SPEECH EMOTION RECOGNITION USING MLPCLASSIFIER

A MINI PROJECT REPORT

Submitted by

HARINEE S 212220040039

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

SAVEETHA ENGINEERING COLLEGE , KANCHEEPURAM

ANNA UNIVERSITY: CHENNAI- 600 025

DECEMBER 2022

BONAFIDE CERTIFICATE

Certified that this Mini Project report **"SPEECH EMOTION RECOGNITION USING MLP CLASSIFIER"** is the bonafide work of **HARINEE S (212220040039),** who carried out the mini project work under my supervision.

SIGNATURE	**SIGNATURE**
Ms. K.S. Rekha, M.E.,	**Dr. G. Nagappan, M.E., PhD**
Assistant Professor	**Professor**
SUPERVISOR	**HEAD OF THE DEPARTMENT**
Dept of Computer Science and Engineering,	Dept of Computer Science and Engineering,
Saveetha Engineering College,	Saveetha Engineering College,
Thandalam, Chennai 602105.	Thandalam, Chennai 602105.

DATE OF THE VIVA VOCE EXAMINATION: ……………………...

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

I express my deep sense of gratitude to our honourable and beloved Founder President **Dr. N. M. VEERAIYAN**, our President **Dr. SAVEETHA RAJESH**, our Director **Dr. S. RAJESH** and other management members for providing the infrastructure needed.

I express my wholehearted gratitude to our principal**, Dr. N. DURAIPANDIAN,** for his wholehearted encouragement in completing this project.

I convey my thanks to **Dr. G. NAGAPPAN**, Professor and Head of the Department of Computer Science and Engineering, Saveetha Engineering College, for his kind support and for providing necessary facilities to carry out the project work.

I would like to express my sincere thanks and deep sense of gratitude to my Supervisor **Ms. K.S.REKHA**, Assistant Professor, Department of Computer Science and Engineering, Saveetha Engineering College and **Dr.M. VIJAY ANAND**, Professor, Department of Computer Science and Engineering, Saveetha Engineering College for their valuable guidance, suggestions and constant encouragement that paved the way for the successful completion of the project work and for providing us necessary support and details at the right time and during the progressive
reviews.

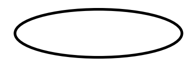
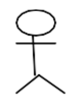
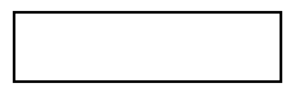
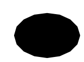
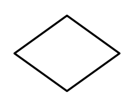
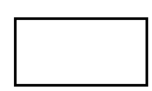
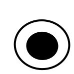
I owe my thanks to all the members of our college, faculty, staff and technicians for their kind and valuable cooperation during the course of the project. I am pleased to acknowledge my sincere thanks to my beloved parents, friends and well-wishers who encouraged me to complete this project successfully.

ABSTRACT

Speech Emotion Recognition, abbreviated as SER, is the act of attempting to recognize human emotion and the associated affective states from speech. This is capitalizing on the fact that voice often reflects underlying emotion through tone and pitch. Emotion recognition is a rapidly growing research domain in recent years. Unlike humans, machines lack the abilities to perceive and show emotions. But human-computer interaction can be improved by implementing automated emotion recognition, thereby reducing the need of human intervention. In this project, basic emotions like calm, happy, fearful, disgust etc. are analyzed from emotional speech signals. We use machine learning techniques like Multilayer perceptron Classifier (MLP Classifier) which is used to categorize the given data into respective groups which are non linearly separated. Mel-frequency cepstrum coefficients (MFCC), chroma and mel features are extracted from the speech signals and used to train the MLP classifier. For achieving this objective, we use python libraries like Librosa, sklearn, pyaudio, numpy and soundfile to analyze the speech modulations and recognize the emotion.

LIST OF FIGURES

LIST OF SYMBOLS

S.NO.	SYMBOL NAME	SYMBOL
1.	Use case	
2.	Actor	
3.	Process	
4.	Start	
5.	Decision	
6.	Unidirectional	
7.	Entity set	
8.	Stop	

LIST OF ABBREVIATIONS

S.NO.	ABBREVIATIONS	EXPANSION
1.	SER	Speech Emotion Recognition
2.	MLP	Multi Layer Perceptron
3.	MFCC	Mel Frequency Cepstrum
4.	SVM	Support Vector Mechanism
5.	ANN	Artificial Neural Networks
6.	LPCC	Linear Predictive Cepstral Model
7.	KNN	K Nearest Neighbour
8.	PRNN	Pattern Recognition Neural Network
9.	STE	Short-Term Energy
10.	RAVDESS	Ryerson Audio-Visual Database of Emotional Speech and Song
11.	DBM	Deep Belief Networks
12.	HMM	Hidden Markov Model

13.	GMM	Gaussian mixture models
14.	BN	Bayesian Networks

CHAPTER-1

INTRODUCTION

1.1 OVERVIEW OF THE PROJECT:

In this project, basic emotions like calm, happy, fearful, disgust etc. are analyzed from emotional speech signals. We use machine learning techniques like Multilayer perceptron Classifier (MLP Classifier) which is used to categorize the given data into respective groups which are non linearly separated. Mel-frequency cepstrum coefficients (MFCC), chroma and mel features are extracted from the speech signals and used to train the MLP classifier. For achieving this objective, we use python libraries like Librosa, sklearn, pyaudio, numpy and soundfile to analyze the speech modulations and recognize the emotion.

1.2 SCOPE AND OBJECTIVE

SCOPE:

Emotion recognition provides benefits to many institutions and aspects of life. It is useful and important for security and healthcare purposes. Also, it is crucial for easy and simple detection of human feelings at a specific moment without actually asking them.

OBJECTIVE:

The primary objective of SER is to improve man-machine interface. It can also be used to monitor the psycho physiological state of a person in lie detectors. In recent time, speech emotion recognition also find its applications in medicine and forensics.

CHAPTER 2
LITERATURE SURVEY

2.1 INTRODUCTION:

A literature survey or a literature review in a project report is that section which shows various analysis and research made in the field of your interest and the results already published, taking into account the various parameters of the project and the extent of project. It is the most important part of your report as it gives you a direction in the area of your research. It helps you set a goal for your analysis - thus giving you your problem of statement.

2.2. LITERATURE SURVEY

1.Girija Deshmukh, (2019), "Speech based emotion recognition using machine learning, 17th European Signal Processing Conference, vol 387.

Girija Deshmukh proposed a system in which they obtained audiosamples of Short-Term Energy (STE), Pitch, and MFCC coefficients infrustration, happiness, and sadness of emotions. Open source North American English served as expression and as feedback was used to record natural speech. Thus, only three emotions i.e., anger, happiness and sadness were recognized. They also identified the speaker's detailed features, such as sound, energy, pitch. The whole Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) dataset is manually split into train and test sets. The multi-class Support vector machine(SVM) takes feature vectors as input, which is turned up as a model corresponding to each emotion.

2.Peng Shi in , (2018)"Speech emotion recognition based on deep belief network", International Journal of Computer Applications, vol. 1, pp.6-9.

Peng Shi in introduced discrete model and continuous model of speech emotion recognition; different characteristics are analysed to make better description of emotions. When compared to Artificial Neural Networks (ANNs) and support vector machines (SVMs), the Deep Belief Networks (DBNs) have about 5% higher accuracy rate than the traditional methods. The output shows that the features which are extracted by Deep Belief Networks is much better than the original feature. DBN-SVM had slightly improved result than DBN-DN because SVM classifies in small size better. DBN converts empty characteristics into deep abstract characteristics, resulting into better classification.

3.M.S. Likitha , (2017) ,"Speech based human emotion recognition using MFCC " , International Journal of Applied Information Systems (IJAIS), (2013), pp. 5.8.

M.S. Likitha observed recognition requires assessment of the verbal communication wave to classify the required feeling, based on the training of its characteristics, like Sound, format, phoneme. On the side of withdrawal of functionality and examination, A good number of algorithms were made of a speech signal. The acoustic precision of the communication kinesics is a feature. Withdrawal of features is the process of removing a compact amount of information from the voice signal employed to reflect each speaker later on. Most Methods of extraction are at one's fingertips but the widely used method is coefficient (MFCC).

4.Edward Jones , (2019), “Speech Emotion Recognition Using Deep Learning Techniques: A Review” , International Conference on Computing for Sustainable Global Development, vol. 12.

Edward Jones considered Speech emotion recognition as exciting ingredient of Human Computer Interaction (HCI). The main approach for SER must be feature extraction and feature classification. Linear And nonlinear classifiers can be used for Feature classification. In linear classifiers, frequently used classifiers are Support Vector Machines (SVMs), Bayesian Networks (BN). Since, Speech signal is considered variating, thus, these types of classifiers work effectively for SER.Deep learning techniques possess more advantages for SER when compared to traditional methods.

5.Michael Neumann ,(2019), “Improving Speech Emotion Recognition with Unsupervised Representation Learning on Unlabeled Speech” , International Conference on Computing for Sustainable Global Development, vol. 134.

Michael Neumann presented their conclusions illustration gaining knowledge on unlabelled voice entity can be appropriate for Speech Emotion Recognition (SER). They have used t-distributed neighbour embeddings (t-SNE) to analyse visualizations of different representations. However, no divisible clusters are found in the 2D projections. These plots are excluded as of capacity they require. The autoencoder is trained on a large dataset. They have incorporated representations generated by autoencoders, which, in turn leads to steady developments in identification accuracy of SER model.

CHAPTER 3

SYSTEM ANALYSIS

3.1 Algorithm-SER:

Our SER system consists of four main steps. First is the voice sample collection. The second features vector that is formed by extracting the features. As the next step, we tried to determine which features are most relevant to differentiate each emotion. These features are introduced to machine learning classifier for recognition.

3.2 Advantages of using MLPClassifier for SER:

1. Provides the flexibility to work with nonlinear values

2. Less number of parameters required

3. Higher performance compared to previous systems

4. Better classification of parameters is shown.

5. Can handle missing values, model complex relationships and support multiple inputs.

3.3 Disadvantages of using MLPClassifier for SER:

1. MLPs always need fixed number of inputs to be provided for fixed number of outputs, there is a fixed mapping function between the inputs and the outputs in these feed-forward neural networks that pose a problem when a sequence of inputs is provided to the model.

2. Network must be retrained when a new emotion is added to the system.

3.4 Existing System:

- The existing work in this area reveals that most of the present work relies on lexical analysis for emotion recognition, that have been used for the purpose of classification of emotions into three categories, i.e., Angry, Happy and Neutral.
- The maximum cross correlation between the discrete time sequences of the audio signals is computed and the highest degree of correlation between the testing audio file and the training audio file is used as an integral parameter for identification of a particular emotion type.
- The second technique is used with the feature extraction of discriminatory features with the Cubic SVM classifier for recognition of Angry, Happy and Neutral emotion segments only.

3.5 Proposed System:

In the project, MFCC has been used as the feature for classifying the speech data into various emotion categories employing artificial neural networks.

The usage of the Neural Networks provides us the advantage of classifying many different types of emotions in a variable length of audio signal in a real time environment.

CHAPTER 4

SYSTEM DESIGN

4.1 Architecture diagram:

An architectural diagram is a visual representation that maps out the physical implementation for components of a software system. It shows the general structure of the software system and the associations, limitations, and boundaries between each element.

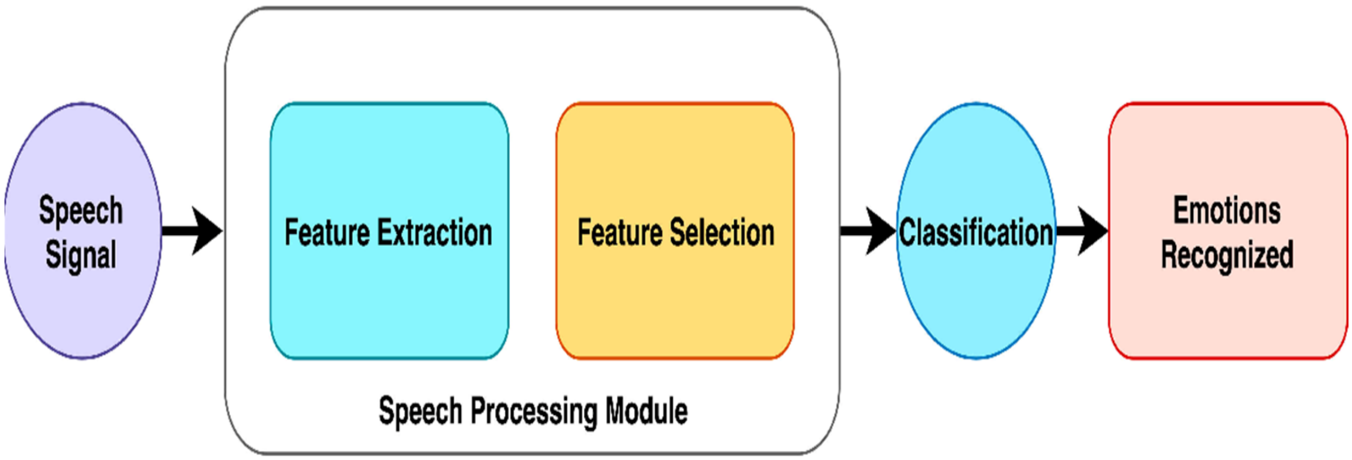


Fig 4.1 Architecture diagram

4.2 UML DIAGRAMS

4.2.1 USE CASE DIAGRAM:

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

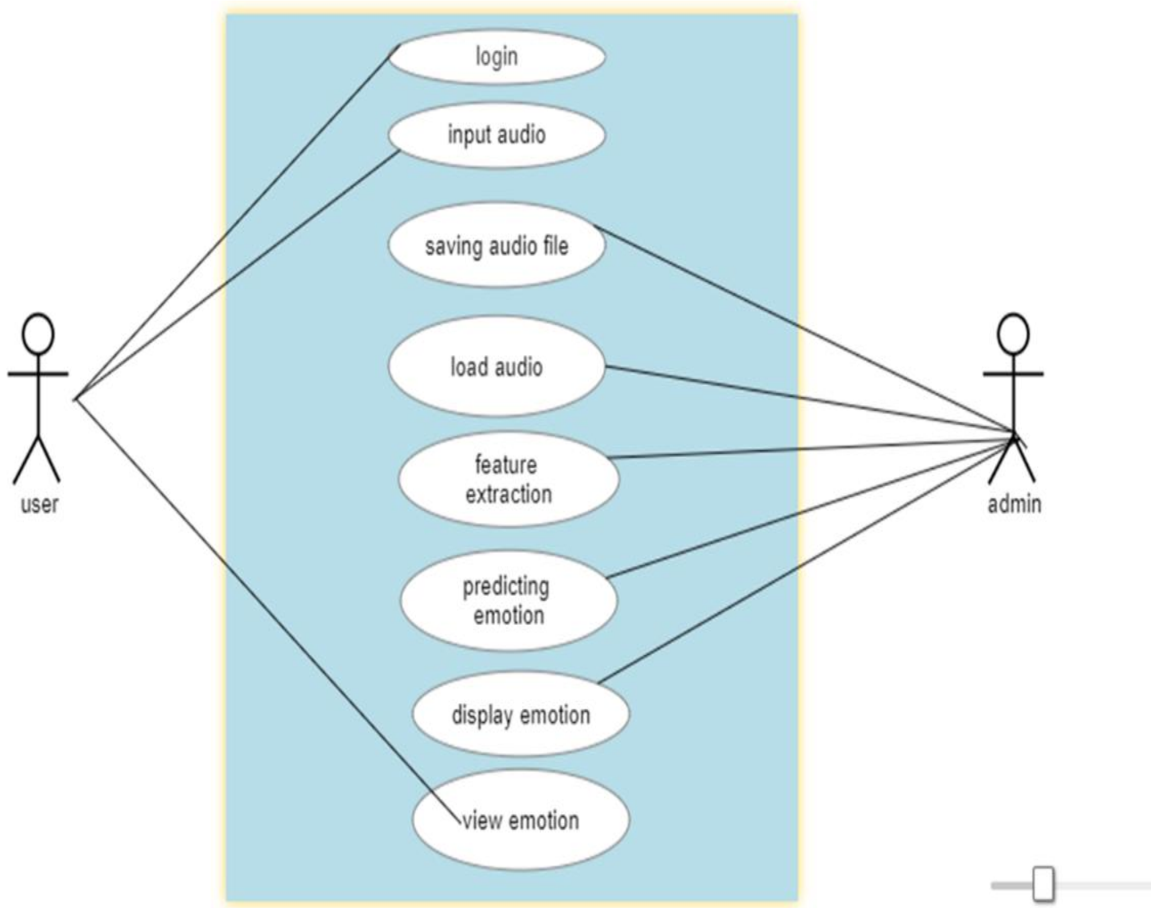


Fig 4.2.1 Use case diagram

4.2.2 SEQUENCE DIAGRAM:

A sequence diagram is a Unified Modeling Language (UML) diagram that illustrates the sequence of messages between objects in an interaction. A sequence diagram consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction.

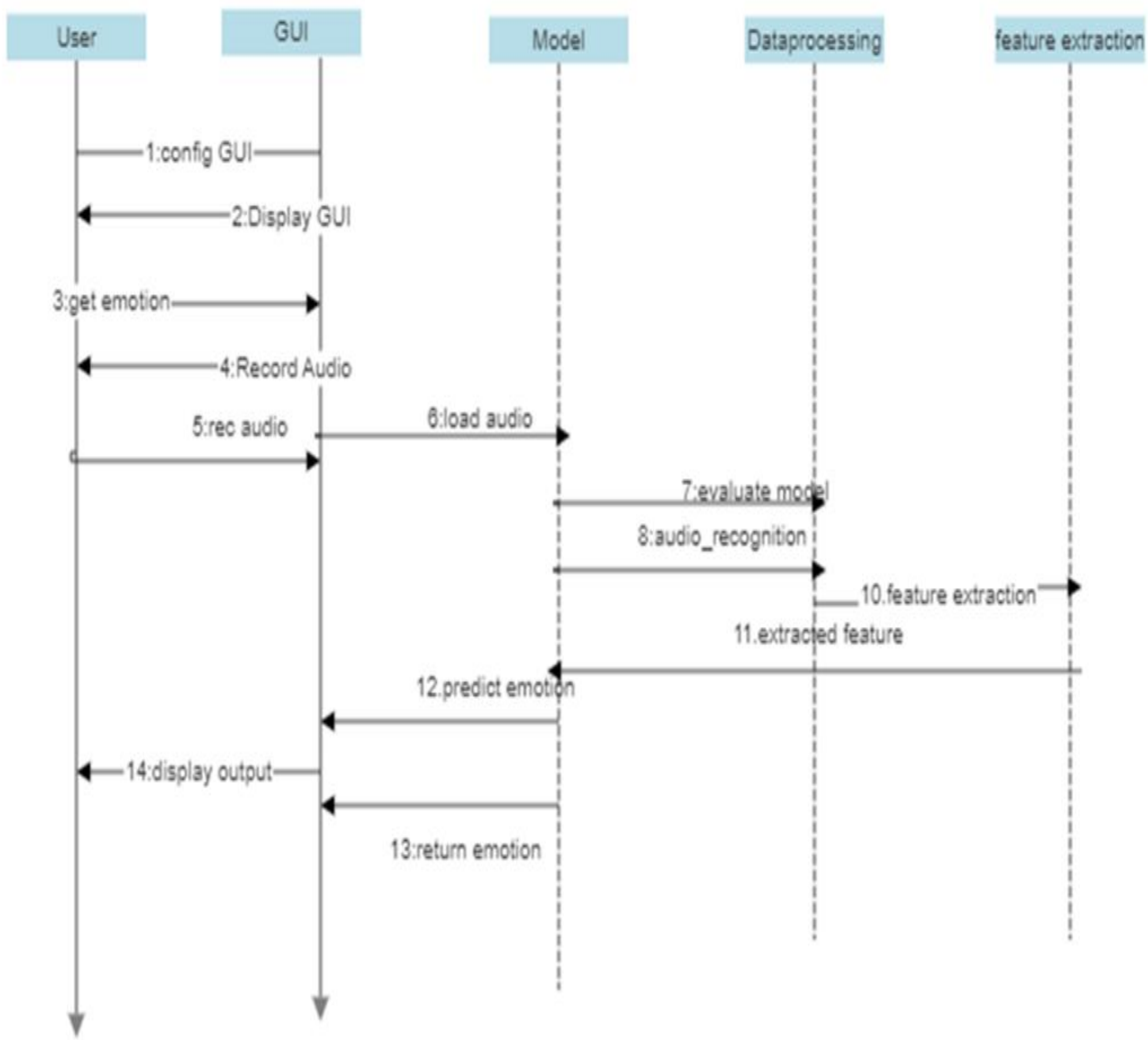


Fig 4.2.2 Sequence diagram

4.2.3 ACTIVITY DIAGRAM:

An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modeling. They can also describe the steps in a use case diagram.

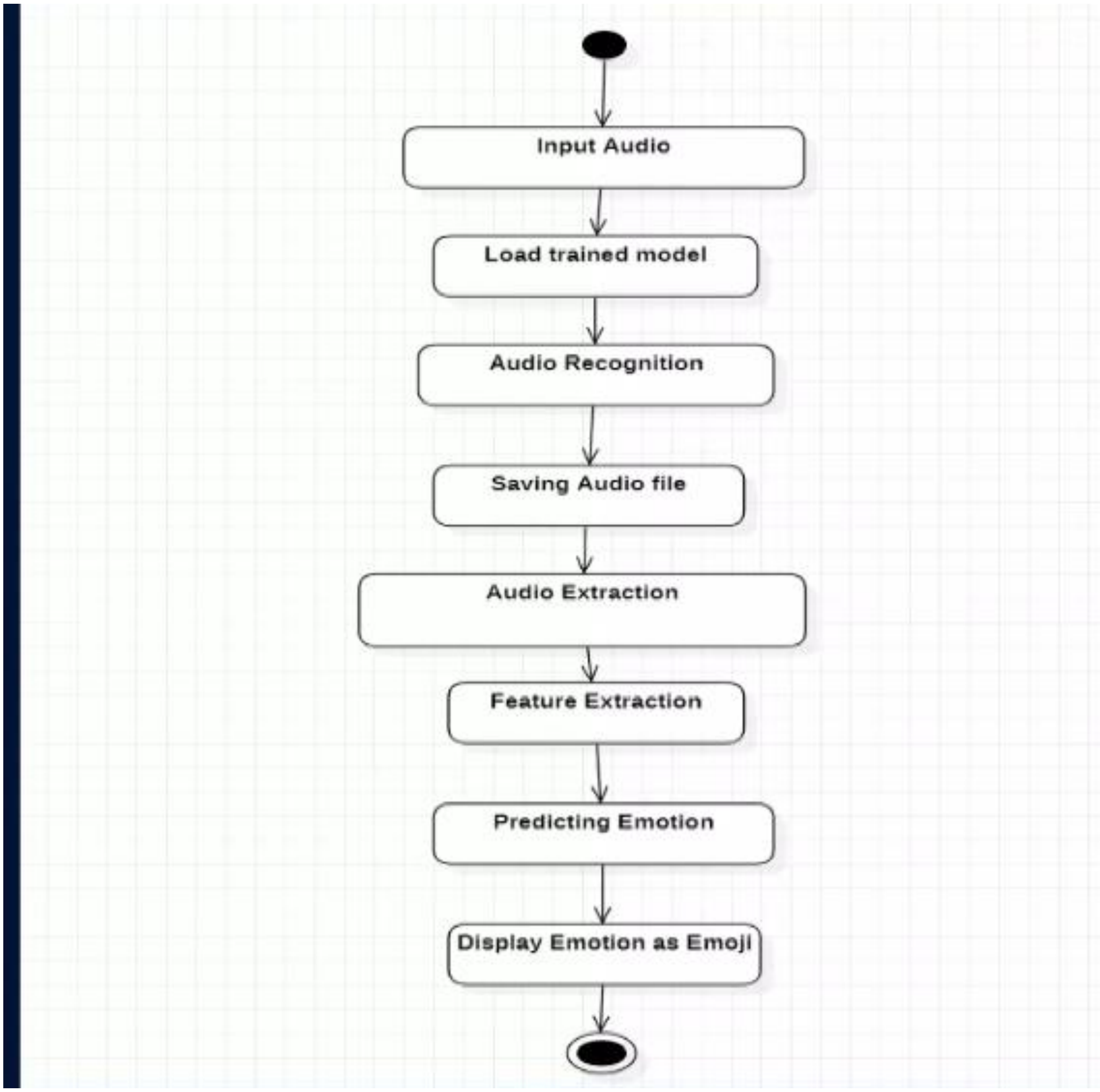


Fig 4.2.3 Activity Diagram

4.3 Hardware Requirements:

- Processor - CORE i3
- Hard disk - 250GB
- RAM -8 GB

4.4 Software Requirements:

- Operating system : WINDOWS 10
- Programming language: Python

CHAPTER 5

IMPLEMENTATION AND ANALYSIS:

5.1 Python Library

NUMPY:

NumPy offers comprehensive mathematical functions, random number generators, linear algebra routines, Fourier transforms, and more. NumPy's high level syntax makes it accessible and productive for programmers from any background or experience level.

PANDAS:

pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool,Built on top of the Python programming language.

MATPLOTLIB:

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible. Create publication quality plots. Make interactive figures that can zoom, pan, update.

PLOTLY:

The plotly Python library is an interactive, open-source plotting library that supports over 40 unique chart types covering a wide range of statistical, financial, geographic, scientific, and 3-dimensional use-cases.

REQUEST:

The request module allows you to send HTTP requests using Python.The HTTP request returns a response object with all the response data (content, encoding, status, etc).

LIBROSA:

Librosa is a Python package for **music and audio analysis**. Librosa is basically used when we work with audio data like in music generation(using LSTM's), Automatic Speech Recognition. It provides the building blocks necessary to create the music information retrieval systems.

STREAMLIT:

Streamlit is an open source app framework in Python language. It helps us create web apps for data science and machine learning in a short time. It is compatible with major Python libraries such as scikit-learn, Keras, NumPy, pandas, Matplotlib etc.

TENSORFLOW:

The TensorFlow platform helps you implement best practices for data automation, model tracking, performance monitoring, and model retraining. Using production-level tools to automate and track model training over the lifetime of a product, service, or business process is critical to success.

5.2 List of Modules:

5.2.1 Feature extraction

Feature extraction is based on partitioning speech into small intervals known as frames. To select suitable features which are carrying information about emotions from speech signal is an important step in SER system. There are two types of features: prosodic features including energy, pitch and spectral features including **MFCC** Mel-Frequency Cepstrum coefficients is the most important feature of speech with simple calculation, good ability of distinction, anti-noise. MFCC in the low frequency region has a good frequency resolution, and the robustness to noise is also very good.

5.2.2 MLP CLASSIFIER:

1. Multilayer perceptrons are often applied to supervised learning problems. They train on a set of input-output pairs and learn to model the correlation (or dependencies) between those inputs and outputs.
2. The network thus has a simple interpretation as a form of input-output model, with the weights and thresholds (biases) the free parameters of the model. Important issues in MLP design include specification of the number of hidden layers and the number of units in these layers. The number of hidden units to use is far from clear. As good a starting point as any is to use one hidden layer, with the number of units equal to half the sum of the number of input and output units

5.2.3 TESTING AND TRAINING:

We are loading the data where it takes in the relative size of the test set as parameter. X and Y are empty lists, functions will checks whether the emotion are in the list of observed emotions. The feature will be send to X and emotions to Y. Now the testing and training function will be called. 75% of audio will be tested at the same time 25% of audio will trained. For classification we are using MLP Classifier.

5.2.4 PREDICTION:

Once a neural network has been trained it can be used to make various predictions. You can make predictions on test data in order to estimate the skill of the model on unseen data. You can also deploy it operationally and use it to make predictions continuously.

5.3 SOFTWARE DESCRIPTION

PYTHON

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics developed by Guido van Rossum. It was originally released in 1991. Designed to be easy as well as fun, the name "Python" is a nod to the British comedy group Monty Python. Python has a reputation as a beginner-friendly language, replacing Java as the most widely used introductory language because it handles much of the complexity for the user, allowing beginners to focus on fully grasping programming concepts rather than minute details.Python is used for server-side web development, software development, mathematics, and system scripting, and is popular for Rapid Application Development and as a scripting or glue language to tie existing components because of its high-level, built-in data structures, dynamic typing, and dynamic binding. Program maintenance costs are reduced with Python due to the easily learned syntax and emphasis on readability. Additionally, Python's support of modules and packages facilitates modular programs and reuse of code. Python is an open source community language, so numerous independent programmers are continually building libraries and functionality for it.

CHAPTER 6

6.1 CONCLUSION

This project is focused on improving the performance of a machine learning model in the speech dataset, which is the Ryerson Audio-Visual Database of Emotional Speech and Song. The dataset contains both speech and song data, with a total of 2452 audio files with different emotions (calm, fearful, happy, surprise, sad, disgust, angry, and neutral emotions). The data are not balanced enough, but theyare also not highly unbalanced .The best performance was obtained by the MLP classifier in previous works. Several parameters of a default MLP classifier are altered in this study, such as the design of the MLP classifier's hidden layer (750,750,750). Since the number of layers is quite high, it can cause overfit for high iteration rates; that is why the number of iterations during training is kept low so that overfitting is avoided. Otherwise, this model will cause overfit on training data for high iteration numbers on data-augmented datasets. For the preprocessing part, different approaches are also applied here. MFCC images are used but did not result in better performance, so the dataset is kept as a one-dimensional array. Feature extraction is the part that takes most of the time in this implementation, which took longer than that taken to train the model. The mean of MFCC features is calculated, and then short-time Fourier transform and Mel spectrogram features are obtained. After the training process of the MLP classifier, the model is tested on the test dataset, which is 25% of the original data that was not used during training. As a result, an overall accuracy of 81% is obtained, whose performance is better than that of both the classification report and confusion matrix that was used. The highest performance is observed in angry emotion while the lowest performance is observed in happy emotion.

APPENDICES

SAMPLE CODING:

```
from sklearn.neural_network import MLPClassifier

from sklearn.metrics import accuracy_score
from utils import load_data

import os
import pickle

# load RAVDESS dataset
X_train, X_test, y_train, y_test = load_data(test_size=0.25)
# print some details
# number of samples in training data
print("[+] Number of training samples:", X_train.shape[0])
# number of samples in testing data
print("[+] Number of testing samples:", X_test.shape[0])
# number of features used
# this is a vector of features extracted
# using utils.extract_features() method
print("[+] Number of features:", X_train.shape[1])
# best model, determined by a grid search
model_params = {
    'alpha': 0.01,
    'batch_size': 256,
    'epsilon': 1e-08,
    'hidden_layer_sizes': (300,),
    'learning_rate': 'adaptive',
    'max_iter': 500,
}
# initialize Multi Layer Perceptron classifier
# with best parameters ( so far )
model = MLPClassifier(**model_params)

# train the model
print("[*] Training the model...")
model.fit(X_train, y_train)

# predict 25% of data to measure how good we are
y_pred = model.predict(X_test)
```

```python
# calculate the accuracy
accuracy = accuracy_score(y_true=y_test, y_pred=y_pred)

print("Accuracy: {:.2f}%".format(accuracy*100))

# now we save the model
# make result directory if doesn't exist yet
if not os.path.isdir("result"):
    os.mkdir("result")

pickle.dump(model, open("result/mlp_classifier.model", "wb"))
import numpy as np
import streamlit as st
import cv2
import librosa
import librosa.display
from tensorflow.keras.models import load_model
import os
from datetime import datetime
import streamlit.components.v1 as components
import matplotlib.pyplot as plt
from PIL import Image
from melspec import plot_colored_polar, plot_melspec
import random
# load models
model = load_model("model3.h5")
# constants
starttime = datetime.now()
CAT6 = ['fear', 'angry', 'neutral', 'happy', 'sad', 'surprise']
CAT7 = ['fear', 'disgust', 'neutral', 'happy', 'sad', 'surprise', 'angry']
CAT3 = ["positive", "neutral", "negative"]
COLOR_DICT = {"neutral": "grey",
              "positive": "green",
              "happy": "green",
              "surprise": "orange",
              "fear": "purple",
              "negative": "red",
              "angry": "red",
              "sad": "lightblue",
              "disgust": "brown"}

TEST_CAT = ['fear', 'disgust', 'neutral', 'happy', 'sad', 'surprise', 'angry']
TEST_PRED = np.array([.3, .3, .4, .1, .6, .9, .1])

# page settings
```

```python
st.set_page_config(page_title="Mini-project speech emotion analyzer",
                   page_icon="images/favicon.ico", layout="wide")
def log_file(txt=None):
    with open("log.txt", "a") as f:
        datetoday = datetime.now().strftime("%d/%m/%Y %H:%M:%S")
        f.write(f"{txt} - {datetoday};\n")
# @st.cache
def save_audio(file):
    if file.size > 4000000:
        return 1
    # if not os.path.exists("audio"):
    #     os.makedirs("audio")
    folder = "audio"
    datetoday = datetime.now().strftime("%d/%m/%Y %H:%M:%S")
    # clear the folder to avoid storage overload
    for filename in os.listdir(folder):
        file_path = os.path.join(folder, filename)
        try:
            if os.path.isfile(file_path) or os.path.islink(file_path):
                os.unlink(file_path)
        except Exception as e:
            print('Failed to delete %s. Reason: %s' % (file_path, e))

    try:
        with open("log0.txt", "a") as f:
            f.write(f"{file.name} - {file.size} - {datetoday};\n")
    except:
        pass

    with open(os.path.join(folder, file.name), "wb") as f:
        f.write(file.getbuffer())
    return 0
# @st.cache
def get_melspec(audio):
    y, sr = librosa.load(audio, sr=44100)
    X = librosa.stft(y)
    Xdb = librosa.amplitude_to_db(abs(X))
    img = np.stack((Xdb,) * 3, -1)
    img = img.astype(np.uint8)
    grayImage = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    grayImage = cv2.resize(grayImage, (224, 224))
    rgbImage = np.repeat(grayImage[..., np.newaxis], 3, -1)
    return (rgbImage, Xdb)
```

```python
# @st.cache
def get_mfccs(audio, limit):
    y, sr = librosa.load(audio)
    a = librosa.feature.mfcc(y, sr=sr, n_mfcc=40)
    if a.shape[1] > limit:
        mfccs = a[:, :limit]
    elif a.shape[1] < limit:
        mfccs = np.zeros((a.shape[0], limit))
        mfccs[:, :a.shape[1]] = a
    return mfccs


@st.cache
def get_title(predictions, categories=CAT6):
    title = f"Detected emotion: {categories[predictions.argmax()]} \
    - {predictions.max() * 100:.2f}%"
    return title
@st.cache
def color_dict(coldict=COLOR_DICT):
    return COLOR_DICT


@st.cache
def plot_polar(fig, predictions=TEST_PRED, categories=TEST_CAT,
               title="TEST", colors=COLOR_DICT):
    # color_sector = "grey"

    N = len(predictions)
    ind = predictions.argmax()

    COLOR = color_sector = colors[categories[ind]]
    theta = np.linspace(0.0, 2 * np.pi, N, endpoint=False)
    radii = np.zeros_like(predictions)
    radii[predictions.argmax()] = predictions.max() * 10
    width = np.pi / 1.8 * predictions
    fig.set_facecolor("#d1d1e0")
    ax = plt.subplot(111, polar="True")
    ax.bar(theta, radii, width=width, bottom=0.0,
           color=color_sector, alpha=0.25)

    angles = [i / float(N) * 2 * np.pi for i in range(N)]
    angles += angles[:1]

    data = list(predictions)
    data += data[:1]
```

```
    plt.polar(angles, data, color=COLOR, linewidth=2)
    plt.fill(angles, data, facecolor=COLOR, alpha=0.25)

    ax.spines['polar'].set_color('lightgrey')
    ax.set_theta_offset(np.pi / 3)
    ax.set_theta_direction(-1)
    plt.xticks(angles[:-1], categories)
    ax.set_rlabel_position(0)
    plt.yticks([0, .25, .5, .75, 1], color="grey", size=8)
    plt.suptitle(title, color="darkblue", size=12)
    plt.title(f"BIG {N}\n", color=COLOR)
    plt.ylim(0, 1)
    plt.subplots_adjust(top=0.75)


def main():
    side_img = Image.open("images/emotion.jpg")
    with st.sidebar:
        st.image(side_img, width=300)
        st.title('Speech Emotion recogonizer')
        st.markdown(
            '<div style="text-align: right;">-by Harinee 🐰</div>',
unsafe_allow_html=True)
    st.sidebar.subheader("Menu")
    website_menu = st.sidebar.selectbox("Menu", ("Emotion Recognition", "Project
description",
                                                 "Relax"))
    st.set_option('deprecation.showfileUploaderEncoding', False)

    if website_menu == "Emotion Recognition":
        st.sidebar.subheader("Model")
        model_type = st.sidebar.selectbox(
            "How would you like to predict?", ("mfccs", "mel-specs"))
        em3 = em6 = em7 = gender = False
        st.sidebar.subheader("Settings")

        st.markdown("## Upload the file")
        with st.container():
            col1, col2 = st.columns(2)
            # audio_file = None
            # path = None
            with col1:
                audio_file = st.file_uploader(
                    "Upload audio file", type=['wav', 'mp3', 'ogg'])
                if audio_file is not None:
```

```
                if not os.path.exists("audio"):
                    os.makedirs("audio")
                path = os.path.join("audio", audio_file.name)
                if_save_audio = save_audio(audio_file)
                if if_save_audio == 1:
                    st.warning("File size is too large. Try another file.")
                elif if_save_audio == 0:
                    # extract features
                    # display audio
                    st.audio(audio_file, format='audio/wav', start_time=0)
                    try:
                        wav, sr = librosa.load(path, sr=44100)
                        Xdb = get_melspec(path)[1]
                        mfccs = librosa.feature.mfcc(wav, sr=sr)
                        # # display audio
                        # st.audio(audio_file, format='audio/wav',
start_time=0)
                    except Exception as e:
                        audio_file = None
                        st.error(
                            f"Error {e} - wrong format of the file. Try
another .wav file.")
                else:
                    st.error("Unknown error")
            else:
                if st.button("Try test file"):
                    wav, sr = librosa.load("test.wav", sr=44100)
                    Xdb = get_melspec("test.wav")[1]
                    mfccs = librosa.feature.mfcc(wav, sr=sr)
                    # display audio
                    st.audio("test.wav", format='audio/wav', start_time=0)
                    path = "test.wav"
                    audio_file = "test"
        with col2:
            if audio_file is not None:
                fig = plt.figure(figsize=(10, 2))
                fig.set_facecolor('#d1d1e0')
                plt.title("Wave-form")
                librosa.display.waveshow(wav, sr=44100)
                plt.gca().axes.get_yaxis().set_visible(False)
                plt.gca().axes.get_xaxis().set_visible(False)
                plt.gca().axes.spines["right"].set_visible(False)
                plt.gca().axes.spines["left"].set_visible(False)
                plt.gca().axes.spines["top"].set_visible(False)
                plt.gca().axes.spines["bottom"].set_visible(False)
```

```
            plt.gca().axes.set_facecolor('#d1d1e0')
            st.write(fig)
        else:
            pass
```

Fig 7.1 Sample code

SAMPLE OUTPUT:

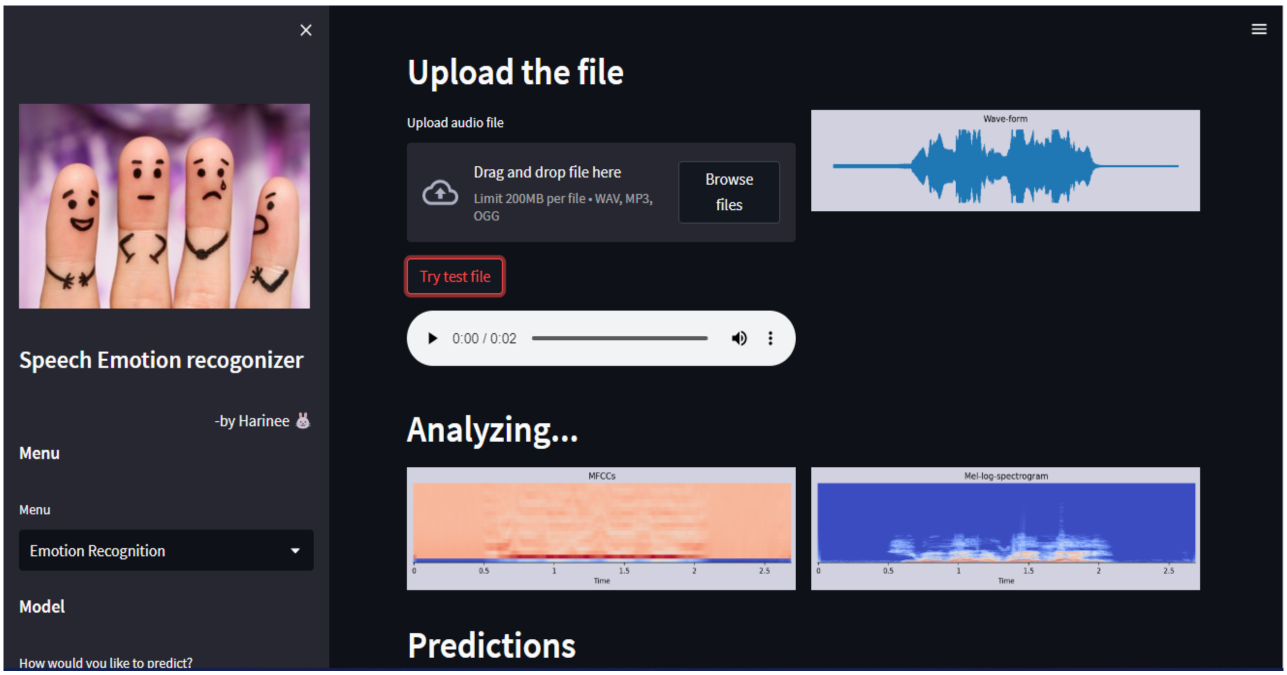
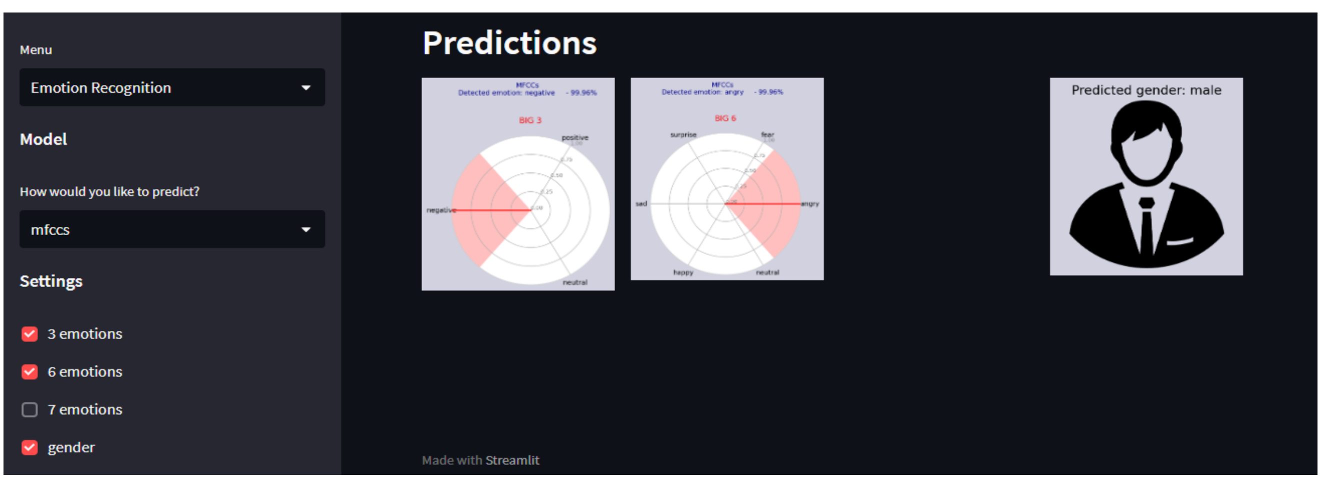


Fig 7.2 Sample output



WAVEFORM:

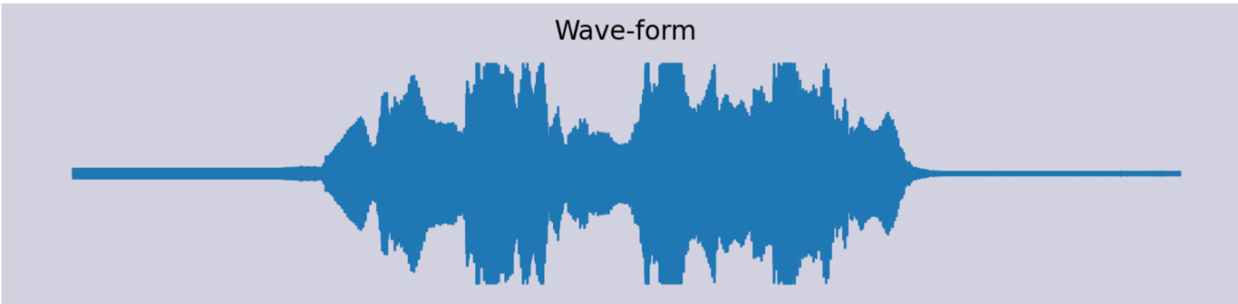


Fig 7.3 Waveform

FEATURE EXTRACTION USING MFCC:

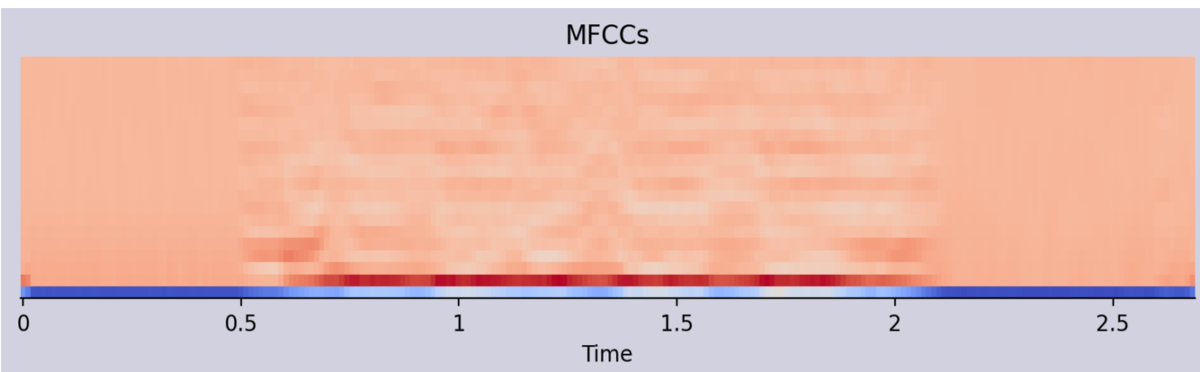


Fig 7.4 Feature extraction using MFCC

MFCC'S PREDICTED EMOTION:

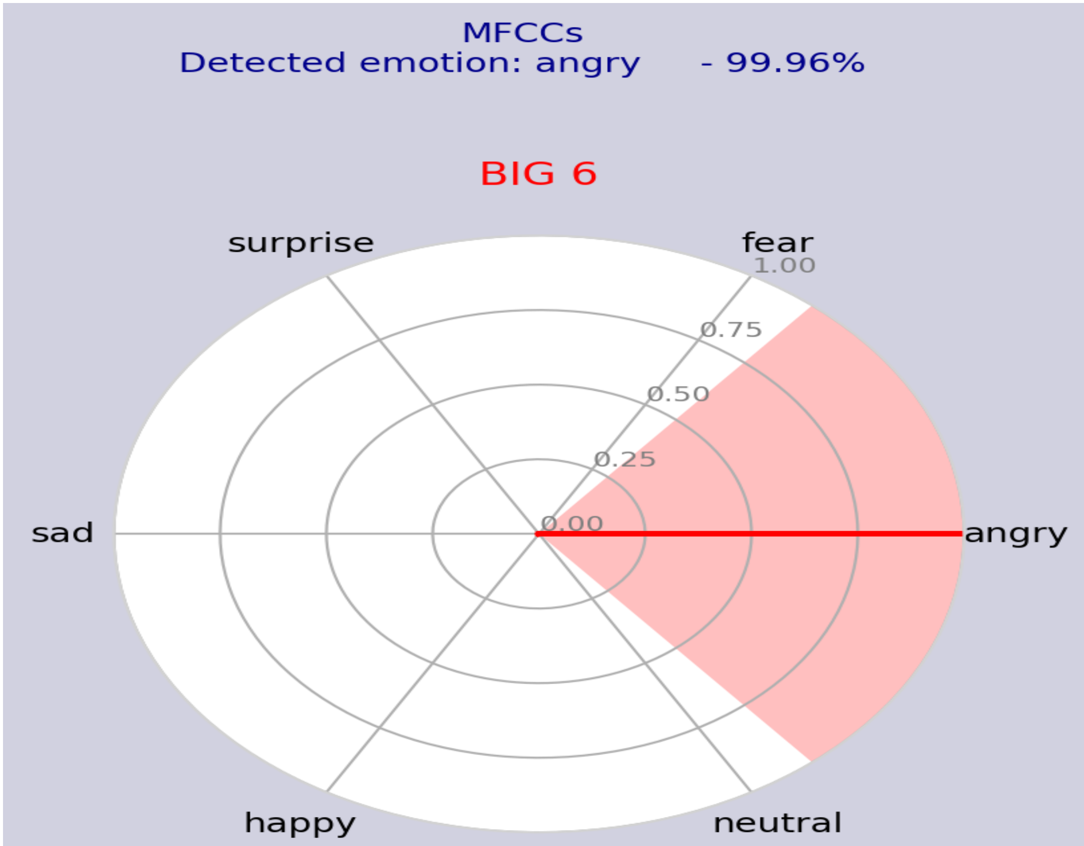


Fig 7.5 MFCC's predicted emotion

MFCC'S PREDICTED GENDER:

Fig 7.6 MFCC's predicted gender

REFERENCES:

1. Ayadi, M. E., Kamel, M. S., & Karray, F. “Survey on speech emotion recognition: features, classification schemes, and databases”. Pattern Recognition, 44, 572–587 (2011)

2. H. Altun *et al.* Boosting selection of speech related features to improve performance of multi-class SVMs in emotion detection(2009)

3. E.M. Albornoz *et al.* Spoken emotion recognition using hierarchical classifiers(Jul. 2011)

4. S. Emerich, E. Lupu, A. Apatean, “Emotions Recognitions by Speech and Facial Expressions Analysis”, 17th European Signal Processing Conference, (2009).

5. Idris I., Salam M.S. “Improved Speech Emotion Classification from Spectral Coefficient Optimization”.Lecture Notes in Electrical Engineering, vol 387. Springer,(2016).

6. A. B Ingale and D. S. Chaudhari, “Speech Emotion Recognition”, International Journal of Soft Computing and Engineering (IJSCE), March (2012), pp. 235-238.

7. A. Joshi, "Speech Emotion Recognition Using Combined Features of HMM & SVM Algorithm," International Journal of Advanced Research in Computer Science and Software Engineering, pp. 387-392, (2013).
8. R.B. Lanjewar *et al.* Implementation and comparison of speech emotion recognition system using gaussian mixture model (GMM) and K- nearest neighbor (K-NN) techniques (2015)
9. P. Laukka *et al.* Expression of affect in spontaneous speech: acoustic correlates and automatic detection of irritation and resignation(Jan. 2011)
10. A. Mohanta and U. Sharma, "Bengali speech emotion recognition," 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, (2016)
11. Pao TL., Chen YT., Yeh JH., Cheng YM., Chien C.S. “Feature Combination for Better Differentiating Anger from Neutral in Mandarin Emotional Speech”, LNCS: Vol. 4738 Berlin:Springer (2007).
12. K.R. Scherer *et al.* Comparing the acoustic expression of emotion in the speaking and the singing voice(Jan. 2015)
13. A.S.Utane and S.L. Nalbalwar, “Emotion Recognition through Speech”, International Journal of Applied Information Systems (IJAIS), (2013), pp. 5.8.

14. A.P. Wanare, S.N. Dandare, "Human Emotion Recognition From Speech", Int. Journal of Engineering Research and Applications, vol. 4, no. 7, pp. 74-78, July(2014).

15. Yashpalsing Chavhan, M. L. Dhore, Pallavi Yesaware, "Speech Emotion Recognition Using Support Vector Machine", International Journal of Computer Applications, vol. 1, pp.6-9,February(2010).