will have to consult flower guidebooks or browse any relevant web pages on the Internet through keywords searching. Our system can recognize the flower in real time using mobile camera. Currently this Android app can identify around 10 flowers. Most important thing is that this app can fully work in offline. We are continuously working to add more flowers to identify. Everyday we see a huge number of flower species in our house, parks, roadsides, in farms, on our rooftop but we have no knowledge of that flower species or their origin. Even we have no idea about its name. There are several guidebooks for flowers knowledge but it becomes quite difficult to find the name when we have the picture. Even the Internet sometimes is not useful. But it is quite difficult for human brain to memorize all the species they see. Even some flower is similar to look at. This application recognizes the flower in real time by using mobile camera.

The purpose of this project is to use Tensorflow, an open-source dataflow and machine learning library, to build an image classifying Convolutional Neural Network (CNN) for classifying the flower image. Tensorflow, in addition to providing developers a simple way to build neural network layers, can also be run on mobile platforms such as Android. The ultimate goal of this project is to design and optimize a convolutional neural network for use with flower classification, and eventually build a simple classification app for mobile devices around the trained network. The mobile app will allow users to try and classify flowers while outdoors or offline [1].

1.2 MOTIVATION

There have a special motivation behind this research and project work. Me and my cousin used to walk along the railway line in afternoon. That time we used to observed a lot of unknown flower around the area. We were curious about that flowers but we could not recognize it. That time we discussed about an idea to make a system which can identify our desired flower automatically. Since then our journey has started and dream have come true now. We developed an android application which can introduce people about that unknown flower which they see but don't identify that. Hope our research & developed project will be helpful for the curious people.

1.3 RATIONALE OF THE STUDY

At the primary stage of our journey looked for research paper and tools related with our idea. Even we deeply searched the Internet to find if there any system that can identify flower automatically. But unfortunately we found very few resources. There exists a handful papers in there. That time we take this as a challenge and start our research to make a system which can recognize flower in real time. After a lot of hard work our project now come to light finally. Now our application can identify around 10 Bangladeshi common flowers with some foreign flowers also. The most unique feature of project is it can identify flower in real time. For some kind of flower it provides 100% accuracy rate. Different flower is same to look at we know. If this occur during identification time that time our application shows 3 or 2 most similar result. Convolutional Neural Network (CNN) and Image Classification technique used in our project which can identify flower with the confidence level from 0 to 1. We use Tensorflow an open source library for training process.

1.4 RESEARCH QUESTIONS

1. Can a machine identify flower before the human eye?
2. Can a system recognize different color of same flower using machine learning? Ex: Pink Rose / White Rose
3. Is convolutional neural network (CNN) technique is perfect to identify flower accurately?

1.5 EXPECTED OUTCOME

1. Identify flower in real time which include native Bangladeshi flowers along with some foreign flowers also.
2. Produce a convolutional neural network which is capable correctly classifying images of flowers with an average confidence level of 95% or more. Sometimes it provides 100% accuracy rate for some kind of flowers.
3. Collected a dataset of over 5000 images of flowers using their genus-species classification as the Google Image search term.

The following figure showing the output of the application which identify Rose with highest confidence value by using mobile camera.

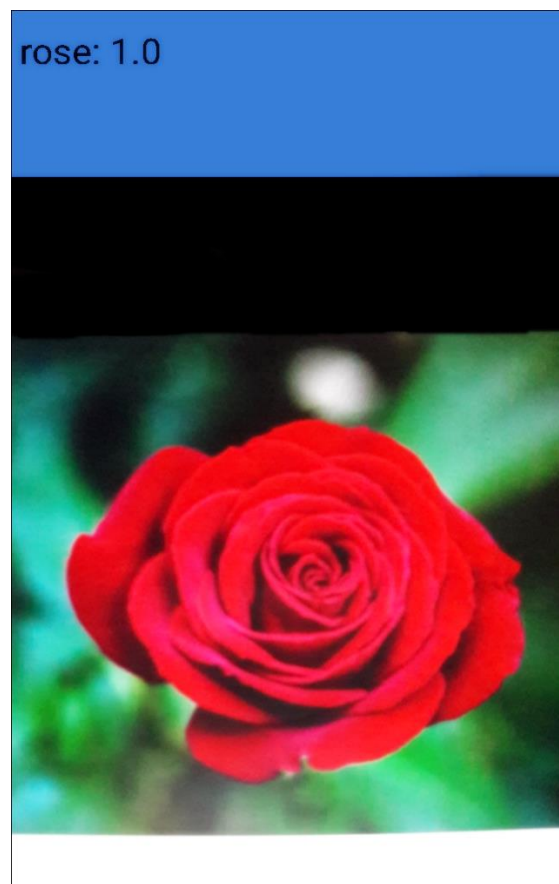


Figure 1.5 : Rose Identification with highest confidence value

CHAPTER 2

BACKGROUND

2.1 INTRODUCTION

Since the invention of computer, it has begun to change our daily life. It improves the quality of our lives. An impressive idea is to let a computer think and learn as like human being. Basically machine learning developed skills to develop the computer itself with the knowledge given. There are many applications of CNNs, one of the most popular of which is image classification, and CNNs have even been used specifically for flower classification. These flower classification neural networks, however generally use datasets with very specific details. Thus, this project is explorative and aimed at learning how to design a neural network using Tensorflow, but ultimately has practical applications for developers, botanists, or nature enthusiasts. The dataset used for training was collected through Google Images, and the images of the plants themselves are very general, rather than of specific characteristics of the plants, and typically include most of the identifying characteristics of a flower [3].

2.2 RELATED WORKS

Convolutional Neural Network (CNN) and Image Classification is the most used technique in machine learning. On the other hand Tensorflow an open source library developed by Google Brain team is the hot topic nowadays in the field of machine learning. There are some good publications of Tensorflow. But It is very little compared to the importance. In the following overview of the related works, we try to cover the works that are mostly related to ours to our best knowledge. Image Classification by Tensorflow is a relatively new topic. But the idea of reliable classification has made it increasingly important [1].

Recent history of modern software technology we can see lots of use of machine learning and image processing in our day to life. Like Police using face detection technology to identify criminal. Facebook using AI technology to match your face ID with any photo of you that has been uploaded. Pinterest use this technology to find similar photos. Traffic police also find out vehicles by using this technology. There is so many big projects on this topic, those are given below,

» PICTURETHIS



Figure 2.2 : PictureThis application feature graphic image

Among the popular software This is a Plant identification software developed by “Glory Software limited” running on both Android and IOS. This Software let you identify plant by taking a picture with 90% accuracy rate less than a second.

» PLANTS NAP



Figure 2.3 : PlantSnap application feature graphic image

This is also a popular software like PictureThis, developed by PlantSnap INC. This software let us to identify plants, flowers, cacti, mushrooms etc. by taking a picture of that plant. There are 585,000 species of plants and trees in their database. So their accuracy rate is 90% accurate.

2.3 RESEARCH SUMMARY

The purpose of this project is to use Tensorflow, an open-source dataflow and machine learning library, to build an image classifying Convolutional Neural Network (CNN) for classifying the flower image. Tensorflow, in addition to providing developers a simple way to build neural network layers, can also be run on mobile platforms such as Android. The ultimate goal of this project is to design and optimize a convolutional neural network for use with flower classification, and eventually build a simple classification app for mobile devices around the trained network. The mobile app will allow users to try and classify flowers while outdoors or offline. After examining, inquiring, researching and comparing all the related works on Flower Identification, we come to have different and many types of conclusion. First of all, none of them are using real time object detection and identification. All of them are just stuck into by taking picture of object and search through their database to identify the object. Secondly is accuracy rate, most of the big project hit the accuracy rate up to 90% so they say. But the community feedback shows the different color. Users complain about the miss detection of plants and less information about it. It also take long time for analyzing and searching their database. Because of our real time flower identification, it is more like instant detection of flower.

2.4 SCOPE OF THE PROJECT

Developers

The project is complete ready for practical use, someone who wishes to learn the basics of CNNs and Tensorflow, substituting their own classes. The project is essential for anyone that wants to improve their project. It have scope to improve both the dataset and the CNN architecture, however, this will require more research and knowledge of CNN design. Additionally, someone who is interested in Tensorflow and Android development or front end development can be benefited from this work.

Scientist or Botanist

This research and project will be helpful for botanists or biologists, although any scientist would likely wish to use their unique dataset. The project would most likely be useful to scientists for classifying large amounts of data and generating statistics for the dataset. Using our Android application Botanists or other scientists dealing with flowers to identify easily.

Traveler or Nature Enthusiasts

Our developed project will be much helpful while spending time outdoors to identify flowers by a mobile application. This application works completely in offline which use mobile camera to identify flowers. It will be really helpful for Traveler, Hiker and Nature Enthusiasts [1].

2.5 CHALLENGES

Duplicate flower identification is the main challenge of our project. Some flowers are same to look at by size and shape. That time it is so tough to provide the accurate result. We are continue our research to solve this problem.

Another challenge is to identify same flower of different specific color. Ex- White Rose, Pink Rose. This is a tough work to identify color using convolutional neural network. But we take it as a challenge.

Other challenge is achieving accurate data. In the market of false and duplicate data, it is very hard to get all of the things right. But we are trying are heart and soul to do better.

CHAPTER 3

RESEARCH METHODOLOGY

AND IMPLEMENTATION

3.1 BASIC INTRODUCTION TO MACHINE LEARNING

In Artificial Intelligence Learning is a very important feature. Many scientists tried to give a proper definition for learning. Many computer scientists, sociologists, logicians and others discussed about this for a long time. Some scientists think that learning is an adaptive skill that can perform the same process better later on (Simon 1987). Others claim that learning is a process of collecting knowledge (Feigenbaum 1977). Although there is no definite definition for learning skills, we still have to define machine learning. In general, machine learning has to be identified on how to improve the computer algorithm automatically through experience (Mitchell 1997).

Machine learning is one of the important field of Artificial Intelligence. At the beginning of development of Artificial Intelligence (AI), the system does not have a thorough learning ability so the whole system is not perfect. For instance when the computer faces problems, it can not be self-adjusting. Moreover, the computer cannot automatically collect and discover new knowledge. Therefore, computer only can conducted by already existing truths. It does not have the ability to discover a new logical theory, rules and so on [3].

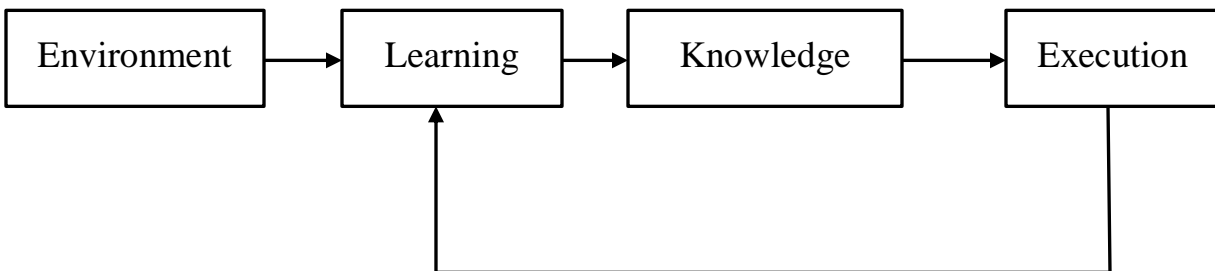


Figure 3.1: Learning System Structure

3.2 CONVOLUTIONAL NEURAL NETWORK

Convolutional neural networks are a class of machine learning networks which are commonly applied to image visualization problems such as classification. CNNs were inspired by the connections of the neurons and synapses in the brain. The design of these networks is made up of series of convolutional, pooling, and fully connected layers. The convolutional layer does what its name describes, it applies a number of convolutional filters to the input images in order to acquire the learning parameters for the network. Pooling layers are placed in between convolutional layers, and are used to reduce the number of parameters used for learning, and thus reduce the computation required. Finally, fully connected layers are full connections to the previous layer, rather than the small window the convolutional layers are connected to in the input. Convolutional neural networks are commonly used for image classification, however, there are limitations to this application. A human can identify the contents of certain images much more quickly than a computer, but CNNs have proven to have a 97.6% success rate when applied to facial recognition [11].

The following figure showing the typical design of convolutional neural network.

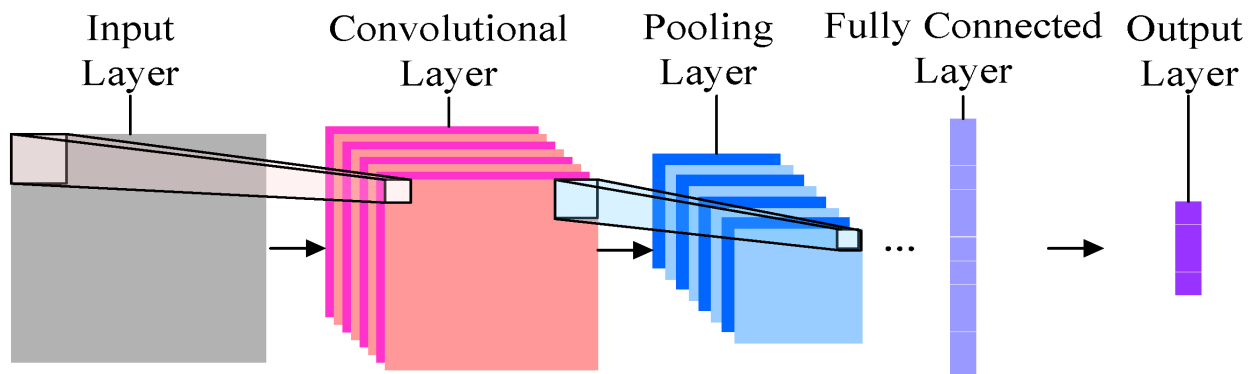


Figure 3.2: Typical CNN design

3.3 DESIGN OF DATASET

The diagram of dataset design which shows different step of data collection and dataset handler.

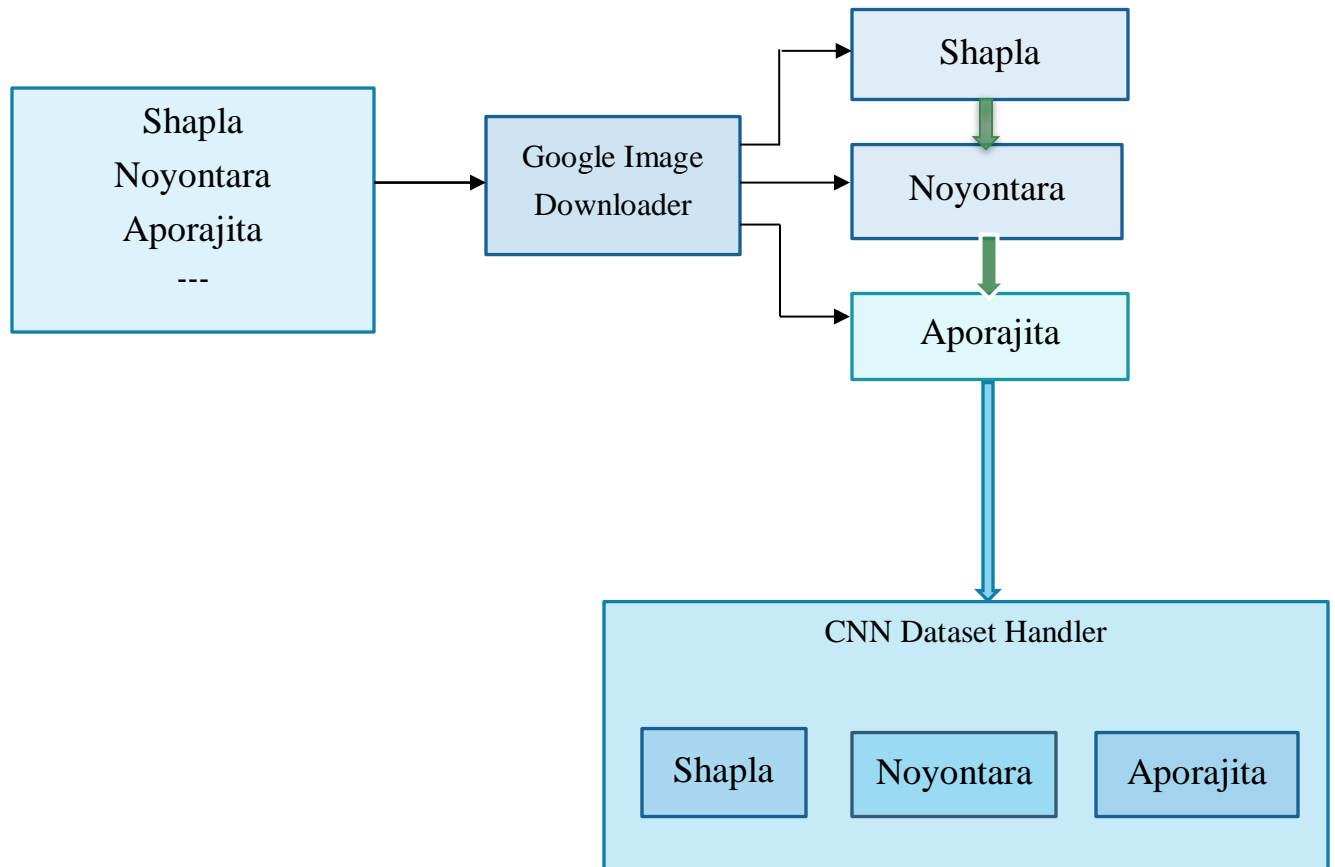


Figure 3.3: Design of Dataset

3.4 DATA COLLECTION PROCEDURE

There are a number of existing datasets which have images of specific flowers. These datasets were generally collected for very specific uses with neural networks that were designed to classify flowers based on certain characteristics. The dataset for this project was produced by searching Google Images using a Python script adapted from a web-crawler created by hardikvasa. As a result, the images of the flowers are a diverse collection of plants in their natural setting. This adds the benefit of training the network for use outdoors. The script had to be modified so that it would receive the list of search keywords from a text file. Google Image Downloader is also helpful to collect specific flower data on windows platform. The search keywords for the dataset are a long list of flower observations in Bangladesh. The search keywords use the full genus-species classification in order to increase the quality of the dataset. For example, rather than searching for the word “Gardenia”, which would produce undesired images, the preferred keyword would be “Gardenia jasminoides”. The labels for classification, however, are grouped by genus so that each class has 400 to over 1000 images.

The following figure shows the sample dataset image of Shapla flower.

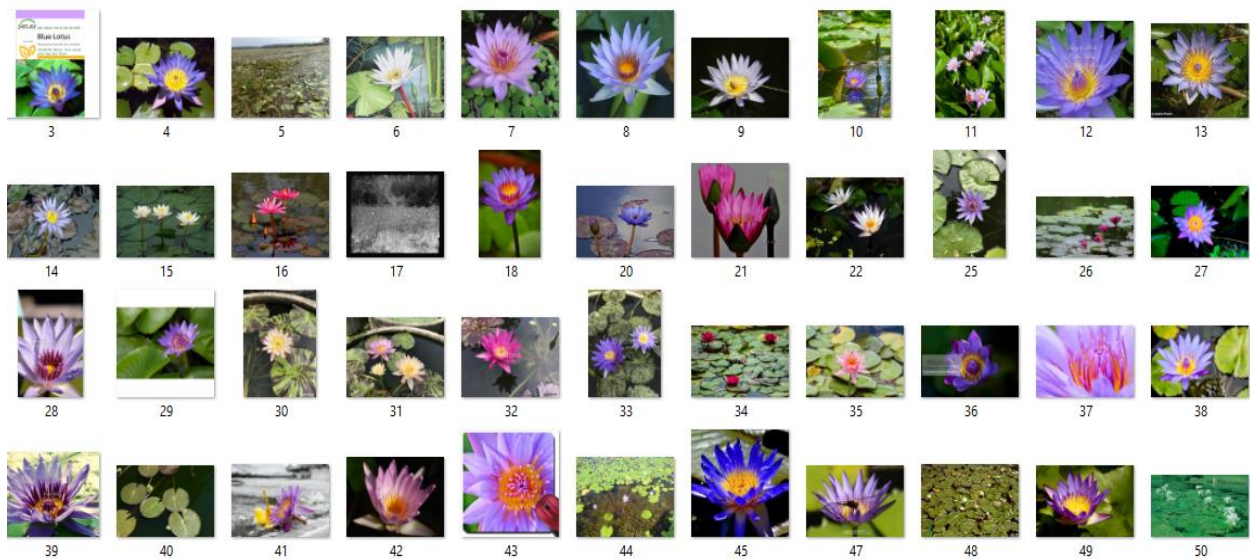


Figure 3.4: Dataset of Shapla

3.4 STATISTICS OF DATA

The following table contains the statistics of raw data

Flower Name	Amount of Training Image
Shapla	400
Noyon Tara	500
Tulip	800
Aporajita	400
Rose	500
Kadam	450
Daisy	700
Cosmos	350
Apple Flower	300
Rangan	300

Table 3.4 : Statistics of Raw Data

3.5 IMPLEMENTATION REQUIREMENTS (Minimum)

Software

- Windows / Linux / Mac OS
- Python pip package / Anaconda
- Tensorflow
- Android Studio

Hardware

- Intel i3 Processor
- 4 GB RAM
- 1TB Hard Disk

3.6 IMPLEMENTATION OF NETWORK DESIGN

Convolutional Layer 1

Input:

The image data is reduced to a size of 128x128 pixels in order to not overwhelm the hardware the program was normally tested on. Batches of 32 images are fed into the convolutional layer and 16 filters of 8x8 pixels are applied to the images.

```
conv_layer1 =  
lb . build_convolutional_layer ( input = image_placeholder ,  
num_channels = NUM_CHANNELS ,  
filter_size = FILTER_SIZE ,  
num_filters = NUM_FILTERS)
```

Pooling layer 1

Input:

Each pooling layer uses a pool size of 2x2 and a stride size of 2.

```
pool_layer1 = tf . layers . max_pooling2d ( inputs = conv_layer1 ,  
pool_size =[ 2 , 2 ], strides = 2)
```

Convolutional Layer 2

The second convolutional layer has the same parameters as the first.

```
conv_layer2 =  
lb . build_convolutional_layer ( input = pool_layer1 ,  
num_channels = NUM_FILTERS ,  
filter_size = FILTER_SIZE ,  
num_filters = NUM_FILTERS)
```

Pooling layer 2

Fully connected layer 1

Each fully connected layer performs an activation on each of its inputs. The first, however, performs a RELU activation function on the data.

```

connected_layer1 =
lb . create_connected_layer ( input = flat_layer,
num_inputs = flat_layer . get_shape ()[ 1 : 4 ]. num_elements (),
num_outputs = 32 ,
use_relu = True)

```

Fully connected layer 2

The second FC layer does not perform the RELU activation.

```

connected_layer2 = \
lb . create_connected_layer ( input = connected_layer1 ,
num_inputs = 32 ,
num_outputs = num_classes ,use_relu = False)

```

Training Step

The number of training steps can be specified as a command line parameter. Each training step is validated and tested, and the results of each step are printed to standard out.

```

is_last_step = (i + 1 == FLAGS.how_many_training_steps)
if (i % FLAGS.eval_step_interval) == 0 or is_last_step:
    train_accuracy, cross_entropy_value = sess.run(
        [evaluation_step, cross_entropy],
        feed_dict={bottleneck_input: train_bottlenecks,
                    ground_truth_input: train_ground_truth})
    tf.logging.info('%s: Step %d: Train accuracy = %.1f%%' %
                    (datetime.now(), i, train_accuracy * 100))
    tf.logging.info('%s: Step %d: Cross entropy = %f' %
                    (datetime.now(), i, cross_entropy_value))
    validation_bottlenecks, validation_ground_truth, _ = (
        get_random_cached_bottlenecks(
            sess, image_lists, FLAGS.validation_batch_size, 'validation',
            FLAGS.bottleneck_dir, FLAGS.image_dir, jpeg_data_tensor,
            decoded_image_tensor, resized_image_tensor, bottleneck_tensor,
            FLAGS.architecture))
    validation_summary, validation_accuracy = sess.run(

```



```
[merged, evaluation_step],
feed_dict={bottleneck_input: validation_bottlenecks,
ground_truth_input: validation_ground_truth})
validation_writer.add_summary(validation_summary, i)
tf.logging.info('%s: Step %d: Validation accuracy = %.1f%% (N=%d)' %
(datetime.now(), i, validation_accuracy * 100,
len (validation_bottlenecks)))
```

3.7 DESIGN DEVELOPMENT

The initial step of this project was to research the available machine learning libraries, convolutional neural network design, and collect datasets. Though there exist some other libraries, but Tensorflow was chosen because there are many tutorials and documentation for the library. After began to have a basic understanding of the Tensorflow library, we retrained the MobileNet model with our own dataset, which proved to be very successful in testing, however, the goal of this project was to learn how to develop and optimize a neural network. The initial designs of the CNN for this project were based on several different tutorials about how to use Tensorflow to design an image classifier. The next step was to begin modifying the initial network to try and find a design that worked for the application of this project. The next step was implementation of android app using Android Studio. Some open source code of Tensorflow collected from github which needed in our implementation work. Final Step was set up the pb file and label file with the application.

The design of CNN which we tested is being submit as a deliverable for this project. This model also was trained using the dataset at the genus-species level, which uses approximately 700 images average per class, which is almost enough. But we are working to add more images per class. This model is currently being trained and tested. The parameters of the network such as number of training steps, output directory, and image input directory can all be specified, however, their defaults will place all the output directory in the current working directory. The input image directory must be specified, and the contents of the directory must be folders of images in tf files folder. The Tensorflow trainer will generates two text files: one containing the labels for the classifier, and the other lists which images were selected for training, testing, and validation which called pb file. The classifier uses these to read result for each image classification and show the output result.

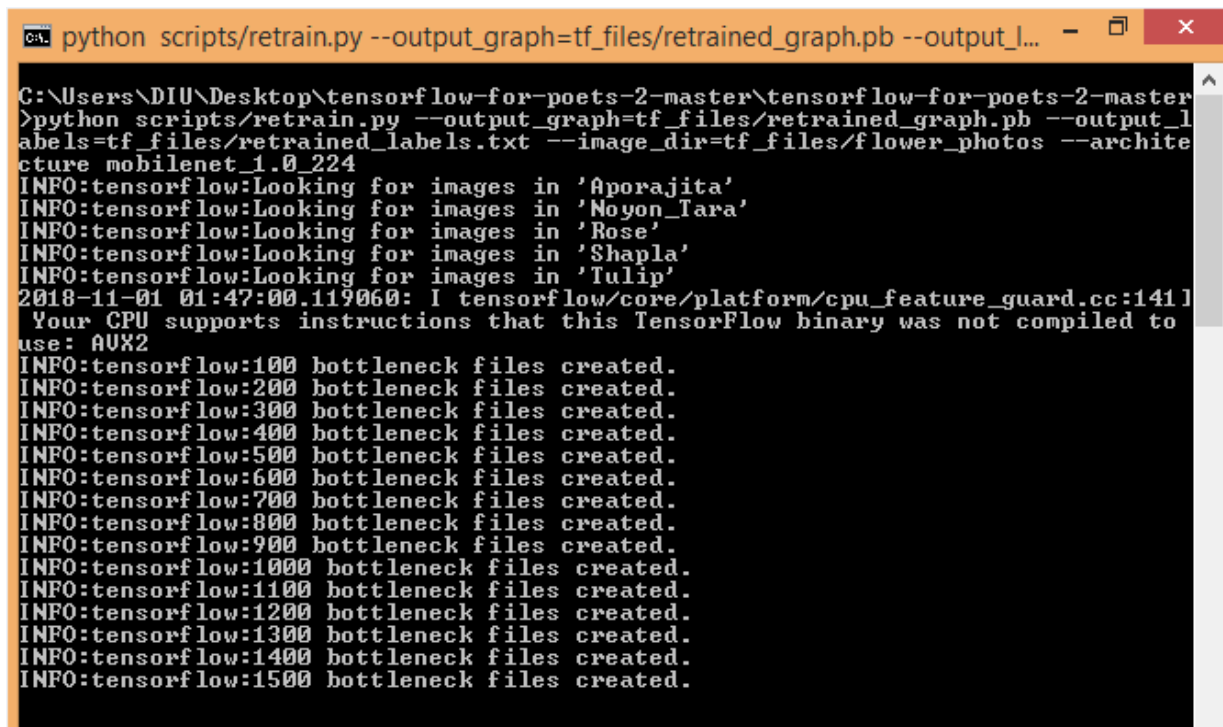
CHAPTER 4

EXPERIMENTAL RESULTS AND DISCUSSION

4.1 INTRODUCTION TO IMAGE CLASSIFICATION

The Python script which uses to the trained Tensorflow model is very simple, because most of the time spent on this project was for collecting the data and learning how to design a CNN. The classifier is designed to take a directory of images, a text file of the labels used in the network, and the trained model itself as inputs. The classifier tests the images with the specified model and displays the results comparing the correct label with the top four classes based on the confidence level of the predictions [1].

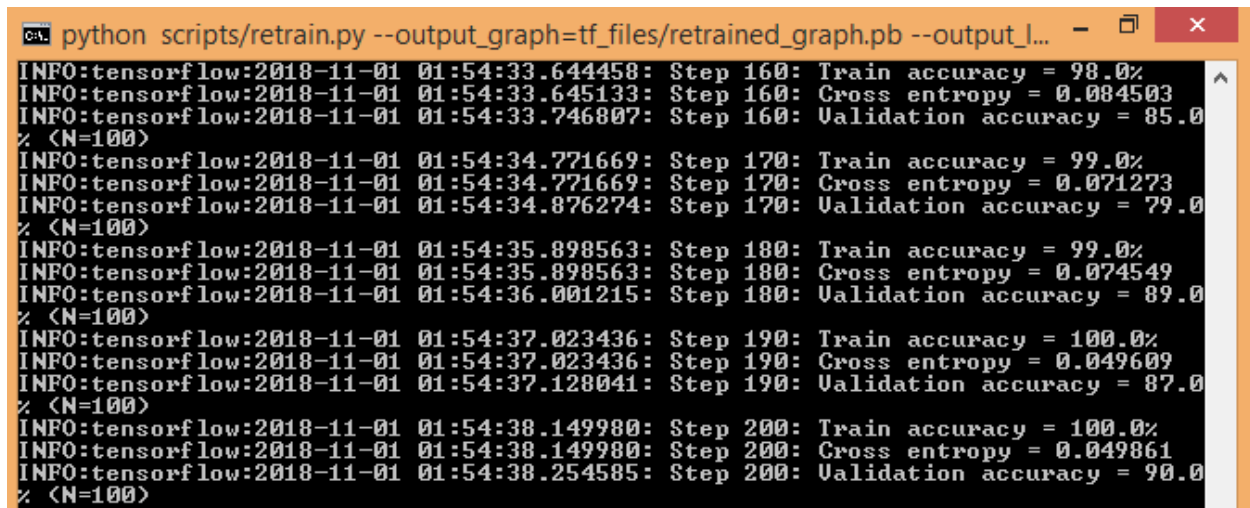
The following figure shows the pre processing step in Tensorflow which create label file and bottleneck file.



```
python scripts/retrain.py --output_graph=tf_files/retrained_graph.pb --output_l...
C:\Users\DIU\Desktop\tensorflow-for-poets-2-master\tensorflow-for-poets-2-master
>python scripts/retrain.py --output_graph=tf_files/retrained_graph.pb --output_l
labels=tf_files/retrained_labels.txt --image_dir=tf_files/flower_photos --archite
cture mobilenet_1.0_224
INFO:tensorflow:Looking for images in 'Aporajita'
INFO:tensorflow:Looking for images in 'Noyon_Tara'
INFO:tensorflow:Looking for images in 'Rose'
INFO:tensorflow:Looking for images in 'Shapla'
INFO:tensorflow:Looking for images in 'Tulip'
2018-11-01 01:47:00.119060: I tensorflow/core/platform/cpu_feature_guard.cc:141]
Your CPU supports instructions that this TensorFlow binary was not compiled to
use: AVX2
INFO:tensorflow:100 bottleneck files created.
INFO:tensorflow:200 bottleneck files created.
INFO:tensorflow:300 bottleneck files created.
INFO:tensorflow:400 bottleneck files created.
INFO:tensorflow:500 bottleneck files created.
INFO:tensorflow:600 bottleneck files created.
INFO:tensorflow:700 bottleneck files created.
INFO:tensorflow:800 bottleneck files created.
INFO:tensorflow:900 bottleneck files created.
INFO:tensorflow:1000 bottleneck files created.
INFO:tensorflow:1100 bottleneck files created.
INFO:tensorflow:1200 bottleneck files created.
INFO:tensorflow:1300 bottleneck files created.
INFO:tensorflow:1400 bottleneck files created.
INFO:tensorflow:1500 bottleneck files created.
```

Figure 4.1: Image pre processing by tensorflow

The following figure shows the training step which generates the train accuracy value, cross entropy and validation accuracy value.



```
python scripts/retrain.py --output_graph=tf_files/retrained_graph.pb --output_l...
INFO:tensorflow:2018-11-01 01:54:33.644458: Step 160: Train accuracy = 98.0%
INFO:tensorflow:2018-11-01 01:54:33.645133: Step 160: Cross entropy = 0.084503
INFO:tensorflow:2018-11-01 01:54:33.746807: Step 160: Validation accuracy = 85.0%
% (N=100)
INFO:tensorflow:2018-11-01 01:54:34.771669: Step 170: Train accuracy = 99.0%
INFO:tensorflow:2018-11-01 01:54:34.771669: Step 170: Cross entropy = 0.071273
INFO:tensorflow:2018-11-01 01:54:34.876274: Step 170: Validation accuracy = 79.0%
% (N=100)
INFO:tensorflow:2018-11-01 01:54:35.898563: Step 180: Train accuracy = 99.0%
INFO:tensorflow:2018-11-01 01:54:35.898563: Step 180: Cross entropy = 0.074549
INFO:tensorflow:2018-11-01 01:54:36.001215: Step 180: Validation accuracy = 89.0%
% (N=100)
INFO:tensorflow:2018-11-01 01:54:37.023436: Step 190: Train accuracy = 100.0%
INFO:tensorflow:2018-11-01 01:54:37.023436: Step 190: Cross entropy = 0.049609
INFO:tensorflow:2018-11-01 01:54:37.128041: Step 190: Validation accuracy = 87.0%
% (N=100)
INFO:tensorflow:2018-11-01 01:54:38.149980: Step 200: Train accuracy = 100.0%
INFO:tensorflow:2018-11-01 01:54:38.149980: Step 200: Cross entropy = 0.049861
INFO:tensorflow:2018-11-01 01:54:38.254585: Step 200: Validation accuracy = 90.0%
% (N=100)
```

Figure 4.2: Training Step

4.2 EXPERIMENT RESULTS

The results of this project is almost successful and has the potential for future improvements. Now the application can identify around 10 Local Flowers of Bangladesh along with some foreign flowers. We deeply focused to the accuracy rate. The accuracy rate depends on the amount of data. So that we use more images with different angel to improve the confidence level. Currently some flower identify with 100% confidence level. It is one of the success of our research and project. We used almost 700 images for per flowers for training step. The dataset contains around 5000 flower images. The CNN and the classifier are inconsistent, with some tests resulting in nearly 100% confidence during a correct classification, and other tests which entirely fail to produce a correct classification. Improvements which can be made are explained in detail later on.

The following figures showing the experiment results in real time by using mobile camera.

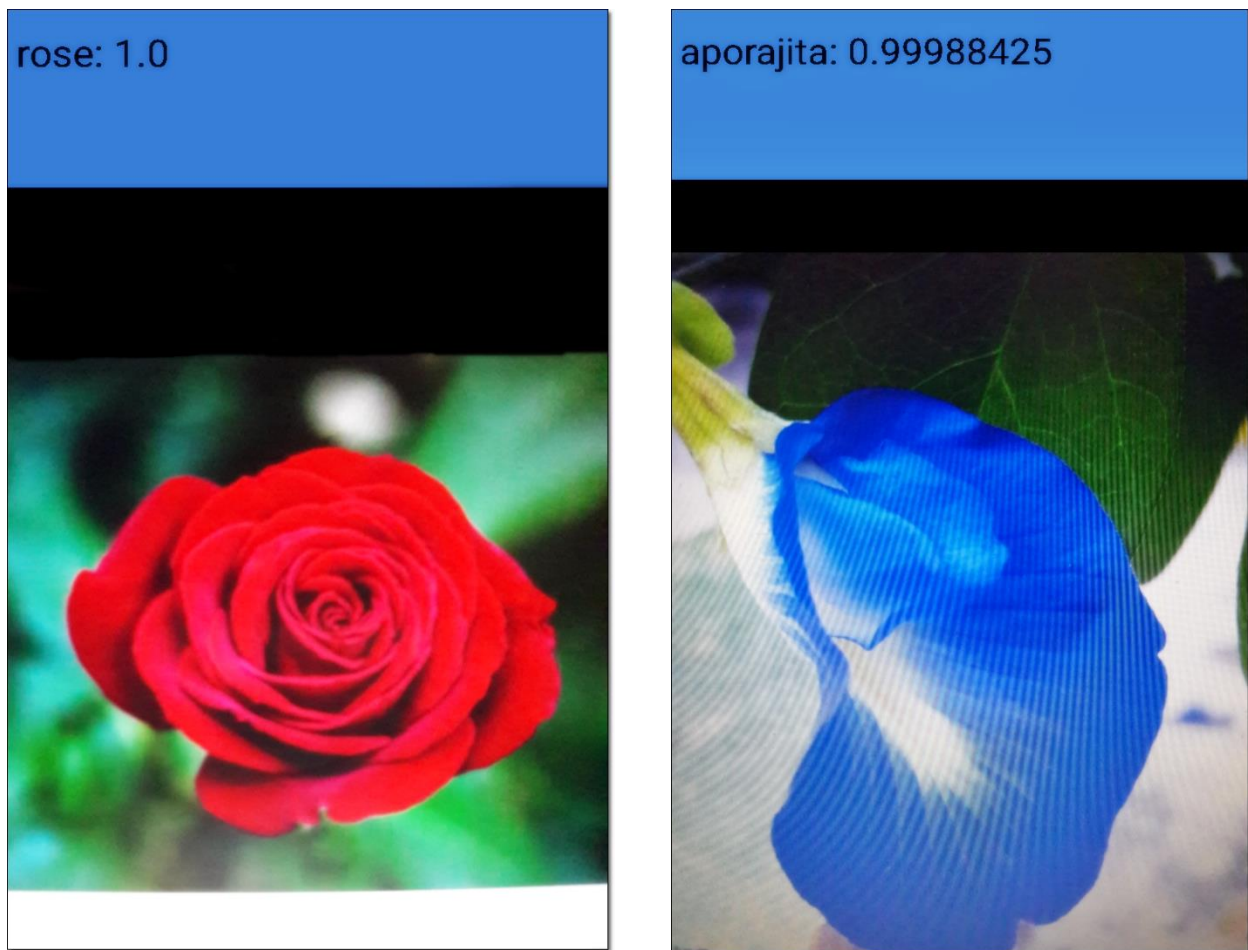


Figure 4.3: Experiment Result using Android Application

4.3 DESCRIPTIVE ANALYSIS

TENSORBOARD

TensorFlow Computation for training a massive deep neural network - can be complex and confusing. To make it easier to understand, debug, and optimize TensorFlow programs, there include a visualization tools called TensorBoard. TensorBoard visualizes the quantitative metrics of the plot, graph, plots and additional information such as passing the images [12].

There are some figures below which shows the histograms of tensorflow computations in tensorboard.

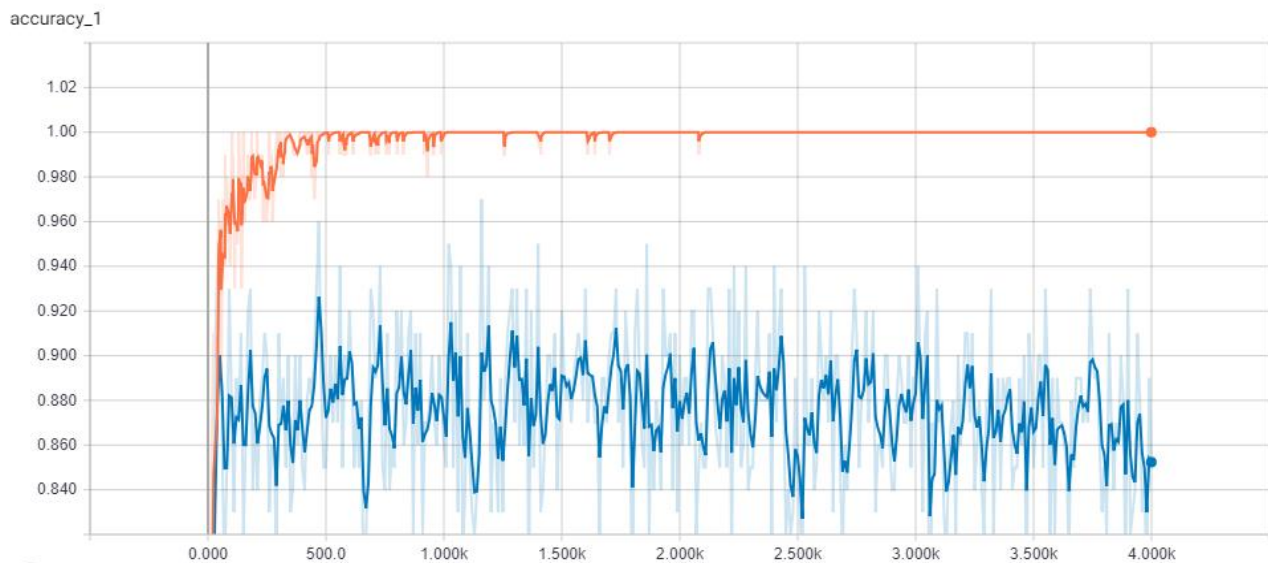


Figure 4.5: Train Accuracy Graph

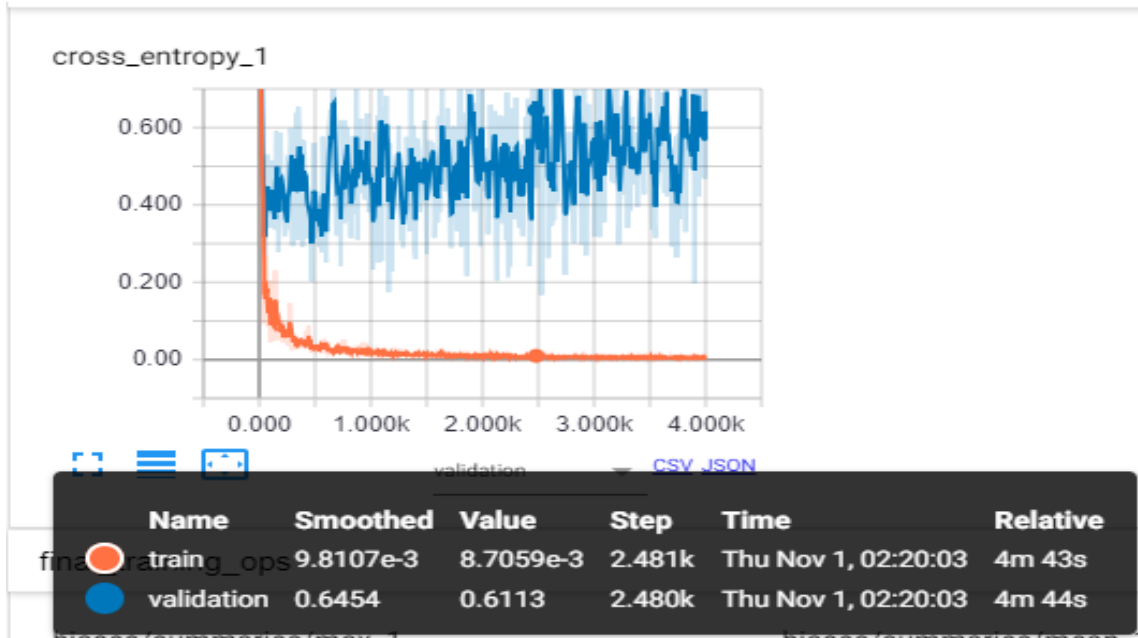


Figure 4.6 : Cross Entropy Graph

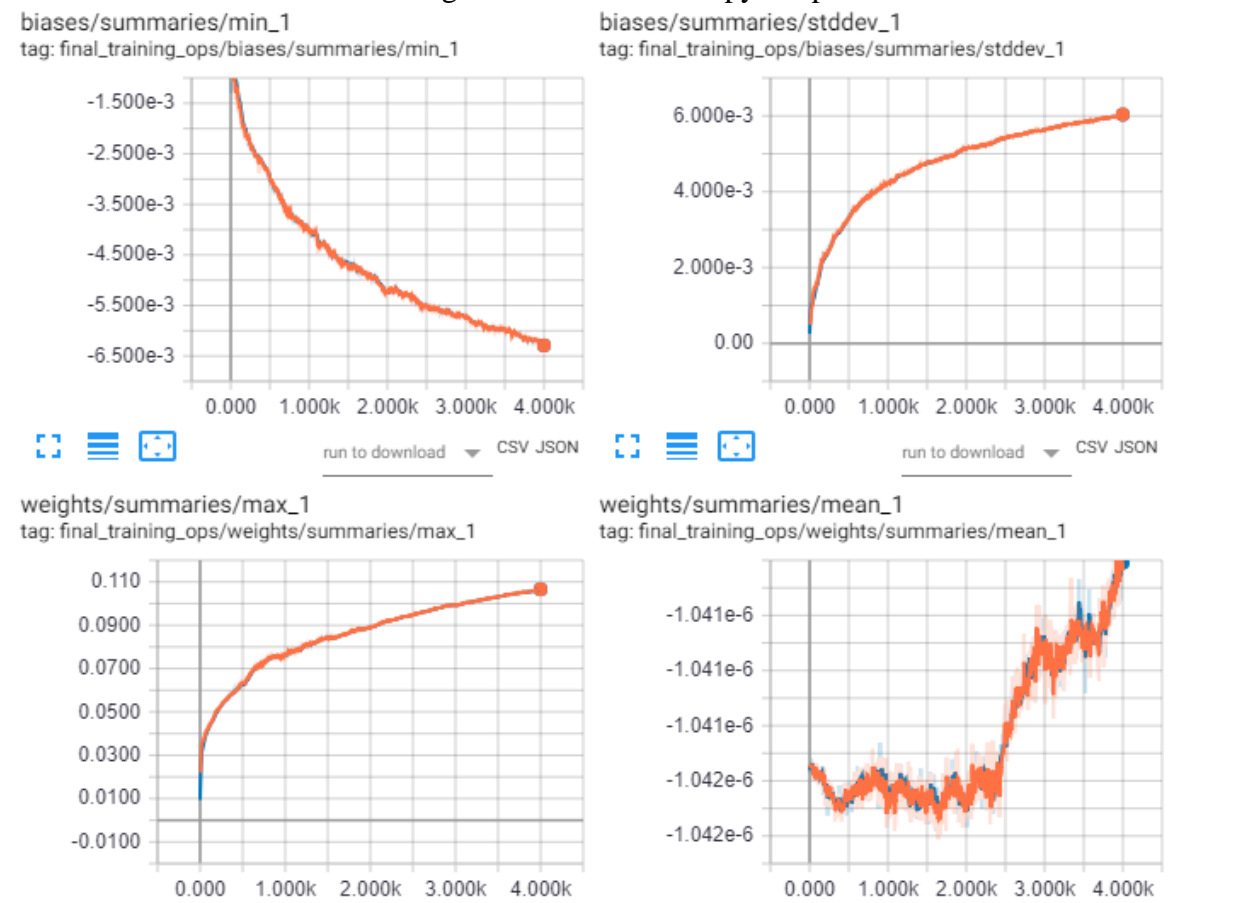


Figure 4.7: Final Training Summaries Graph

The following figures shows the activation histograms and summaries for train and validation data.



Figure 4.8: Activation Graph

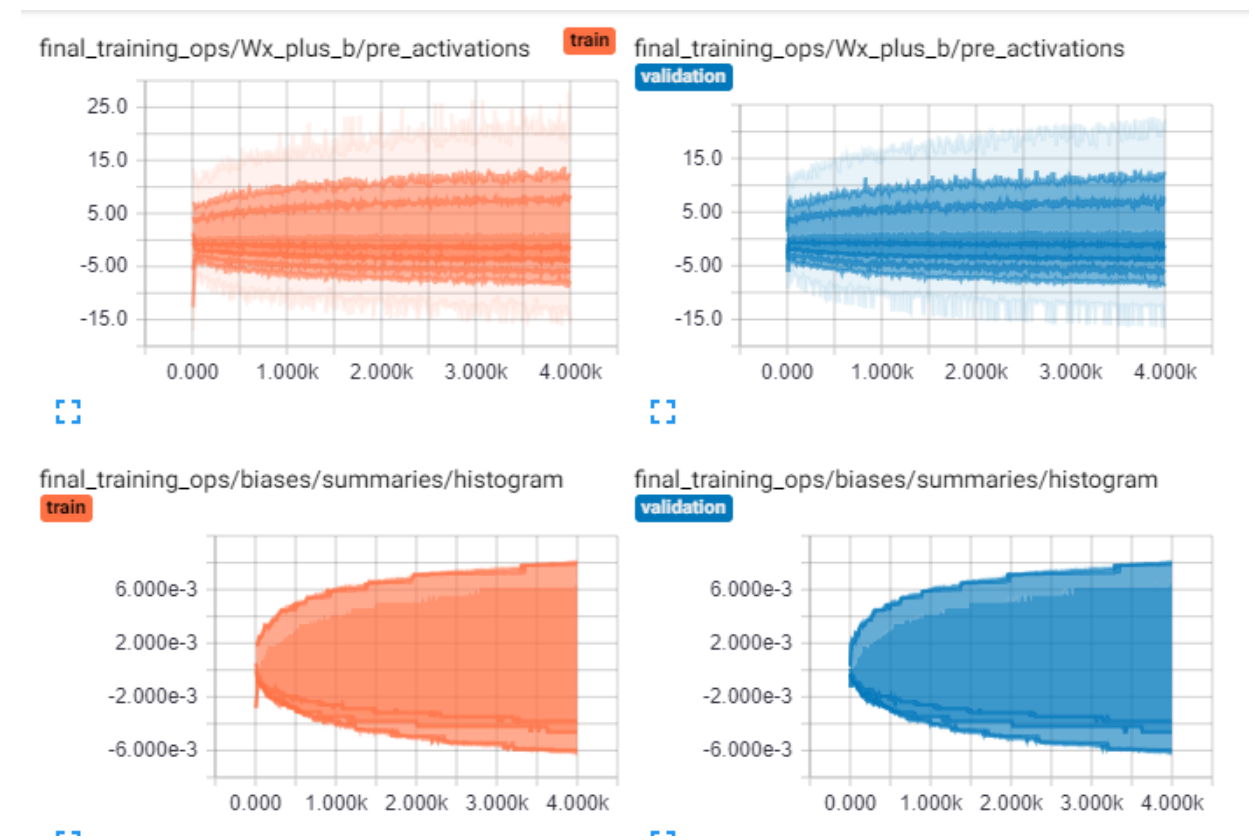


Figure 4.9: Final Training Summaries Histogram

The following figure shows the final training summaries histogram.

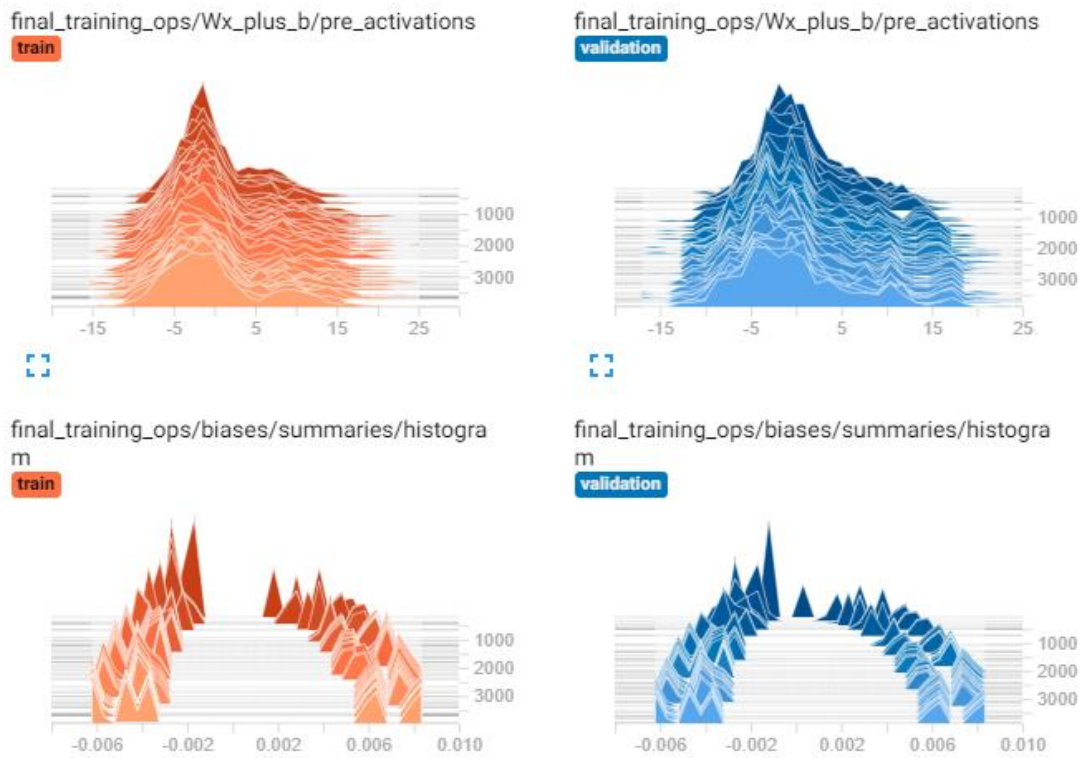


Figure 4.10: Final Training Summaries Histogram

	A	B	C
1	Wall time	Step	Value
2	1.54E+09	5	0.46
3	1.54E+09	15	0.85
4	1.54E+09	18	0.88
5	1.54E+09	24	0.91
6	1.54E+09	40	0.91
7	1.54E+09	42	0.95
8	1.54E+09	44	0.95
9	1.54E+09	45	0.97
10	1.54E+09	47	0.96
11	1.54E+09	53	0.96
12	1.54E+09	54	0.96
13	1.54E+09	56	0.89
14	1.54E+09	65	0.97
15	1.54E+09	71	0.94
16	1.54E+09	72	0.95
17	1.54E+09	73	0.99
18	1.54E+09	74	0.95
19	1.54E+09	79	0.98
20	1.54E+09	87	0.96
21	1.54E+09	94	0.94
22	1.54E+09	102	1
23	1.54E+09	105	0.97
24	1.54E+09	106	0.96

Figure 4.11 : Train Accuracy Value

	A	B	C
1	Wall time	Step	Value
2	1.54E+09	5	2.200136
3	1.54E+09	15	0.459741
4	1.54E+09	18	0.294452
5	1.54E+09	24	0.327483
6	1.54E+09	40	0.247127
7	1.54E+09	42	0.105551
8	1.54E+09	44	0.195391
9	1.54E+09	45	0.091672
10	1.54E+09	47	0.155764
11	1.54E+09	53	0.161887
12	1.54E+09	54	0.140675
13	1.54E+09	56	0.267811
14	1.54E+09	65	0.143229
15	1.54E+09	71	0.18873
16	1.54E+09	72	0.151813
17	1.54E+09	73	0.119465
18	1.54E+09	74	0.144008
19	1.54E+09	79	0.083966
20	1.54E+09	87	0.140156
21	1.54E+09	94	0.205061
22	1.54E+09	102	0.046524
23	1.54E+09	105	0.093426
24	1.54E+09	106	0.065484

Figure 4.12: Cross Entropy Value

4.4 EXPERIMENT SUMMARY

Our developed application can identify local Bangladeshi flowers with some foreign flowers also. We deeply focused on the accuracy level. The accuracy rate depends on the amount of data. So that we use more images with different angle to improve the confidence level. Currently some flower identify with 100% confidence level. It is one of the success of our research and project. We used almost 700 images for per flowers for training step. The dataset contains around 5000 flower images.

CHAPTER 5

5.1 SUMMARY OF THE STUDY

Flower is a very important part of nature. Mostly we identify a plant through its flower. Experienced botanists do this identification of flower but a naive person will have to consult flower guidebooks or browse any relevant web pages on the Internet through keywords searching. This is a system that recognizes the flower in real time using mobile camera. Presently this Android app can identify around 10 flowers. Most important thing that this app fully works in offline. We are continuously working to add more flowers to identify. Everyday we see a huge number of flower species in our house, parks, roadsides, in farms, on our rooftop but we have no knowledge of that flower species or their origin. Even we have no idea about its name. There are several guidebooks for flowers knowledge but it becomes quite difficult to find the name when have the picture. Even the Internet sometimes is not useful. But it is quite difficult for human brain to memorize all the species they see. Even some flower is similar to look at. This software recognizes the flower in real time by using mobile camera [5].

The purpose of this project is to use Tensorflow, an open-source dataflow and machine learning library, to build an image classifying Convolutional Neural Network (CNN) for classifying the flower image. Tensorflow, in addition to providing developers a simple way to build neural network layers, can also be run on mobile platforms such as Android. The ultimate goal of our project is to design and optimize a convolutional neural network for use with flower classification, and eventually build a simple classification app for mobile devices around the trained network. The mobile app will allow users to try and classify plants while outdoors or offline [12].

5.2 FUTURE WORK

This project has plenty of room for future work, by myself or a future interested student.

1. Improved CNN design. There much more research and practice is needed to optimize the design.
2. Improve the dataset and add more data.
3. Specific Identification of duplicate flower which is same to look at.
4. We have plan to release this app on Google Playstore.

5.3 CONCLUSIONS

With the rapid development of technology, AI is being used in various fields. Machine learning is the most basic method to achieve AI. This research describes the work principle of machine learning and an application of machine learning. At the beginning of development of Artificial Intelligence (AI), the AI system does not have a thorough learning ability so the whole system is not perfect. For instance when the computer faces problems, it can not be self-adjusting. Moreover, the computer cannot automatically collect and discover new knowledge. Therefore, computer only can conducted by already existing truths. It does not have the ability to discover a new logical theory, rules and so on. The ultimate goal of this project is to design and optimize a convolutional neural network for use with flower classification, and eventually build a simple classification app for mobile devices around the trained network. The mobile app will allow users to try and classify plants while outdoors or offline. We will continue our research to make the system more efficient.

REFERENCES

- [1] Kody G. Dangtongdee and Dr. Franz Kurfess, "Plant Identification Using Tensorflow, ". Computer Engineering Department, California Polytechnic State University, San Luis Obispo, Senior Project Final Report. CPE 462, Winter 2018, pp. 5-15.
- [2] Xuanxin Liu, Fu Xu, Yu Sun, Haiyan Zhang, and Zhibo Chen, "Convolutional Recurrent Neural Networks for Observation-Centered Plant Identification," Journal of Electrical and Computer Engineering, vol. 2018, pp. 1–7, 2018.
- [3] Yu Yang and Patric Granholm, "A study of pattern recognition of Iris flower based on Machine Learning, ". Turku University of Applied Sciences, Turku Finland , Bachelor's Thesis, 2013.
- [4] Wang-Su Jeon and Sang-Yong Rhee, "Plant Leaf Recognition Using a Convolution Neural Network, ". International Journal of Fuzzy Logic and Intelligent Systems Vol. 17, No. 1, March 2017, pp. 26-34.
- [5] Er. Gurleen Kaur Marwa and Er. Salam Din, "Development of Software for Flower Recognition Using Image Processing Technique, ". Dept. of SEEIT, COAET, PAU, Ludhiana, India, vol. 3, Issue 5, May 2015.
- [6] Vijaya Bylaiah, "Leaf Recognition and Matching with MATLAB, ". A Thesis Presented to the Faculty of San Diego State University, Spring 2014.
- [7] Y. Li, Q. Zhu, Y. Cao, and C. Wang, "A leaf vein extraction method based on snakes technique," in Proceedings of the International Conference on Neural Networks and Brain (ICNN&B '05), pp. 885–888, 2005.
- [8] Bache, K.& Lichman, M. 2013. UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml>]. Irvine, CA: University of California, School of Information and Computer Science.
- [9] Bishop, C. 2006. Pattern Recognition and Machine Learning. New York:Springer, pp.424-428.
- [10] Robin. 2010. Machine-Learning - Artificial Intelligence. Available at: <http://intelligence.worldofcomputing.net/category/machine-learning>. Consulted 22 Nov 2018.
- [11] "Convolutional Neural Network." Wikipedia, Wikimedia Foundation, Available: en.wikipedia.org/wiki/Convolutional_neural_network. Accessed on: Nov. 2, 2018.
- [12] "TensorFlow." Wikipedia, Wikimedia Foundation, Available:en.wikipedia.org/wiki/TensorFlow. Accessed on: Nov. 2, 2018.

APPENDICES

Accuracy Value For Some Training Steps.

Number	Step	Value
1	5	0.46
2	15	0.85
3	18	0.88
4	24	0.91
5	40	0.91
6	42	0.95
7	44	0.95
8	45	0.97
9	47	0.96
10	53	0.96
11	54	0.96
12	56	0.89
13	65	0.97
14	71	0.94
15	72	0.95
16	73	0.99
17	74	0.95
18	79	0.98
19	87	0.96
20	94	0.94
21	102	1
22	105	0.97
23	106	0.99
24	113	0.93
25	118	0.96
26	126	0.95
27	129	0.98
28	130	1

29	139	0.97
30	141	0.98
31	142	0.93
32	150	1
33	155	0.96
34	158	0.97
35	167	0.98
36	168	0.99
37	177	0.97
38	178	0.97
39	183	0.99
40	185	1
41	186	0.99
42	190	0.99
43	198	0.97
44	206	0.98
45	207	0.98
46	208	1
47	212	0.99
48	223	0.98
49	229	0.99
50	234	0.96
51	237	0.98
52	249	0.96
53	255	0.97
54	260	1
55	263	0.98
56	267	0.99
57	271	0.98
58	273	0.96
59	292	1
60	297	1

61	298	0.99
62	303	1
63	307	0.99
64	308	1
65	310	0.98
66	318	0.99
67	319	0.98
68	325	1
69	326	0.99
70	327	1
71	330	1
72	339	1
73	347	1
74	363	0.99
75	367	0.99
76	373	0.99
77	377	0.99
78	379	0.99
79	391	1
80	392	1
81	408	1
82	421	0.99
83	422	1
84	424	0.99
85	430	1
86	439	1
87	441	0.98
88	442	0.99
89	445	1
90	451	0.97