**EMOTION RECOGNIZER**

*A*

*Mini Project Report Submitted in partial fulfilment of the*

*Requirements for the award of the Degree of*

## BACHELOR OF ENGINEERING

IN

## INFORMATION TECHNOLOGY

By

## R. SAI SATHVIK 1602-20-737-035

**B. SURESH KUMAR 1602-20-737-051**

## M. SAI LAXMI 1602-20-737-034



**Department of Information Technology Vasavi College of Engineering (Autonomous)**

## VASAVI COLLEGE OF ENGINEERING (AUTONOMOUS)

### (AFFILIATED TO OSMANIA UNIVERSITY) HYDERABAD - 500 030

**Department of Information Technology**



### DECLARATION BY CANDIDATE

We, **R. SAI SATHVIK, B. SURESH KUMAR, M. SAI LAXMI,** bearing hall ticket number, **1602-20-737-035, 1602-20-737-051, 1602-20-737-034** hereby declare that the project report entitled **”EMOTION RECOGNIZER”** Department of Information Technology, Vasavi College of Engineering, Hyderabad, is submitted in partial fulfillment of the requirement for the award of the degree of **Bachelor of Engineering** in **Information Technology**

This is a record of bonafide work carried out by me and the results embodied in this project report has not been submitted to any other university or institute for the award of any other degree or diploma.

### R. SAI SATHVIK 1602-20-737-035

**B. SURESH KUMAR 1602-20-737-051**

**M. SAI LAXMI 1602-20-737-034**

## VASAVI COLLEGE OF ENGINEERING (AUTONOMOUS)

### (AFFILIATED TO OSMANIA UNIVERSITY) HYDERABAD - 500 030

**Department of Information Technology**



### BONAFIDE CERTIFICATE

This is to certify that the project entitled “**EMOTION RECOGNIZER**” being submitted by **R. SAI SATHVIK, B. SURESH KUMAR, M. SAI LAXMI** bearing **1602-20-737-035,** **1602-20-737-051, 1602-20-737-034**, in partial fulfillment of the requirements for the completion of MINI PROJECT of Bachelor of Engineering in Information Technology is a record of bonafide work carried out by them under my guidance.

Mrs. C.Sireesha External Examiner Dr.K Ram Moahn Rao IInternal Guide HOD, IT

## ACKNOWLEDGEMENT

We thank the department of INFORMATION TECHNOLOGY, for introducing the subject “Mini Project-2” in BE fifth semester.

We would also like to show our appreciation to our Honorable principal, Dr S V Ramana sir, our HOD K. Ram Mohan Rao for supporting us and our mini project lecturer, Mrs C.Sireesha ma’am, for letting us properly understand the process of doing a project and for providing valuable insight and expertise that has greatly assisted us in the making of the project.

Table of Contents:

[CHAPTER 1: INTRODUCTION 7](#_Toc123249257)

[1.1 PURPOSE: 7](#_Toc123249258)

[1.2 INTENDED AUDIENCE: 7](#_Toc123249259)

[1.3 PRODUCT SCOPE: 7](#_Toc123249260)

[1.4 PROBLEM DEFINITION 7](#_Toc123249261)

[CHAPTER 2: RELATED WORK 8](#_Toc123249262)

[CHAPTER 3: PROPOSED WORK 8](#_Toc123249263)

[3.1 Use cases: 8](#_Toc123249264)

[3.2 UI prototypes or screenshots: 9](#_Toc123249265)

[3.3 Architecture: 10](#_Toc123249266)

[3.4 Technology used: 11](#_Toc123249267)

[3.4 Design: 12](#_Toc123249268)

[3.4.1 DATA FLOW DIAGRAM: 12](#_Toc123249269)

[3.4.2 USE CASE DIAGRAM: 12](#_Toc123249270)

[3.5 Implementation: 13](#_Toc123249271)

[3.5.1 Algorithm used: 13](#_Toc123249272)

[3.5.2 Code: 13](#_Toc123249273)

[CHAPTER 4: RESULTS 24](#_Toc123249274)

[CHAPTER 5: DISCUSSION AND FUTURE WORK 26](#_Toc123249275)

[CHAPTER 6: REFERENCES 26](#_Toc123249277)

# ABSTRACT:

Humans can detect emotions from multiple domains for example speech and visual.

The primary objective of our project is to detect human emotions through machines similarly like how humans detect, which has become an essential requirement in the field of social intelligence, also increases the human-machine interactions.

The future generations of computers thus must be able to interact with a human being just like another.

With advancement of deep learning technology there has been significant improvement of speech and video recognition.

Recognizing emotion from speech and video is important aspect and with deep learning technology emotion recognition has improved in accuracy and latency.

There are still many challenges to improve accuracy. In this work, we attempt to explore different neural networks to improve accuracy of emotion recognition. With different architectures explored.

# CHAPTER 1: INTRODUCTION

**What is an emotion recognizer?**

Emotion recognizer is used to detect human emotions through machines similarly like how humans detect, which has become an essential requirement in the field of social intelligence, also increases the human-machine interactions.

**Why does a user need a smart prescription interpreter?**

1. To detect human emotions through audio.
2. To detect human emotions through video.

## PURPOSE:

**Detect emotion in audio:** Audio feeds are transcribed so that they are converted into text and then this is analyzed for the sentiment expressed. Audio feeds could range from sources such as podcasts, sales calls, customer service calls, interviews, telehealth calls, or any other medium.

**Detect emotion in video:** Sentiment can be identified and analyzed in videos through machine learning algorithms that can capture text from caption overlays in the videos as well as the audio in it.

## 1.2 INTENDED AUDIENCE:

The intended audience are people who wish to know the facial expression of a person using a camera or a mic by recording their expressions for further analysis to get a review or a dataset over it for further improvement or developments.

## 1.3 PRODUCT SCOPE:

Emotion Recognizer has a scope to detect emotions through audio or video that help companies to detect the emotions of their target audience and to improve machine human interactions.

## 1.4 PROBLEM DEFINITION

Through various modes such as audio and video different emotions such as anger, happiness, excitement, sadness, frustration, fear, surprise and neutral state are detected using deep learning algorithms.

# CHAPTER 2: RELATED WORK

Emotion recognition through Audio signal and Picture videos is a recent research topic in the Human Computer Interaction. The demand was risen for increasing communication interface between the humans and digital media. Many researchers are currently working in order to improve their accuracy. But still there is a lack of complete system which can recognize emotions from speech. In order to make the human and the digital machine interaction more natural, the computer should be able to recognize emotional states in the same way as human.

# CHAPTER 3: PROPOSED WORK

## 3.1 Use cases:

Diagram, schematic

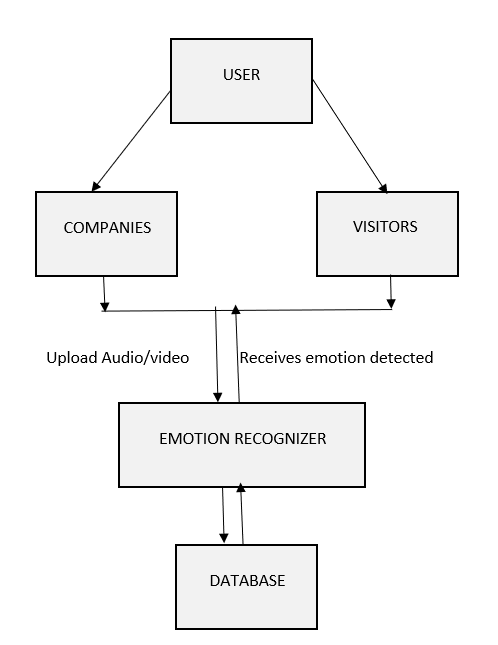
Description automatically generated

## 3.2 UI prototypes or screenshots:

A screenshot of a computer

Description automatically generated with medium confidence

## 3.3 Architecture:

****

## 3.4 Technology used:

The tool using which emotion recognizer was made is Tensor flow, keras.

Tensor Flow is an end-to-end open-source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources that let’s researchers push the state-of-the-art in ML and developers easily build and deploy ML- powered applications.

TensorFlow was originally developed by researchers and engineers working on the Google Brain team within Google's Machine Intelligence Research organization to conduct machine learning and deep neural networks research. The system is general enough to be applicable in a wide variety of other domains, as well.

**User**- A user is any human being who uses emotion detection technology. They can play any role: detect emotion through audio or detect emotion through video. As long as they are human, they are termed ‘user’.

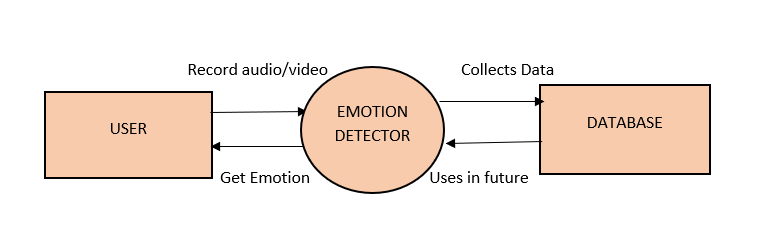
**Emotions** – Emotions are reactions that human beings experience in response to events or situations. The type of emotion a person experiences is determined by the circumstance that triggers the emotion. For instance, a person experiences joy when they receive good news. A person experiences fear when they are threatened.1

Emotions have a strong influence on our daily lives. We make decisions based on whether we are happy, angry, sad, bored, or frustrated. We choose activities and hobbies based on the emotions they incite. Understanding emotions can help us navigate life with greater ease and stability.

**Response:** This is the emotion recognizer output that is aimed at satisfying the user’s intent. The most accurate responses occur when a proper range of emotions have been correctly grouped. Accurate and simple responses are important traits for a good emotion detector.

## 3.4 Design:

### 3.4.1 DATA FLOW DIAGRAM:

****

### 3.4.2 USE CASE DIAGRAM:

**Diagram, schematic

Description automatically generated**

## Implementation:

### 3.5.1 Algorithm used:

TensorFlow was originally developed by researchers and engineers working on the Google Brain team within Google's Machine Intelligence Research organization to conduct machine learning and deep neural networks research. The system is general enough to be applicable in a wide variety of other domains, as well.

### 3.5.2 Code:

<!DOCTYPE html>

<html lang="en">

<head>

<title>Emotional recognizer using deep learning </title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<div class="main">

<div class="navbar">

<div class="icon">

<h2 class="logo">EMOTION RECOGNIZER</h2>

</div>

<div class="menu">

<ul>

<li><a href="HELP.html">HELP</a></li>

</ul>

</div>

</div>

<div class="content">

<h1>Emotion <span>Recognizer</span></h1>

<p class="par"><br>Welcome you to EMOTION RECOGNIZER Website where we make machine learn emotion.

<br>Emotion's speak louder than words.Recognize Emotions as they start to arise and smile <br>

at yourself inwardly.</p>

<button class="cn"><a href="index.html">START</button>

</div>

</div>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<title>HOME PAGE</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<div class="main">

<div class="navbar">

<div class="icon">

<h2 class="logo">EMOTION RECOGNIZER</h2>

</div>

<div class="menu">

<ul>

<li><a href="Start.html">HOME</a></li>

<li><a href="ABOUT.html">ABOUT</a></li>

</ul>

</div>

</div>

<div class="content">

<h1> <span>Description:</span></h1>

<p class="par"><br>Multimodal Emotion Recognition is

a relatively new discipline that aims to include text inputs, <br>as well as sound

and video.This field has been rising with the development of social networks<br>

that gave researchers access to a vast amount of data.Recent studies have been

exploring<br> potential metrics to measure the coherence between emotions from the

different channels.</p>

<h2><span>Information:</span></h2>

<p class="par"><br>-->In this project two different inputs audio or video could be given as input.

<br> <br>

-->If audio was to be given as input then an local mp3 file is to be uploaded.

<br> <br>

-->The system recognizes different emotions that are observed in the audio.</p>

<div class="form">

<h1>Let's detect Emotion</h1>

<p> <br>Click on the "Choose file" to upload AUDIO file:<br></p>

<form action ="#">

<br>

<input type="file" id ="myFile" name="filename">

<br>

<p> <br>Click on the "Live camera" to access camera:<br></p>

<br>

<button class="cn"><a href="#">Live camera</button>

</form>

</div>

</div>

</div>

</div>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<title>HELP </title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<div class="main">

<div class="navbar">

<div class="icon">

<h2 class="logo">EMOTION RECOGNIZER</h2>

</div>

</div>

</div>

<div class="content">

<h1>INFO</h1>

<p class="par"><br></p>

</div>

\*{

margin: 0;

padding: 0;

}

.main{

width: 100%;

background: linear-gradient(to top,rgb(0, 0, 0,0.5)50%,rgb(0, 0, 0,0.5)50%),url(2.jpg);

background-position: center;

background-size: cover;

height:109vh;

}

.navbar{

width: 1200px;

height:75px;

margin: auto;

}

.icon{

width: 200px;

float: left;

height: 70px;

}

.logo{

color: #ff7200;

font-size: 30px;

font-family: 'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans', Arial, sans-serif;

padding-left: 10px;

float: left;

padding-top: 10px;

}

.menu{

width: 400px;

float: left;

height: 70px;

}

ul{

float: left;

display: flex;

justify-content: center;

align-items: center;

}

ul li{

list-style: none;

margin-left: 62px;

margin-top: 27px;

font-size: 14px;

}

ul li a{

text-decoration: none;

color: #fff;

font-family: Arial;

font-weight: bold;

transition: 0.4 ease-in-out;

}

ul li a:hover{

color:#ff7200;

}

.content{

width: 1200px;

height:auto;

margin: auto;

color: #fff;

position: relative;

}

.content .par{

padding-left:20px;

padding-bottom:25px;

font-family: Arial;

letter-spacing: 1.2px;

line-height: 3opx;

}

.content h1{

font-family:'Times New Roman' ;

font-size: 60px;

padding-left: 20px;

margin-top: 9%;

letter-spacing: 2px;

}

.content .cn{

width:120px;

height:40px;

background: #ff7200;

border: none;

margin-bottom: 10px;

margin-left: 20px;

font: size 18px;

border-radius: 10px;

cursor:pointer;

transition: .4 ease

}

.content .cn a{

text-decoration:none;

color:black;

transition: .3s ease;

}

.cn:hover{

background-color:#fff;

}

.content span{

color:#ff7200;

font-size: 60px;

}

.form {

width: 250px;

height:350px;

background: linear-gradient(to top,rgba(0,0,0,0.8)50%,rgba(0,0,0,0.8)50%);

position:absolute;

top:-20px;

left:870px;

border-radius:10px;

padding:25px;

}

.form h1{

width: 220px;

font-family: 'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans', Arial, sans-serif;

text-align: center;

color: #ff7200;

font-size: 35px;

background-color: #fff;

border-radius:10px;

margin: 2px;

padding: 8px;

}

/\* The flip card container - set the width and height to whatever you want. We have added the border property to demonstrate that the flip itself goes out of the box on hover (remove perspective if you don't want the 3D effect \*/

.flip-card {

background-color: transparent;

width: 300px;

height: 200px;

/\* border: 1px solid #f1f1f1; \*/

perspective: 1000px; /\* Remove this if you don't want the 3D effect \*/

}

/\* This container is needed to position the front and back side \*/

.flip-card-inner {

position: relative;

width: 100%;

height: 100%;

text-align: center;

transition: transform 0.8s;

transform-style: preserve-3d;

}

/\* Do an horizontal flip when you move the mouse over the flip box container \*/

.flip-card:hover .flip-card-inner {

transform: rotateY(180deg);

}

/\* Position the front and back side \*/

.flip-card-front, .flip-card-back {

position: absolute;

width: 100%;

height: 100%;

-webkit-backface-visibility: hidden; /\* Safari \*/

backface-visibility: hidden;

}

/\* Style the front side (fallback if image is missing) \*/

.flip-card-front {

background-color: #bbb;

color: black;

}

/\* Style the back side \*/

.flip-card-back {

background-color: dodgerblue;

color: white;

transform: rotateY(180deg);

}

from \_future\_ import division

import numpy as np

import pandas as pd

import cv2

from time import time

from time import sleep

import re

import os

import argparse

from collections import OrderedDict

### Image processing ###

from scipy.ndimage import zoom

from scipy.spatial import distance

import imutils

from scipy import ndimage

import dlib

from tensorflow.keras.models import load\_model

from imutils import face\_utils

import requests

global shape\_x

global shape\_y

global input\_shape

global nClasses

def show\_webcam() :

shape\_x = 48

shape\_y = 48

input\_shape = (shape\_x, shape\_y, 1)

nClasses = 7

thresh = 0.25

frame\_check = 20

def eye\_aspect\_ratio(eye):

A = distance.euclidean(eye[1], eye[5])

B = distance.euclidean(eye[2], eye[4])

C = distance.euclidean(eye[0], eye[3])

ear = (A + B) / (2.0 \* C)

return ear

def detect\_face(frame):

#Cascade classifier pre-trained model

cascPath = 'Models/face\_landmarks.dat'

faceCascade = cv2.CascadeClassifier(cascPath)

#BGR -> Gray conversion

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

#Cascade MultiScale classifier

detected\_faces = faceCascade.detectMultiScale(gray,scaleFactor=1.1,minNeighbors=6,

minSize=(shape\_x, shape\_y),

flags=cv2.CASCADE\_SCALE\_IMAGE)

coord = []

for x, y, w, h in detected\_faces :

if w > 100 :

sub\_img=frame[y:y+h,x:x+w]

cv2.rectangle(frame,(x,y),(x+w,y+h),(0, 255,255),1)

coord.append([x,y,w,h])

return gray, detected\_faces, coord

def extract\_face\_features(faces, offset\_coefficients=(0.075, 0.05)):

gray = faces[0]

detected\_face = faces[1]

new\_face = []

for det in detected\_face :

#Region dans laquelle la face est détectée

x, y, w, h = det

#X et y correspondent à la conversion en gris par gray, et w, h correspondent à la hauteur/largeur

#Offset coefficient, np.floor takes the lowest integer (delete border of the image)

horizontal\_offset = np.int(np.floor(offset\_coefficients[0] \* w))

vertical\_offset = np.int(np.floor(offset\_coefficients[1] \* h))

#gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

#gray transforme l'image

extracted\_face = gray[y+vertical\_offset:y+h, x+horizontal\_offset:x-horizontal\_offset+w]

#Zoom sur la face extraite

new\_extracted\_face = zoom(extracted\_face, (shape\_x / extracted\_face.shape[0],shape\_y / extracted\_face.shape[1]))

#cast type float

new\_extracted\_face = new\_extracted\_face.astype(np.float32)

#scale

new\_extracted\_face /= float(new\_extracted\_face.max())

#print(new\_extracted\_face)

new\_face.append(new\_extracted\_face)

return new\_face

(lStart, lEnd) = face\_utils.FACIAL\_LANDMARKS\_IDXS["left\_eye"]

(rStart, rEnd) = face\_utils.FACIAL\_LANDMARKS\_IDXS["right\_eye"]

(nStart, nEnd) = face\_utils.FACIAL\_LANDMARKS\_IDXS["nose"]

(mStart, mEnd) = face\_utils.FACIAL\_LANDMARKS\_IDXS["mouth"]

(jStart, jEnd) = face\_utils.FACIAL\_LANDMARKS\_IDXS["jaw"]

(eblStart, eblEnd) = face\_utils.FACIAL\_LANDMARKS\_IDXS["left\_eyebrow"]

(ebrStart, ebrEnd) = face\_utils.FACIAL\_LANDMARKS\_IDXS["right\_eyebrow"]

model = load\_model('Models/video.h5')

face\_detect = dlib.get\_frontal\_face\_detector()

predictor\_landmarks = dlib.shape\_predictor("Models/face\_landmarks.dat")

#Lancer la capture video

video\_capture = cv2.VideoCapture(0)

while True:

# Capture frame-by-frame

ret, frame = video\_capture.read()

face\_index = 0

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

rects = face\_detect(gray, 1)

#gray, detected\_faces, coord = detect\_face(frame)

for (i, rect) in enumerate(rects):

shape = predictor\_landmarks(gray, rect)

shape = face\_utils.shape\_to\_np(shape)

# Identify face coordinates

(x, y, w, h) = face\_utils.rect\_to\_bb(rect)

face = gray[y:y+h,x:x+w]

#Zoom on extracted face

face = zoom(face, (shape\_x / face.shape[0],shape\_y / face.shape[1]))

#Cast type float

face = face.astype(np.float32)

#Scale

face /= float(face.max())

face = np.reshape(face.flatten(), (1, 48, 48, 1))

#Make Prediction

prediction = model.predict(face)

prediction\_result = np.argmax(prediction)

# Rectangle around the face

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

cv2.putText(frame, "Face #{}".format(i + 1), (x - 10, y - 10), cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, (0, 255, 0), 2)

for (j, k) in shape:

cv2.circle(frame, (j, k), 1, (0, 0, 255), -1)

# 1. Add prediction probabilities

cv2.putText(frame, "----------------",(40,100 + 180\*i), cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, 155, 0)

cv2.putText(frame, "Emotional report : Face #" + str(i+1),(40,120 + 180\*i), cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, 155, 0)

cv2.putText(frame, "Angry : " + str(round(prediction[0][0],3)),(40,140 + 180\*i), cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, 155, 0)

cv2.putText(frame, "Disgust : " + str(round(prediction[0][1],3)),(40,160 + 180\*i), cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, 155, 0)

cv2.putText(frame, "Fear : " + str(round(prediction[0][2],3)),(40,180 + 180\*i), cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, 155, 1)

cv2.putText(frame, "Happy : " + str(round(prediction[0][3],3)),(40,200 + 180\*i), cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, 155, 1)

cv2.putText(frame, "Sad : " + str(round(prediction[0][4],3)),(40,220 + 180\*i), cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, 155, 1)

cv2.putText(frame, "Surprise : " + str(round(prediction[0][5],3)),(40,240 + 180\*i), cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, 155, 1)

cv2.putText(frame, "Neutral : " + str(round(prediction[0][6],3)),(40,260 + 180\*i), cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, 155, 1)

# 2. Annotate main image with a label

if prediction\_result == 0 :

cv2.putText(frame, "Angry",(x+w-10,y-10), cv2.FONT\_HERSHEY\_SIMPLEX, 1, (0, 255, 0), 2)

elif prediction\_result == 1 :

cv2.putText(frame, "Disgust",(x+w-10,y-10), cv2.FONT\_HERSHEY\_SIMPLEX, 1, (0, 255, 0), 2)

elif prediction\_result == 2 :

cv2.putText(frame, "Fear",(x+w-10,y-10), cv2.FONT\_HERSHEY\_SIMPLEX, 1, (0, 255, 0), 2)

elif prediction\_result == 3 :

cv2.putText(frame, "Happy",(x+w-10,y-10), cv2.FONT\_HERSHEY\_SIMPLEX, 1, (0, 255, 0), 2)

elif prediction\_result == 4 :

cv2.putText(frame, "Sad",(x+w-10,y-10), cv2.FONT\_HERSHEY\_SIMPLEX, 1, (0, 255, 0), 2)

elif prediction\_result == 5 :

cv2.putText(frame, "Surprise",(x+w-10,y-10), cv2.FONT\_HERSHEY\_SIMPLEX, 1, (0, 255, 0), 2)

else :

cv2.putText(frame, "Neutral",(x+w-10,y-10), cv2.FONT\_HERSHEY\_SIMPLEX, 1, (0, 255, 0), 2)

# 3. Eye Detection and Blink Count

leftEye = shape[lStart:lEnd]

rightEye = shape[rStart:rEnd]

# Compute Eye Aspect Ratio

leftEAR = eye\_aspect\_ratio(leftEye)

rightEAR = eye\_aspect\_ratio(rightEye)

ear = (leftEAR + rightEAR) / 2.0

# And plot its contours

leftEyeHull = cv2.convexHull(leftEye)

rightEyeHull = cv2.convexHull(rightEye)

#cv2.drawContours(frame, [leftEyeHull], -1, (0, 255, 0), 1)

#cv2.drawContours(frame, [rightEyeHull], -1, (0, 255, 0), 1)

# 4. Detect Nose

nose = shape[nStart:nEnd]

noseHull = cv2.convexHull(nose)

#cv2.drawContours(frame, [noseHull], -1, (0, 255, 0), 1)

# 5. Detect Mouth

mouth = shape[mStart:mEnd]

mouthHull = cv2.convexHull(mouth)

#cv2.drawContours(frame, [mouthHull], -1, (0, 255, 0), 1)

# 6. Detect Jaw

jaw = shape[jStart:jEnd]

jawHull = cv2.convexHull(jaw)

#cv2.drawContours(frame, [jawHull], -1, (0, 255, 0), 1)

# 7. Detect Eyebrows

ebr = shape[ebrStart:ebrEnd]

ebrHull = cv2.convexHull(ebr)

#cv2.drawContours(frame, [ebrHull], -1, (0, 255, 0), 1)

ebl = shape[eblStart:eblEnd]

eblHull = cv2.convexHull(ebl)

#cv2.drawContours(frame, [eblHull], -1, (0, 255, 0), 1)

cv2.putText(frame,'Number of Faces : ' + str(len(rects)),(40, 40), cv2.FONT\_HERSHEY\_SIMPLEX, 1, 155, 1)

cv2.imshow('Video', frame)

if cv2.waitKey(1) & 0xFF == ord('q'):

break

# When everything is done, release the capture

video\_capture.release()

cv2.destroyAllWindows()

def main():

show\_webcam()

if \_name\_ == "\_main\_":

main()

# CHAPTER 4: RESULTS

A screenshot of a computer

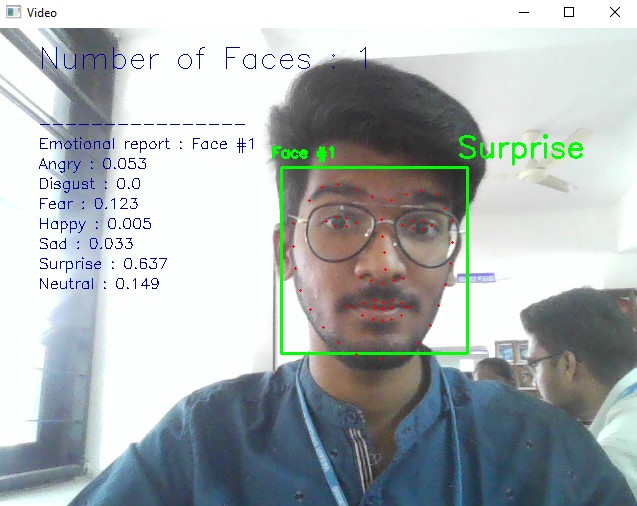
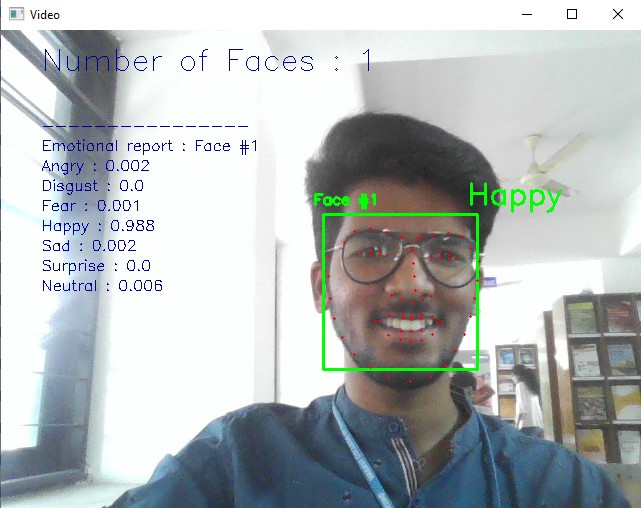
Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a video game

Description automatically generated



# CHAPTER 5:

# DISCUSSION AND FUTURE WORK

* Our project Emotion Recognizer helps to detect human emotions through machines by allowing user to record live video or audio, which increases the human-machine interactions.
* This project has a future scope of increase in accuracy of detecting the emotions.

# CHAPTER 6: REFERENCES

<https://www.overleaf.com/read/xvtrrfpvzwhf>

[The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) | Zenodo](https://zenodo.org/record/1188976/?f=3.XAcEs5NKhQK#.Y2FhCmlBy5c)

[Challenges in Representation Learning: Facial Expression Recognition Challenge | Kaggle](https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data)

<https://github.com/maelfabien/Multimodal-Emotion-Recognition>