Speech Emotion Recognition Using Librosa

[1]Kopparapu Nikhil, [2]Kattubadi Drutesh, [3]Abhishek Kumar Reddy Manchuri,
[4]Bikki Hari Kiran, [5]Vakil Sai Teja, [6]Manjula. R

[1,2,3,4,5]Student, [6]Assistant Professor (Guide)
Electronics and Communication Department,
GITAM School of Technology, Bangalore.

Abstract: **Speech emotion recognition (SER) has numerous uses in industries like psychology, entertainment, and healthcare and is a critical component of human-computer interaction. Deep learning techniques have advanced recently, and SER has drawn a lot of interest from the research community. Use of the librosa Python package for music and audio analysis, which offers a number of functions for feature extraction from voice signals, is one such method. In this study, we explain the key features and techniques for feature extraction and classification, and we examine the state-of-the-art methodologies for SER utilising librosa. We also point out the difficulties and restrictions associated with using librosa for SER and offer some potential paths for further study in this area. Overall, this paper sheds light on librosa's potential.**

Keywords: Speech Emotion Recognition, Librosa, Kaggle, Spectrogram, Mel-Spectrogram, RAVDEES dataset, MLP.

INTRODUCTION:

Speech emotion recognition (SER) has a wide range of applications in industries like psychology, entertainment, and healthcare and is a key study subject in the realm of human-computer interaction. Automatically identifying a speaker's emotional state might provide important details about that person's mental and physical health as well as their objectives and motivations.
Deep learning techniques have advanced recently, and SER has drawn a lot of interest from the research community. Using machine learning algorithms to extract features from speech signals and categorise them into emotional states like happy, sorrow, anger, and fear is one of the most well-liked methods for SER[1].
A Python audio and music analysis library called Librosa offers a number of tools for extracting features from spoken signals. For tasks like speech recognition, speaker identification, and music genre categorization, it is frequently employed in the research community. A variety of features, such as spectral features, pitch features, and temporal features, can be efficiently and conveniently extracted from speech signals using Librosa.
The capability of librosa to handle complicated speech signals with various sources of variation is one of the main benefits of utilising it for SER[2]. Speech signals, for instance, may differ depending on the gender, age, and accent of the speaker as well as the presence of background noise and speech problems. For addressing these causes of variation, Librosa offers a complete and adaptable collection of tools, enabling researchers to create SER systems that are more precise and dependable.
In this study, we explain the key features and techniques for feature extraction and classification, and we examine the state-of-the-art methodologies for SER utilising librosa. We also point out the difficulties and restrictions associated with using librosa for SER and offer some potential paths for further study in this area. Our goal is to give a thorough analysis of librosa's potential for SER and its significance for creating accurate and effective SER systems.

Libraries Used:

Librosa Library: Librosa is a Python library for analysing audio and music. It provides many tools for processing and analysing audio signals, including the capacity to load audio files, glean information from audio signals, and see spectrograms and waveforms[3]. Speech processing, sound event recognition, and music information retrieval are some typical use cases for Librosa.
Pandas Library: Librosa is a Python library for analysing audio and music. It provides many tools for processing and analysing audio signals, including the capacity to load audio files, glean information from audio signals, and see spectrograms and waveforms. Librosa's typical use cases include speech processing, sound event detection, and music information retrieval. Pandas can organise disorganised data sets, making them readable and useful.

Visual Representation Techniques:

Spectrogram: A spectrogram is a visual representation of frequencies plotted against time that shows the signal's strength at a particular instant. In essence, a spectrogram is a visual representation of sound. Sound signals are converted into spectrograms using Fourier Transforms. A Fourier Transform decomposes the signal into its constituent frequencies and displays the magnitude of each frequency.

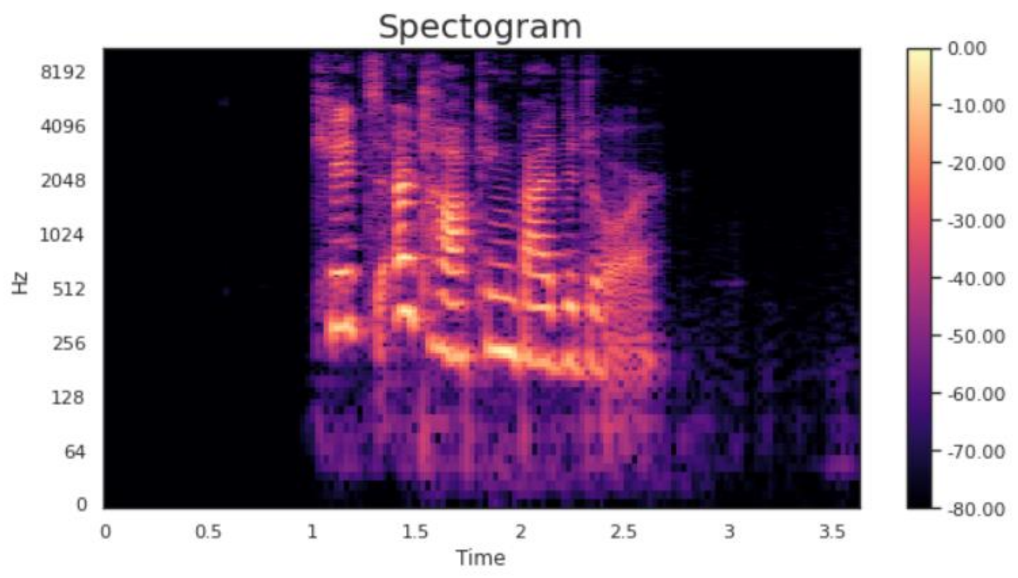


Fig.1: Plot of Spectrogram

Mel-Spectrogram: Usable features must be retrieved in order to detect audio, which often has complex properties[4]. Each audio sample has a high sampling rate, and the Mel-spectrogram is one of the efficient methods for audio processing.

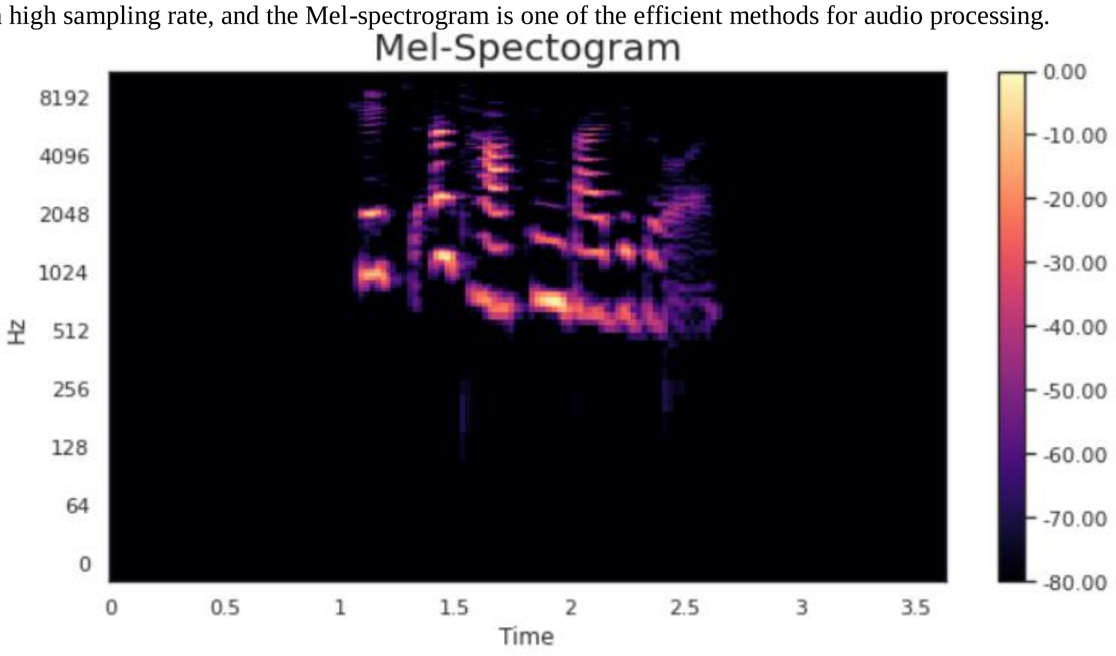
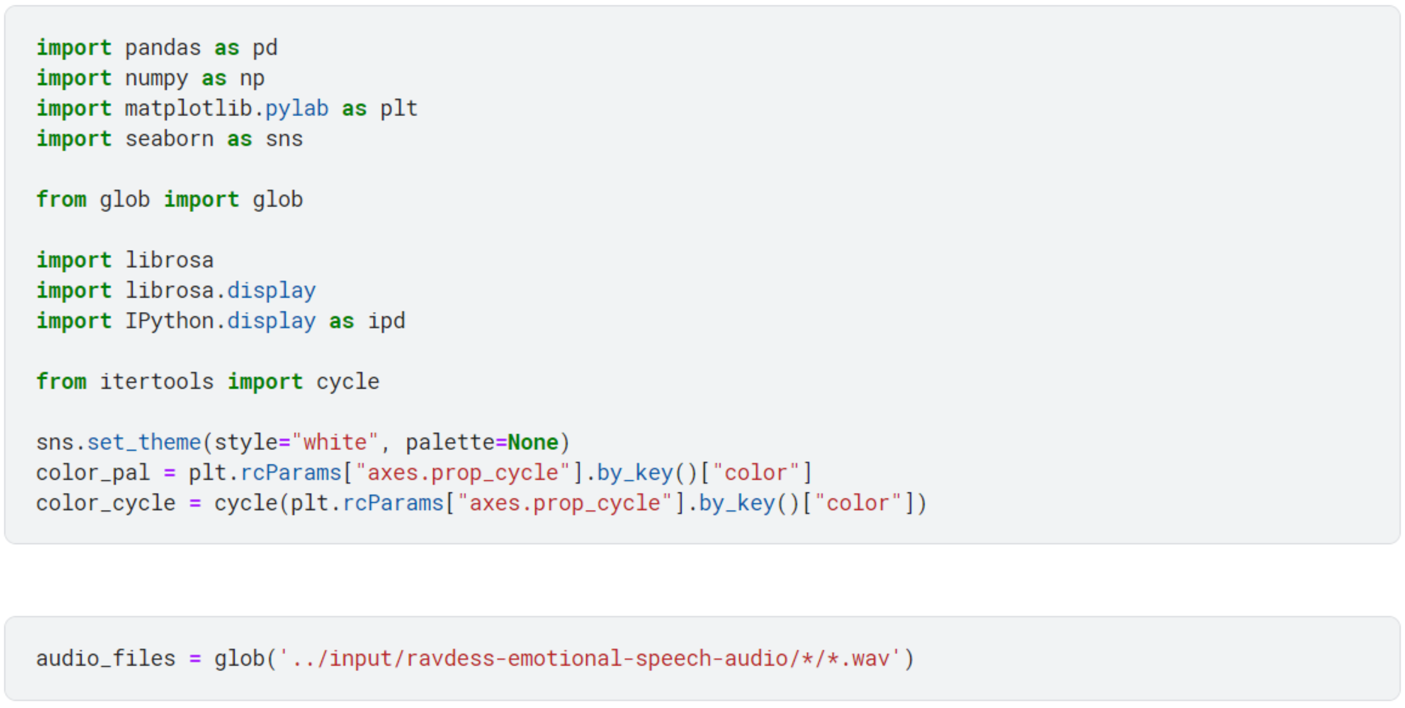


Fig.2: Plot of Mel-Spectrogram

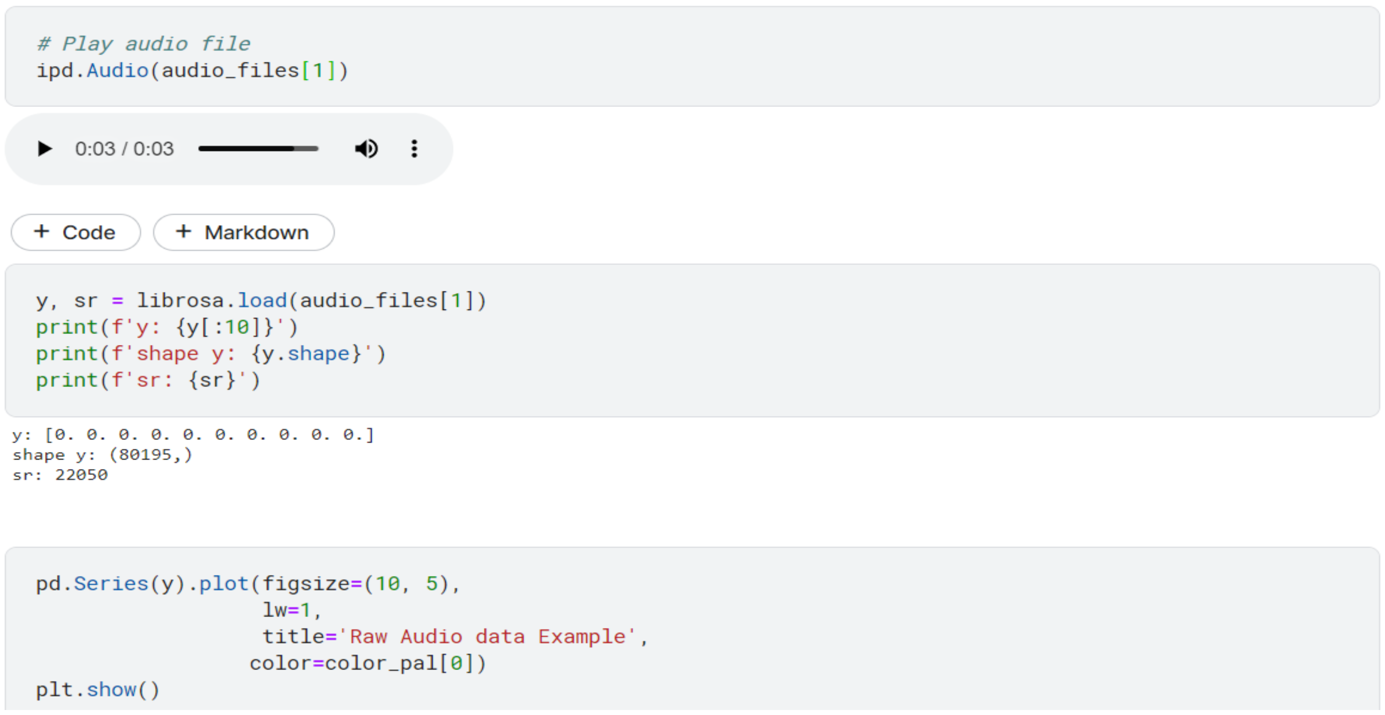
Librosa Working:



```
import pandas as pd
import numpy as np
import matplotlib.pylab as plt
import seaborn as sns

from glob import glob

import librosa
import librosa.display
import IPython.display as ipd

from itertools import cycle

sns.set_theme(style="white", palette=None)
color_pal = plt.rcParams["axes.prop_cycle"].by_key()["color"]
color_cycle = cycle(plt.rcParams["axes.prop_cycle"].by_key()["color"])
```

```
audio_files = glob('../input/ravdess-emotional-speech-audio/*/*.wav')
```

Fig 3: Import all libraries



```
# Play audio file
ipd.Audio(audio_files[1])
```

```
y, sr = librosa.load(audio_files[1])
print(f'y: {y[:10]}')
print(f'shape y: {y.shape}')
print(f'sr: {sr}')
```

```
y: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
shape y: (80195,)
sr: 22050
```

```
pd.Series(y).plot(figsize=(10, 5),
                  lw=1,
                  title='Raw Audio data Example',
                 color=color_pal[0])
plt.show()
```

Fig 4: Load audio file

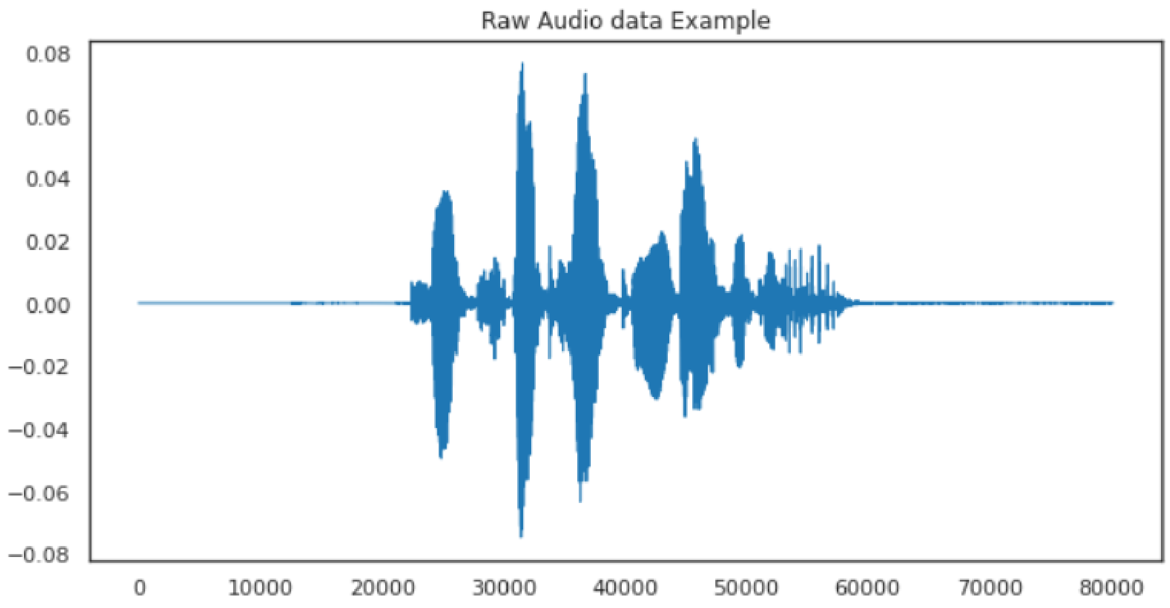


Fig 5: Plot of Raw Audio data

```
# Trimming leading/lagging silence
y_trimmed, _ = librosa.effects.trim(y, top_db=20)
pd.Series(y_trimmed).plot(figsize=(10, 5),
                  lw=1,
                  title='Raw Audio Trimmed Example',
                 color=color_pal[1])
plt.show()
```

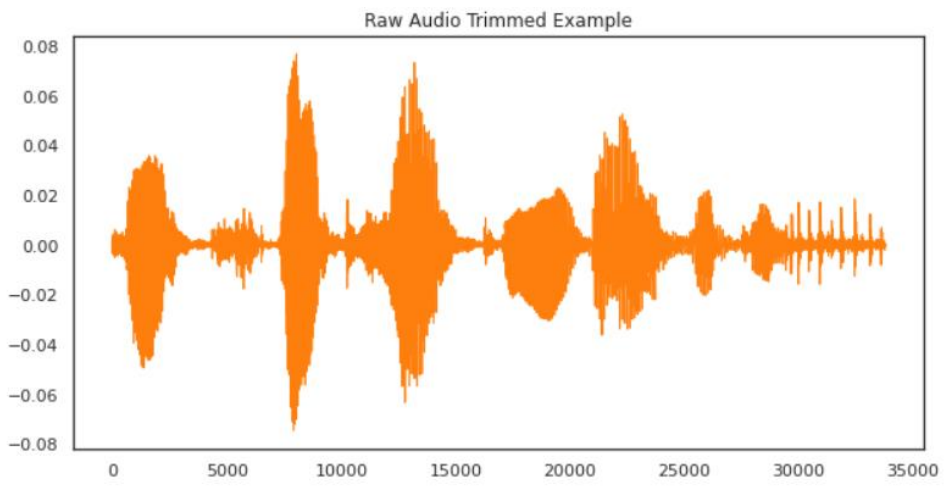


Fig 6: Plot of Raw Trimmed Audio

```
pd.Series(y[30000:30500]).plot(figsize=(10, 5),
                  lw=1,
                  title='Raw Audio data Zoomed In Example',
                 color=color_pal[3])
plt.show()
```

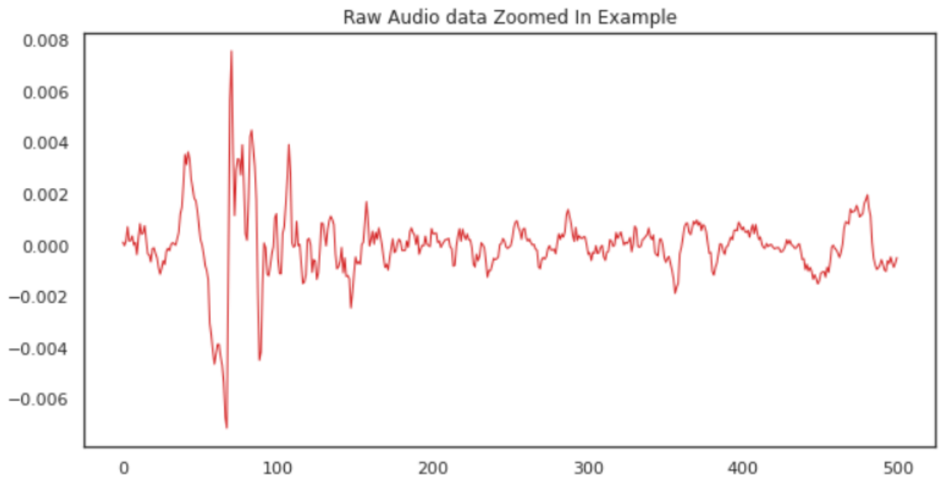


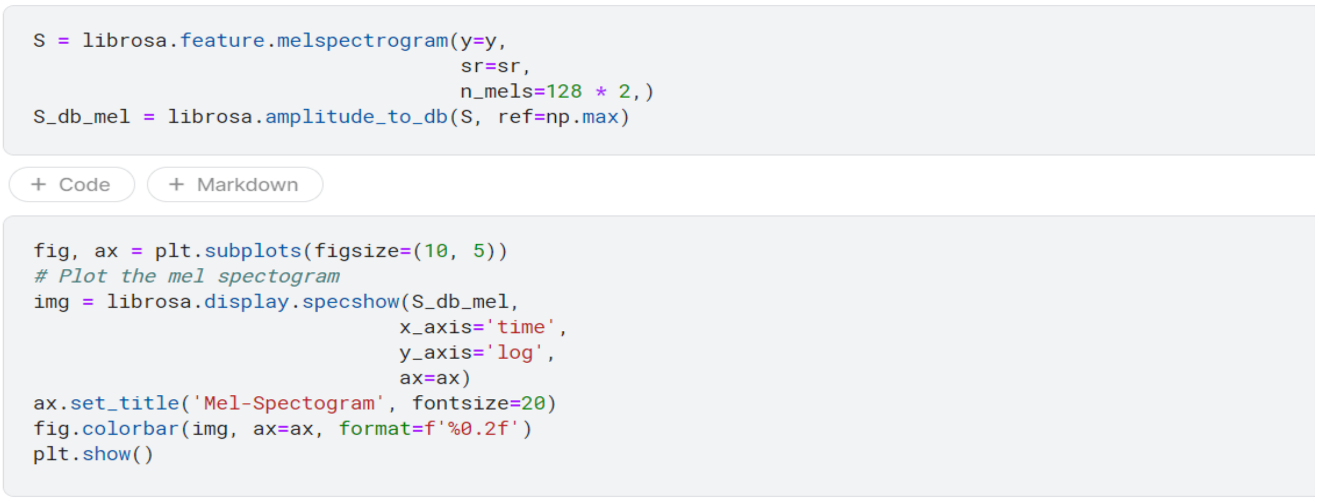
Fig 7: Plot of Zoomed in Audio data

```
D = librosa.stft(y)
S_db = librosa.amplitude_to_db(np.abs(D), ref=np.max)
S_db.shape
```

```
(1025, 157)
```

```
# Plot the transformed audio data
fig, ax = plt.subplots(figsize=(10, 5))
img = librosa.display.specshow(S_db,
                               x_axis='time',
                               y_axis='log',
                               ax=ax)
ax.set_title('Spectogram', fontsize=20)
fig.colorbar(img, ax=ax, format=f'%0.2f')
plt.show()
```

Fig 8: Amplitude to Decibles Transformation and plots spectrogram


```
S = librosa.feature.melspectrogram(y=y,
                                   sr=sr,
                                   n_mels=128 * 2,)
S_db_mel = librosa.amplitude_to_db(S, ref=np.max)
```

```
fig, ax = plt.subplots(figsize=(10, 5))
# Plot the mel spectogram
img = librosa.display.specshow(S_db_mel,
                              x_axis='time',
                              y_axis='log',
                              ax=ax)
ax.set_title('Mel-Spectogram', fontsize=20)
fig.colorbar(img, ax=ax, format=f'%0.2f')
plt.show()
```

Fig 9: Melodian Spectrogram

Accuracy and Emotion Prediction using Google Colab:

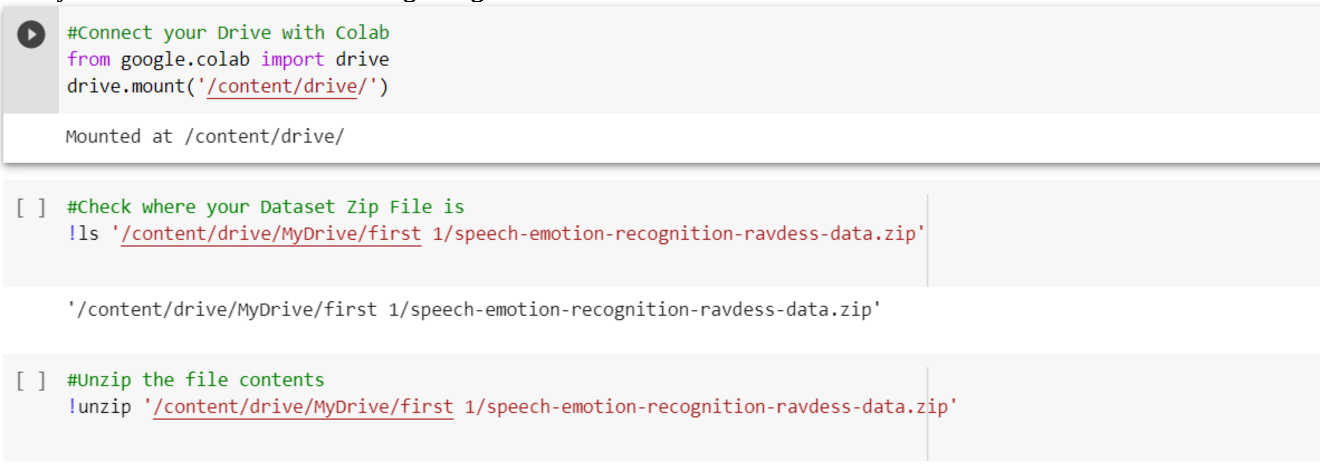

```
#Connect your Drive with Colab
from google.colab import drive
drive.mount('/content/drive/')
```
```
Mounted at /content/drive/
```
```
[ ] #Check where your Dataset Zip File is
    !ls '/content/drive/MyDrive/first 1/speech-emotion-recognition-ravdess-data.zip'
```
```
'/content/drive/MyDrive/first 1/speech-emotion-recognition-ravdess-data.zip'
```
```
[ ] #Unzip the file contents
    !unzip '/content/drive/MyDrive/first 1/speech-emotion-recognition-ravdess-data.zip'
```

Fig 10: Import Dataset

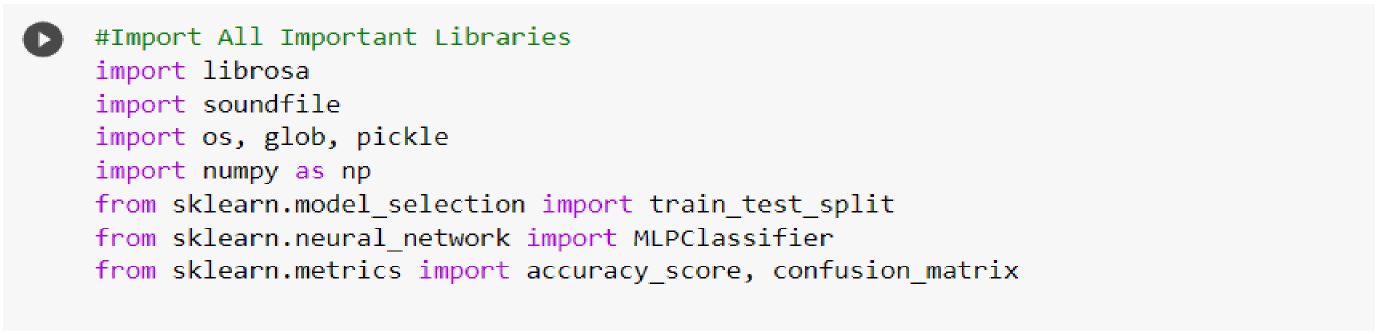

```
#Import All Important Libraries
import librosa
import soundfile
import os, glob, pickle
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
```

Fig 11: Import Libraries

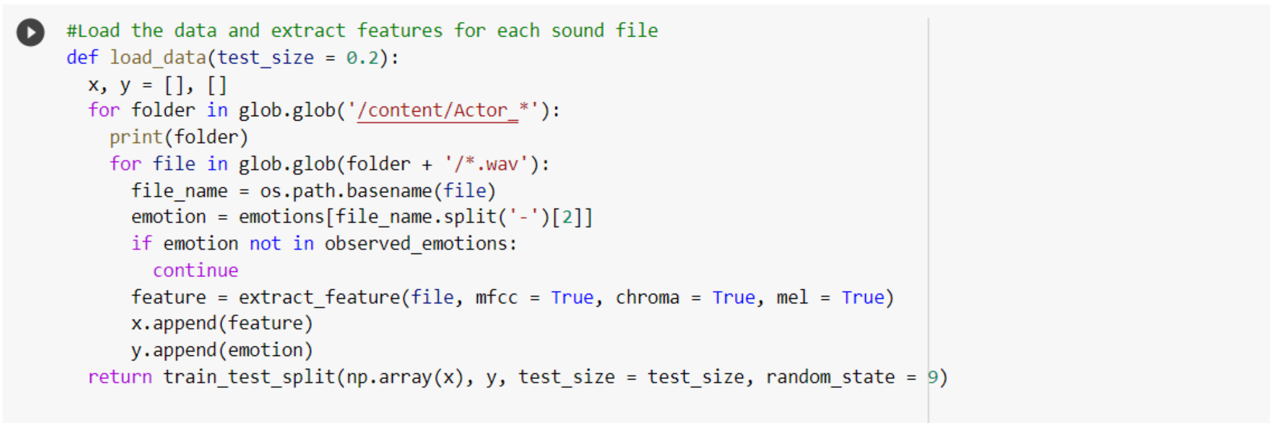
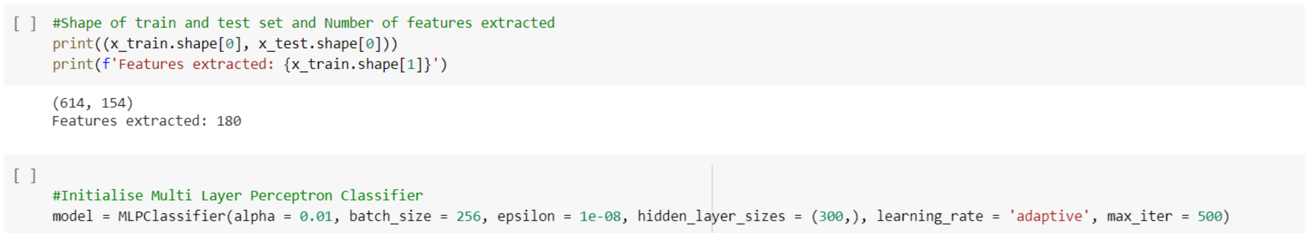

```
#Load the data and extract features for each sound file
def load_data(test_size = 0.2):
  x, y = [], []
  for folder in glob.glob('/content/Actor_*'):
    print(folder)
    for file in glob.glob(folder + '/*.wav'):
      file_name = os.path.basename(file)
      emotion = emotions[file_name.split('-')[2]]
      if emotion not in observed_emotions:
        continue
      feature = extract_feature(file, mfcc = True, chroma = True, mel = True)
      x.append(feature)
      y.append(emotion)
  return train_test_split(np.array(x), y, test_size = test_size, random_state = 9)
```

Fig 12: Extracting the features



```
[ ] #Shape of train and test set and Number of features extracted
    print((x_train.shape[0], x_test.shape[0]))
    print(f'Features extracted: {x_train.shape[1]}')

    (614, 154)
    Features extracted: 180

[ ]
    #Initialise 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)
```

Fig 13: MLP classification

METHODOLOGY:

The main objective of SER is to enhance the human-machine interface. To utilise the RAVDESS dataset and the librosa libraries to construct a model for voice emotion recognition.

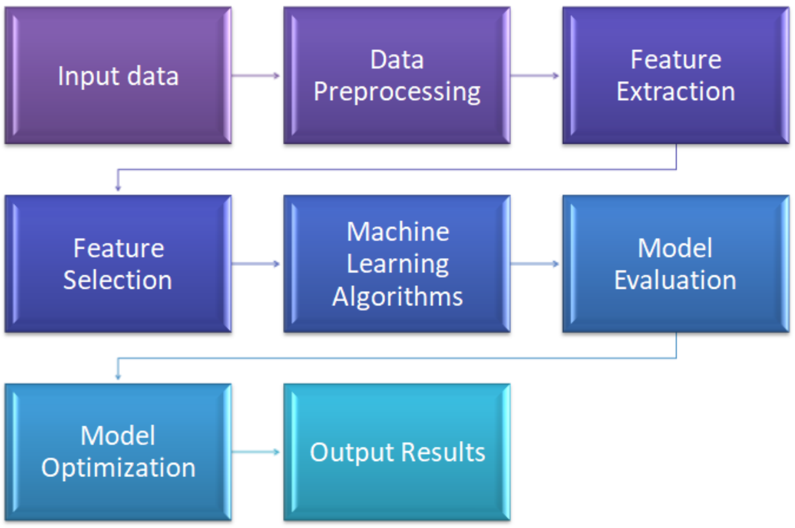


Fig.14: Working Block Diagram

WORKING MODEL:

To begin with, there are numerous platforms available for running and execute the Python codes. Some platforms include:

1. Google Colab
2. Kaggle
3. Pycharm
4. Jupyter Notebook

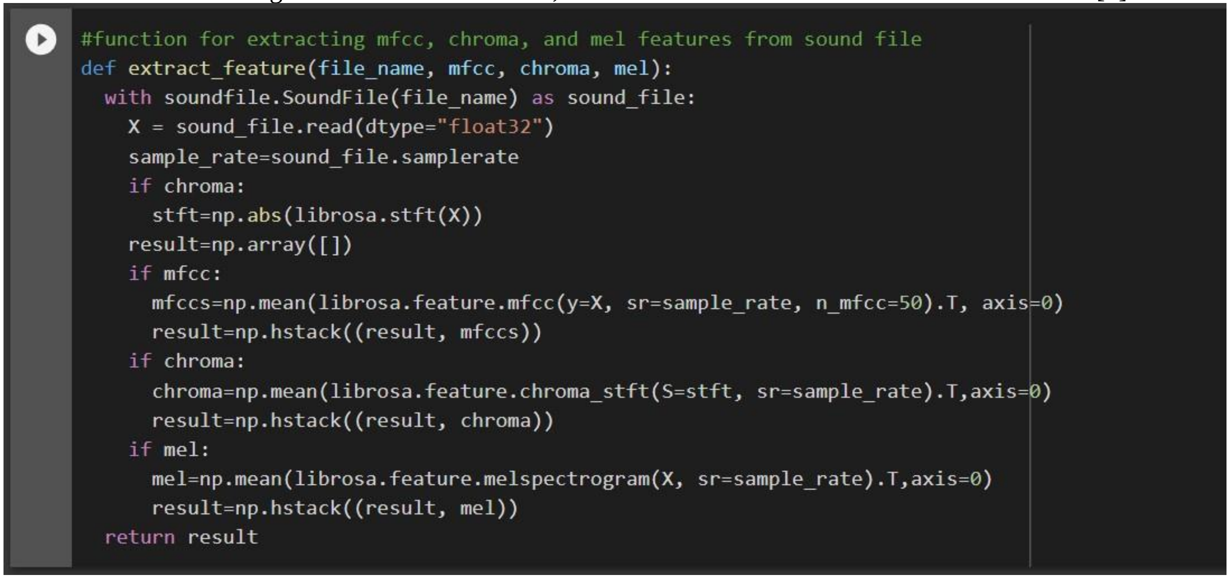
In this paper mainly our attention is on Kaggle. Kaggle Notebooks is a cloud computing platform that facilitates collaborative and repeatable analysis. You have access to cutting-edge data analysis and machine learning programmes that are already installed and compatible. Also, the RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song) database contains 7,356 files (total size: 24.8 GB). The collection contains recordings of 24 professional actors, 12 men and 12 women, voicing two lexically comparable sentences in a neutral North American accent. Speech and music may both express a variety of emotions, such as joy, sorrow, rage, fear, surprise, and contempt. For each expression, one neutral expression and two emotional intensity levels (strong and normal) are produced.

Once the Ravdees dataset has been downloaded on your computer, open Kaggle Notebook and import the dataset. Import the necessary packages, including librosa, pandas, numpy, and matplotlib, among others. run the cell first, then use glob to extract the audio files. Take one of those huge audio files, and with the aid of IPPD, we can display and listen to the audio file. Now, extract some information from that audio file, such as the sample rate and shape.

The graph can then be modified as needed, for example, by altering the colour or the scale, depending on the original audio data. Now plot a second one that has been edited; in this instance, the audio file was edited to remove any unnecessary signals. Plot a graph for the audio file that has been zoomed in to show what it looks like at a specific point. Plot the spectrogram signal, which is used to depict frequencies visually against time to indicate the signal strength at a specific moment. Now that the mel spectrogram has been drawn, it is required to extract useful features in order to identify the audio. And final one is accuracy checking we achieved accuracy.

IMPLEMENTATIONS:

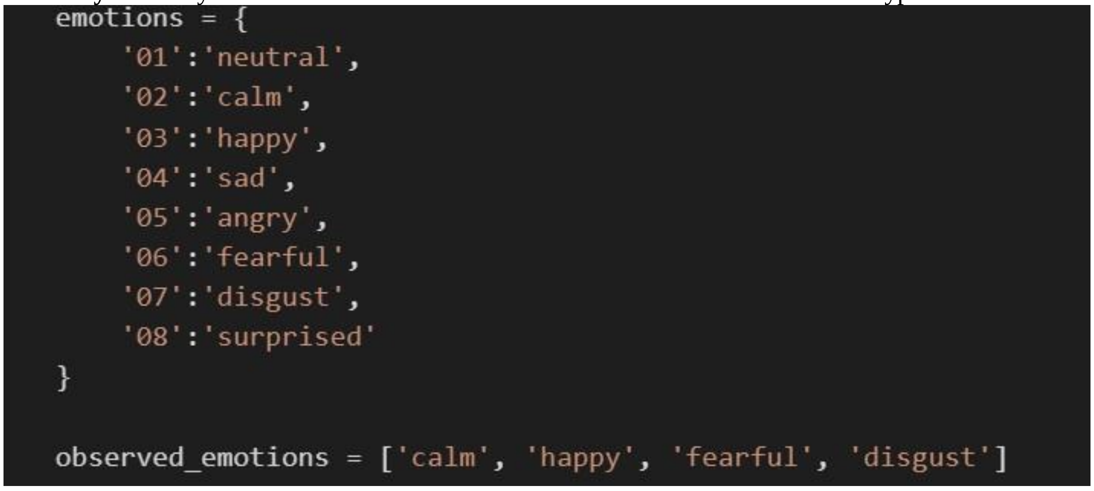
1.Feature Extraction: Extracting the all features like mfcc, mel and chroma that will store in a result variable[5].



```python
#function for extracting mfcc, chroma, and mel features from 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=50).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
```

Fig 15. Extracting features

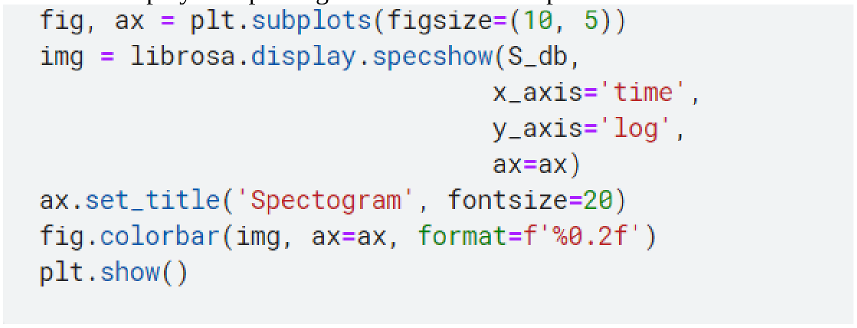
2.Emotions Dictionary: Classify the all emotions which are used to declare the result as which type of emotion.



```python
emotions = {
      '01':'neutral',
      '02':'calm',
      '03':'happy',
      '04':'sad',
      '05':'angry',
      '06':'fearful',
      '07':'disgust',
      '08':'surprised'
}

observed_emotions = ['calm', 'happy', 'fearful', 'disgust']
```

Fig16.Emotions

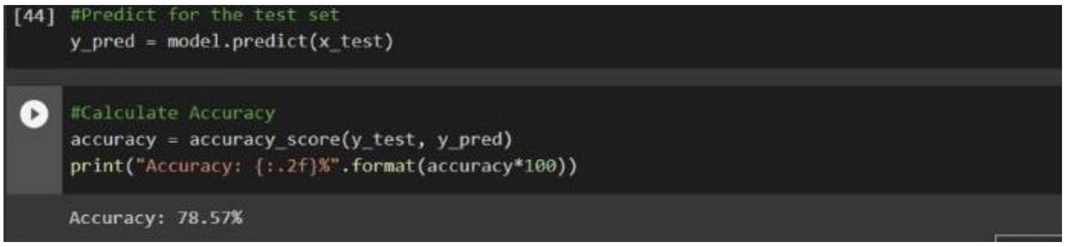
3.Visualization: This function will display the spectrogram of an audio sample.



```python
fig, ax = plt.subplots(figsize=(10, 5))
img = librosa.display.specshow(S_db,
                              x_axis='time',
                              y_axis='log',
                              ax=ax)
ax.set_title('Spectogram', fontsize=20)
fig.colorbar(img, ax=ax, format=f'%0.2f')
plt.show()
```

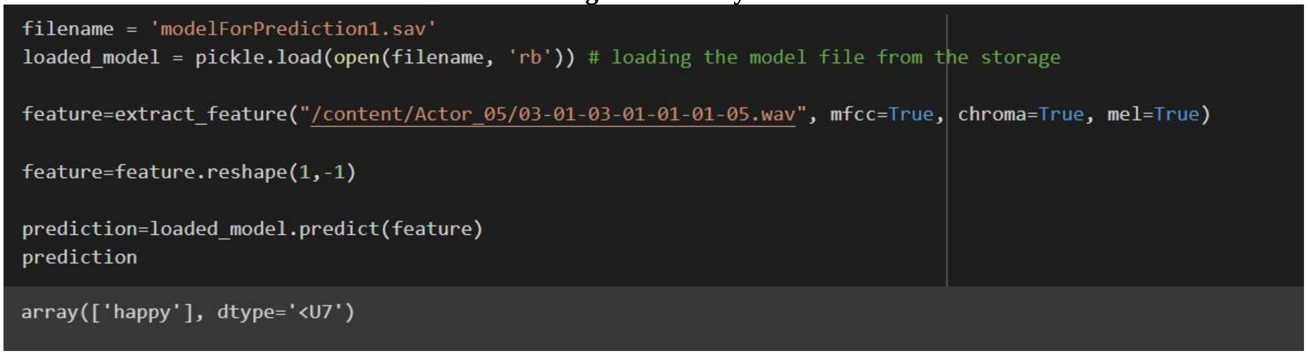
Fig17. Audio Visualization

RESULTS: In the below fig9 it will shows the accuracy of an emotion recognition. And in fig10 it will display the final result that is what type of emotion it is.



```
[44] #Predict for the test set
     y_pred = model.predict(x_test)

     #Calculate Accuracy
     accuracy = accuracy_score(y_test, y_pred)
     print("Accuracy: {:.2f}%".format(accuracy*100))

     Accuracy: 78.57%
```

Fig18. Accuracy



```
filename = 'modelForPrediction1.sav'
loaded_model = pickle.load(open(filename, 'rb')) # loading the model file from the storage

feature=extract_feature("/content/Actor_05/03-01-03-01-01-01-05.wav", mfcc=True, chroma=True, mel=True)

feature=feature.reshape(1,-1)

prediction=loaded_model.predict(feature)
prediction

array(['happy'], dtype='<U7')
```

Fig.19 Emotion prediction

CHALLENGES:

Many people can display their emotions in various ways, and they may react differently emotionally to the same stimulus. It is challenging because of the variety in emotional expression. Noise in speech signals is a common problem that can make emotion identification models less accurate. Moreover, the quality of the speech stream may be impacted by signal distortion brought on by elements like reverberation and compression. For recognising voice emotions, a variety of machine learning techniques and deep learning architectures can be applied. Finding the ideal model for a given dataset can be difficult.

ACKNOWLEGEMENT:

We are very much thankful for our faculty Manjula.R for their valuable guidance and suggestions to improve the quality of manuscript.

CONCLUSION AND FUTURE SCOPE:

In conclusion, speech emotion reognition using Librosa is a promising method for examining the emotional content of audio data. A selection of audio processing and feature extraction tools from Librosa make it possible to extract pertinent information from audio inputs.
To increase the reliability and accuracy of speech emotion recognition models utilising Librosa, more research is therefore required. This can involve examining alternate feature extraction techniques, creating more complex machine learning algorithms, and examining the effects of various speech and language traits on the effectiveness of emotion recognition. Overall, Librosa's voice emotion detection technology has a lot of promise for use in a variety of industries, including psychology, medicine, and human-computer interaction.

REFERENCES:

1. Z. Tang, C. Jin, X. Yang, and W. Liu, "Speech Emotion Recognition Based on Deep Convolutional Neural Network and Data Augmentation with Librosa," in 2020 IEEE 20th International Conference on Communication Technology (ICCT), 2020, pp. 510-514.
2. F. Liu, X. Zhang, J. Liu, and Q. Zhu, "Speech Emotion Recognition Based on Improved MFCC and SVM Using Librosa," in 2020 6th International Conference on Information Management (ICIM), 2020, pp. 305-311.
3. S. Saha, S. Saha, S. Roy, and S. Bhaumik, "Speech Emotion Recognition Using Librosa and SVM Classifier," in 2021 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), 2021, pp. 1-5.
4. V. P. N, P. T. P, and K. K. K, "Speech Emotion Recognition using Convolutional Neural Networks with Librosa Feature Extraction," in 2021 IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), 2021, pp. 1-6.
5. N. Kaur and H. Gupta, "Speech Emotion Recognition using Deep Learning with Librosa Feature Extraction," in 2021 International Conference on Artificial Intelligence and Sustainable Technologies (ICAIST), 2021, pp. 251-256.