RV Educational Institutions ®
RV College of Engineering ®

Autonomous Institution Affiliated to Visvesvaraya Technological University, Belagavi

Approved by AICTE, New Delhi

Go, change the world

A Project Report on

SPEECH EMOTION RECOGNITION USING MLP CLASSIFIER

Submitted in Partial Fulfilment of the Requirement

for the IV Semester MCA Academic Minor Project – I 18MCA46

MASTER OF COMPUTER APPLICATIONS
By

USN	STUDENT NAME
1RV19MCA21	CHAITRA B V
1RV19MCA35	JANVI NAGESH NAIK

Under the Guidance of

Dr S Anupama Kumar

Associate Professor

Department of MCA

RV College of Engineering®

Department of Master of Computer Applications
RV College of Engineering®,
Accredited by National Board of Accreditation,
Mysuru Road
RV Vidyanikethan Post, Bengaluru – 560059

June -2021

RV Educational Institutions®
RV College of Engineering®
Autonomous Institution Affiliated to Visvesvaraya Technological University, Belagavi
Approved by AICTE, New Delhi

Go, change the world

DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

CERTIFICATE

This is to certify that the project entitled "SPEECH EMOTION RECOGNITION USING MLP CLASSIFIER" submitted in partial fulfillment of Minor Project-I (18MCA46) of IV Semester MCA is a result of the bonafide work carried out by CHAITRA B V-1RV19MCA21 and JANVI NAGESH NAIK -1RV19MCA35 during the Academic year 2020-21.

Dr S Anupama Kumar	Dr. Andhe Dharani
Associate Professor	Professor and Director
Department of MCA,	Department of MCA,
RV College of Engineering®	RV College of Engineering®

UNDERTAKING BY THE STUDENT

We, CHAITRA B V-1RV19MCA21 and JANVI NAGESH NAIK-1RV19MCA35, hereby declare that the Minor project-I SPEECH EMOTION RECOGNISITION USING MLP CLASSIFIER is carried out and completed successfully by us and is our original work.

SIGNATURE

(CHAITRA B V)

(JANVI NAGESH NAIK)

Acknowledgement

We would like to thank all those who are involved in this endeavour, for their kind cooperation for its successful completion. At the outset, we wish to express our sincere gratitude to all those people who have helped us to complete this project in an efficient manner.

We offer our special thanks to our Project guide Dr S Anupama Kumar Associate Professor, Department of MCA, RVCE without whose help and support this project would not have been this success.

Most of all and more than ever, we would like to thank our family members for their warmness, support, encouragement, kindness and patience. We are really thankful to all our friends who always advised and motivated us throughout the course.

Abstract

Speech Emotion Recognition, abbreviated as SER, is the act of attempting to recognize human emotion and affective states from speech. This is capitalizing on the fact that voice often reflects underlying emotion through tone and pitch. Speech Emotion Recognition can be used by multiple industries to offer different services like marketing company suggesting you to buy products based on your emotions, automotive industry can detect the persons emotions and adjust the speed of autonomous cars as required to avoid any collisions etc.

Speech Emotion Recognition involves speech feature extraction and voice activity detection. The process involves Using MLP Classifier for analysing speech features to include tone, energy, pitch, format frequency, etc. and identifying emotions through changes in these. Nowadays personalization is something that is needed in all the things experienced every day. This Project will be implemented in python using resources such as Anaconda for Python 3.6.5 and Spyder. The library such as librosa, soundfile, and sklearn (among others) to build a model using an MLP Classifier. This will recognize emotion from sound files. Data will be loaded and features will be extracted from it, then dataset will be divided into training and testing sets. Then, it will be initialized by MLP Classifier and model will be trained. Finally, the accuracy of the model will be calculated.

Detection and Analysis of Emotion from Speech Signals will improvise man-machine interface. This project builds a model that could detect emotions from the speech. It can also be used to monitor the psycho physiological state of a person in lie detectors. In recent time, Speech Emotion Recognition also finds its applications in medicine and forensics

Table Of Contents

LIST OF FIGURES

LIST OF TABLES

Chapter 1: Introduction

1.1 Project Description:

Speech Emotion Recognition, abbreviated as SER, is the act of attempting to recognize human emotion and affective states from speech. This is capitalizing on the fact that voice often reflects underlying emotion through tone and pitch. This is also the phenomenon that animals like dogs and horses employ to be able to understand human emotion. SER is tough because emotions are subjective and annotating audio is challenging. In machine interaction with human being is yet challenging task that machine should be able to identify and react to human non-verbal communication such as emotions which makes the human computer interaction become more natural. In present research area automatic emotion recognition using speech is an essential task which paid close attention. Speech signal is a rich source of information and it is an attractive and efficient medium due to its numerous features of expressing approach & extracting emotions through speech is possible. In most of In this paper emotions is recognized through speech using spectral features such as Mel frequency spectrum co-efficient prosodic features like pitch , energy and were utilized & study is carried out using MLP classifiers which is used for detection of six basic emotional states of speaker’s like anger ,happiness, sadness, fear, disgust and neutral using RAVDESS dataset.

Chapter 2: Literature Review

2.1: Literature Survey

S No	Author and Paper title	Details of Publication	Summary of the Paper
1	Deepak Bharti, A Hybrid Machine Learning Model for Emotion Recognition From Speech Signals	Proceedings of the International Conference on Smart Electronics and Communication (ICOSEC 2020) IEEE Xplore Part Number: CFP20V90-ART; ISBN: 978-1-7281-5461-9	• The feature extraction method using Gammatone Frequency Cepstral Co-efficient (GFCC) • Dataset used in this paper is RAVDESS data set • It analyses how the classification and feature extraction performance in the recognition rate of emotions in speech (Sad, Happy, and Angry)
2	Pavol Harár , Radim Burget and Malay Kishore Dutta, Speech Emotion Recognition with Deep Learning	2017 4th International Conference on Signal Processing and Integrated Networks (SPIN)	• SER is used to predict the emotional state of a person from a short voice recording split into 20 millisecond segments. • This approach is context independent which means that all audio segments were classified independently.

3	Girija Deshmukh, Apurva Gaonkar, Gauri Golwalkar, Sukanya Kulkarni, Speech based Emotion Recognition using Machine Learning	Proceedings of the Third International Conference on Computing Methodologies and Communication (ICCMC 2019) IEEE Xplore Part Number: CFP19K25-ART; ISBN: 978-1-5386-7808-4	• In this paper, three emotions- anger, happiness, and sadness, were classified using three feature vectors. • Pitch, Mel frequency cepstral coefficients, Short Term Energy were the • three feature vectors extracted from audio signals. • Open source North American English acted speech corpus and recorded natural speech corpus were used as input.
4	Saikat Basu, Jaybrata Chakraborty , Arnab Bag and Md. Aftabuddin, A Review on Emotion Recognition using Speech	International Conference on Inventive Communication and Computational Technologies (ICICCT 2017)	• The study reveals the fact that identification of emotion of a person is a task yet to have complete and general solution. • Till now, most of the work has been done on the fixed size speech segment for classification of emotion, that means on the off line speech.
5	Esther Ramdinmawii , Abhijit Mohanta and Vinay Kumar Mittal, Emotion Recognition from Speech Signal	Proc. of the 2017 IEEE Region 10 Conference (TENCON), Malaysia, November 5-8, 2017	• This paper is about analyzing speech for emotion states (Anger, Fear, Neutral, Happy) using speech signals. • The analysis of these emotion states has been done using features, namely, instantaneous fundamental frequency using Zero Frequency Filtering, Formant frequencies (F1, F2, F3), signal energy, and dominant frequencies. • The speech files are sampled at 16 kHz for F0 and signal energy, and at 10 kHz for Formant frequencies and dominant frequencies.

6	W. Q. Zheng, J. S. Yu, Y. X. Zou, An Experimental Study of Speech Emotion Recognition Based on Deep Convolutional Neural Networks	2015 International Conference on Affective Computing and Intelligent Interaction (ACII)	• In this paper, a deep convolution neural networks-based approach has been developed to learn the effective features for speech emotion recognition from audio spectrogram data • A speech emotion recognition algorithm termed as PCA-DCNNs-SER is proposed. • Preliminary experiments have been conducted to evaluate the performance of PCA-DCNNs-SER on the IEMOCAP database.
7	RUHUL AMIN KHALIL , EDWARD JONES Speech Emotion Recognition Using Deep Learning Techniques	Received July 25, 2019, accepted August 5, 2019, date of publication August 19, 2019, date of current version September 4, 2019	• This paper has provided a detailed review of the deep learning techniques for SER. • Deep learning techniques such as DBM, RNN, DBN, CNN, and AE have been the subject of much research in recent years. • These deep learning methods and their layer-wise architectures are briefly elaborated based on the classification of various natural emotion such as happiness, joy, sadness, neutral, surprise, boredom, disgust, fear, and anger.
8	LILI GUO, LONGBIAO WANG, JIANWU DANG Exploration of Complementary Features for Speech Emotion Recognition Based on Kernel Extreme Learning Machine	Received May 10, 2019, accepted June 2, 2019, date of publication June 6, 2019, date of current version June 24, 2019	• This paper focused on improving speech emotion recognition by using complementary features. • To utilize the potential advantages of two types of features (i.e., the spectrogram-based statistical features and auditory-based empirical features),a dynamic fusion framework to extract the complementary features based on spectrograms and the auditory-based features

9	MUSTAQEEM , MUHAMMAD SAJJAD , AND SOONIL KWON Clustering-Based Speech Emotion Recognition by Incorporating Learned Features and Deep BiLSTM	Mustaqeem et al.: Clustering-Based SER by Incorporating Learned Features and IEEE Access, vol. 6, pp. 52227–52237, 2018.	• The existing CNNs system of SER has too many challenges such as improvement in accuracy and reduce the computational complexity of the whole model. • Due to these limitations, we planned a novel approach for SER to improve the recognition accuracy and reduce the overall model cost computation and processing time.
10	A Pramod Reddy and V. Vijayarajan2, Extraction of Emotions from Speech	International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 16 (2017) pp. 5760-5767	• This paper presents Most common emotions searched and extracted are Happiness', Sadness, Disgust, Neutral along with other features such as joy, Borden, fear and surprise. • The extraction rate depends on the classifier used.

2.1 Existing and Proposed System

Existing System:

Speech Emotion Recognition is a subject under research. Speech emotion recognition abbreviated as SER. It creates a natural Human Computer interaction. There are various kinds of methods used to identify the emotion from the speech, such as using support vector machine (SVM), Recurrent Neural Network, K-nearest neighbour, Hidden Markov Model (HMM).

Support Vector Machine (SVM): Support Vector Machine approach computes the audio parameters to identify the emotion and has high accuracy in predicting the emotion from the speech. But this approach can only classify the dataset into 2 classes only. That means we can only identify among the 2 emotions trained to the classifier. The other disadvantages of this approach are long processing time, background noise leading to error and it has low accuracy.

K-nearest neighbour: The other classifier is k nearest neighbour classifier. This is the simplest classification algorithm which identifies the emotion of speech. This classifier uses pitch and energy of the audio to predict emotion and has accuracy of 64% for 4 emotions audio.

Hidden Markov Model (HMM): HMM models temporal sequencing from the audio. This modelling is useful in predicting the emotion from the speech. The main limitation of this classifier is feature selection process. As the features don’t carry the complete information of the emotion of the speech. But it has good classification accuracy compare to other classifiers.

Disadvantages of Existing System

1. It can only tell only limited number of emotions for instance only two emotions.

2. The model trained heavily depends on the language used, words used in that particular language rather than depending on features of the language such as pitch, tone, pauses etc.

3. There are smaller number of features extracted from the test data which makes it difficult to predict accurate emotion on test data

4. Audio Visual enhancements are not considered in the Existing system to predict the emotion

5. Song and Speech are not differentiated distinctly in Existing system, which leads to the problem Song is confused with Speech Emotion and vice versa

Proposed System:

The proposed system extracts feature such as pitch, tone, frequency from the input audio and plot the features against properties such as MFCC, Mel and Chroma to generate a feature vector of float type 32-bit floating point integers.

- The feature vector represents the features of the audio, then those features are used to train a multi-layer perceptron model which internally makes use of an Artificial Neural Network with binary Inputs and Outputs.

- It has 3 layers Input Layer, hidden Layer and the Output Layer, there might be N number of layers under the hidden layer but Input and Output layers are just one, the feature vector goes through input of Artificial Neural Network and it gives us output as the observed emotion which matches the audio's features most or the closest correct emotion

- The proposed system can identify can understand seven emotions 1. Neutral 2. Calm 3. Happy 4. Sad 5. Angry 6. Fearful 7. Disgust 8. Surprised

- If any emotion does not match any of the observable emotions, then the closest observed emotion is predicted, emotions are recognized through speech using spectral features such as Mel frequency cepstrum coefficient

- Prosodic features like pitch, energy and were utilized & study is carried out using MLP classifiers which is used for detection of six basic emotional states of speakers such as anger, happiness, sadness, fear, disgust and neutral using RAVDESS dataset.

- RAVDESS dataset has recordings of 24 actors, 12 male actors and 12 female actors, the actors are numbered from 01 to 24. The male actors are odd numbered and female actors are even numbered. The emotions contained in the dataset are as sad, happy, neutral, angry, disgust, surprised, fearful and calm expressions. The dataset contains all expressions in three formats, which are: Only Audio, Audio-Video and Only Video.

Advantages of Proposed System

• The Human computer interaction will be improved and will be more natural.

• We can detect the emotion of the speaker irrespective of the language.

• Proposed system can detect 8 different emotions (calm, happy, sad, angry, fearful, surprise, and disgust expressions)

• We can detect the correct emotion of the speaker approximately since the accuracy of the model high.

2.3 Tools and Technologies used

- Anaconda for python 3.6.5
- Jupyter Notebook
- NetBeans
- Programming Language: Python 3.6.5, Html, CSS, JavaScript, JSP
- Classifier: MLPClassifier
- Libraries: librosa, soundfile, neural network, sklearn

2.4 Hardware and Software Requirements

Software Requirements

1. Python 3 or above (with IDE)

2. Pip to install python packages

3. AI ML Libraries like librosa, sklearn 7

4. RAVDESS dataset (Audio-Visual Database)

Hardware Requirements

1. Minimum 4GB RAM (8GB recommended)

2. Windows/UNIX based OS

3. JupyterLab (Cloud).

Chapter 3: Software Requirement Specifications

3.1 Introduction

The Speech emotion recognition systems have the aim of recognizing emotions, in this case, from the speech. The Speech emotion recognition aims to recognize the underlying emotional state of a speaker from the voice signal.
The problems introduced to these systems are:
How the emotions are presented inside an audio signal?
How can a classifier use labelled samples to classify the emotion of a new one?

3.2 General Description

In proposed system, we are using RAVDESS dataset as a data for the system. The data present in the dataset is pre-processed to clean the audio and remove the disturbance from the audio to reduce the error in the output. The audio is divided into equal time intervals frames. Then the dataset is divided into 2 parts as training data and testing data. Training data is 80% of the dataset and testing data is 20% of the dataset. The features are extracted from the audio and given to the classifier to predict the emotion. The model is created by training data inputs to the classifier then this model is tested with the testing data inputs. We get the accuracy by calculating the output of the model and the actual emotion in dataset.

Objectives of SER are:

- To build a model to recognize emotion from speech using the librosa and sklearn libraries and the RAVDESS dataset
- To identification of the emotional state of humans from their voice with maximum accuracy.
- To classify emotions in speech (sad, happy, and angry)
- To improve man-machine interface.
- It can also be used to monitor the psycho physiological state of a person in lie detectors

3.3 Functional requirements:

They are the list of functionalities that are provided by the system or its component. It is the reaction of the system to a particular input or the behaviour of the system in a particular situation to a particular input.

The Main functional requirements are:

- The user has to record the input using UI interface (the input will be audio file)
- This input will be used to predict the emotions in speech

Input:

Loading the required RAVDESS Data set with length of 1439 Audio RAVDESS:
This dataset includes around1500 audio file input from 24 different actors (12 male and 12 female) where these actors record short audios in 8 different Emotions.
1 = neutral, 2 = calm, 3 = happy, 4 = sad, 5 = angry, 6 = fearful,7 = disgust, 8 = surprised.
Each audio file is named in such a way that the 7th character is consistent with the different emotions that they represent.

Process:

- Use the Speech-Recognition API to get the Raw Text from Audio Files. Though Speech Recognition is less strong for large chunk of files, so used Error Handling, where when it is not be able to produce the text of a particular Audio File it prints the statement error just for understanding Audio
- Masking and Cleaning:
    - Down sampling of audio files is done and Put mask over it and direct into clean folder.
    - Mask is to remove unnecessary empty voices around the main audio voice.

- Feature Extraction of Audio Files Function:
    - Extract features from a sound file
- Labels Classification:
    - Emotions in the RAVDESS dataset to be classified Audio Files

- Loading of data and splitting of dataset (training and testing data):
    - Load the data and extract features for each sound file
- Applying the multi-layer perceptron classifier:
    - Initialize the Multi-Layer Perceptron Classifier
- Train the model
- Saving the model

Output:

- Predicting the test data using the saved model:
- Store the Prediction probabilities into CSV file
- Applying extract feature function on random file and then loading model to predict the result
- Output displayed in GUI

3.4 NON-FUNCTIONAL REQUIREMENTS

A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours.

- User Friendly- It is user friendly in nature and easy to understand.
- Availability- It can be accessed from anywhere at any time. It means 24 X 7 availability.
- Flexibility-The application is capable to support any web-browser.
- Memory Utilization- Data is stored at central database so memory is completely utilized.
- Portability- An end-user can use this system on any OS; either it is Windows or Linux. The system shall run on PC, Laptops, and PDA etc.
- Security-The application asks for the permission of the microphone by the user to record the audio

Chapter 4: System Design

4.1 System Perspective /Architectural Design:

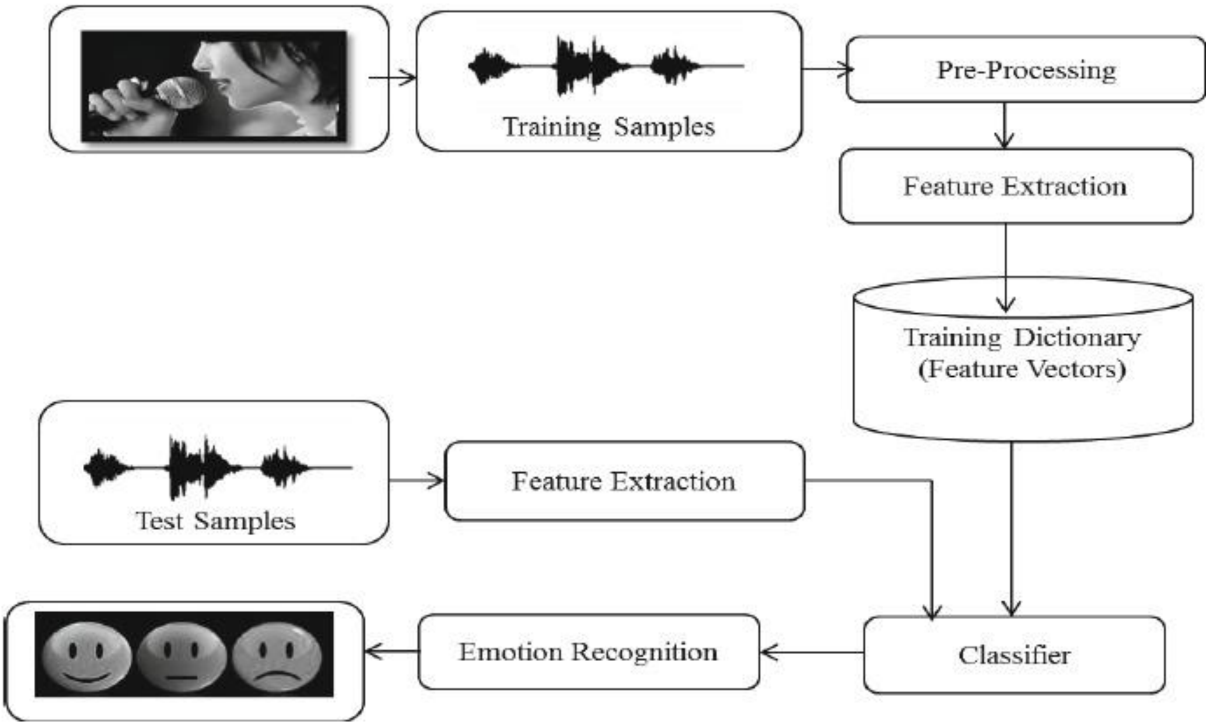


Fig 1.1-Block Diagram

Training data is pre-processed and features are extracted from it. These are used to train SER Model. The model is given with audio inputs and Emotion is recognised from it

4.2 Context Diagram

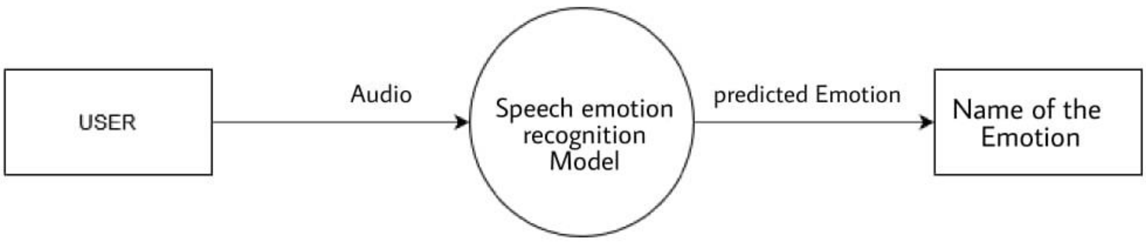


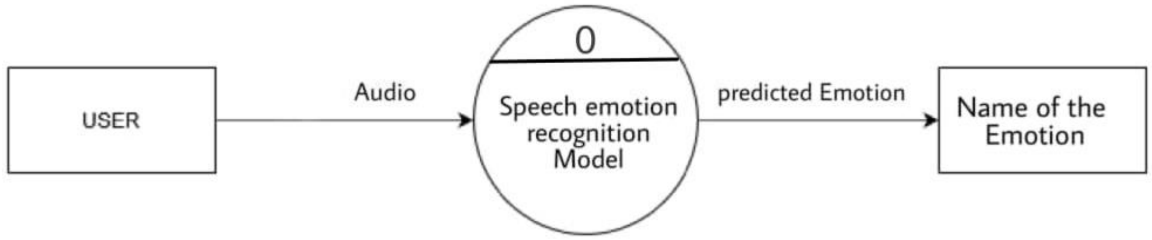
Fig 2.1-Context Diagram

The user will give Audio as input and the model will predict the emotion and it will be displayed

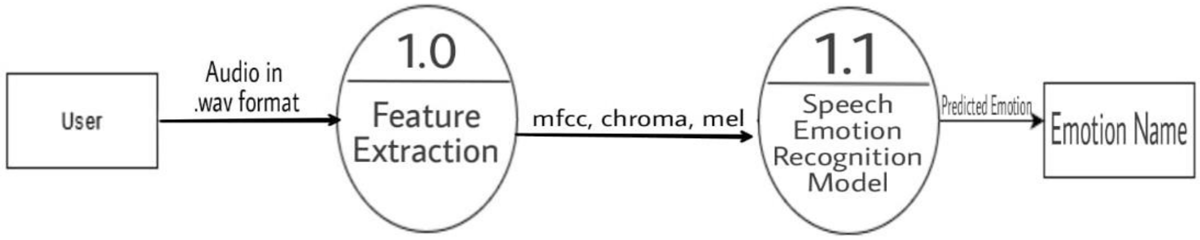
Chapter 5: Detailed Design

5.1 System Design

Data Flow Diagram:



Level 0



Level 1

Fig 3.1-Data Flow Diagram

The flow of data starts from user by giving audio as a input data and it ends with the output data as predicted emotion

5.2 Detailed design

Class Diagram:

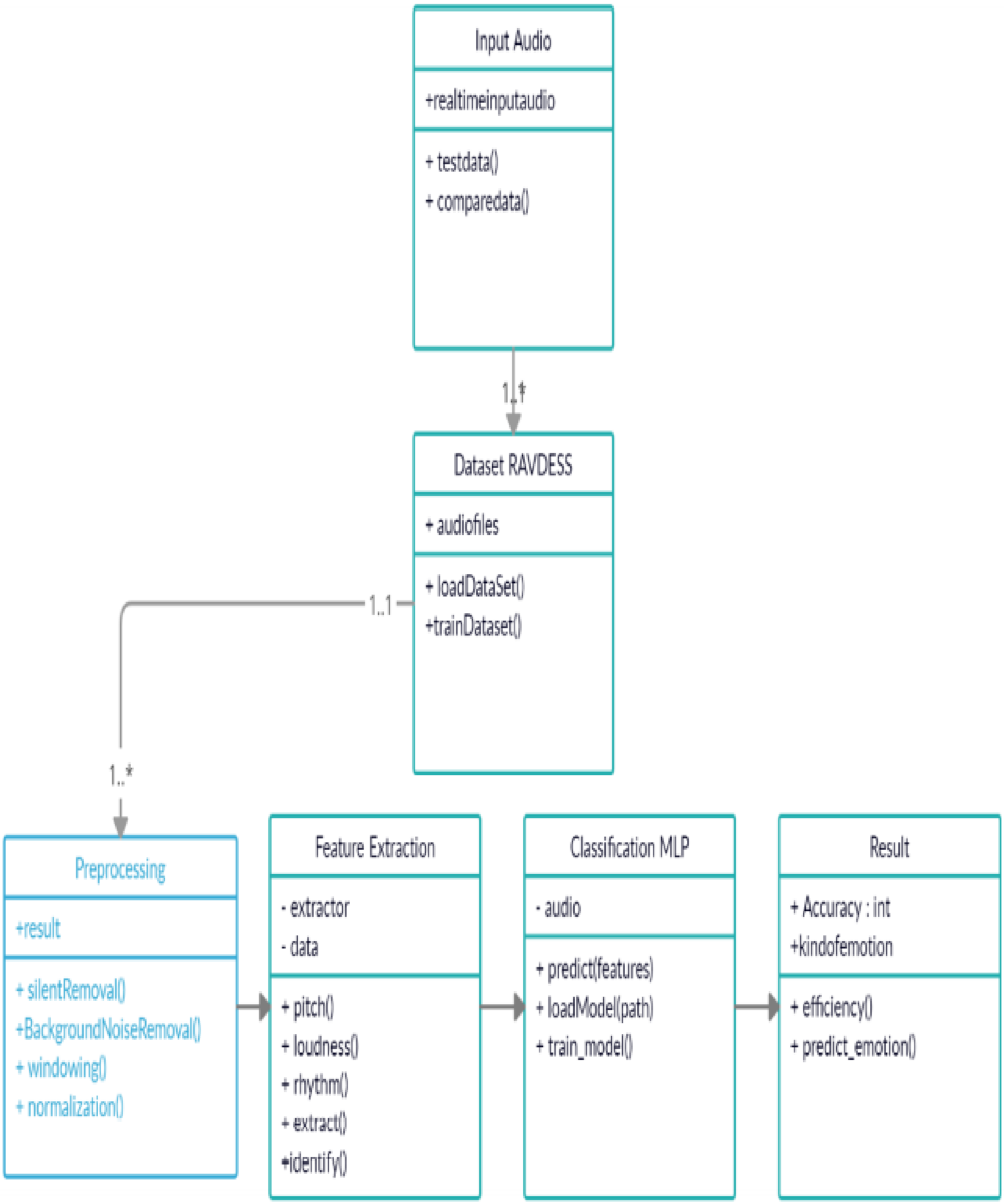


Fig 4.1: Class Diagram

Sequence Diagram

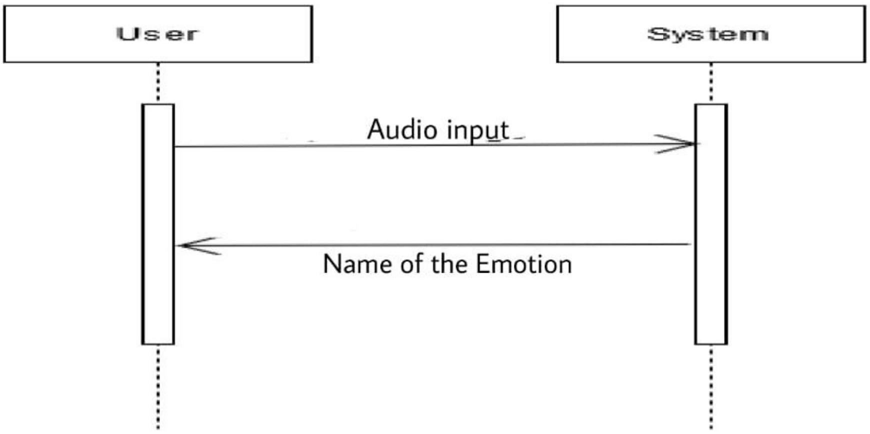


Fig 5.1-Sequence Diagram

The user will give Audio as input and the model will predict the emotion and it will be displayed

Activity Diagram:

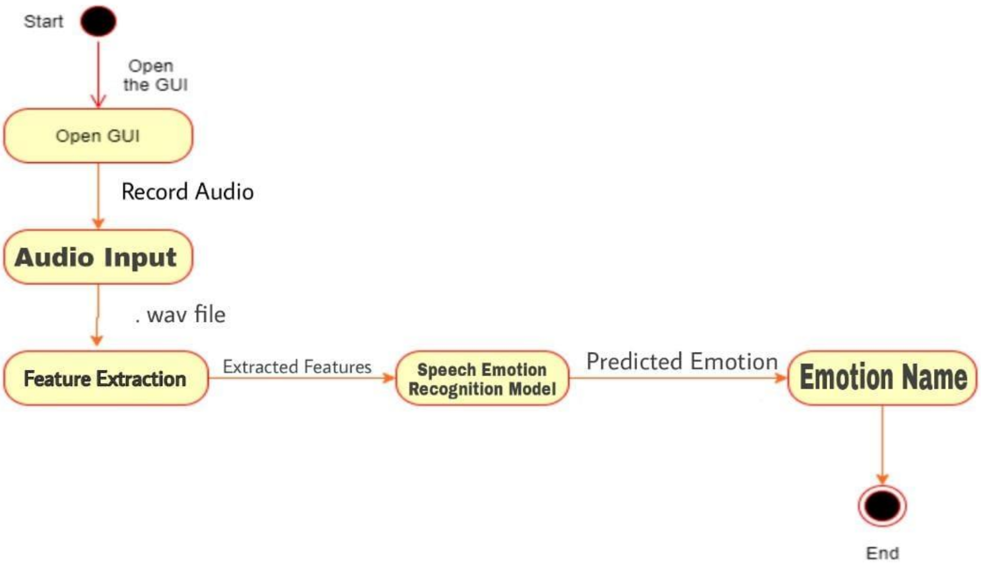


Fig 6.1-Activity Diagram

The User will open GUI and record the audio and give it as input the features will be extracted and will be given as parameters to SER model the model will predict the emotion and display it to user

Chapter 6 Implementation

6.1 Code Snippets

#INSTALL ALL THE REQUIRED LIBRARIES AND PACKAGES

```
import os
import glob
import tqdm
from tqdm.autonotebook import tqdm
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.io import wavfile
from python_speech_features import mfcc,logfbank
import librosa as lr
import os, glob, pickle
import librosa
from scipy import signal
import noisereduce as nr
from glob import glob
import librosa
get_ipython().magic('matplotlib inline')
#All the Required Packages and Libraies are installed.
import soundfile
from sklearn.utils.class_weight import compute_class_weight
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score
print("All the Libraries have been Imported")
```

 # LOADING THE REQUIRED DATASET:-

#Loading the required RAVDESS DataSet with length of 1439 Audio Files

```
#os.listdir(path='.\speech-emotion-recognition-ravdess-data')
os.listdir(path='C:\\Users\\Nethra\\Desktop\\Speech_Emotion_Detection-master\\speech-emotion-recognition-ravdess-data')
def getListOfFiles(dirName):
    listOfFile=os.listdir(dirName)
    allFiles=list()
    for entry in listOfFile:
        fullPath=os.path.join(dirName, entry)
        if os.path.isdir(fullPath):
            allFiles=allFiles + getListOfFiles(fullPath)
        else:
            allFiles.append(fullPath)
    return allFiles
dirName = './speech-emotion-recognition-ravdess-data'
listOfFiles = getListOfFiles(dirName)
```

 # USING SPEECH RECOGNITION API TO CONVERT AUDIO INTO TEXT: -

```
#Use the Speech-Recognition API to get the Raw Text from Audio Files, Though Speech Recognition
#is less strong for large chunk of files , so used Error Handling , where when it is not be able to
#produce the text of a particular Audio File it prints the statement 'error'.Just for understanding Audio
import speech_recognition as sr
r=sr.Recognizer()
for file in range(0 , len(listOfFiles) , 1):
    with sr.AudioFile(listOfFiles[file]) as source:
        audio = r.listen(source)
        try:
            text = r.recognize_google(audio)
            print(text)
        except:
            print('error')
```

 # PLOTTING TO UNDERSTAND RAW AUDIO FILES: -

```
#Plotting the Basic Graphs for understanding of Audio Files :
```

```
for file in range(0 , len(listOfFiles) , 1):
    audio , sfreq = lr.load(listOfFiles[file])
    time = np.arange(0 , len(audio)) / sfrez
    fig ,ax = plt.subplots()
    ax.plot(time , audio)
    ax.set(xlabel = 'Time (s)' , ylabel = 'Sound Amplitude')
    plt.show()
#PLOT THE SEPCTOGRAM
for file in range(0 , len(listOfFiles) , 1):
     sample_rate , samples = wavfile.read(listOfFiles[file])
     frequencies , times, spectrogram = signal.spectrogram(samples, sample_rate)
     plt.pcolormesh(times, frequencies, spectrogram)
     plt.imshow(spectrogram)
     plt.ylabel('Frequency [Hz]')
     plt.xlabel('Time [sec]')
     plt.show()
```

 # VISUALISATION OF AUDIO DATA:-

```
#Next Step is In-Depth Visualisation of Audio Fiels and its certain features to plot for.
#They are the Plotting Functions to be called later.
def plot_signals(signals):
    fig , axes = plt.subplots(nrows=2, ncols=5,sharex =False , sharey=True, figsize=(20,5))
    fig.suptitle('Time Series' , size=16)
    for x in range(2):
        for y in range(5):
            axes[x,y].set_title(list(signals.keys())[i])
            axes[x,y].plot(list(signals.values())[i])
            axes[x,y].get_xaxis().set_visible(False)
            axes[x,y].get_yaxis().set_visible(False)
            i +=1
def plot_fft(fft):
    fig , axes = plt.subplots(nrows=2, ncols=5,sharex =False , sharey=True, figsize=(20,5))
```

```
    fig.suptitle('Fourier Transform' , size=16)
    i=0
    for x in range(2):
        for y in range(5):
            data = list(fft.values())[i]
            Y,freq = data[0] , data[1]
            axes[x,y].set_title(list(fft.keys())[i])
            axes[x,y].plot(freq , Y)
            axes[x,y].get_xaxis().set_visible(False)
            axes[x,y].get_yaxis().set_visible(False)
            i +=1

def plot_fbank(fbank):
    fig , axes = plt.subplots(nrows=2, ncols=5,sharex =False , sharey=True, figsize=(20,5))
    fig.suptitle('Filter Bank Coefficients' , size=16)
    i=0
    for x in range(2):
        for y in range(5):
            axes[x,y].set_title(list(fbank.keys())[i])
            axes[x,y].imshow(list(fbank.values())[i],cmap='hot', interpolation = 'nearest')
            axes[x,y].get_xaxis().set_visible(False)
            axes[x,y].get_yaxis().set_visible(False)
            i +=1

def plot_mfccs(mfccs):
    fig , axes = plt.subplots(nrows=2, ncols=5,sharex =False , sharey=True, figsize=(20,5))
    fig.suptitle('Mel Frequency Capstrum  Coefficients' , size=16)
    i=0
    for x in range(2):
        for y in range(5):
            axes[x,y].set_title(list(mfccs.keys())[i])
            axes[x,y].imshow(list(mfccs.values())[i],
```

```
                    cmap='hot', interpolation = 'nearest')
            axes[x,y].get_xaxis().set_visible(False)
            axes[x,y].get_yaxis().set_visible(False)
            i +=1

def calc_fft(y,rate):
    n = len(y)
    freq = np.fft.rfftfreq(n , d= 1/rate)
    Y= abs(np.fft.rfft(y)/n)
    return(Y,freq)
```

 # HERE THE DATA SET IS LOADED AND PLOTS ARE VISUALISED BY CALLING THE PLOTTING FUNCTIONS .

```
import matplotlib.pyplot as plt
from scipy.io import wavfile as wav
from scipy.fftpack import fft
import numpy as np
for file in range(0 , len(listOfFiles) , 1):
    rate, data = wav.read(listOfFiles[file])
    fft_out = fft(data)
    get_ipython().run_line_magic('matplotlib', 'inline')
    plt.plot(data, np.abs(fft_out))
    plt.show()
signals={}
fft={}
fbank={}
mfccs={}
# load data
for file in range(0 , len(listOfFiles) , 1):
#     rate, data = wavfile.read(listOfFiles[file])
     signal,rate =librosa.load(listOfFiles[file] , sr=44100)
     mask = envelope(signal , rate , 0.0005)
```

```
    signals[file] = signal
    fft[file] = calc_fft(signal , rate)
    bank = logfbank(signal[:rate] , rate , nfilt = 26, nfft = 1103).T
    fbank[file] = bank
    mel = mfcc(signal[:rate] , rate , numcep =13 , nfilt = 26 , nfft=1103).T
    mfccs[file]=mel
plot_signals(signals)
plt.show()
plot_fft(fft)
plt.show()
plot_fbank(fbank)
plt.show()
plot_mfccs(mfccs)
plt.show()
print("over")
```

#CLEANING AND MASKING:-

```
#Now Cleaning Step is Performed where:
#DOWN SAMPLING OF AUDIO FILES IS DONE  AND PUT MASK OVER IT AND DIRECT INTO CLEAN FOLDER
#MASK IS TO REMOVE UNNECESSARY EMPTY VOIVES AROUND THE MAIN AUDIO VOICE
def envelope(y , rate, threshold):
   mask=[]
   y=pd.Series(y).apply(np.abs)
   y_mean = y.rolling(window=int(rate/10) ,  min_periods=1 , center = True).mean()
   for mean in y_mean:
      if mean>threshold:
         mask.append(True)
      else:
         mask.append(False)
   return mask
print("Finished Executing")
```

#The clean Audio Files are redirected to Clean Audio Folder Directory

```
import glob,pickle

for file in tqdm(glob.glob('C:/Users/Nethra/Desktop/Speech_Emotion_Detection-master/speech-emotion-recognition-ravdess-data//**//**.wav')):

    file_name = os.path.basename(file)

    signal , rate = librosa.load(file, sr=16000)

    mask = envelope(signal,rate, 0.0005)

    wavfile.write(filename= 'C:/Users/Nethra/Desktop/Speech_Emotion_Detection-master//clean//'+str(file_name), rate=rate,data=signal[mask])
```

 # FEATURE EXTRACTION

```
#Feature Extraction of Audio Files Function

#Extract features (mfcc, chroma, mel) from a sound file

def extract_feature(file_name, mfcc, chroma, mel):

    with soundfile.SoundFile(file_name) as sound_file:

        X = sound_file.read(dtype="float32")

        sample_rate=sound_file.samplerate

        if chroma:

            stft=np.abs(librosa.stft(X))

        result=np.array([])

        if mfcc:

            mfccs=np.mean(librosa.feature.mfcc(y=X, sr=sample_rate, n_mfcc=40).T, axis=0)

        result=np.hstack((result, mfccs))

        if chroma:

            chroma=np.mean(librosa.feature.chroma_stft(S=stft, sr=sample_rate).T,axis=0)

        result=np.hstack((result, chroma))

        if mel:

            mel=np.mean(librosa.feature.melspectrogram(X, sr=sample_rate).T,axis=0)

        result=np.hstack((result, mel))

    return result
```

 # LABELS CLASSIFICATION

```
#Emotions in the RAVDESS dataset to be classified Audio Files based on .
```

```
emotions={
  '01':'neutral',
  '02':'calm',
  '03':'happy',
  '04':'sad',
  '05':'angry',
  '06':'fearful',
  '07':'disgust',
  '08':'surprised'
}
#These are the emotions User wants to observe more :
observed_emotions=['calm', 'happy', 'fearful', 'disgust','neutral','sad','angry','surprised']
print("Executed this block")
```

 # LOADING OF DATA AND SPLITTING OF DATASET

```
# (TRAINING AND TESTING DATA)
#Load the data and extract features for each sound file
from glob import glob
import os
import glob
def load_data(test_size=0.33):
    x,y=[],[]
    answer = 0
    for file in glob.glob('C:/Users/Nethra/Desktop//Speech_Emotion_Detection-master/clean//*.wav'):
        file_name=os.path.basename(file)
        emotion=emotions[file_name.split("-")[2]]
        if emotion not in observed_emotions:
            answer += 1
            continue
        feature=extract_feature(file, mfcc=True, chroma=True, mel=True)
        x.append(feature)
        y.append([emotion,file_name])
```

```
    return train_test_split(np.array(x), y, test_size=test_size, random_state=9)
print("Loading the data and extract features for each sound file")
```

 # MAPPING OF TESTING DATA TO THEIR CORRESPONDING FILENAMES AS LABELS

```
#Split the dataset
import librosa
import numpy as np
x_train,x_test,y_trai,y_tes=load_data(test_size=0.30)
print(np.shape(x_train),np.shape(x_test), np.shape(y_trai),np.shape(y_tes))
y_test_map = np.array(y_tes).T
y_test = y_test_map[0]
test_filename = y_test_map[1]
y_train_map = np.array(y_trai).T
y_train = y_train_map[0]
train_filename = y_train_map[1]
#print(np.shape(y_train),np.shape(y_test))
print(*test_filename,sep="\n")
```

#Get the shape of the training and testing datasets

```
# print((x_train.shape[0], x_test.shape[0]))
print((x_train[0], x_test[0]))
```

#Get the number of features extracted

```
print(f'Features extracted: {x_train.shape[1]}')
```

 # APPLYING THE MLP CLASSIFIER

```
# Initialize the Multi Layer Perceptron Classifier
model=MLPClassifier(alpha=0.01, batch_size=256, epsilon=1e-08, hidden_layer_sizes=(300,),
learning_rate='adaptive', max_iter=500)
```

#Train the model

```
model.fit(x_train,y_train)
```

#SAVING THE MODEL

```
import pickle
# Save the Modle to file in the current working directory
#For any new testing data other than the data in dataset
Pkl_Filename = "Emotion_Voice_Detection_Model.pkl"
with open(Pkl_Filename, 'wb') as file:
    pickle.dump(model, file)
```

 # Load the Model back from file

```
with open(Pkl_Filename, 'rb') as file:
    Emotion_Voice_Detection_Model = pickle.load(file)
Emotion_Voice_Detection_Model
Pkl_Filename = "Pickle_sorted_Model.pkl"
```

 # PREDICT THE TEST DATA USING THE SAVED MODEL

```
#predicting :
y_pred=Emotion_Voice_Detection_Model.predict(x_test)
y_pred
```

 # STORE THE PREDICTED FILE IN .CSV FILE

```
#Store the Prediction probabilities into CSV file
import numpy as np
import pandas as pd
y_pred1 = pd.DataFrame(y_pred, columns=['predictions'])
y_pred1['file_names'] = test_filename
print(y_pred1)
y_pred1.to_csv('predictionfinal.csv')
```

 # REAL TIME IMPLEMENATION

```
data, sampling_rate = librosa.load('audio.wav')
```

```
get_ipython().run_line_magic('matplotlib', 'inline')
import os
import pandas as pd
import librosa.display
import glob
plt.figure(figsize=(15, 5))
librosa.display.waveplot(data, sr=sampling_rate)
audiofile = 'audio.wav'
# data , sr = librosa.load(file)
# data = np.array(data)
feature=extract_feature(audiofile, mfcc=True, chroma=True, mel=True)
#print(feature)
ans = np.array(feature)
print("Emotion in this speech is ",Emotion_Voice_Detection_Model.predict([ans]))
```

 # STORE IN DATABASE

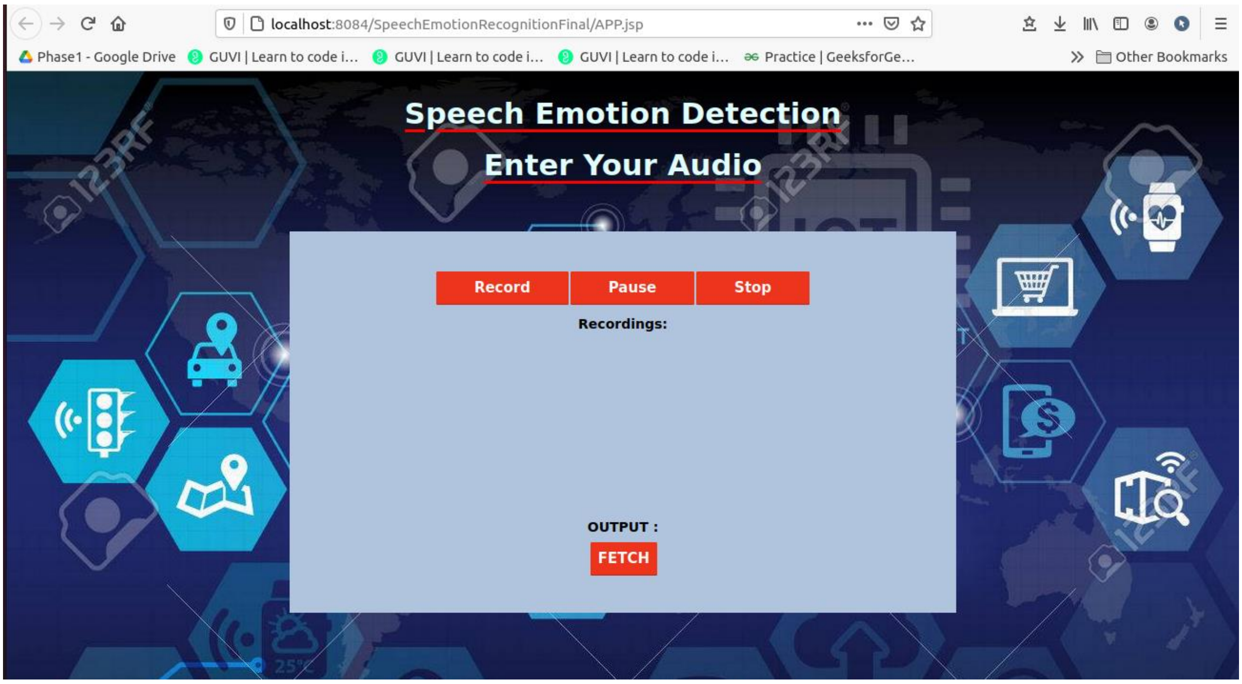
```
import pymysql
con=pymysql.connect (host="localhost", user="root",passwd="1234",db="speech")
print("Database Successfully connected")
a=Emotion_Voice_Detection_Model.predict([ans])
class mydb:
    def insert(self,audio,output):
        self.audio=audio
        self.output=output
        cur=con.cursor()
        cur.execute(''' insert into store3(audioname,output)values('%s','%s')'''%(audio,output))
        con.commit()
        print("inserted")
obj=mydb()
audio=audiofile
output=a[0]
print("The emotion detected from the speech is:",output)
```

```
print(audio)
obj.insert(audio,output)
```

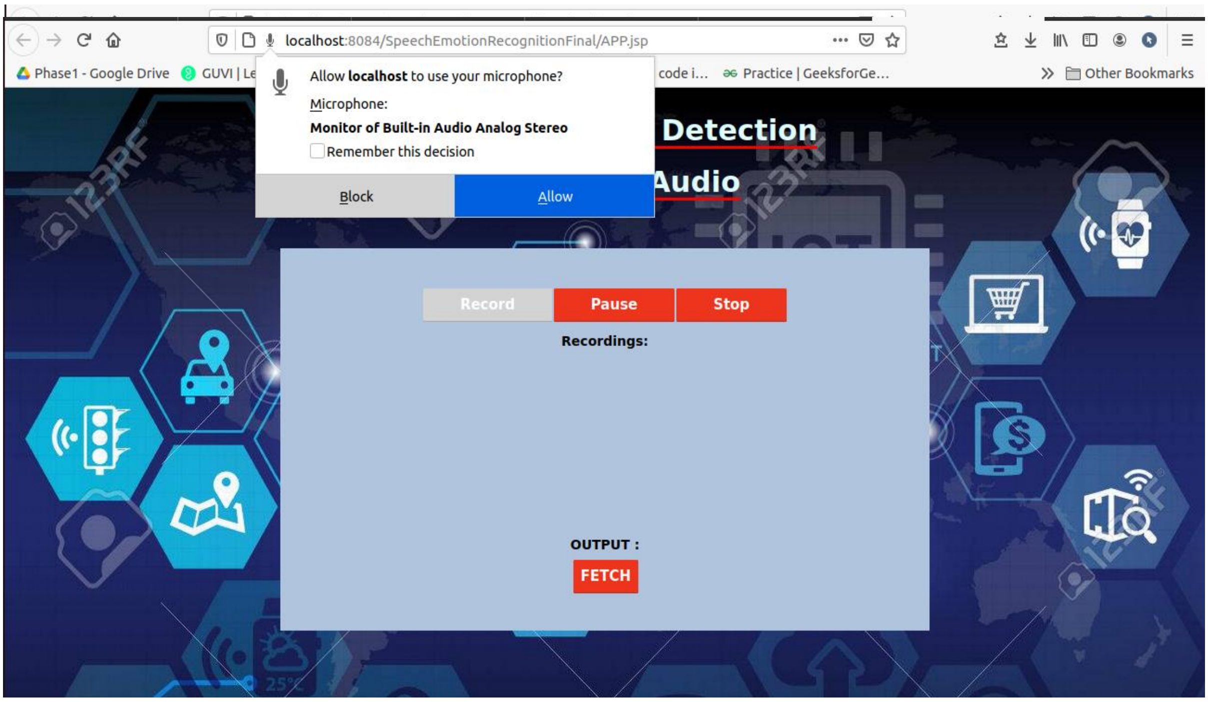
#Confusion Matrix

```
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
from sklearn.metrics import plot_confusion_matrix
#tn, fp, fn, tp =confusion_matrix(y_test, y_pred).ravel()
fig, ax = plt.subplots(figsize=(10, 10))
 plot_confusion_matrix(Emotion_Voice_Detection_Model, x_test, y_test,ax=ax)
```

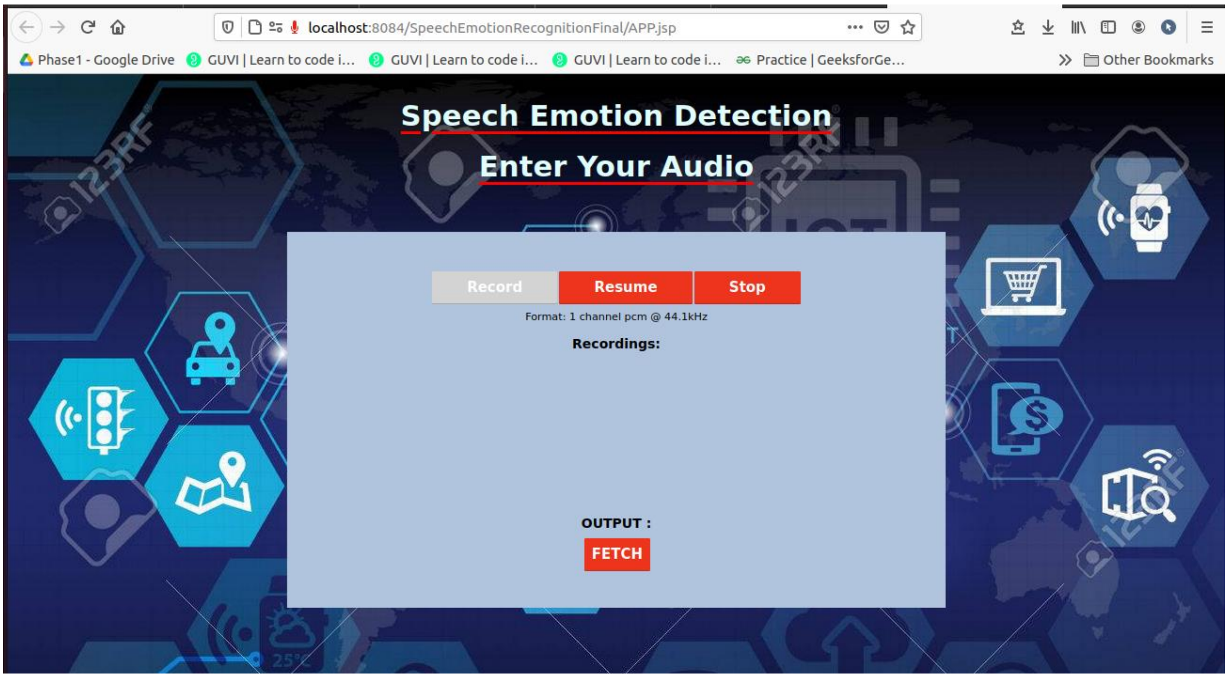
6.2 Implementation



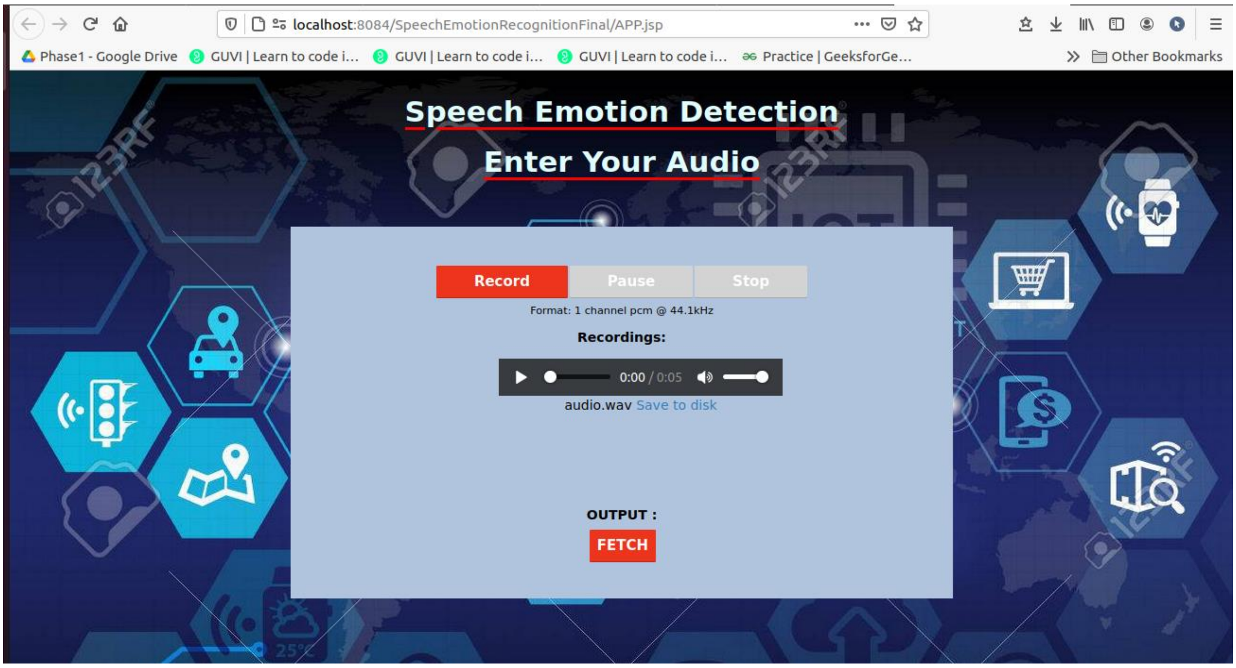
Screenshot 1: The GUI page of the SER Project



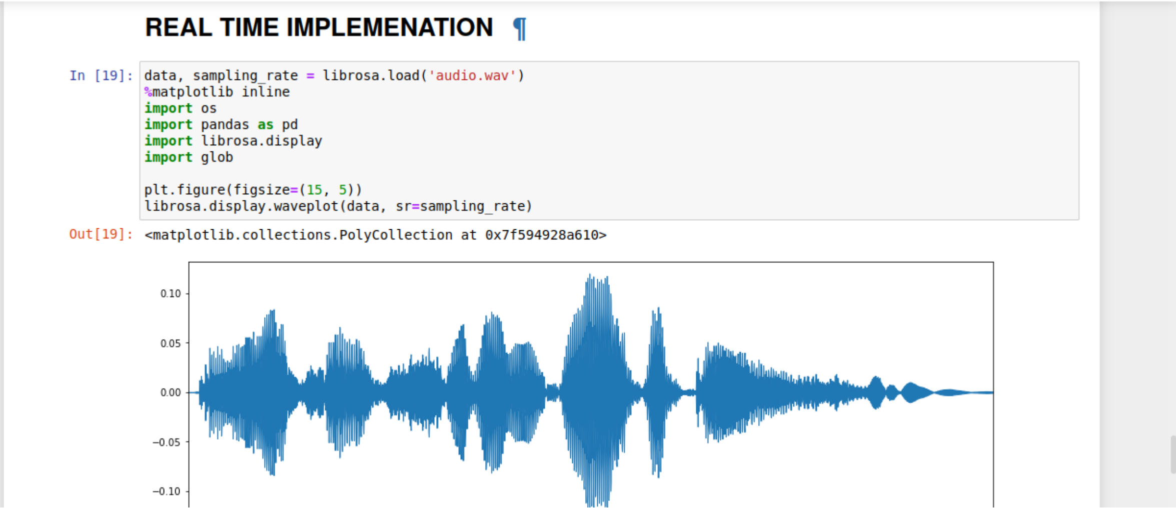
Screenshot 2: The user will be asked for permission



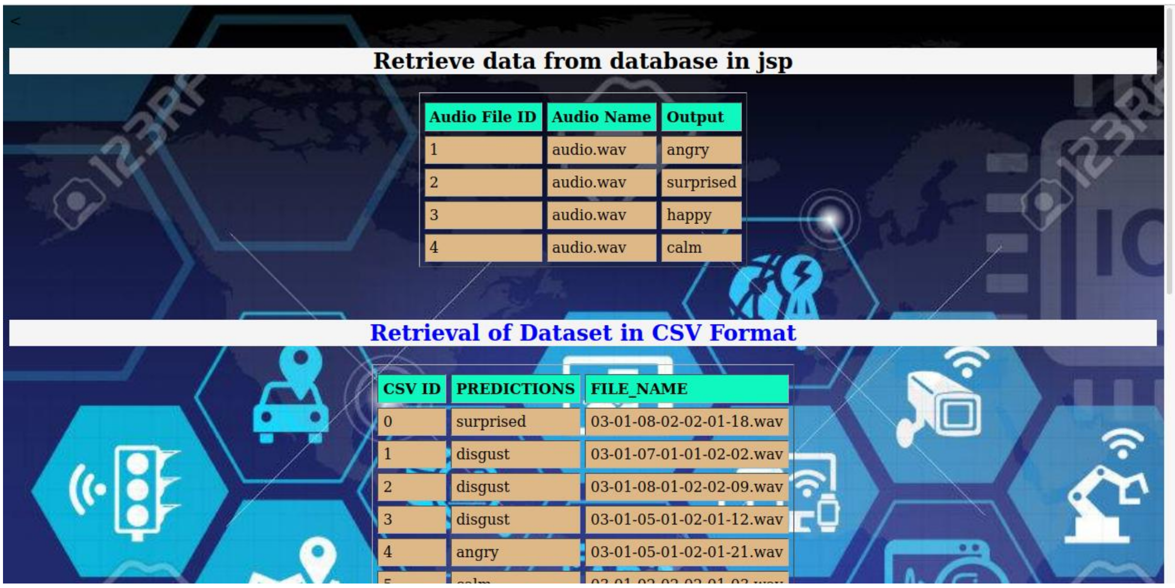
Screenshot 3: The Audio is being recorded



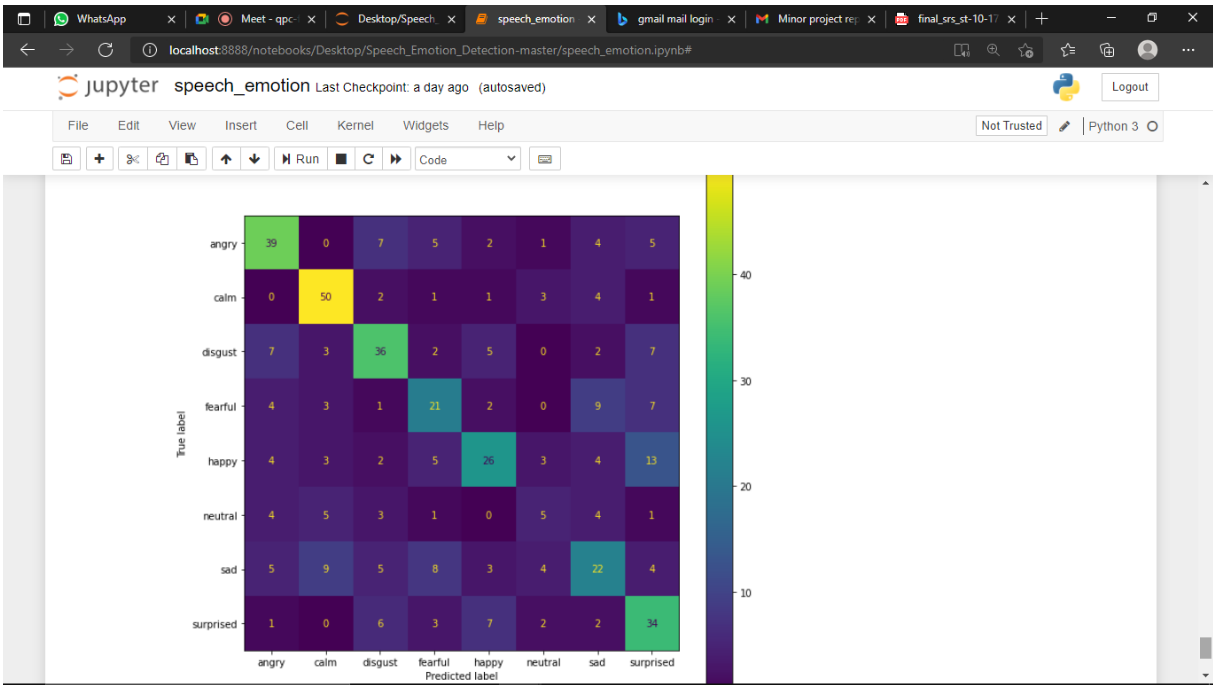
Screenshot 4: Recorded audio will be displayed



Screenshot 5: Plotting of Recorded Audio



Screenshot 6: Emotion Displayed on GUI



Screenshot 7: Confusion matrix of the SER model

Chapter 7: Software Testing

7.1 Test cases

White Box Testing: Here we are testing internal operations of system, it involves testing of software code forflow of specific inputs through the code.

Unit Testing: Here individual units or components of software are tested the purpose is to validate that each unit of software code performs as expected .It is done in developmentphase of an application by developers.

Integration Testing: Here individual units or components of the application's source codes are combined and tested as a group. The purpose is to expose errors in the interactions of different interfaces with one another. It is done after unit testing.

7.2 Test cases and Validation

Test Case Id	Description	Input	Expected Output	Actual Output	Remark
1	Able to import required libraries	Import statements	Able to import	Able to import	Pass
2	Check whether correct path is given for audio files	Path	Correct path	Correct path	Pass
3	Check whether able to load ravdess dataset	Load ravdess dataset and audio files	Able to load ravdess dataset	Able to load ravdess dataset	Pass
4	Check Use the Speech-Recognition API	Get all audio from audio files	Able to get audio into text	Able to get audio into text	Pass

	to get the Raw Text from Audio Files				
5	Check Data cleaning	Function for removing unnecessary empty voices around the main audio files	Background noises are removed.	Background noises are removed.	Pass
6.	Check is The clean Audio Files are redirected to Clean Audio Folder Directory	Function that will clean the Audio Files then be given to Clean Audio Folder Directory	Able to clean audio	Able to clean audio	Pass
7.	Check for Extraction of feature vector from all audios	function extract_features() will extract features for all audios and we will map audio names with their respective feature array. Then we will dump the features dictionary into a "Emotion_Voice_Detection_Model.pkl" pickle file.	Emotion_Voice_Detection_Model.pkl file	Emotion_Voice_Detection_Model.pkl file	Pass
8.	Check for loading the data for training model	Load clean folder	Data loaded	Data loaded	Pass
9.	Check for applying the MLP Classifier	MLP classifier will be used to initialise and train the model	Classifier initialized	Classifier initialized	Pass

10.	Check Training of model	define model function	Models contain trained model	Models contain trained model	Pass
11.	Testing the model	Saved model will again be loaded which will load the model and generate predictions	Output predicted	Output predicted	Pass
12.	Check the module loaded properly	Load website	Loaded properly	Loaded properly	pass
13.	If only Record Button is enabled	Record button on module	Enabled	Enabled	pass
14.	Stop button is disabled	Stop button on module	Disabled	Disabled	pass
15.	Check whether record button is clickable	Module and record button	Clickable	Clickable	Pass
16.	After clicking record asking microphone permission expected	Dialgue box in module	Dialogue box appeared	Dialogue box appeared	pass
17.	If permission allowed then record audio	Dialgue box in module	Allow clickable	Allow clickable	pass
18.	Pause and Stop Button Enabled	Pause and stop button in module	Enabled	Enabled	pass
19.	Record button to be disabled	Record button in module	Disabled	Disabled	pass

20.	Pause, Resume and Stop button working	Module	Working	Working	pass
21.	After Stop button audio stored in .wav format with name.	Audio playing module	Saved	Saved	pass
22.	Check whether audio is been playing which is recorded	Audio playing module	Audio is playing	Audio is playing	pass
23.	A link to save the file on the module should be generated	Audio playing module	Link generated	Link generated	pass
24.	Link is clickable	link	Clickable	Clickable	Pass
25.	Permission asked to stored the file	Dialogue box on model	Dialogue box appeared	Dialogue box appeared	Pass
26.	Check the output for the given audio file in ipynb file	Ipynb module with the function loaded to predict output	Output predicted and stored in database	Output predicted and stored in database	Pass
27.	Check whether fetch button is working and loading the output from the database	Database connected and web browser loading the module and displaying the output	Output displayed on the browser	Output displayed on the browser	Pass

28	Loading the required libraries	Import <library name>	Imported successfully	No library found	Fail
29	Loading the Dataset	Ravdees dataset	Dataset loaded	Dataset not found	Fail
30	Cleaning the dataset	Ravdees dataset	Cleaned successfully	Something went wrong	Fail
31	Checking for permission	web page	Permission granted	Permission denied	Fail
32	Writing to csv file	Csv file	Successful	Permission denied	Fail

Chapter 8: Conclusion

It can be used in virtual assistants such as google assistant, Siri, Alexa etc. Machine can also understand the emotion of the human and can respond in corresponding way. Speech emotion model is improving the human computer interaction. The proposed model achieved an accuracy of 80.67%. Calm was the best identified emotion. The model gets confused between similar emotions like calm-neutral, happy-surprised. The model was tested using many test cases. The failed test cases were corrected. The system could take into consideration multiple speakers from different geographic locations speaking with different accents. Though standard feed forward MLP is powerful tool for classification problems. Study shows that people suffering with autism have difficulty expressing their emotions explicitly. Image based speech processing in real time can prove to be of great assistance.

Chapter 9: Future Enhancements

Some of the drawbacks can be resolved in the future to make the model more accurate and efficient.

- The model can be improved by training the model a variety of datasets which increases the accuracy of the model.
- Removing the disturbance from the input audio which deviates from correct prediction.
- Adding more emotions to the system since this system can identify only 8 emotions. Extracting more features from the speech to improve the classification process

Bibliography

[1] Proceedings of the International Conference on Smart Electronics and Communication (ICOSEC 2020) IEEE Xplore Part Number: CFP20V90-ART; ISBN: 978-1-7281-5461-9

[2] 2017 4th International Conference on Signal Processing and Integrated Networks (SPIN)

[3] Proceedings of the Third International Conference on Computing Methodologies and Communication (ICCMC 2019) IEEE Xplore Part Number: CFP19K25-ART; ISBN: 978-1-5386-7808-4

[4] International Conference on Inventive Communication and Computational Technologies (ICICCT 2017)

[5] Proc. of the 2017 IEEE Region 10 Conference (TENCON), Malaysia, November 5-8, 2017

[6] 2015 International Conference on Affective Computing and Intelligent Interaction (ACII)

[7] Received July 25, 2019, accepted August 5, 2019, date of publication August 19, 2019, date of current version September 4, 2019

[8] Received May 10, 2019, accepted June 2, 2019, date of publication June 6, 2019, date of current version June 24, 2019

[9] Mustaqeem et al.: Clustering-Based SER by Incorporating Learned Features and IEEE Access, vol. 6, pp. 52227–52237, 2018.

[10] International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 16 (2017) pp. 5760-5767