## Project-Assignment 2

**Student Name:**SUSHANT KUMAR

**Branch:** BE-ECE

**Semester:** 4th

**UID:** 19BEC1078

**Section / Group:** 2 /A

**Subject Code**: 21-ECR-279

**Task 1:**

## Project Title: Emotional Face Recognition System Using Python Group Member:

1. **Anmol Gupta(19BEC1102)**
2. **Devesh Sehgal(19BEC1082)**
3. **Paras Kumar(19BEC1099)**
4. **Sushant Kumar(19BEC1078)**

**Group mentor**: **Dr. Yogendra Narayan**

* **Why we choose Emotional Face Recognition System Using Python as a project?**

**Ans:** As, we know that people are digging under Anxiety, Depression, Bad mood swings, etc… So, now we are coming with this project, it makes people Happy and turned their sadness into calmness and happy vibes. And we are planning to introduce this project as a Patent soon. So that we figure out the Market

## Work Distribution:

**Anmol and Devesh** Brainstormed the problem faced by the people having difficulty in communication especially when it comes to special cases such as autism where the individual is unable to share problems which sometimes can lead to problems, **Anmol, Sushant and Paras** solved the problem and shaped the solution and made it viable.

## Task 2

## Problem definition:

Detecting the real-time emotion of the person with a camera input is one of the advanced features in the machine learning process. The detection of emotion of a person using a camera is useful for various research and analytics purposes. The detection of emotion is made by using the machine learning concept. You can use the trained dataset to detect the emotion of the human being. For detecting the different emotions, first, you need to train those different emotions, or you can use a dataset already available on the internet.

In this article, we will discuss creating a Python program to detect the real- time emotion of a human being using the camera.

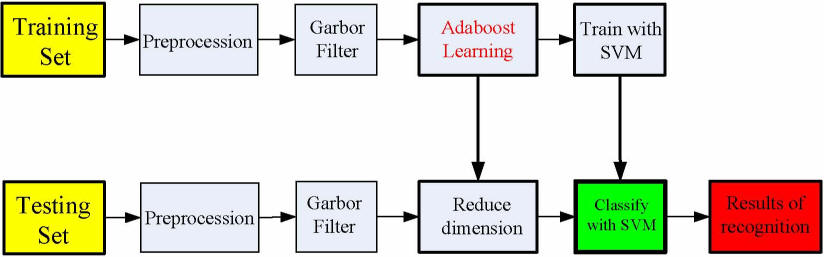
## Objective:

The main goal of this project is to develop a software which is not only able to understand and recognize the facial features and the act accordingly for example if the person is in extreme distress, then and call to the concerned people and inform the about the current situation while it triggers something which will el the affected safe calm.

**Financial Model:**

Since this Project only utilizes the software and no hardware is needed for it , financially the overall cost for the development was nil , everything needed such as software etc was obtained for free from the source websites

**Detailed schematics/ Coding:**

****

**Source code for the Model set :** **import sys, os**

import pandas as pd

import numpy as np

from keras.models import Sequential

from keras.layers import Dense, Dropout, Activation, Flatten

from keras.layers import Conv2D, MaxPooling2D, BatchNormalization,AveragePooling2D

from keras.losses import categorical\_crossentropy

from keras.optimizers import Adam

from keras.regularizers import l2

from keras.utils import np\_utils

df=pd.read\_csv('fer2013.csv')

X\_train,train\_y,X\_test,test\_y=[],[],[],[]

for index, row in df.iterrows():

val=row['pixels'].split(" ")

try:

if 'Training' in row['Usage']:

X\_train.append(np.array(val,'float32'))

train\_y.append(row['emotion'])

elif 'PublicTest' in row['Usage']:

X\_test.append(np.array(val,'float32'))

test\_y.append(row['emotion'])

except:

print(f"error occured at index :{index} and row:{row}")

num\_features = 64

num\_labels = 7

batch\_size = 64

epochs = 30

width, height = 48, 48

X\_train = np.array(X\_train,'float32')

train\_y = np.array(train\_y,'float32')

X\_test = np.array(X\_test,'float32')

test\_y = np.array(test\_y,'float32')

train\_y=np\_utils.to\_categorical(train\_y, num\_classes=num\_labels)

test\_y=np\_utils.to\_categorical(test\_y, num\_classes=num\_labels)

X\_train -= np.mean(X\_train, axis=0)

X\_train /= np.std(X\_train, axis=0)

X\_test -= np.mean(X\_test, axis=0)

X\_test /= np.std(X\_test, axis=0)

X\_train = X\_train.reshape(X\_train.shape[0], 48, 48, 1)

X\_test = X\_test.reshape(X\_test.shape[0], 48, 48, 1)

model = Sequential()

model.add(Conv2D(64, kernel\_size=(3, 3), activation='relu', input\_shape=(X\_train.shape[1:])))

model.add(Conv2D(64,kernel\_size= (3, 3), activation='relu'))

model.add(MaxPooling2D(pool\_size=(2,2), strides=(2, 2)))

model.add(Dropout(0.5))

model.add(Conv2D(64, (3, 3), activation='relu'))

model.add(Conv2D(64, (3, 3), activation='relu'))

model.add(MaxPooling2D(pool\_size=(2,2), strides=(2, 2)))

model.add(Dropout(0.5))

model.dd(Conv2D(128, (3, 3), activation='relu'))

model.add(Conv2D(128, (3, 3), activation='relu'))

model.add(MaxPooling2D(pool\_size=(2,2), strides=(2, 2)))

model.add(Flatten())

model.add(Dense(1024, activation='relu'))

model.add(Dropout(0.2))

model.add(Dense(1024, activation='relu'))

model.add(Dropout(0.2))

model.add(Dense(num\_labels, activation='softmax'))

model.compile(loss=categorical\_crossentropy,

optimizer=Adam(),

metrics=['accuracy'])

model.fit(X\_train, train\_y,

batch\_size=batch\_size,

epochs=epochs,

verbose=1,

validation\_data=(X\_test, test\_y),

shuffle=True)

fer\_json = model.to\_json()

with open("fer.json", "w") as json\_file:

json\_file.write(fer\_json)

model.save\_weights("fer.h5")

**Source code for the Project :** **import cv2**

import numpy as np

from keras.models import model\_from\_json

from keras.preprocessing import image

import pygame

from pygame import mixer

import time

model = model\_from\_json(open("fer.json", "r").read())

model.load\_weights('fer.h5')

face\_haar\_cascade = cv2.CascadeClassifier('haarcascade\_frontalface\_default.xml')

cap=cv2.VideoCapture(0)

while True:

ret,test\_img=cap.read()

if not ret:

continue

gray\_img= cv2.cvtColor(test\_img, cv2.COLOR\_BGR2GRAY)

faces\_detected = face\_haar\_cascade.detectMultiScale(gray\_img, 1.32, 5)

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

cv2.rectangle(test\_img,(x,y),(x+w,y+h),(255,0,0),thickness=7)

roi\_gray=gray\_img[y:y+w,x:x+h]

roi\_gray=cv2.resize(roi\_gray,(48,48))

img\_pixels = image.img\_to\_array(roi\_gray)

img\_pixels = np.expand\_dims(img\_pixels, axis = 0)

img\_pixels /= 255

predictions = model.predict(img\_pixels)

max\_index = np.argmax(predictions[0])

emotions = ('Angry', 'Disgust', 'Fear', 'Happy', 'Sad', 'Surprise', 'Neutral')

predicted\_emotion = emotions[max\_index]

cv2.putText(test\_img, predicted\_emotion, (int(x), int(y)), cv2.FONT\_HERSHEY\_SIMPLEX, 1, (0,0,255), 2)

print (predicted\_emotion)

if predicted\_emotion == ("Sad"):

mixer.init()

mixer.music.load("3.mp3")

mixer.music.set\_volume(0.3)

mixer.music.play()

resized\_img = cv2.resize(test\_img, (1000, 700))

cv2.imshow('Facial emotion analysis ',resized\_img)

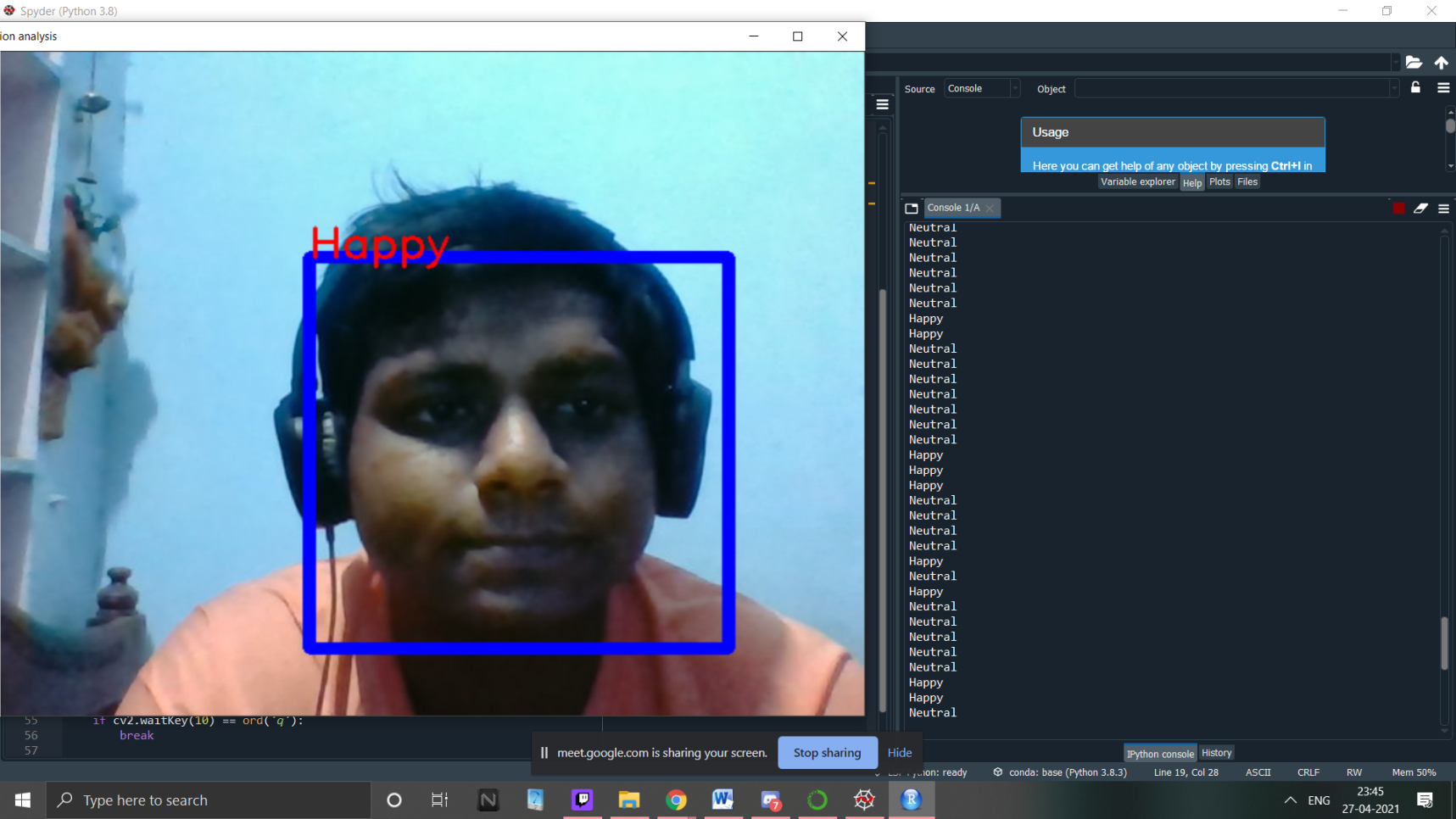
if cv2.waitKey(10) == ord('q'):

break

cap.release()

cv2.destroyAllWindows

**Output:**

****

## Project Sketch/Techniques:

https://github.com/sushassassin/Facial\_emotions

BEFARTME9T0F

cu )

Discover. Learn. Empower.

NGRAAADCE¥A\*W«I•

ACCREDITED UNIVERSITY