**FACIAL EXPRESSION RECOGNITION**

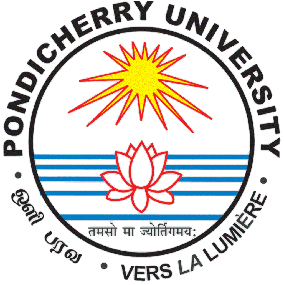
By

**ARPIT KUMAR**

**(Reg. No. 15352116)**

**Project report submitted in partial fulfillment of the requirements for the award of the degree of**

**MASTER OF COMPUTER APPLICATIONS**



**DEPARTMENT OF COMPUTER SCIENCE**

**SCHOOL OF ENGINEERING AND TECHNOLOGY**

**PONDICHERRY UNIVERSITY**

**PUDUCHERRY-605014**

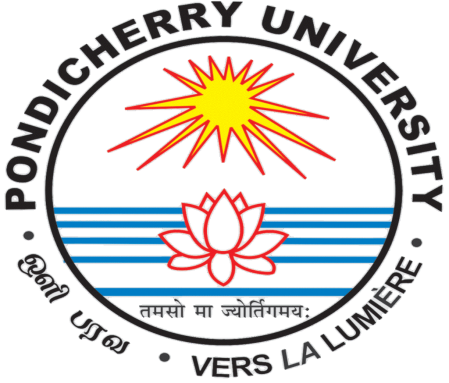
**MAY 2018**

**PONDICHERRY UNIVERSITY**

**(A Central university)**

**R.V. Nagar, Kalapet**

**PUDUCHERRY-605014**



**SCHOOL OF ENGINEERING AND TECHNOLOGY**

**DEPARTMENT OF COMPUTER SCIENCE**

**Master of Computer Applications**

**Bonafide Certificate**

*Certified that this is a bonafide report of project work done by* **Arpit Kumar** *with register number* **15352116** *of M.C.A in semester* **VI** *during the year 2017-2018.*

**Guide Head of Department**

*Submitted for the viva-voce examination held on 02nd May, 2018 at the Department of Computer Science, Pondicherry University, Puducherry-14.*

**INTERNAL EXAMINER** **EXTERNAL EXAMINER**

**To Whomsoever It May Concern**

It is to notify that **Arpit Kumar**, Reg. no. **15352116**, MCA 3rd Year, is doing a project titled “Facial Expression Recognition” under me as a part of his academic curriculum of the final semester.

Further, his attendance percentage as per 27th April 2018 is calculated to be

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

I mark his performance during entire project as: Bad / Average / Good

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Dr. K. Suresh Joseph**

Assistant Professor,

Dept. of Computer Science

Pondicherry University

**TABLE OF CONTENTS**

**TITLE PAGE NO.**

**SYNOPSIS 7**

**1. INTRODUCTION 8-9**

1.1. About the Project

1.2. Plan of the Report

**2. PROBLEM DEFINITION AND FEASIBILITY ANALYSIS 10-11**

2.1. Problem Definition

2.2. Feasibility Analysis

2.2.1. Economic Feasibility

2.2.2.Technical Feasibility

2.2.3. Time Feasibility

2.3 Facial Expression Analysis

**3. SOFTWARE REQUIREMENTS SPECIFICATION 12**

3.1. Introduction

3.2. Functional and Non-Functional Requirements

3.2.1 Functional Requirements

3.2.2 Non-Functional Requirements

3.3. Hardware Requirements

3.4 Software Requirements

**4. SYSTEM DESIGN 13-14**

4.1. Introduction

4.2. Flow-Chart Diagram

4.3. Data-Flow Diagram

4.4. Interface

4.5 Search Algorithm (K- Nearest Neighbor)

**5. SYSTEM IMPLEMENTATION AND TESTING 15 - 20**

5.1. System Implementation Details

5.2. Algorithms

5.2.1 K-Means Clustering

5.2.2 SVM

5.2.3 CNN (Convolutional Neural Networks)

5.2.3.1 Dropout Layer

5.2.3.2 Max Pooling Layer

5.2.3.3 Activation Layer

5.2.3.4 Convolutional 2D Layer

5.2.3.5 Dense Layer

5.2.3.6 Flatten Layer

5.2.3.7 Batch Normalization Layer

5.2.4 K- Nearest Neighbour

5.3 System Testing

**6. CONCLUSION AND FORSEEABLE ENHANCEMENTS 21**

6.1 Conclusions

6.2 Future Enhancements

**APPENDIX A- SCREENSHOTS 22-26**

**APPENDIX B- SYSTEM CODE 27-28**

**ACKNOWLEDGEMENTS**

The Satisfaction and Euphoria that accompany the successful completion of any task would be incomplete without mention of the people who made it possible, whose constant guidance and encouragement that had made our effort a success.

I would like to express my sincere and profound gratitude to **Dr. K. Suresh Joseph** my supervisor whose valued guidance, knowledge, encouragement, enthusiasm, and support enabled me to do extensive work, without which this project would have been for less comprehensive.

I am extremely grateful to **Dr. T. Chithralekha, Head of the Department of Computer Science** for her continuous support and to other faculties in the department who provided me valuable suggestion which accounted for the completion of the project in the provided time.

I would like to extend my gratitude to the **faculties** of this department who provided me a platform to learn various things and giving me the courage to try different things and go forward with the ideas that I had.

Last but not the least I would like me to thank **Parents** who stood by me in all the circumstances.

**With true regards,**

**Arpit Kumar**

**Master of Computer Applications**

**SYNOPSIS**

The problem of automatic recognition of facial expressions is still an ongoing research, and it relies on advancements in image processing and computer vision techniques. Such systems have a variety of interesting applications from human-computer interaction to robotics and computer animations. Their aim is to provide robustness and high accuracy, and adapt to real time scenarios.

This paper proposes an automatic facial expression recognition system, capable of distinguishing the six universal emotions: Neutral, Happy, Sad, Angry, Disgust and Surprise. It is designed to be person independent and tailored only for static images. The system integrates a face detection mechanism using K-means, CNN, KNN algorithms uses uniform Local Binary Patterns for feature extraction and performs classification using multiclass Support Vector Machine (SVM) model.

The most information for machine perception of emotions is through facial expressions. Effective human-computer intelligent interaction needs the computer to detect emotions through facial expression. This project aims to develop automatic emotion detecting system by evaluating machine learning algorithms for facial expression recognition. It successfully detection of facial features including eyes, eyebrows, nose and mouth. The program can also able to find motion distribution of different facial features and sends back image fusion facial features shaded with colors.

**CHAPTER 1**

**INTRODUCTION**

Facial expressions are important cues for non verbal communication among human beings. This is only possible because humans are able to recognize emotions quite accurately and efficiently. An automatic facial emotion recognition system is an important component in human machine interaction. Apart from the commercial uses of automatic facial emotion recognition system it might be useful to incorporate some cues from the biological system in the model and use the model to develop further insights into the cognitive processing of our brain.

In facial emotion recognition use Active Appearance Models (AAMs), FACS (Facial Action Coding System) labels or some other sophisticated feature extraction scheme. AAMs can be learned from a set of training images and can be fitted on a new face to generate the landmark positions which can further be used to design features. Thus, in an automatic setting either the availability of landmark point on face images is assumed or can be obtained by fitting the model. Trying to interpret a person's emotional state in a nonverbal form, usually requires decoding his/hers facial expression. Many times, body languages and especially facial expressions, tell us more than words about one's state of mind.

For this project I have performed an experiment which serves multiple purposes:

1. Finding out, once and for all, who "reads" facial expressions better- Men or Women, and if so, suggesting an answer for the question- why do those differences exist?
2. Revealing special features for recognizing classically defined facial expressions and answering the question- which facial cues help us the most decipher facial expressions?

**1.1 About the Project**

Understanding the human facial expressions and the study of expressions has many aspects, from computer analysis, emotion recognition, lie detectors, airport security, choose the music depending on the mood, nonverbal communication and even the role of expressions in art.

A facial expression is a gesture executed with the facial muscles, which convey the emotional state of the subject to observers. An expression sends a message about a person's internal feeling. Although human developed a very wide range and powerful of verbal languages, facial expression role in interactions remains essential, and sometimes even critical. Expressions and emotions go hand in hand, i.e. special combinations of face muscular actions reflect a particular emotion. For certain emotions, it is very hard, and maybe even impossible, to avoid it's fitting facial expression.

**1.2 Plan of the Report**

The main objective of this work is the integration and optimization of an automatic face detection and recognition system for video indexing application. The recognition stage is based on the Principal Component Analysis (PCA) approach.

**Chapter 1 Introduction**

An introduction has been given about the project which containing information about the application developed.

**Chapter 2 Problem Definition and Feasibility Analysis**

This part deals with the problem definition and how feasible the developed application is in terms of technique, operation, economy and time. Both of the important aspects has been covered in the second chapter.

**Chapter 3 Software Requirement Specification**

This Chapter includes information about various requirements for this application so developed. Mostly, it includes the details of the hardware requirements and software requirements.

**Chapter 4 System Design**

This Chapter includes information about the algorithm used to implement such as K-Means clustering, CNN and K- Nearest Neighbour algorithm. It also includes the usage of Data-Flow-Diagrams to give a better understanding about the project.

**Chapter 5 System Implementation and Testing**

This chapter contains the details about the technologies used for implementing the application as well as different testing criteria.

**Chapter 6 Conclusion and Foreseeable Enhancement**

This chapter concludes the project and gives some suggestions for the future enhancements which can be done to make the application work better.

**CHAPTER 2**

**PROBLEM DEFINITION AND FEASIBILITY ANALYSIS**

**2.1 Problem Definition**

To make this project, the problem occurs to take the data as an input using the opencv library. Developing such Facial Expression Recognition system is not trivial task, due to the high variability of data. Images are represented under various conditions such as resolution, quality, illumination or size. When we were using tensor flow which is implemented in machine learning then the program shown error again and again. All these constraints have to be taken into consideration for selecting appropriate methods, in order to deliver a system that is robust, person independent and that ideally works in real time scenarios.

**2.2 Feasibility Analysis**

This paper proposes a system capable of performing automatic recognition of six emotions, considered to be universal across cultures: disgust, anger, fear, happiness, sadness and surprise. Such system would analyze the image of a face and produce a calculated prediction of the expression. To integrate a module for face detection by using the algorithms KNN (Nearest Neighbour). The system extract discriminant features with a method called Local Binary Patterns. Lastly, the solution performs expression classification by incorporating the widely used Machine Learning model called Support Vector Machines, which is trained on a standard dataset of examples.

Three key considerations involved in the feasibility analysis are:

1. Economic Feasibility

2. Technical Feasibility

3. Time Feasibility

**2.2.1 Economic Feasibility**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. The application does not require enormous amount of money to be developed. This can be done economically if planned judicially, so it is economically feasible. The cost of project depends upon the number of man-hour required.

**2.2.2 Technical Feasibility**

This study is carried out to check the technical feasibility, that is, the technical

requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

1. Processor -- Intel(R) core i3-2.30GHz

2. Hard Disk -- 500 GB

3. RAM -- 6 GB

4. Monitor -- 15’’ 2’ colour

**2.2.3 Time Feasibility**

Time evaluation is the most important consideration in the development of project. The time

schedule required for the developed of this project is very important since more development time effect machine time, cost and cause delay in the development of other systems. A reliable Secure Data Sharing in cloud can be developed in the considerable amount of time.

**2.3 Facial Expressions Analysis**

The ﬁrst step in any facial expressions analysis system is to recognize facial expressions, and facial expression recognition is a fairly mature domain in computer vision with techniques that boast a high level of accuracy and robustness.

**CHAPTER 3**

**SOFTWARE REQIREMENTS SPECIFICATION**

**3.1 Introduction**

It is very important to grasp the scope of the system, what is the core functionality. The elicitation step involves gathering clear and precise requirements, in order to model the system and its characteristics, a process that can be very complex in software development. Due to the fact that this project is mainly focused on research and less on providing a user oriented tool, it only uses the main techniques for analyzing the system’s requirements.

**3.2 Functional and Non-Functional Requirements**

When collecting and analyzing the requirements of software, there are two aspects that need to be considered. The functional one refers to the features that the system needs to deliver, while the non-functional aspect takes into account constraints and how the system should behave.

**3.2.1 Functional Requirements**

1. The system should classify an image into one of 6 emotions.

2. The system should include an automatic face detection algorithm.

3. The system should include techniques for extraction of meaningful facial features.

4. The system should deliver a trained classifier.

5. The system should deliver a simple GUI.

**3.2.2 Non- Functional Requirements**

1. The system should be implemented in Tensorflow.

2. The system’s GUI should be simple and clear.

**3.3 Hardware Requirements**

System : Intel Core i3 2.30 GHz

Hard Disk : 500GB

RAM : 2 GB

**3.4 Software Requirements**

Operating System : Windows, Linux (any)

Coding Language : Python

Editor : Sublime

Libraries : Opencv – Image Processing

Numpy- Matrix Manipulation

Pandas – Data Analysis

Tensorflow- Machine Learning

Keras- Deep Learning

**CHAPTER 4**

**SYSTEM DESIGN**

**4.1 Introduction**

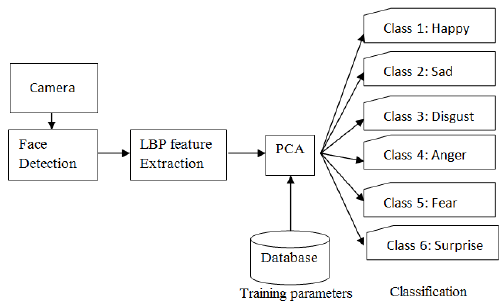
**Opencv**

Opencv is a cross-platform library using which we can develop real-time computer vision applications. It mainly focuses on image processing, video capture and analysis including features like face detection and object detection. Computer Vision can be defined as a discipline that explains how to reconstruct, interrupt, and understand a 3D scene from its 2D images, in terms of the properties of the structure present in the scene. It deals with modeling and replicating human vision using computer software and hardware.

Computer Vision overlaps significantly with the following fields −

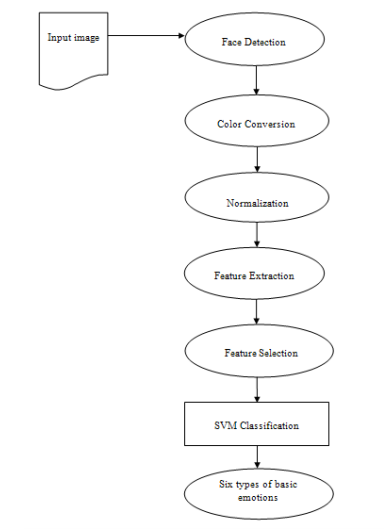
* **Image Processing** − It focuses on image manipulation.
* **Pattern Recognition** − It explains various techniques to classify patterns.

**4.2 Flowchart**



**4.3 Data-Flow-Diagram**

Here, I am showing how the facial expression system works logically with the acquired real world data.



**4.4 Interface**

I am not providing any GUI here, but using Sublime editor. The interface receives input from the user at real time and provide the output as emotion.

**4.5 Search Algorithm (K-Nearest Neighbour)**

The k-nearest neighbours algorithm is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether KNN is used for classification or regression.

1. In KNN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors. If k=1, then the object is simply assigned to the class of that single nearest neighbor.

2. In KNN regression, the output is property value for the object. This value is the average of the values of its k nearest neighbors.

**CHAPTER 5**

**SYSTEM IMPLEMENTATION AND TESTING**

**5.1 System Implementation Details**

The entire system was developed in a high level language and scientific environment. Its capabilities are enhanced by using the integration with Opencv, a library functions mainly aimed towards the computer vision usage. Used Numpy for numerical calculation, Pandas for the data analysis and Tensorflow libraries for fast numerical computing.

**5.2 Algorithms**

**5.2.1 K-Means Clustering**

**5.2.2 SVM**

**5.2.3 CNN (Convolutional Neural Networks)**

**5.2.4 K-Nearest Neighbour**

**5.2.1 K-Means Clustering**

K-means clustering is a type of unsupervised learning, which is used when you have unlabeled data (i.e., data without defined categories or groups). The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable K. The algorithm works iteratively to assign each data point to one of K groups based on the features that are provided. Data points are clustered based on feature similarity. The results of the K-means clustering algorithm are:

1. The centroids of the K clusters, which can be used to label new data.

2. Labels for the training data.

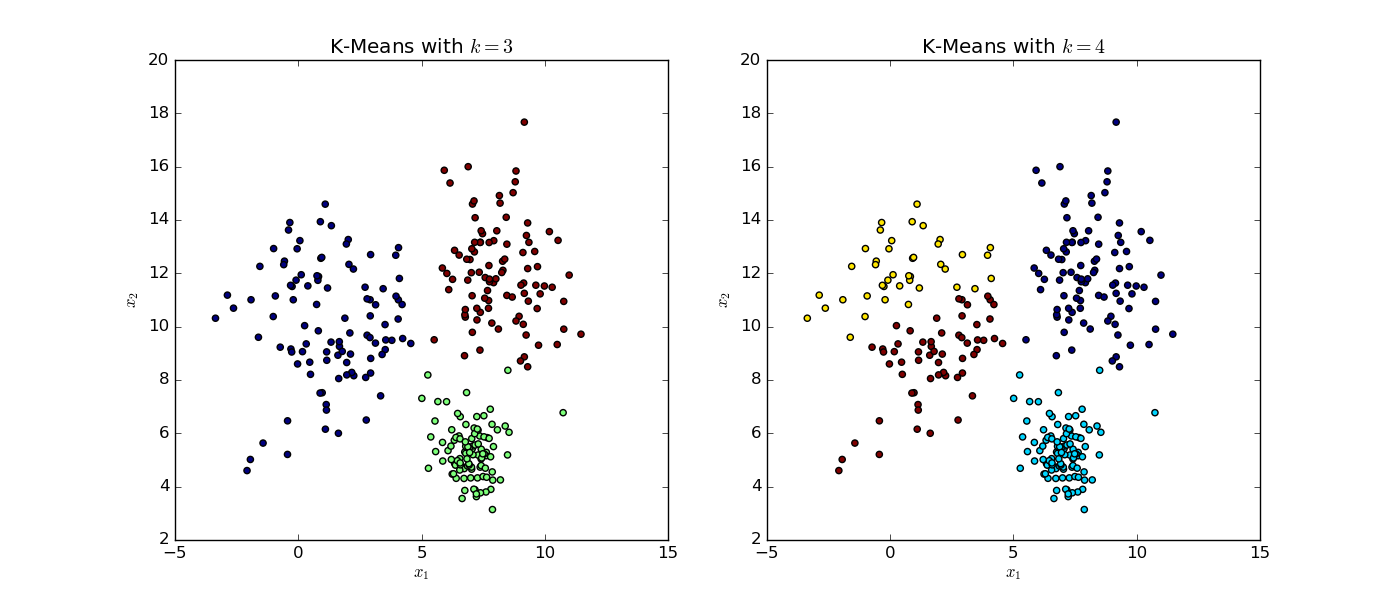


Fig. 1.1 Fig. 1.2

**5.2.2 Support Vector Machine**

SVM is a supervised machine learning algorithm which can be used for both classification and regression challenges. However, it is mostly used in classification problems. SVM are used in a variety of classification scenarios, such as image recognition and hand-writing pattern recognition. Image classification can be greatly improved with the use of SVM.

In this algorithm, we plot each data item as a point in n-dimensional space with the value of each feature being the value of particular coordinate. Then, we perform classification by finding the hyper plane that differentiate or segregate the two classes very well. Hyper-plane to maximizing the distances between nearest data point. Hyper-plane helps us to decide the right hyper-plane. This distance is called the Margin.

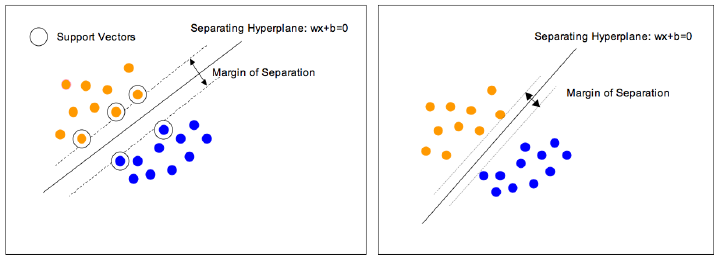


Fig. 1.3 Fig. 1.4

**5.2.3 CNN (Convolutional Neural Network)**

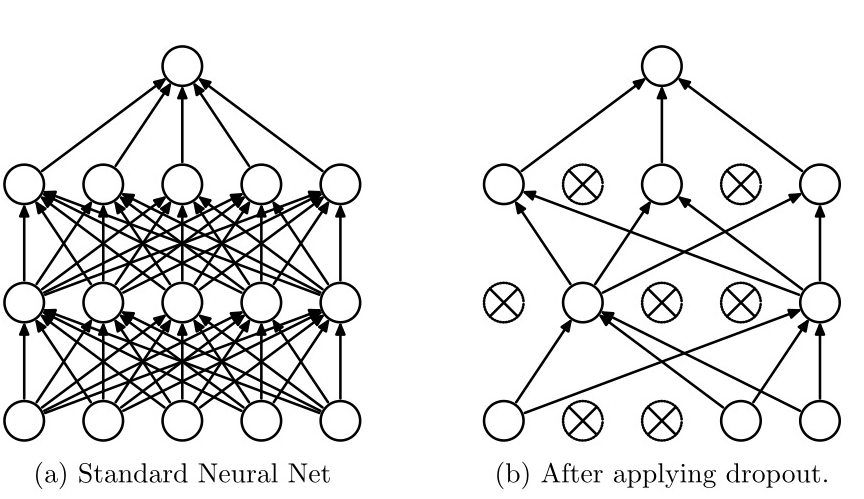
In machine learning, CNN is a class of deep and feed forward learning. It has been successfully applied to analyze the visual imagery.

CNN is made up of neurons that have learnable weights and biases. As it each neuron receives some inputs. Further, performs a dot product.

**5.2.3.1 Dropout Layer**

Dropout is a regularization technique, which aims to reduce the complexity of the model with the goal to prevent overfitting.

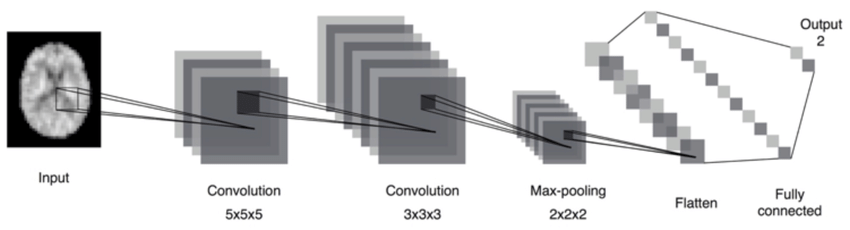
Using dropout, you randomly deactivate certain units in a layer with certain probability p from a Bernoulli distribution. **Dropout layers**have a very specific function in neural networks. The problem of overfitting, where after training, the weights of the network are so tuned to the training examples they are given that the network doesn’t perform well when given new examples. The idea of dropout is simplistic in nature. This layer “drops out” a random set of activations in that layer by setting them to zero.



**5.2.3.2 Max Pooling Layer**

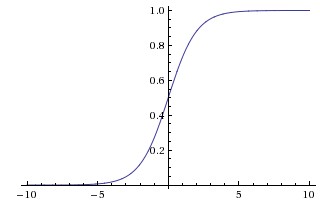
Max pooling is a sample-based discretization process. The objective is to down-sample an input representation (image, hidden-layer output matrix, etc.), reducing its dimensionality and allowing for assumptions to be made about features contained in the sub-regions binned.

This is done to in part to help over-fitting by providing an abstracted form of the representation. As well, it reduces the computational cost by reducing the number of parameters to learn and provides basic translation invariance to the internal representation. Max pooling is done by applying a max filter to (usually) non-overlapping subregions of the initial representation.



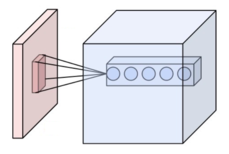
**5.2.3.3 Activation Layer**

Activation Layer is an activation function that decides the final value of a neuron. Suppose a cell value should be 1 ideally, however it has a value of 0.85, since you can never achieve a probability of 1 in CNN thus we apply an activation function.



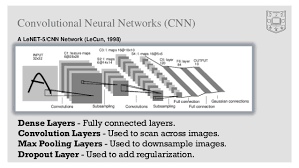
**5.2.3.4 Convolutional 2D Layer**

This layer creates a convolutional kernel that is convolved with the layer input to produce a tensor of outputs. If “use\_bias” is True, a bias vector is created and added to the outputs. When using this layer as the first layer in a model, provide the keyword argument “input\_shape” for 128\*128 RGB pictures in data format.



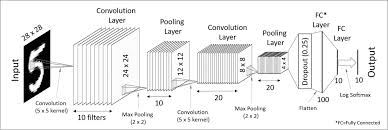
**5.2.3.5 Dense Layer**

The last stage of a convolutional neural network (CNN) is a classifier. It is called a dense layer, which is just an artificial neural network (ANN) classifier.



**5.2.3.6 Flatten Layer**

You need to convert the output of the convolutional part of the CNN into a 1D feature vector, to be used by the ANN part of it. This operation is called flattening. It gets the output of the convolutional layers, flattens all its structure to create a single long feature vector to be used by the dense layer for the final classification.



**5.2.3.7 Batch Normalization Layer**

To increase the stability of a neural network, batch normalization normalizes the output of a previous activation layer by subtracting the batch mean and dividing by the batch standard deviation.

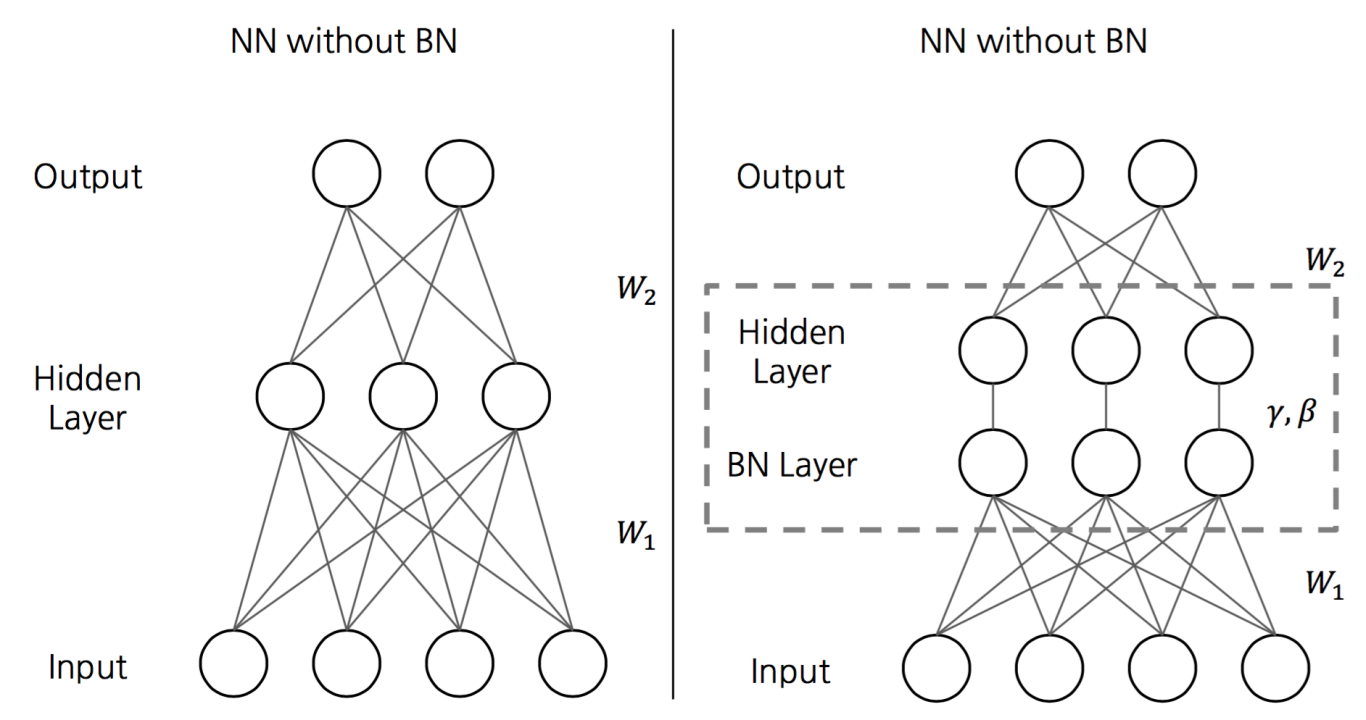


Fig. 1 Batch Normalization

**5.2.4 K-Nearest Neighbors**

The k-nearest neighbours algorithm is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether KNN is used for classification or regression.

1. In KNN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors. If k=1, then the object is simply assigned to the class of that single nearest neighbor.

2. In KNN regression, the output is property value for the object. This value is the average of the values of its k nearest neighbors.



Fig.1

K-nearest neighbour algorithm is a supervised learning algorithm where the result of a new query is classified based on majority of K-nearest neighbour category. The purpose of this algorithm is to classify a new object based on attributes and training samples. The classifier do not use any model to fit and only based on memory. The classification is using majority vote among the classification of the K objects.

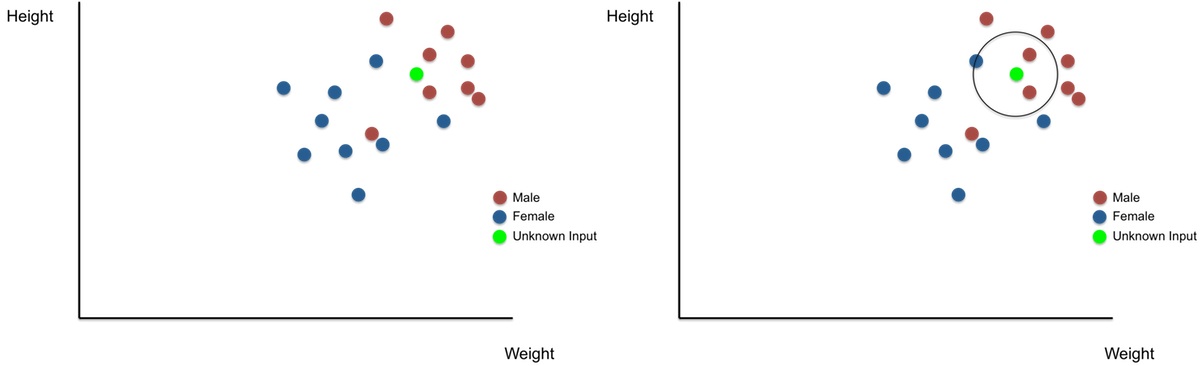


Fig.1 Fig.2

**5.3 System Testing**

The purpose of testing is to discover errors. Testing is the process of trying to discover every

conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product it is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner.

**CHAPTER 6**

**CONCLUSION AND FORESEEABLE ENHANCEMENT**

**6.1 Conclusion**

Facial Expression Recognition is an extensively researched area in computer science along with the technique of expression recognition. Concerning the evolution of facial expressions, it seems most likely that the capacity for facial expression evolved in humans for similar reasons that the capacity for speech evolved. Communication aids survival and reproduction, and facial expressions serve to promote interpersonal communication. This conclusion implies that humans did not evolve the capacity to produce facial expressions solely for the purpose of revealing their emotions, although emotions could be part of the messages communicated facially.

Therefore, this idea suggests that facial expressions do not necessarily correspond perfectly with certain specific inner dispositions. However, facial communications can reflect internal feelings or other physiological processes. There has been a large amount of research conducted that has shown that there is a very strong correlation between our human biology and the ability to make facial expressions. The facial expression recognition system presented in this research work contributes a resilient face recognition model based on the mapping of behavioural characteristics. The physiological characteristics of the human face with relevance to various expressions such as happiness, sadness, fear, anger, surprise and disgust are associated with geometrical structures which restored as base matching template for the recognition system. The behavioural aspect of this system relates the attitude behind different expressions as property base. The property bases are alienated as exposed and hidden category in genetic algorithmic genes. The gene training set evaluates the expressional uniqueness of individual faces and provide a resilient expressional recognition model. This study describes content-based filtering and implementation of three algorithm which is KNN, CNN and K-means clustering, used in facial expression recognition system. The facial expression based face recognition system is made efficient with genetic algorithm invariants of the facial surface resulting to a recognition rate of 95.4%.

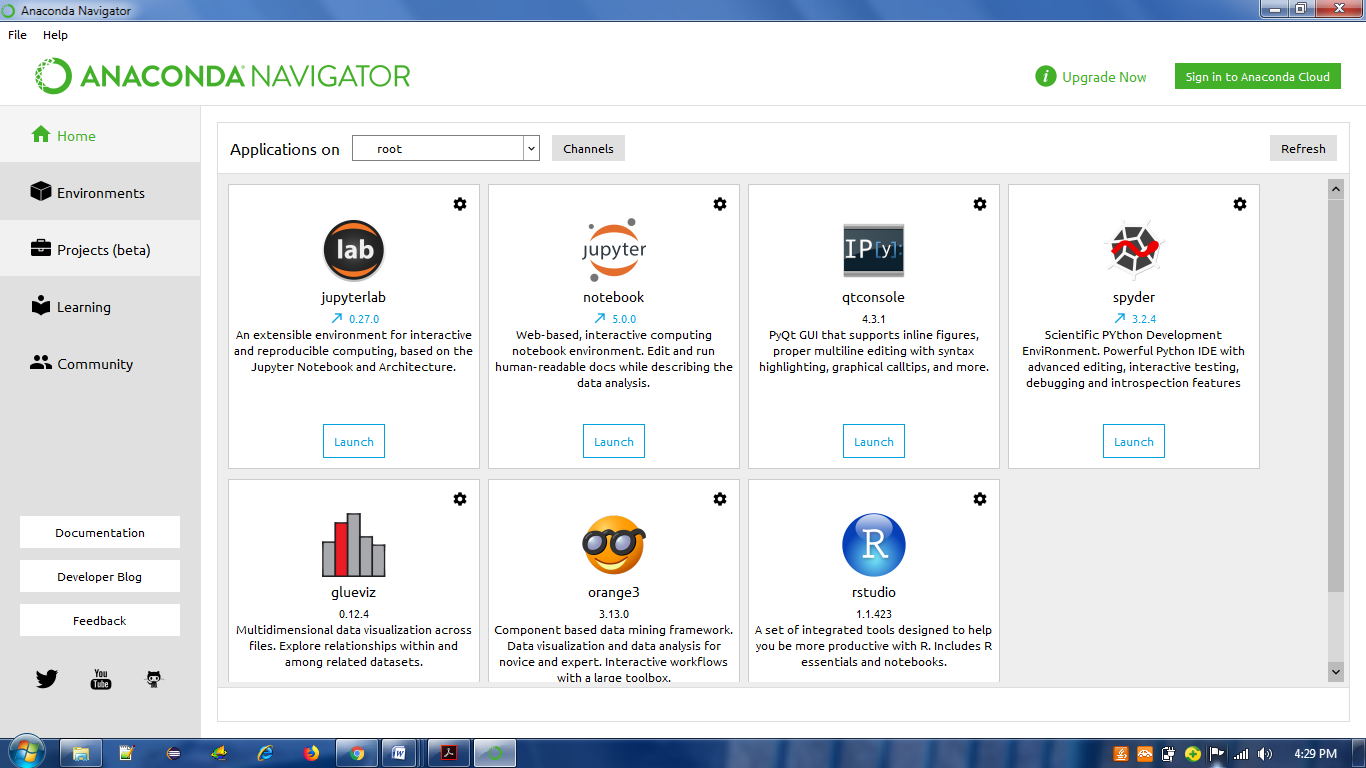
**6.2 Future Enhancement**

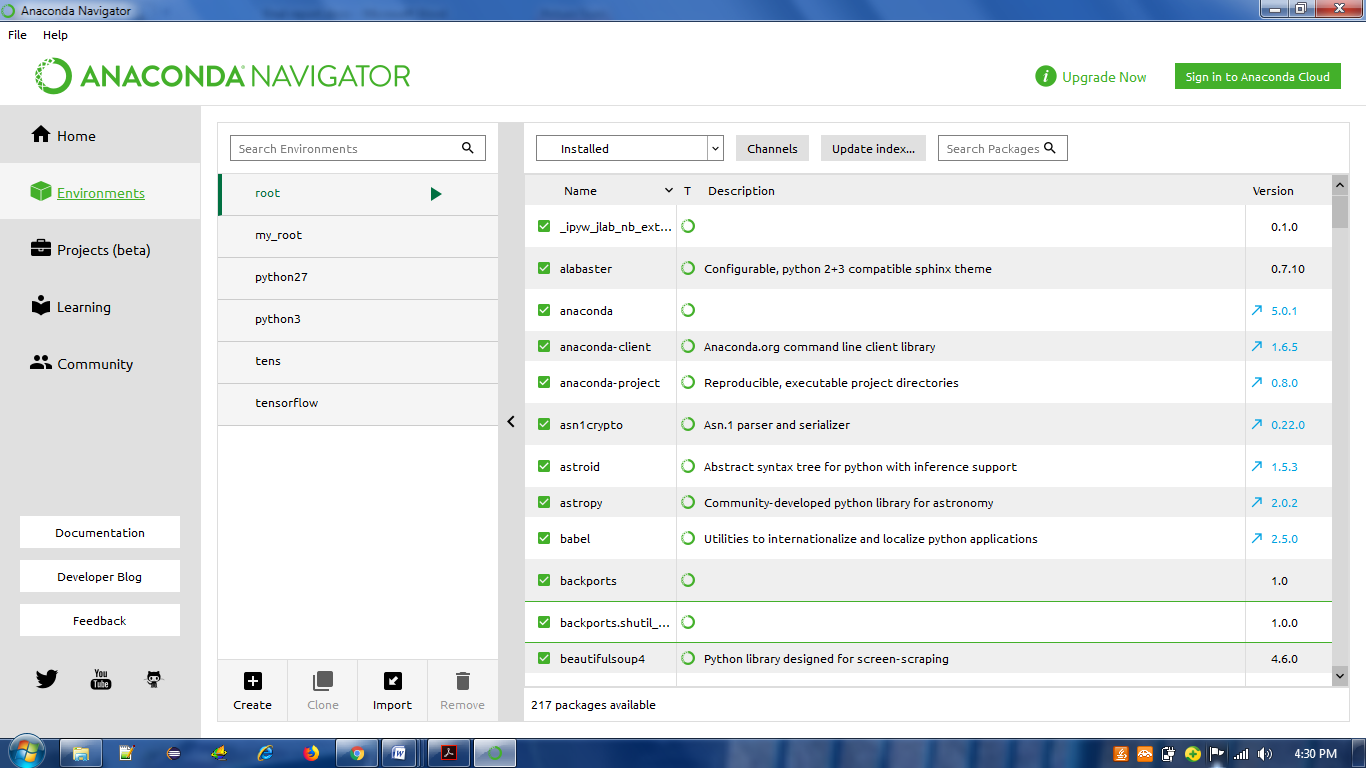
For the future work, it will use to make a real time emoji, it will play or select the song according to the mood.

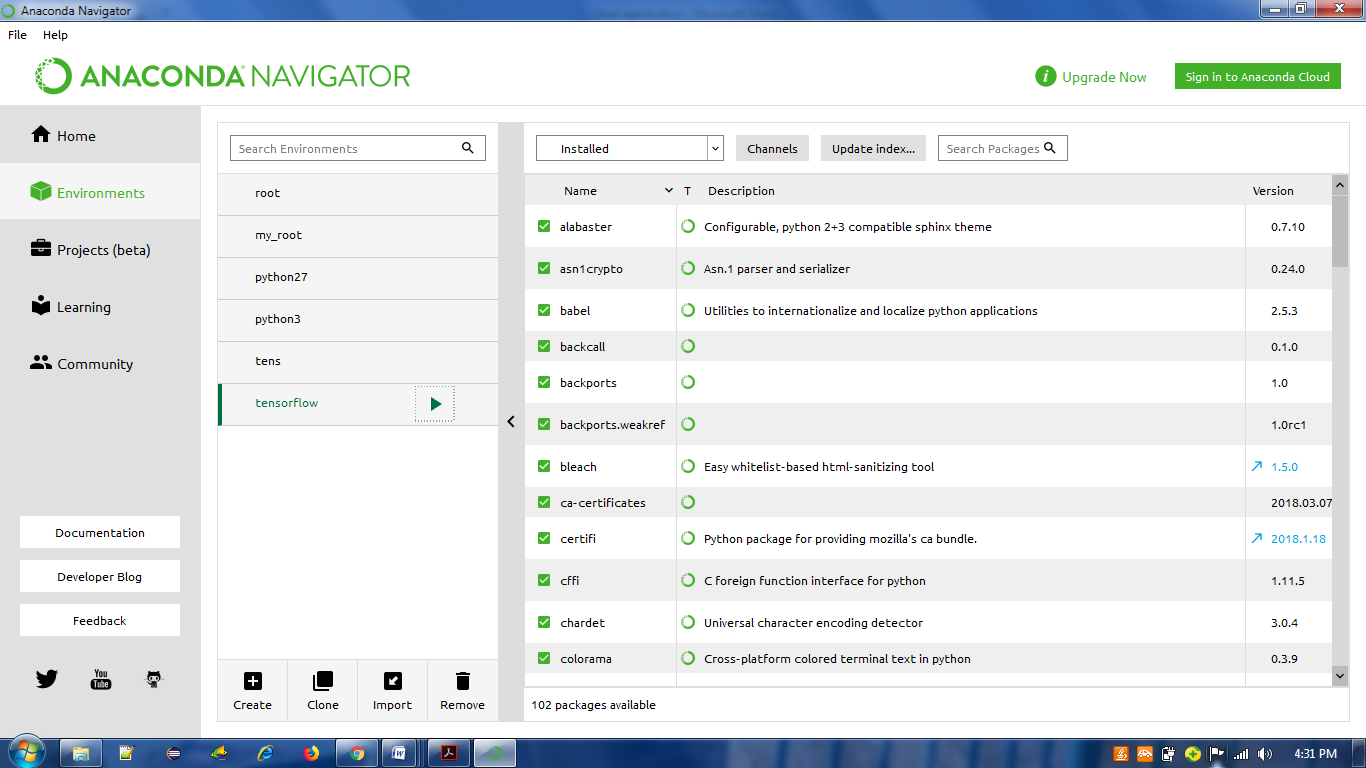
It will be helpful to recognize the expression at the time click the photo of the person.

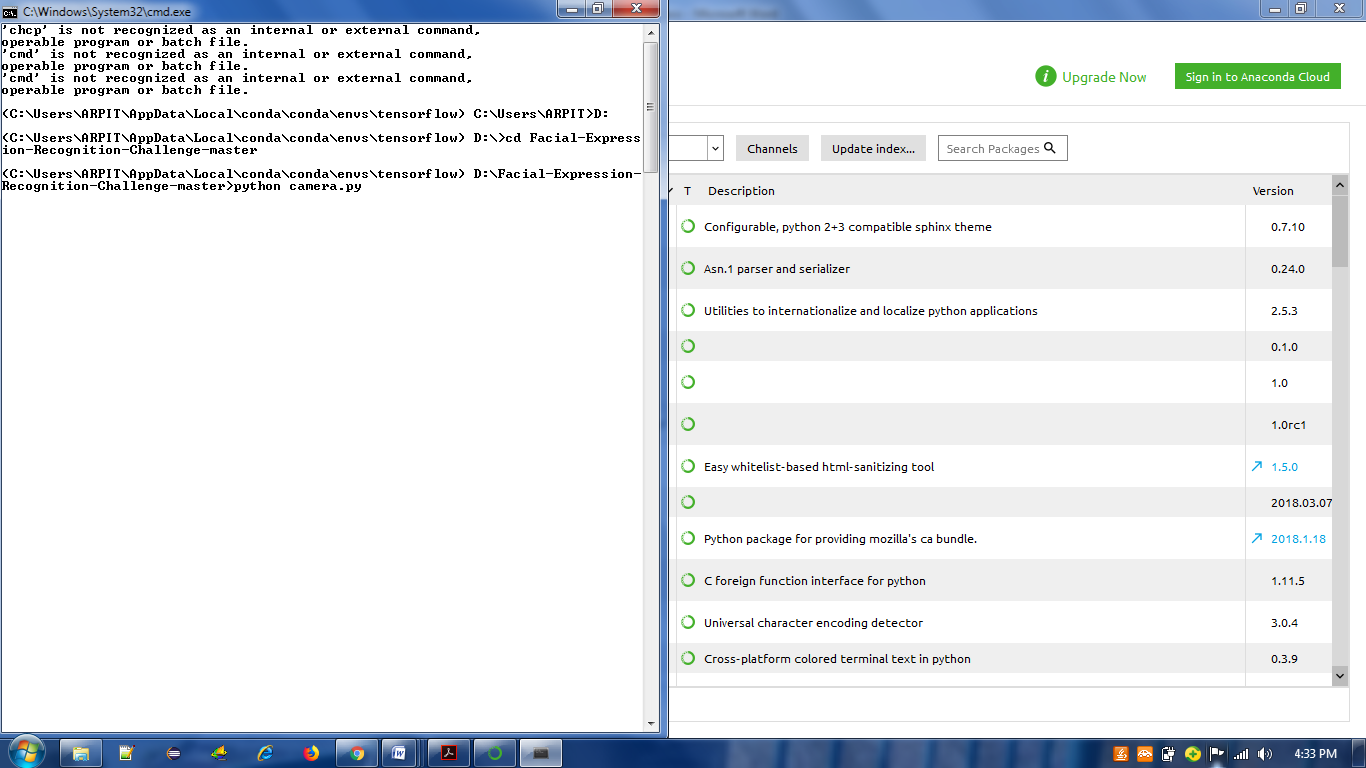
**APPENDIX – A**

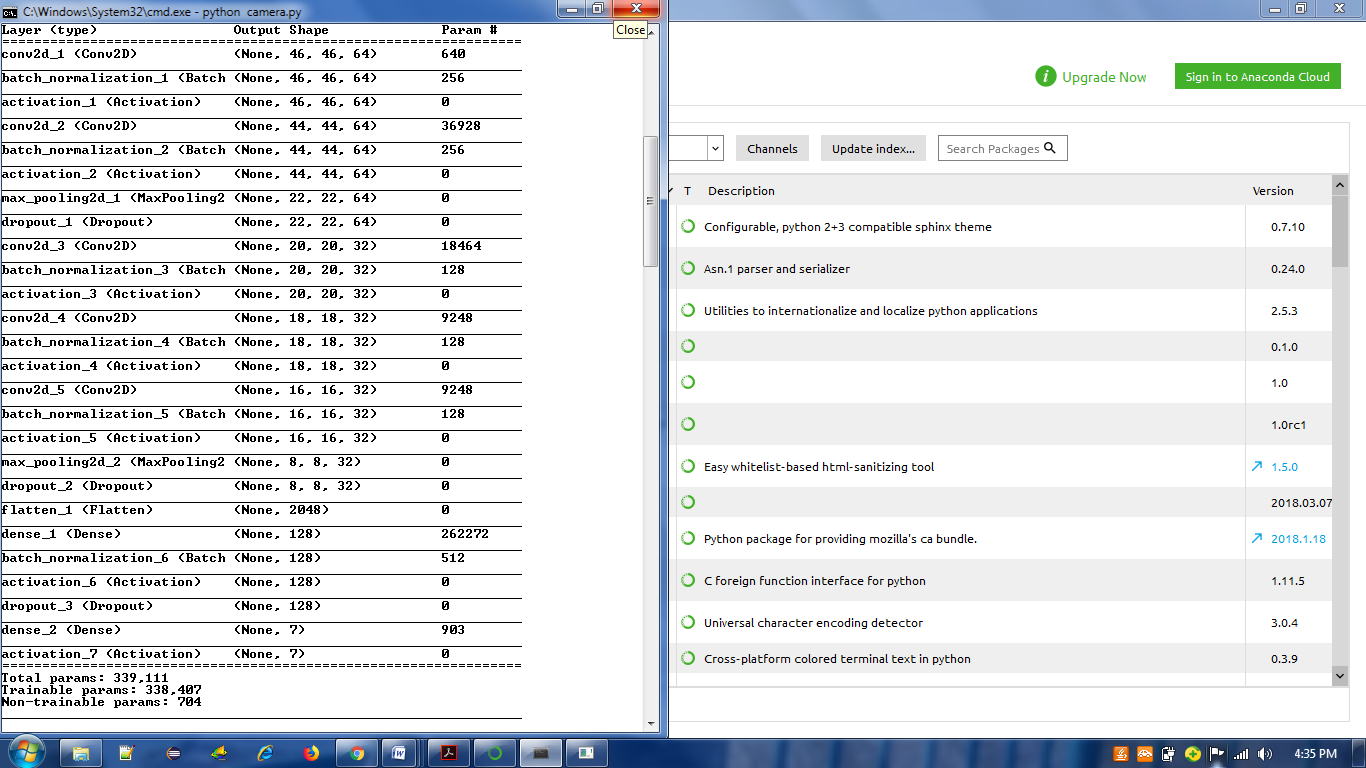
**SCREENSHOT**

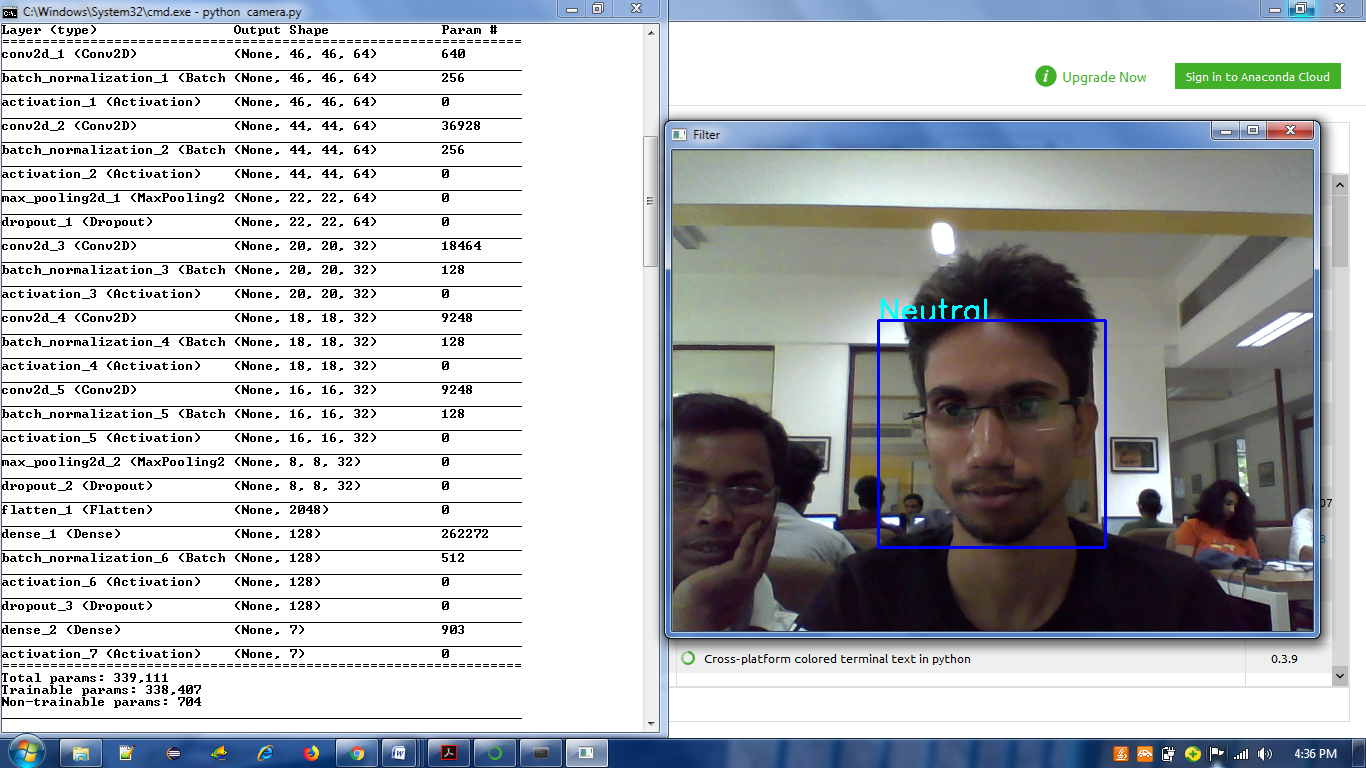
****

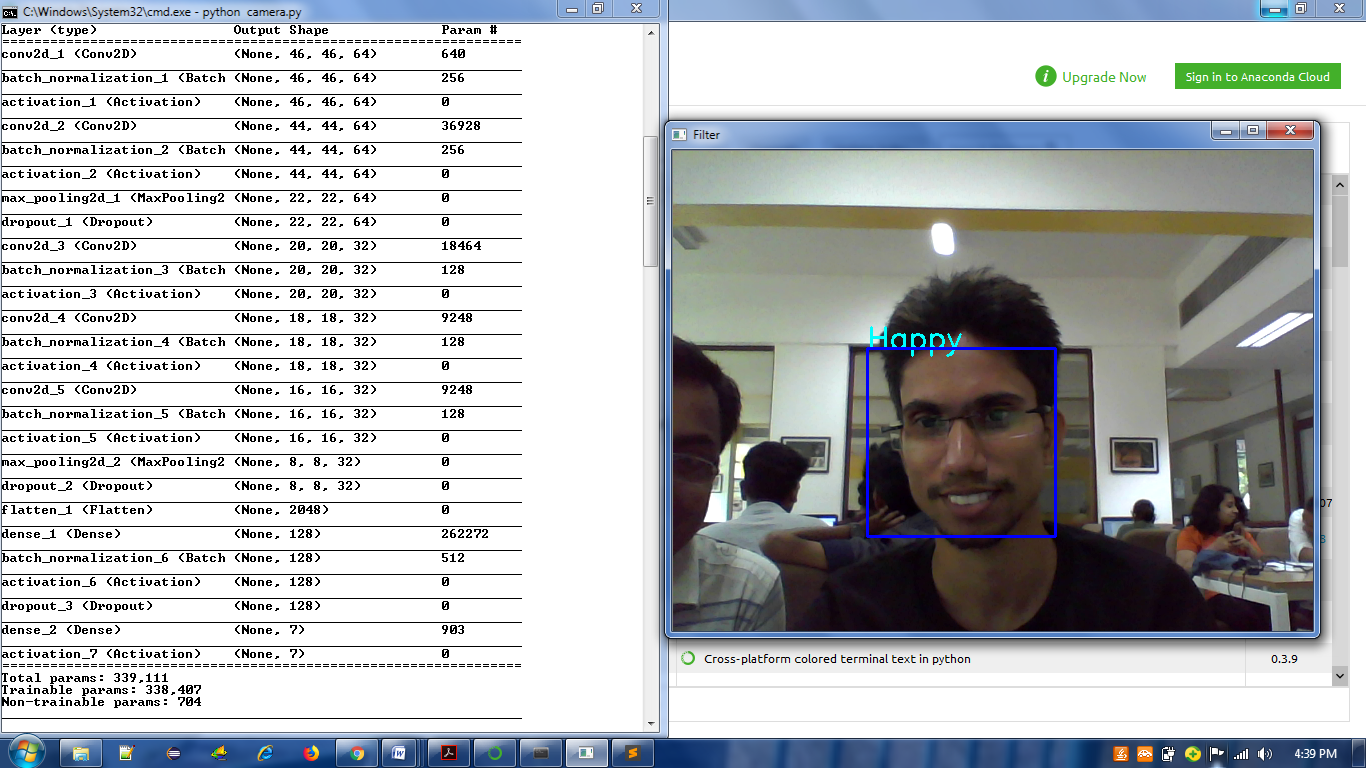
****

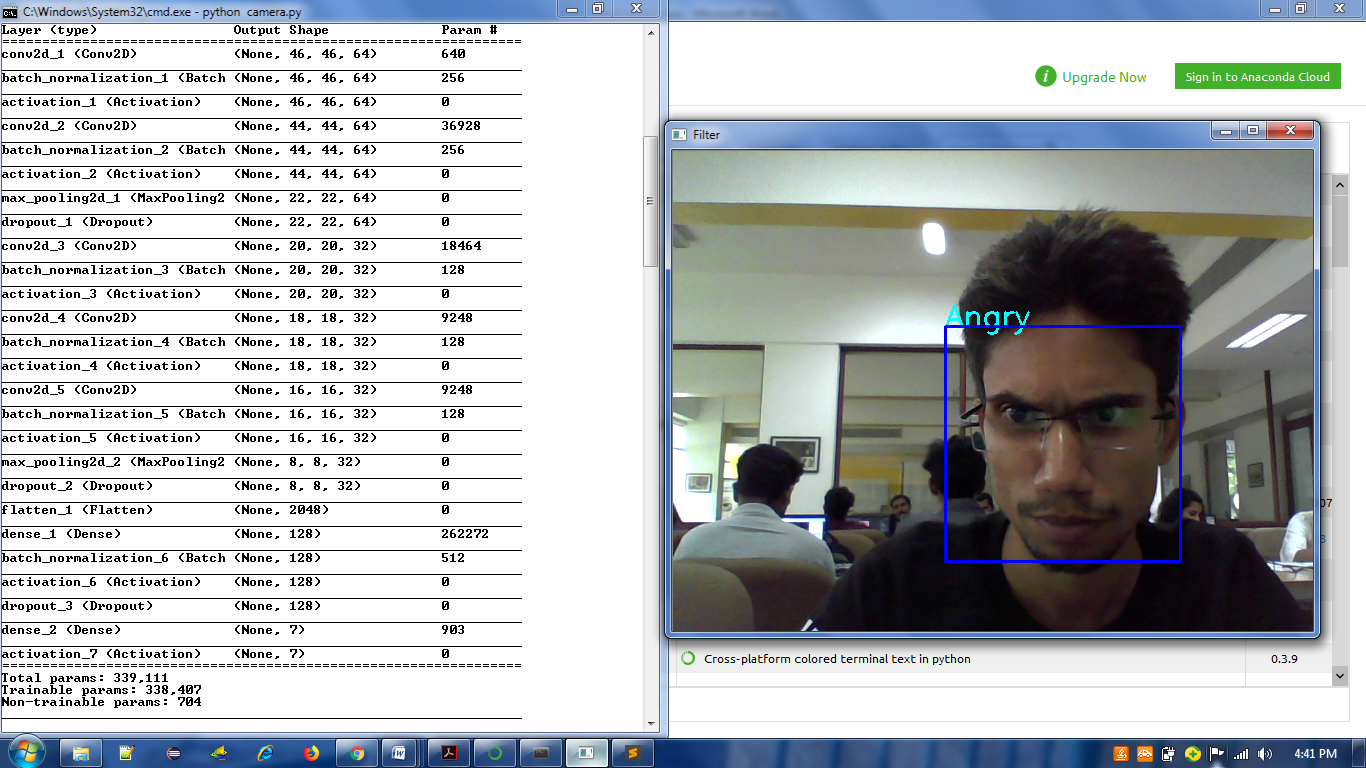
****

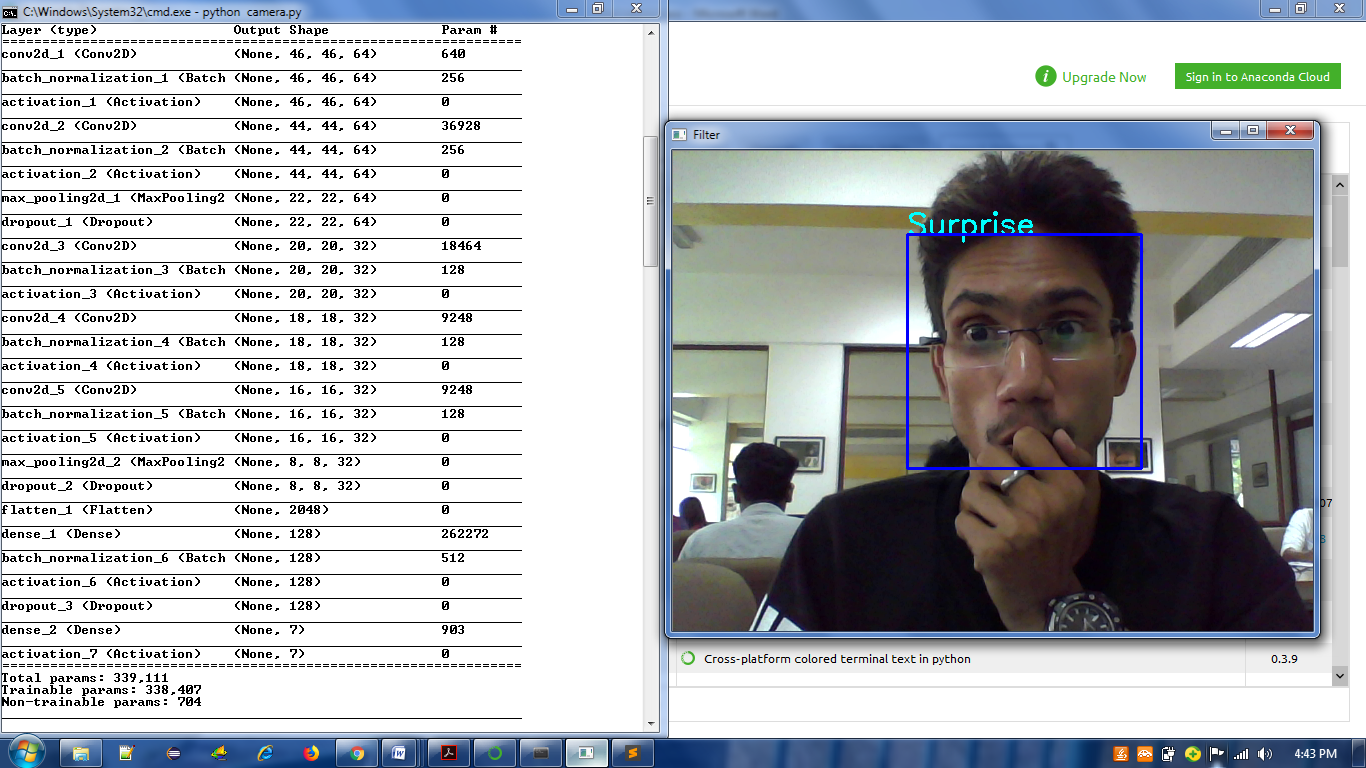
****

****

****

****

****

****

**APPENDIX – B**

**SYSTEM CODE**

**System Code**

**1. Camera.py**

import cv2

from model import FacialExpressionModel

import numpy as np

rgb = cv2.VideoCapture(0)

facec = cv2.CascadeClassifier('haarcascade\_frontalface\_default.xml')

font = cv2.FONT\_HERSHEY\_SIMPLEX

def \_\_get\_data\_\_():

"""

\_\_get\_data\_\_: Gets data from the VideoCapture object and classifies them

to a face or no face.

returns: tuple (faces in image, frame read, grayscale frame)

"""

\_, fr = rgb.read()

gray = cv2.cvtColor(fr, cv2.COLOR\_BGR2GRAY)

faces = facec.detectMultiScale(gray, 1.3, 5)

return faces, fr, gray

def start\_app(cnn):

skip\_frame = 10

data = []

flag = False

ix = 0

while True:

ix += 1

faces, fr, gray\_fr = \_\_get\_data\_\_()

for (x, y, w, h) in faces:

fc = gray\_fr[y:y+h, x:x+w]

roi = cv2.resize(fc, (48, 48))

pred = cnn.predict\_emotion(roi[np.newaxis, :, :, np.newaxis])

cv2.putText(fr, pred, (x, y), font, 1, (255, 255, 0), 2)

cv2.rectangle(fr,(x,y),(x+w,y+h),(255,0,0),2)

if cv2.waitKey(1) == 27:

break

cv2.imshow('Filter', fr)

cv2.destroyAllWindows()

if \_\_name\_\_ == '\_\_main\_\_':

model = FacialExpressionModel("face\_model.json", "face\_model.h5")

start\_app(model)

**2. Model.py**

from keras.models import model\_from\_json

import numpy as np

class FacialExpressionModel(object):

EMOTIONS\_LIST = ["Angry", "Disgust",

"Fear", "Happy",

"Sad", "Surprise",

"Neutral"]

def \_\_init\_\_(self, model\_json\_file, model\_weights\_file):

# load model from JSON file

with open(model\_json\_file, "r") as json\_file:

loaded\_model\_json = json\_file.read()

self.loaded\_model = model\_from\_json(loaded\_model\_json)

# load weights into the new model

self.loaded\_model.load\_weights(model\_weights\_file)

print("Model loaded from disk")

self.loaded\_model.summary()

def predict\_emotion(self, img):

self.preds = self.loaded\_model.predict(img)

return FacialExpressionModel.EMOTIONS\_LIST[np.argmax(self.preds)]

if \_\_name\_\_ == '\_\_main\_\_':

pass