AFFECTIVE ANALYSIS

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

Under the guidance of
K. Srinivasa Chakravarthy
(Assistant Professor)

Department of Information Technology

Vasavi College of Engineering

(Autonomous)

VASAVI COLLEGE OF ENGINEERING (AUTONOMOUS)

(AFFILIATED TO OSMANIA UNIVERSITY)

HYDERABAD - 500 031

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 **"AFFECTIVE ANALYSIS"** 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 universityor 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 - 500 031

Department of Information Technology

DECLARATION BY CANDIDATE

I, **R. SAI SATHVIK** bearing hallticket number, **1602-20-737-035** hereby declare that the project report entitled **"AFFECTIVE ANALYSIS"** 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 universityor institute for the award of any other degree or diploma.

R. SAI SATHVIK
1602-20-737-035

VASAVI COLLEGE OF ENGINEERING (AUTONOMOUS)

(AFFILIATED TO OSMANIA UNIVERSITY)

HYDERABAD - 500 031

Department of Information Technology

DECLARATION BY CANDIDATE

I, **B. SURESH KUMAR** bearing hallticket number, **1602-20-737-051** hereby declare that the project report entitled **"AFFECTIVE ANALYSIS"** 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 universityor institute for the award of any other degree or diploma.

B. SURESH KUMAR

1602-20-737-051

VASAVI COLLEGE OF ENGINEERING (AUTONOMOUS)

(AFFILIATED TO OSMANIA UNIVERSITY)

HYDERABAD - 500 031

Department of Information Technology

DECLARATION BY CANDIDATE

I, **M. SAI LAXMI** bearing hallticket number, **1602-20-737-034** hereby declare that the project report entitled **"AFFECTIVE ANALYSIS"** 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 universityor institute for the award of any other degree or diploma.

M. SAI LAXMI

1602-20-737-034

VASAVI COLLEGE OF ENGINEERING (AUTONOMOUS)

(AFFILIATED TO OSMANIA UNIVERSITY)

HYDERABAD - 500 031

Department of Information Technology

BONAFIDE CERTIFICATE

This is to certify that the project entitled "**AFFECTIVE ANALYSIS**" 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 completionof THEME BASED PROJECT of Bachelor of Engineering in Information Technology is a record of bonafide work carried out by them under my guidance.

Mr. K. Chakravarthy — Internal Guide

External Examiner

Dr.K Ram Mohan Rao — HOD, IT

ACKNOWLEDGEMENT

We thank the department of INFORMATION TECHNOLOGY, for introducing thesubject “Theme Based Project” in BE sixth semester.

We would also like to show our appreciation to our Honorable principal, Dr S V Ramana sir, our HOD K. Ram Mohan Rao sir for supporting us and our mini project lecturer, Mr. K. Chakravarthy sir, for letting us properly understand the process ofdoing a project and for providing valuable insight and expertise that has greatlyassisted us in the making of the project.

Table of Contents:

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 Affective Analysis?

Affective Analysis 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 Affective Analysis?

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

1.1 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.
Detect emotion in video and audio: In similar audience to above two methods we wish to integrate both the models in order to gain a overall accurate model taking into account of audio and video.

1.2 INTENDED AUDIENCE:

The intended audience for this project is everyone who wants to detect emotions of audience.

1.3 PRODUCT SCOPE:

Affective Analysis 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

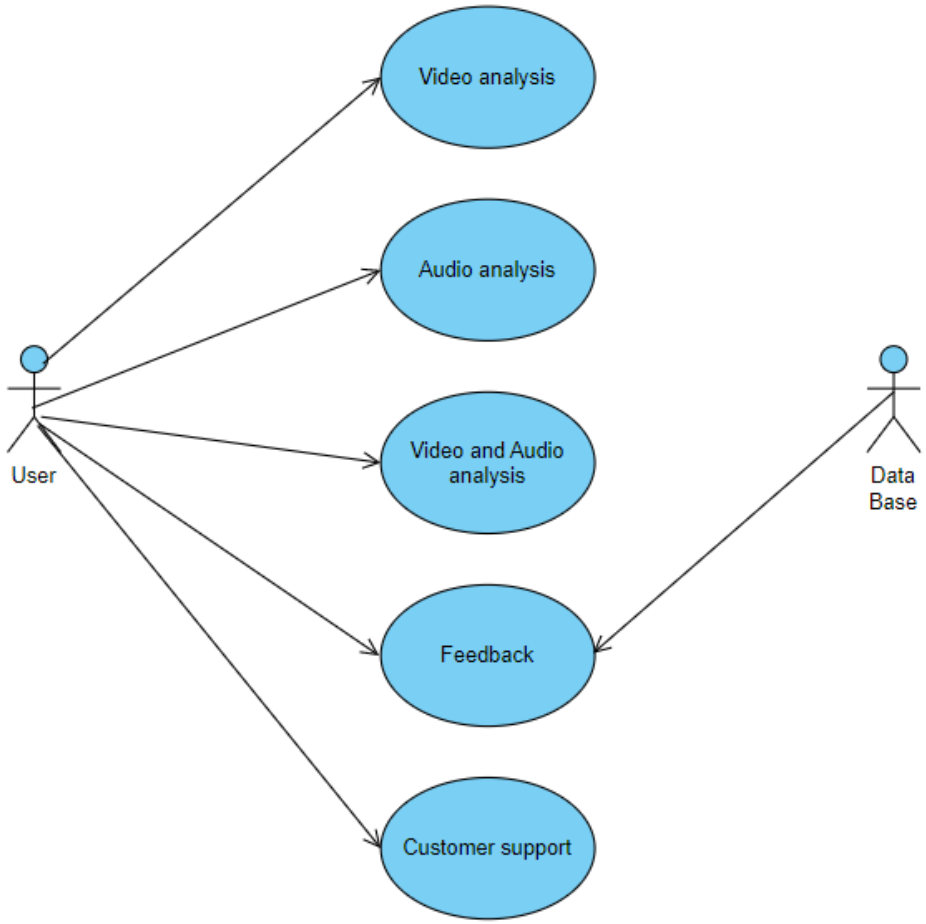
Through various modes such as audio and video different emotions such as anger, happiness, excitement, sadness, frustration, fear, surprise and neutral state are detected using deep learning algorithms.

CHAPTER 2: RELATED WORK

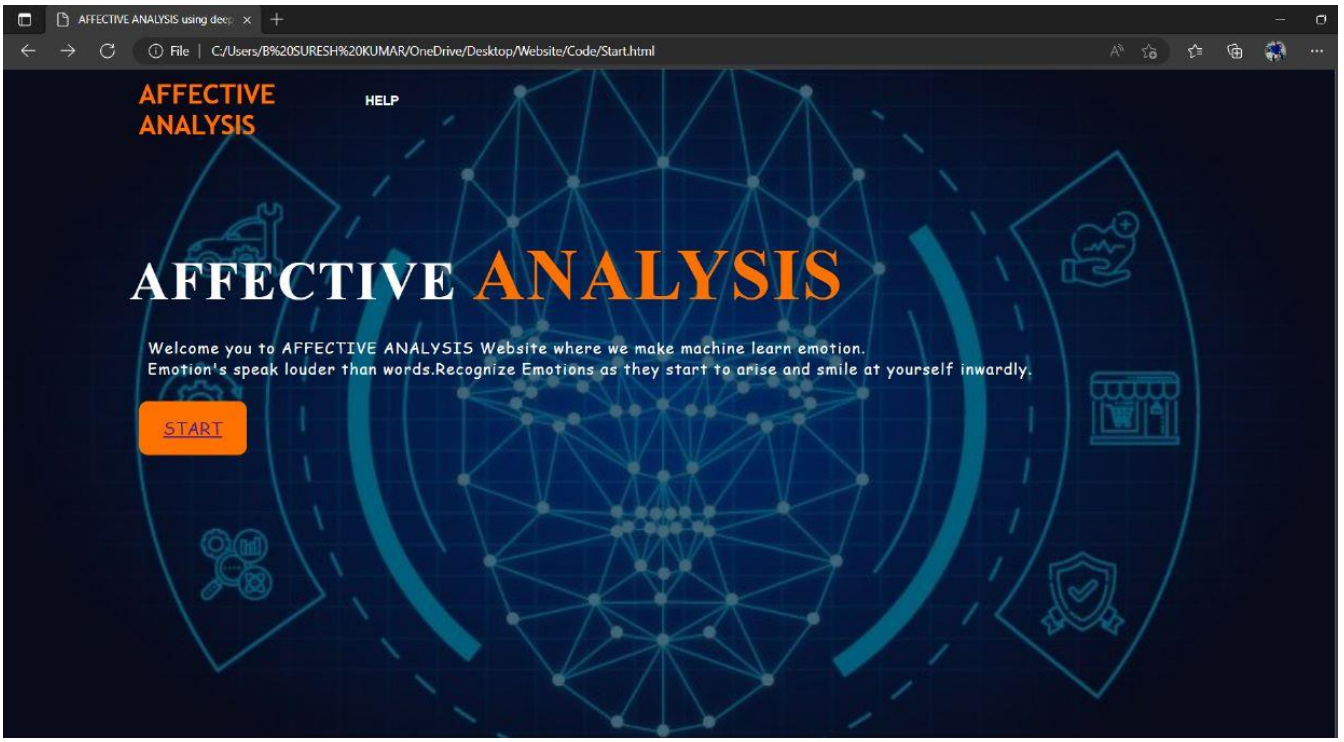
Affective Analysis through Audio signal is a recent research topic in the Human Computer Interaction. The demand was risen for increasing communication interface between the humans and digital media. Many researchers are currently working in order to improve their accuracy. But still there is a lack 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.

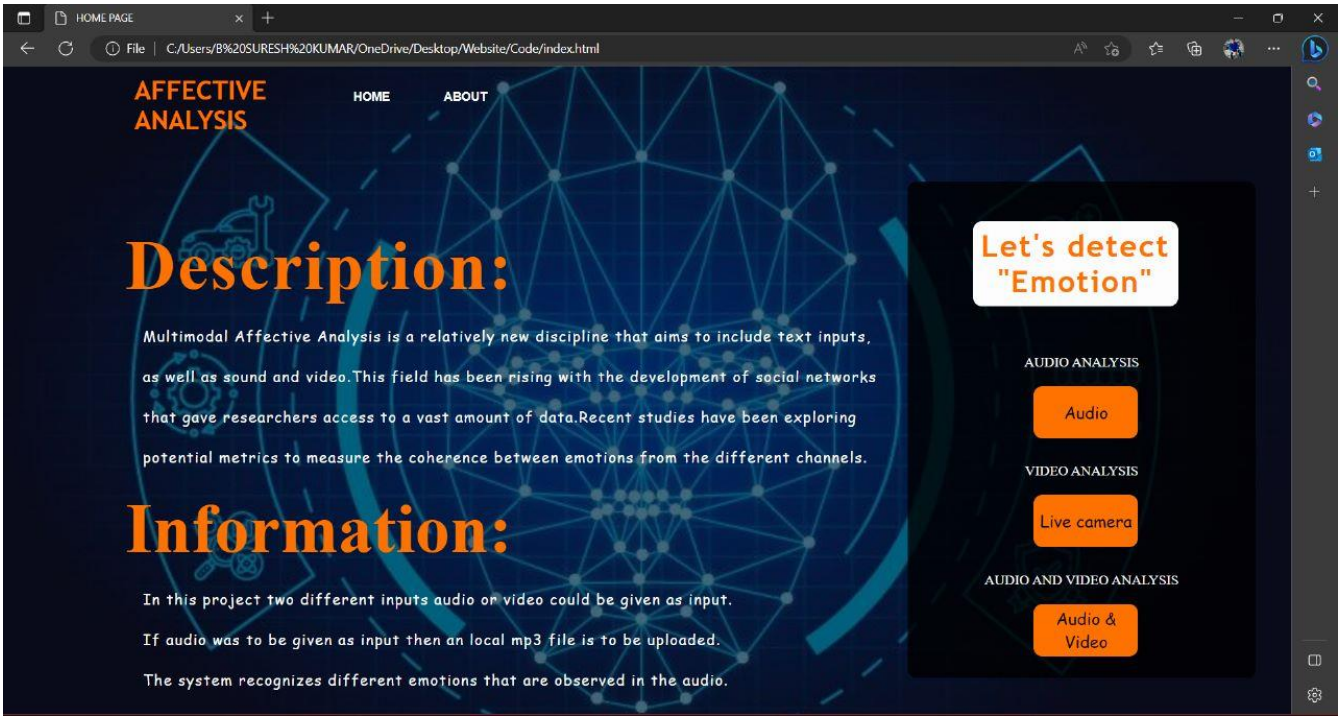
CHAPTER 3: PROPOSED WORK

3.1 Use cases:

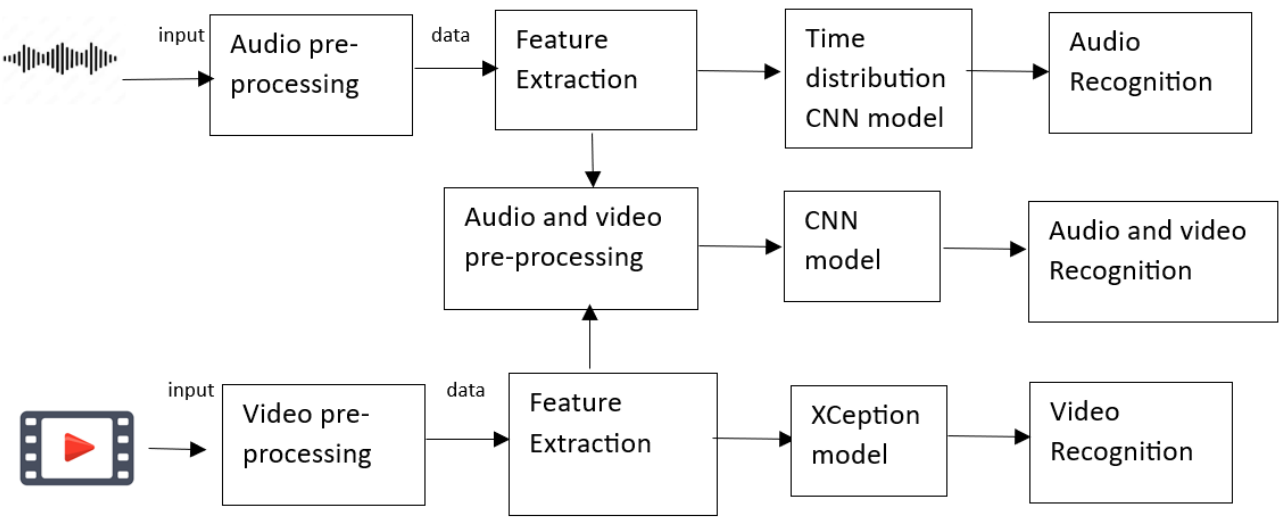


3.2 UI prototypes or screenshots:





3.3 Architecture and Technology used:



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.

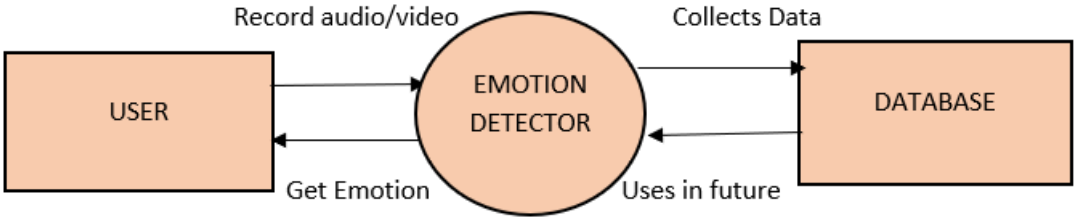
User- A user is any human being who uses emotion detection technology. They can play any role: detect emotion through audio or detect emotion through video. As long as they are human, they are termed 'user'.

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.

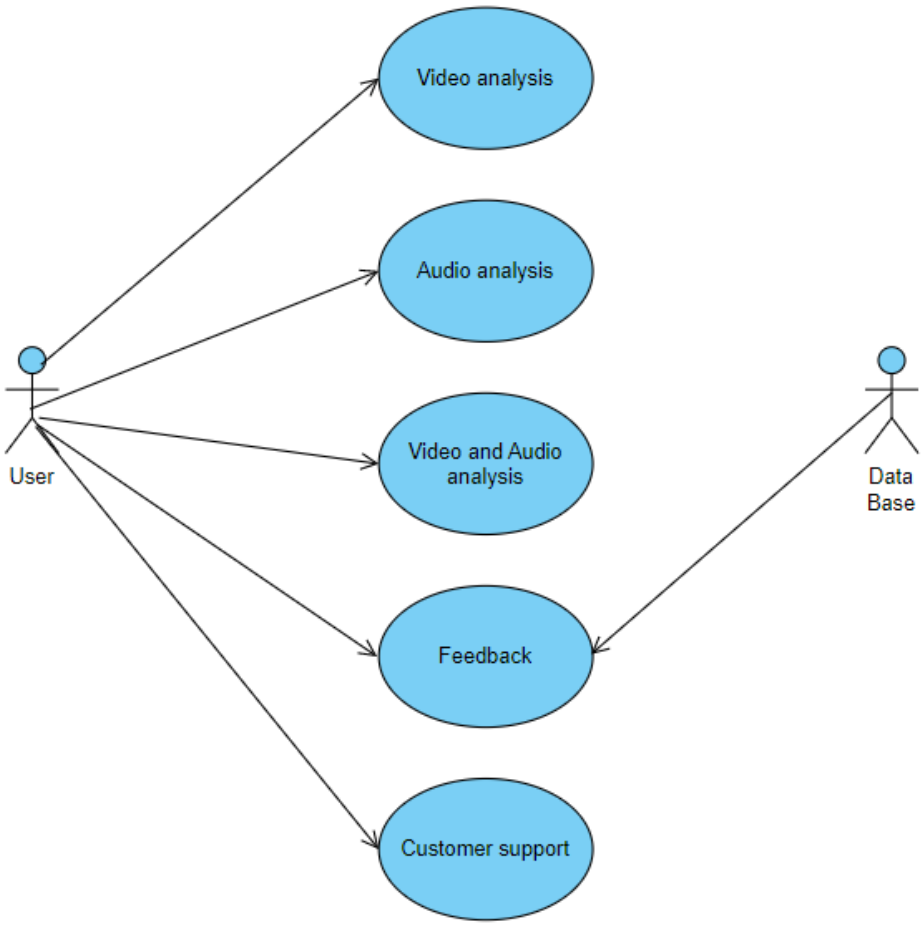
Response: This is the emotion recognizer output that is aimed at satisfying the user's intent. The most accurate responses occur when a proper range of emotions have been correctly grouped. Accurate and simple responses are important traits for a good emotion detector.

3.4 Design:

3.4.1 DATA FLOW DIAGRAM:



3.4.2 USE CASE DIAGRAM:



3.5 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>AFFECTIVE ANALYSIS</title>
   <link rel="stylesheet" href="style.css">
</head>
<body>
   <div class="main">
      <div class="navbar">
         <div class="icon">
            <h2 class="logo">AFFECTIVE ANALYSIS</h2>
         </div>

         <div class="menu">
            <ul>
               <li><a href="/HELP">HELP</a></li>
            </ul>
         </div>
      </div>
      <div class="content">
         <h1>AFFECTIVE <span>ANALYSIS</span></h1>
         <p class="par"><br> Welcome you to AFFECTIVE ANALYSIS Website where
we make machine learn emotion.
            <br>Emotion's speak louder than words.Recognize Emotions as they start to
arise and smile
            at yourself inwardly.</p></font>

         <button class="cn"><a href="/main">START</button>
      </div>
   </div>
</body>
</html>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
   <title>Affective Analysis - Help</title>
   <link rel="stylesheet" href="style.css">
</head>
<body>
   <div class="main">
      <div class="navbar">
         <div class="icon">
            <h2 class="logo">Affective Analysis</h2>
         </div>
      </div>
   </div>
   <div class="content">
      <h1>INFO</h1>
      <p class="par"><br></p>
   </div>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Affective Analysis</title>
</head>
<body>
   <h1>Audio Analysis</h1>
   <table>
     <tbody>
       {% for row in data %}
         <tr>
           <td>{{ row[0] }}</td>
           <td>{{ row[1] }}</td>
         </tr>
       {% endfor %}
     </tbody>
   </table>
</body>
</html>
```

```
from flask import Flask, render_template#,Response

app = Flask(_name_)

@app.route("/")
def home():
    return render_template("Start.html")

@app.route("/main")
def main():
    return render_template("index.html")

@app.route("/HELP")
def help():
    return render_template("HELP.html")

@app.route("/about")
def about():
    return render_template("ABOUT.html")
import video,audio



from app import app

from flask import render_template,Response

import pandas as pd

from SpeechEmotionRecognition import speechEmotionRecognition
import os
import csv

from watchdog.observers import Observer
from watchdog.events import FileSystemEventHandler

class CSVHandler(FileSystemEventHandler):
    def on_modified(self, event):
        if event.src_path.endswith('.csv'):
            with open(event.src_path, 'r') as f:
                reader = csv.reader(f)
                data = list(reader)
            return render_template('AudioAnalysis.html', data=data)

@app.route('/main/audio')
```

```
def audio():
    speechEmotionRecognition(os.path.join('Models', 'audio.hdf5'))
    with open('data.csv', 'r') as f:
        reader = csv.reader(f)
        data = list(reader)
    return render_template('AudioAnalysis.html', data=data)


event_handler = CSVHandler()
observer = Observer()
observer.schedule(event_handler, path='.', recursive=False)
observer.start()



## Basics ##
import time
import os
import numpy as np

## Audio Preprocessing ##
import pyaudio
import wave
import librosa
from scipy.stats import zscore

## Time Distributed CNN ##
import tensorflow as tf
from tensorflow.keras import backend as K
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input, Dense, Dropout, Activation, TimeDistributed
from tensorflow.keras.layers import Conv2D, MaxPooling2D, BatchNormalization,
Flatten
from tensorflow.keras.layers import LSTM


'''
Speech Emotion Recognition
'''
class speechEmotionRecognition:

    '''
    Voice recording function
    '''
    def _init_(self, subdir_model=None):
```

```
        # Load prediction model
        if subdir_model is not None:
            self._model = self.build_model()
            self._model.load_weights(subdir_model)
        # Emotion encoding
        self._emotion = {0:'Angry', 1:'Disgust', 2:'Fear', 3:'Happy', 4:'Neutral', 5:'Sad',
6:'Surprise'}
        self.voice_recording(filename = "audio.wav", duration=8)
        predict,timestamp = self.predict_emotion_from_file("audio.wav")
        # print(predict.shape())
        # print(timestamp.shape())
        self.prediction_to_csv(predictions=predict,timestamp = timestamp, filename=
"data.csv")
        for emotion in predict:
            print(str(emotion))
        # for t in timestamp:
        #     print(str(t))

    '''
    Voice recording function
    '''
    def voice_recording(self, filename, duration=5, sample_rate=16000, chunk=1024,
channels=1):

        # Start the audio recording stream
        p = pyaudio.PyAudio()
        stream = p.open(format=pyaudio.paInt16,
                        channels=channels,
                        rate=sample_rate,
                        input=True,
                        frames_per_buffer=chunk)

        # Create an empty list to store audio recording
        frames = []

        # Determine the timestamp of the start of the response interval
        print('* Start Recording *')
        stream.start_stream()
        start_time = time.time()
        current_time = time.time()

        # Record audio until timeout
        while (current_time - start_time) < duration:

            # Record data audio data
            data = stream.read(chunk)
```

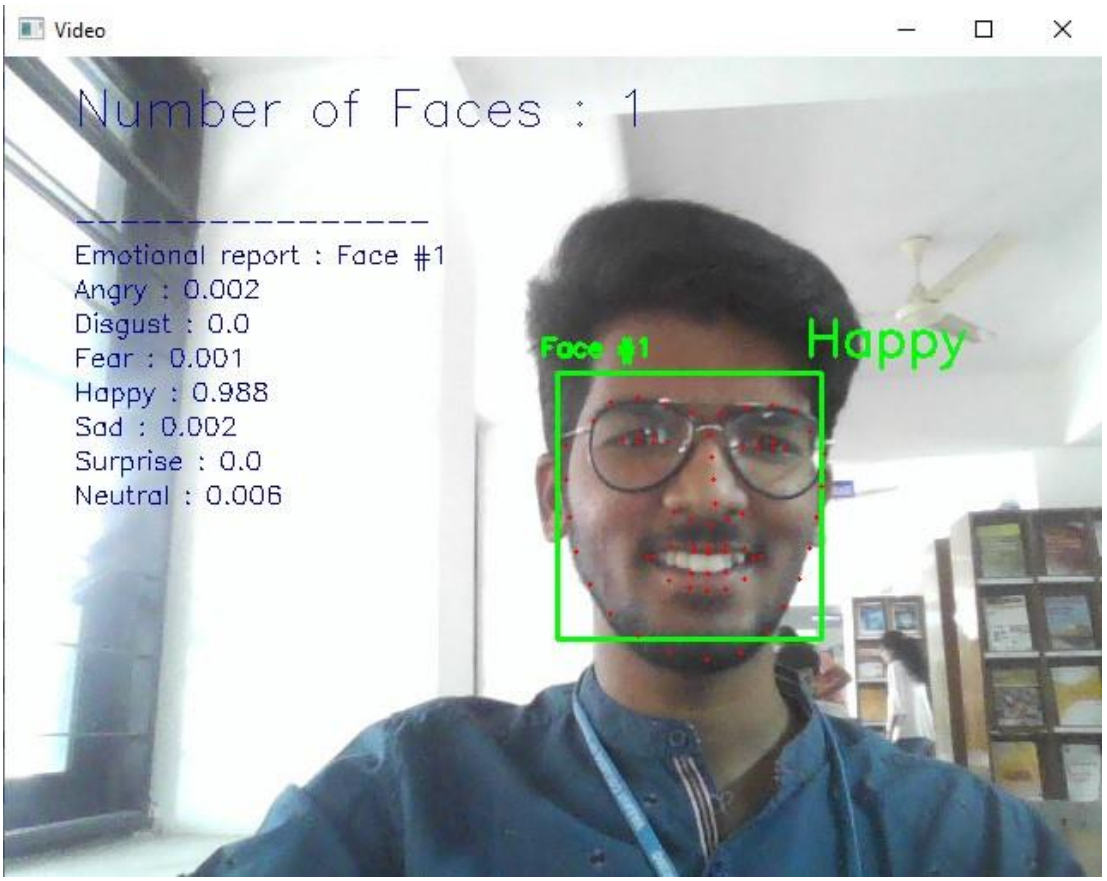
```
            # Add the data to a buffer (a list of chunks)
            frames.append(data)

            # Get new timestamp
            current_time = time.time()

        # Close the audio recording stream
        stream.stop_stream()
        stream.close()
        p.terminate()
        print('* End Recording * ')

        # Export audio recording to wav format
        wf = wave.open(filename, 'w')
        wf.setnchannels(channels)
        wf.setsampwidth(p.get_sample_size(pyaudio.paInt16))
        wf.setframerate(sample_rate)
        wf.writeframes(b''.join(frames))
        wf.close()


    '''
    Mel-spectogram computation
    '''
    def mel_spectrogram(self, y, sr=16000, n_fft=512, win_length=256, hop_length=128,
window='hamming', n_mels=128, fmax=4000):

        # Compute spectogram
        mel_spect = np.abs(librosa.stft(y, n_fft=n_fft, window=window,
win_length=win_length, hop_length=hop_length)) ** 2

        # Compute mel spectrogram
        mel_spect = librosa.feature.melspectrogram(S=mel_spect, sr=sr, n_mels=n_mels,
fmax=fmax)

        # Compute log-mel spectrogram
        mel_spect = librosa.power_to_db(mel_spect, ref=np.max)

        return np.asarray(mel_spect)


    '''
    Audio framing
    '''
    def frame(self, y, win_step=64, win_size=128):
```

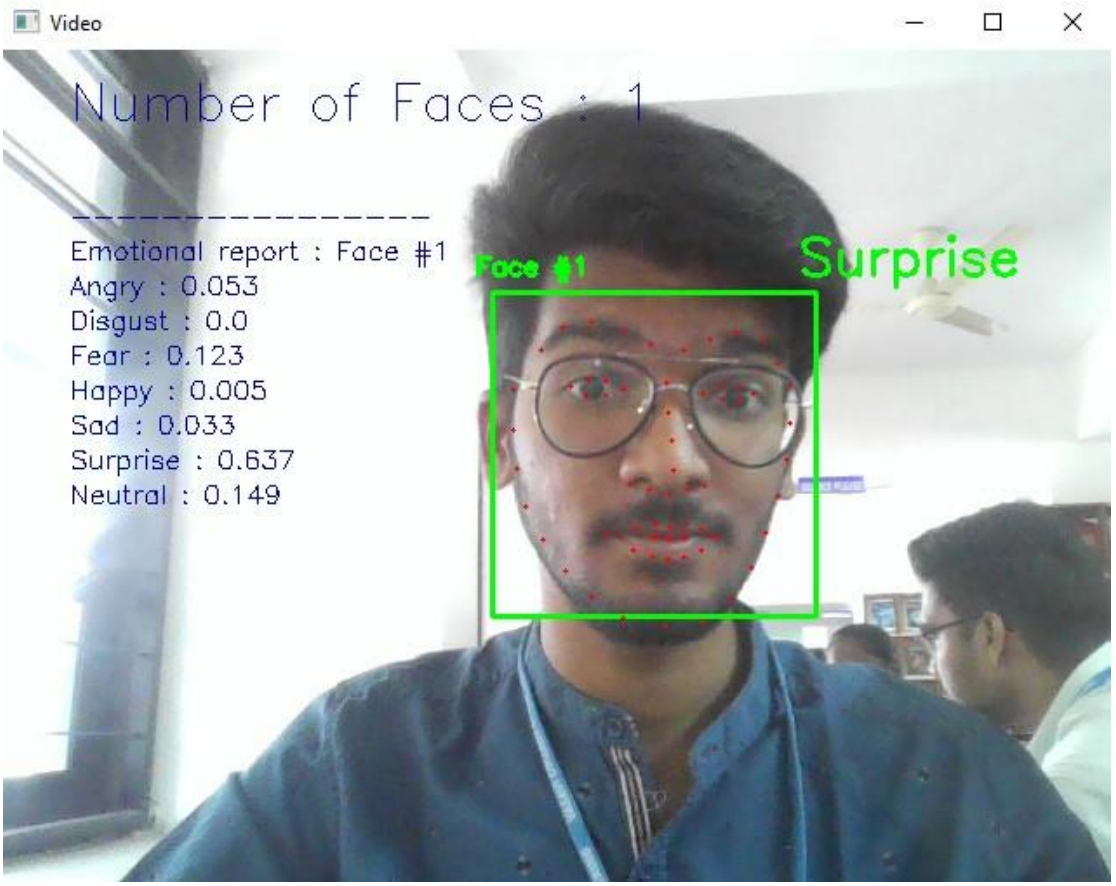
```
        # Number of frames
        nb_frames = 1 + int((y.shape[2] - win_size) / win_step)

        # Framming
        frames = np.zeros((y.shape[0], nb_frames, y.shape[1], win_size)).astype(np.float16)
        for t in range(nb_frames):
            frames[:,t,:,:] = np.copy(y[:,:,(t * win_step):(t * win_step +
win_size)]).astype(np.float16)

        return frames


    '''
    Time distributed Convolutional Neural Network model
    '''
    def build_model(self):

        # Clear Keras session
        K.clear_session()

        # Define input
        input_y = Input(shape=(5, 128, 128, 1), name='Input_MELSPECT')

        # First LFLB (local feature learning block)
        y = TimeDistributed(Conv2D(64, kernel_size=(3, 3), strides=(1, 1),
padding='same'), name='Conv_1_MELSPECT')(input_y)
        y = TimeDistributed(BatchNormalization(), name='BatchNorm_1_MELSPECT')(y)
        y = TimeDistributed(Activation('elu'), name='Activ_1_MELSPECT')(y)
        y = TimeDistributed(MaxPooling2D(pool_size=(2, 2), strides=(2, 2),
padding='same'), name='MaxPool_1_MELSPECT')(y)
        y = TimeDistributed(Dropout(0.2), name='Drop_1_MELSPECT')(y)

        # Second LFLB (local feature learning block)
        y = TimeDistributed(Conv2D(64, kernel_size=(3, 3), strides=(1, 1),
padding='same'), name='Conv_2_MELSPECT')(y)
        y = TimeDistributed(BatchNormalization(), name='BatchNorm_2_MELSPECT')(y)
        y = TimeDistributed(Activation('elu'), name='Activ_2_MELSPECT')(y)
        y = TimeDistributed(MaxPooling2D(pool_size=(4, 4), strides=(4, 4),
padding='same'), name='MaxPool_2_MELSPECT')(y)
        y = TimeDistributed(Dropout(0.2), name='Drop_2_MELSPECT')(y)

        # Third LFLB (local feature learning block)
        y = TimeDistributed(Conv2D(128, kernel_size=(3, 3), strides=(1, 1),
padding='same'), name='Conv_3_MELSPECT')(y)
        y = TimeDistributed(BatchNormalization(), name='BatchNorm_3_MELSPECT')(y)
```

```
        y = TimeDistributed(Activation('elu'), name='Activ_3_MELSPECT')(y)
        y = TimeDistributed(MaxPooling2D(pool_size=(4, 4), strides=(4, 4),
padding='same'), name='MaxPool_3_MELSPECT')(y)
        y = TimeDistributed(Dropout(0.2), name='Drop_3_MELSPECT')(y)

        # Fourth LFLB (local feature learning block)
        y = TimeDistributed(Conv2D(128, kernel_size=(3, 3), strides=(1, 1),
padding='same'), name='Conv_4_MELSPECT')(y)
        y = TimeDistributed(BatchNormalization(), name='BatchNorm_4_MELSPECT')(y)
        y = TimeDistributed(Activation('elu'), name='Activ_4_MELSPECT')(y)
        y = TimeDistributed(MaxPooling2D(pool_size=(4, 4), strides=(4, 4),
padding='same'), name='MaxPool_4_MELSPECT')(y)
        y = TimeDistributed(Dropout(0.2), name='Drop_4_MELSPECT')(y)

        # Flat
        y = TimeDistributed(Flatten(), name='Flat_MELSPECT')(y)

        # LSTM layer
        y = LSTM(256, return_sequences=False, dropout=0.2, name='LSTM_1')(y)

        # Fully connected
        y = Dense(7, activation='softmax', name='FC')(y)

        # Build final model
        model = Model(inputs=input_y, outputs=y)

        return model


    '''
    Predict speech emotion over time from an audio file
    '''
    def predict_emotion_from_file(self, filename, chunk_step=16000, chunk_size=49100,
predict_proba=False, sample_rate=16000):

        # Read audio file
        y, sr = librosa.core.load(filename, sr=sample_rate, offset=0.5)

        # Split audio signals into chunks
        chunks = self.frame(y.reshape(1, 1, -1), chunk_step, chunk_size)

        # Reshape chunks
        chunks = chunks.reshape(chunks.shape[1],chunks.shape[-1])

        # Z-normalization
        y = np.asarray(list(map(zscore, chunks)))
```

```
        # Compute mel spectrogram
        mel_spect = np.asarray(list(map(self.mel_spectrogram, y)))

        # Time distributed Framing
        mel_spect_ts = self.frame(mel_spect)

        # Build X for time distributed CNN
        X = mel_spect_ts.reshape(mel_spect_ts.shape[0],
                                 mel_spect_ts.shape[1],
                                 mel_spect_ts.shape[2],
                                 mel_spect_ts.shape[3],
                                 1)

        # Predict emotion
        if predict_proba is True:
            predict = self._model.predict(X)
        else:
            predict = np.argmax(self._model.predict(X), axis=1)
            predict = [self._emotion.get(emotion) for emotion in predict]

        # Clear Keras session
        K.clear_session()

        # Predict timestamp
        timestamp = np.concatenate([[chunk_size], np.ones((len(predict) - 1)) *
chunk_step]).cumsum()
        timestamp = np.round(timestamp / sample_rate)

        return [predict, timestamp]

    '''
    Export emotions predicted to csv format
    '''
    def prediction_to_csv(self, predictions, timestamp, filename, mode='w'):

        # Write emotion in filename
        with open(filename, mode) as f:
            if mode == 'w':
                f.write("EMOTIONS, TIMESTAMP"+'\n')
            for i in range(len(predictions)):
                f.write(str(predictions[i])+", "+str(timestamp[i])+'\n')
            f.close()

# speechEmotionRecognition(os.path.join('Models', 'audio.hdf5'))
```

CHAPTER 4: RESULTS





CHAPTER 5:
DISCUSSION AND FUTURE WORK

- Our project Affective Analysis 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.
- This project can also be extended to better accuracy by integrating human neural pulses and understand them further and improve it.

CHAPTER 6: REFERENCES

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

The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) | Zenodo

Challenges in Representation Learning: Facial Expression Recognition Challenge | Kaggle

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