SENTIMENT ANALYSIS TO DETECT SUBSTANCE DEPENDENCY USING ARTIFICIAL INTELLIGENCE

A PROJECT PHASE I REPORT

SUBMITTED BY

SIMRAN SHAIKH (BE22152)

AKSHAY SABLE (BE22104)

KUNAL KANASE (BE22169)

UNDER THE GUIDANCE OF

Dr. S.M.M.NAIDU

BE (ELECTRONICS AND TELECOMMUNICATION)

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION

HOPE FOUNDATION'S

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY,

HINJAWADI, PUNE(MH)-411057

SAVITRIBAI PHULE PUNE UNIVERSITY

A.Y. 2022-23

CERTIFICATE

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION

HOPE FOUNDATION'S

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY,

HINJAWADI, PUNE-411057

This is to certify that

SIMRAN SHAIKH

KUNAL KANASE

AKSHAY SABLE

Class: BE(E&TC) have partially completed the Project titled, '**SENTIMENT ANALYSIS TO DETECT SUBSTANCE DEPENDENCY USING ARTIFICIAL INTELLIGENCE**' under my supervision as a part of Semester I of Final Year of Bachelor of Engineering in **Electronics and Telecommunication (A.Y. 2022-2023)** of Savitribai Phule Pune University.

Dr. S.M.NAIDU — Project Guide

Prof. RISIL CHATHRALA — HOD(E&TC)

Principal

Place : Pune

Date :

External Examiner

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed. We take sole responsibility of the work presented by us in this report. We also declare that we will submit our completed project along with all necessary hardware and software to the department at the end of the 2nd semester.

Signature.....

STUDENT NAME .....

Signature.....

STUDENT NAME .....

Signature.....

STUDENT NAME .....

Place : Pune

Date :

Abstract

Mental health issue is prevalent in the current digital era. People often get dependent on external factors to cope with stress, anxiety, depression, loneliness, etc. This usually results in dependency on the substance. Socioeconomic conditions affect the quality of human life. People living in underprivileged communities have low standards and satisfaction with life due to poor socioeconomic status.

Lack of quality education and healthcare results in addiction problems in marginalized communities. It is difficult to treat patients with substance dependency if it is severe. The issue of substance dependence is a huge burden in the healthcare sector and on the economy. Rehabilitation centres run by governments, non-profit organizations, and for-profit organizations provide support to patients addicted to substances which include alcohol, tobacco, etc. Health professionals treat patients in rehabilitation with medical and psychological interventions. They monitor the admitted patients throughout their healing journey by performing quantitative analysis to see changes in behavioural and biological aspects.

The central objective of this project is to analyze the sentiments of the patients to detect levels of substance dependency using artificial intelligence techniques. This will provide quantitative insights to help healthcare professionals to monitor patients' health throughout their treatment duration. We are developing a system to detect the underlying emotions of patients in the recorded speech by analyzing the acoustic features of the audio data of recordings by implementing an Artificial Neural Network (ANN) method and comparing its results with the results of conventional methods.

The audio signals and their content are the input. These signals are analyzed to detect emotions. The signal pre-processing is done before extracting audio features that involve getting the dataset, importing libraries, importing the dataset, encoding categorical data, and splitting the dataset into training, testing, and feature scaling.

Feature extraction aims to reduce the number of features in a dataset by creating new features from the existing ones (and then discarding the original features). We utilize a spectrogram to visualize signals in a picture, which depicts time on the x-axis, frequency on the y-axis, and amplitude on the x-axis for more comprehensive information. It can also be in different hues, with the color density indicating the signal's strength. Finally, it provides a summary of the signal, describing how the signal's strength is divided among frequencies. The result from this analysis will be helpful to determine changes in patients' responses to the treatment through quantitative insights provided by audio signals.

Keywords:

MFCC, Spectrogram, Convolutional Neural Network, StandardScaler, RAVDESS

Contents

List of Figures

Chapter 1

Introduction

Advances in speech and text analysis have been enabling a variety of applications that identify emotional content using machine learning techniques. This project aims at using a technique that does not require the automatic speech recognition (ASR) method for the alignment of text and speech but instead recognizing the emotional state of a person is done using machine learning algorithms. More than 50% of the urban population lives in slums where substance dependency is severely affecting people from underprivileged communities. The central objective of this project is to analyze sentiments to detect substance dependency using artificial intelligence techniques. In this study, we attempt to detect underlying emotions in a recorded speech by analyzing the acoustic features of the audio data of recordings. For this project, we are implementing an Artificial Neural Network (ANN) method and comparing its results with the results of conventional methods. Feature extraction used Mel Frequency Cepstral Coefficient (MFCC), Mel spectrogram, and short-term Fourier transform (STFT) to save them as feature vectors by performing feature merge with MFCC coefficients. Emotions are expressed in various forms including human speech facial expression and body postures. The proposed model can be used in rehabilitation centers to detect the level of substance dependency through sentiment analysis. It can be helpful in qualitative analysis for medical professionals to determine patients' responses to the treatment in rehabilitation. Researchers are constantly working on the mode of speech

emotional recognition to enhance human-machine interactions. Applications of speech-based emotion recognition range from detecting the severity of calls in emergency call centers to automatic safety protocols to detect 'lazy' emotions. Speaker's emotions can be determined using a multitude of factors like the contents of the speech, physical attributes of the speaker like age and gender, and characteristics of the sound itself. This makes determining emotion difficult. Extensive research has been done in increasing the accuracy of the classification models.

1.1 Overview of the Monitoring System

The Speech Emotion Recognition (SER) process goes through five stages: input, preprocessing, feature extraction, classification, and output. The audio signals and their content are the input. These signals are analyzed to detect emotions. The signal preprocessing is done before extracting audio features. It involves getting the dataset, importing libraries, importing the dataset, encoding categorical data, and splitting the dataset into training, testing, and feature scaling. Feature extraction aims to reduce the number of features in a dataset by creating new features from the existing ones (and then discarding the original features).

These new reduced sets of features should then be able to summarize most of the information contained in the original set of features. In classification, speech emotions are classified into five classes, calm, happy, neutral, fearful, and disgusted. The classification has been done using a StandardScaler, which is a class of Artificial Neural Networks (ANN). Lastly, the accuracy of the model was calculated.

Chapter 2

Literature Survey

Badshah et al. (2017) directly fed spectrograms to the classification model. A spectrogram is a 2D time-frequency representation of an audio signal. The intensity of the audio signal at any given point can be determined by the amplitude as well as the colour at that point. This spectrogram is obtained only by applying a fast Fourier transform (FFT) to speech signals Conventional Neural Network (CNN) was used for feature extraction and the softmax layer for the classification of emotions. Extraction of distinct features from spectrograms is harder than in images, Berlin speech emotion dataset was used. 7 emotions: anger, boredom, disgust, fear, sadness, happiness, and neutral were classified. Multiple spectrograms were created per audio file. Prediction after analyzing individual spectrograms was used to update the belief values of that audio file for all emotions Two experiments were performed: traditional learning and transfer learning. Traditional learning performed better than transfer learning [1].

Hadhani Aouani proposed Feature extraction using Mel-frequency cepstral coefficients (MFCC), harmonics-to-noise ratio (HNR), Teager energy operator (TEO) zero crossing rate (ZCR) with auto-encoding, Classification using support vector machine (SVM), Recognition rates on the test corpus obtained using SVM with radial basis function (RBF) kernel: anger: 75% disgust: 88% fear: 42% sad: 71% happy: 72% and surprise: 50% Application of auto-encoder dimension reduction improves the identifi-

cation rate.

A separate database was created by Schaller et al (2003) The corpus was collected in an acoustically isolated room, and the recordings were in English and German language Two methods have been used to collect speech samples of f people, resulting in a total of $250 sample the larger test set of 4 speakers consists of actual emotions. Another method collected spontaneous emotions based on the ability to compare recognition results to that of acted emotions. The probability distribution function of each feature is approximated by the means of up to 4 mixtures of Gaussian distribution. Each Emotion is modelled by one single state Hidden Markov model (HMM) and Gaussian Mixture Modelling(GMM), and the model with the maximum likelihood will be considered as a recognized emotion. The Continuous HMMs(CHMMs) are formed using Bauen welsh estimation 14). Four Gaussian mixtures have been used to estimate the probability density function. One model was chosen for each emotional state, resulting in 7 overall models, and the maximum likelihood model was chosen for the assumed emotion. Anger, fear, sensory pleasure, sadness, excitement, pleasure, amusement, satisfaction, contentment, pride, shame, guilt, disgust, contempt, embarrassment and relief are basic emotions Eckman (1992).

Selecting a suitable database is crucial in determining the performance of a speech recognition system Ayadi et al (2011). Important factors to be considered while selecting a database include the number of samples, type of database (natural/simulated), number of people or actors, language, etc [8]. Girija Deshmukh proposed the methods of preprocessing, MFCC for feature extraction, and classification using SVM. The accuracy of the emotion classes was nos follows anger- 60% happiness 61% and sadness-69% The classification accuracy for all three emotions was increased by 20% by using three features as against using two features. Classification of emotions for regional Indian languages namely Hindi and Marathi was implemented by using The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) database as their audio database which consisted of male and female audio samples to test for 3 emotions: anger, happiness, and sadness. The databases were separated into training and

testing sets. Firstly, audio signals are preprocessed.

The continuous time and signal were converted to discrete-time goals to better analyse them by using sampling. The parts of there were one variation with lime we removed as they didn't contribute towards the determination of mentioned Therefore high pass filter is applied to emphasise the high-frequency components which represent the rapidly changing signal.
Parts of the audio which consist of absolute silence were removed as salary doesn't tribute to emotional determination. To properly analyze the signal, it being in stationary form is preferable. To achieve this, the input signal is divided into all cor of a specific time interval. Generally, for speech processing, it was observed that a duration of 20 or 30 ms is ideal. As digital signals are large and infinite, they cannot be analyzed in their entirely at the same time. Therefore, windowing is performed using techniques like rectangular, hamming, Blackman, etc.

After the dataset is pre-processed, feature extraction is performed. Firstly, energy is extracted from each frame. This energy value is then used) for obtaining a short-term energy plot of the signal which indicates the high-frequency frames using larger poss After that, MFCC feature vector extraction is performed for each frame. MFCC resents the short-term power spectrum envelope. This envelope represents the shape of the human vocal tract which determines various sound characteristics. MFCC is not extractable in the time domain. Hence, it is converted to the frequency domain using FFT Power Spectral density is calculated. Mel filter banks viz.

Trigeorgis et al.(2016) segmented raw waveforms into 6 sec long sequences. These 6-sec sequences were further divided into 150 smaller subsequences. ReCOLA date, which is a french dataset, was used. Raw waveforms (16kHz) were fed to the convolutional layer. The output of the convolutional layer was input to recurrent long short-term memory (LSTM). The convolution layers replace the requirement of hard-

engineering features which were used till now. The researched method performs significantly better in comparison to traditionally designed features on the ReCOLA database (11) Determination of emotion using speech is more difficult than using, say, facial expression. Emotions, like anger and happiness, are highly misclassified Ghosh et al. (2016)

This is because anger has a high fundamental FrequencyAlso, it depends on factors whether the date is in coal or simulated Differences in language and gender could also lend to misclassification of emotions. Some dates are developed in one actor, others are developed using days from call centres, TV and radio programs. Selecting proper features for speech-led emotion region is principal in- creating the clasification accuracy

Ajay Gupta proposed the methods of Random Decision Forest (RDF), SVM, CNN, and MFCC for Feature Extraction. The accuracy obtained was 88.21% with the ugliest and lowest classification accuracy for angry and fear clam respectively Exotics were classified into 6 emotion output classes Sutures were extracted for motion classification. After the initial classification by SVM additional copies of a are fitted on the same dataset but the weights of incorrectly classified samples are adjusted to make successive classifiers focus on more complex cases. For this gradient boosting was used. Random forest initialized multiple decision trees during training an outs men/mode of the class of individual trees. They prevent decision trees from user files ting the training data. An accuracy of 81.05 per cent was obtained for the Berlin database by using the Random Decision Forest classifier which was best suited and gave the highest accuracy

K.Wang et al. (2015) considered German Emotional Corpus (EMODB) Chinese Emotional Database (CASIA), and Chinese Elderly Enjotinal Speech Database (EESDB) EMO- DB comprises 10 sentences that cover seven classes of emotion from everyday communication, namely, anger, fear, happiness, sadness, disgust, boredom and neutral CASIA is composed of 9,000 wave files that represent different emotional

states of happiness, anger, random, surprise, fear and neutrality, with 4 actors. EESDB includes seven classes of emotions (including mutual). 11 Chinese Elderly males and females, over the age of 60 are the source of this data. The first 20 Harmonic Coefficients were extracted. The minimum, maximum, mean, median and standard deviation of Fourier parameters feature vector was calculated.

Mel-frequency cepstral coefficient (MFCC) features were extracted for Comparison Variability with Fourier parameters features MFCC features ally male minimum ki mn, medians and standard deviation. To eliminate speaker and recording while keeping the effectiveness of emotional discrimination.

Ghai et al. (2017) used a Berlin database which has 535 utterances from 1 speaker in the German language. The total length of 535 emotional utterances is 1457 sec and the average length is 277 sec Each file consists of 16-bit pulse code modulation (PCM) and mono channels. After the audio file is divided into frames, feature extraction takes place. The audio signal is sampled at 16000 Hz and the frame duration is selected as 0.025 Large part of the audio signal has a frequency of 8 kHz. Ideally 16kHz. A short part of audio signals remains stationary, thus the division into frames MFCC represents the logarithmic perception of stress and pitch. Mel scale gives a unit of pitch such that equal distances in pitch sound are equally distant to the listener. Energy represents the intensity of the speech. It is calculated by the sum of the square of the amplitude of each frame.

Chapter 3

Proposed Methodology

3.1 Problem Statement

Detecting Substance dependency by using machine learning methods and audio analysis techniques by corresponding emotions.

3.2 Objectives

The project objective is to create an SER system utilizing machine learning techniques and learn about data processing, feature extraction, and classification.
Add more feature extraction, and create a real dataset through the trained model classification into whether a person is addicted.

3.3 Proposed Methodology

For creating a model which can understand the emotion of subjects. Input is needed which can display the Emotion that can be detected from various channels such as electroencephalography (EEG) signals, acoustic, visual, text, and gestures. Speech is the most natural way of communication between humans and robots.
Therefore, speech emotion recognition (SER) is the research hotspot in natural human-

robot interaction (HRI). Nevertheless, effective SER is still a very challenging problem, partly due to cultural differences, various expression types, context, ambient noise, etc. Speech is the input, this input will be further analyzed and classified. Prepossessing data is an important step where the process of preparing the raw data and making it suitable according to the machine learning model. Extraction of features is a very important part of analyzing and finding relations between different things. As we already know that the data provided by audio cannot be understood by the models directly so we need to convert them into an understandable format for which feature extraction is used for better human interpretations.

The frequency scale is first converted into Mel Scale. Mel Scale:-It is much harder for humans to be able to differentiate between higher frequencies, and easier for lower frequencies. So, even though the distance between the two sets of sounds is the same, our perception of the distance is not. This is what makes the Mel Scale fundamental in Machine Learning applications to audio, as it mimics our perception of sound.

The transformation from the Hertz scale to the Mel Scale is the following:

$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right)$$

Figure 3.1: Conversion of Hertz scale to Mel Scale

The Short Term Fourier Transform(STFT) divides a longer signal into shorter segments of equal length and then computes the Fourier transform separately on each shorter segment. STFT is two types: continuous time STFT and discrete time STFT. Continuous time STFT:- Simply, in the continuous-time case, the function to be transformed is multiplied by a window function which is nonzero for only a short period of time.

$$\mathbf{STFT}\{x(t)\}(\tau,\omega) \equiv X(\tau,\omega) = \int_{-\infty}^{\infty} x(t)w(t-\tau)e^{-i\omega t}\,dt$$

Figure 3.2: Continuous-time STFT

Discontinuous time STFT:- In the discrete-time case, the data to be transformed could be broken up into chunks or frames (which usually overlap each other, to reduce artifacts at the boundary). Each chunk is Fourier transformed, and the complex result is added to a matrix, which records the magnitude and phase for each point in time and frequency. This can be expressed as:

$$\mathbf{STFT}\{x[n]\}(m,\omega) \equiv X(m,\omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-i\omega n}$$

Figure 3.3: Discontinuous-time STFT

3.4 Requirement analysis

3.4.1 Software Requirement

3.4.2 Modern Engineering Tools and Software Requirement

Python3

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL).

Python 3.0 was released in 2008. Although this version is supposed to be backward incompatibles, later on, many of its important features have been backported to be compatible with version 2.7. This tutorial gives enough understanding of the Python 3 version of the programming language.

Characteristics of Python

Following are important characteristics of Python Programming:

• It supports functional and structured programming methods as well as OOPS.

• It can be used as a scripting language or can be compiled to byte code for building large applications.

• It provides very high-level dynamic data types and supports dynamic type checking.

• It supports automatic garbage collection.

• It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Spyder

Spyder is a powerful scientific environment written in Python, for Python, and designed by scientists, engineers, and data analysts.

It features a unique combination of the advanced editing, analysis, debugging, and profiling functionality of a comprehensive development tool with the data exploration, interactive execution, deep inspection, and beautiful visualization capabilities of a scientific package.

3.4.3 Techniques

StandardScaler

Standardization of a dataset is a common requirement for many machine learning estimators: they might behave badly if the individual features do not more or less look like standard normally distributed data (e.g. Gaussian with 0 mean and unit variance).

For instance, many elements used in the objective function of a learning algorithm (such as the RBF kernel of Support Vector Machines or the L1 and L2 regularizers of linear models) assume that all features are centered around 0 and have variance in the same order. If a feature has a variance that is orders of magnitude larger than others,

it might dominate the objective function and make the estimator unable to learn from other features correctly as expected.

This scaler can also be applied to sparse CSR or CSC matrices by passing with _ mean=False to avoid breaking the sparsity structure of the data.

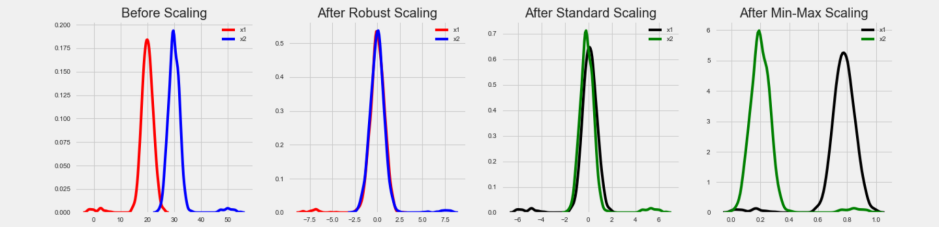


Figure 3.4: Example of StandardScaler

CNN

CNN or the convolutional neural network (CNN) is a class of deep learning neural networks. In short think of CNN as a machine learning algorithm that can take in an input image, assign importance (learn-able weights and biases) to various aspects/objects in the image, and be able to differentiate one from the other.

CNN works by extracting features from the images. Any CNN consists of the following:

The input layer which is a gray-scale image The Output layer which is a binary or multi-class labels Hidden layers consisting of convolution layers, ReLU (rectified linear unit) layers, the pooling layers, and a fully connected Neural Network.

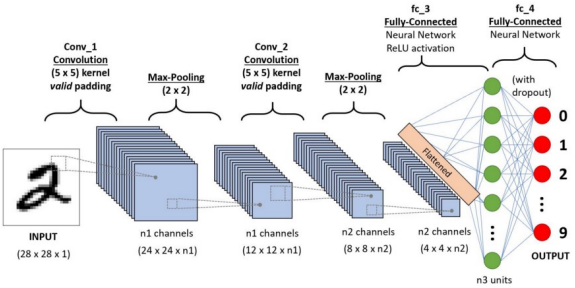


Figure 3.5: Architecture of Convolutional neural network(CNN)

3.4.4 Libraries Requirement

Librosa

Librosa is a Python tool that analyses music and audio. It contains the components required to construct music information retrieval systems. This library has been used by a lot of people for machine learning. Using advanced algorithmic techniques, BONeral approaches for visualizing sounds can be viewed.

Soundfile

SoundFile is a NumPy-based audio library that uses libindfile, CFFI, and NumPy as its foundation. SoundFile is a program that can read and write sound files. libsndfile, a free, cross-platform, open-source (LGPL) library for reading and writing many different sampled sound file formats that run on many platforms including Windows, OS X., and Unix, is used to read and write files. It's accessed by CFFI, which is a Python-to-C code foreign function interface. For CPython 2.6+, 3x, and PyPy 2.0+, CFFI is supported. NumPy arrays are used to represent audio data in SoundFile.

Scikit learn

Scikit-learn: (previously scikits learn, and also known as skirstu) is a fee Pytlum machine learning package. It inchales support sector machines, random, formis, gradient boosting, k-means, and DBSCAN, xoong other classification, grin, and cluster ing techniques, and is designed to work with the Python merical and wicatific besties NumPy and SciPy Seilot-learn is a financially supported NemFOCUS project Scikit-learn is mostly built in Python, and in heavily relies on NumPy for high-speed linear algebra and array operations. In addition to boot performance, som k algorithans are written in Cython. A Cythm wrapper around LIBSVM implements support vector machines; a similar wrapper around LIBLINEAR implements optic regression and linear support vector machines. It may not be posible to expand the methods with Python in such instances operating system.

OS

In Python, the OS module has functions for dealing with the Python's standard utility modules include OS. This module allows you to use operating system-dependent functions on the go. Many functions to interface with the file system are included in the os and os.path module

Glob

Glob is a generic name that refers to ways for matching given patterne wing Unix shell rules Glob is supported by Lanux and Unix systems and shells, and the function glob) is available in system libraries. The glob module uom the Unix shell's rules to identify all pathnamen that match a particular pattern, though the results are returned in any order. There is no tilde expansion, but character rangs indicated with 7, and will be accurately matched. This is accomplished by combining the osseandie() and fumatch furmatch() routines rather than running a subshell.

Pydub

The main class in Pydub is AudioSegment. An AudioSegment acts as a container to load, manipulate, and save audio.

To create our first audio script, we need a test audio file, this can be any supported format such as WAV, MP3, or AIFF. For the purposes of this tutorial, we're going to download a file as part of the script using urllib.request.

- Loading and saving different types of audio files.
- Splitting or appending audio in segments.
- Mixing audio from two different audio files.
- Changing audio levels or pan settings.
- Applying simple effects such as filters.
- Generating audio tones.

3.5 Impact analysis

3.5.1 Impact of project on society

Positive Impact of project on society:

- Detect substance dependency of a person.
- Development and also brings awareness about mental health in slum areas.
- Reduction in suicide rates.

Negative Impact of project on society:

- Misidentifying emotion.
- Identity\details can be at risk.

3.5.2 Impact of project on environment

Positive Impact of project on environment:

- The software can be able to identify one's emotions and substance dependency of the subject

Negative Impact of project on environment:

- Data like silence and pauses would take up unnecessary space in storage.

3.6 Professional ethical practices to be followed

- All the Research papers used for referring should be cited to give due credits.
- No identity of a subject should be revealed or misused, for their privacy concerns.
- Integrity while Documentation.

Chapter 4

Project Implementation

This project is implemented in a Machine Learning environment. We used python programming language to design the Machine Learning Model.

4.1 Software Implementation

The necessary libraries for the module are imported.

LIBROSA

- Librosa is a Python library for analyse audio and music. It can be used to extract the data from the audio files.
- It used to load audio file and provides various features such as audio I/O.
- Librosa convert the original signal into mono signal (Only one channel is used or one feature).
- Represent the audio signal w.r.t. normalise pattern (-1 to +1).
- Librosa by default converts the sample rate to 22 Khz.

PANDAS

- Pandas is the library that is used for creating and manipulating fast and efficient dataframe objects.

- It is ideal for working with tabular data.

SEABORN

- Seaborn is a python data visualization library, which provides a high level interface for presenting stastitical graphs.

OS

- OS is the module that is udes to interact with operating system like deleting a directory / folder etc.

GLOB

- Glob is using the module for path names can be searched using patterns.

MATPLOTLIB

- Matplotlib is the library used for creating visualizations in python.

NUMPY

- Numpy is a python library for advanced manipulation of arrays, Is specially useful while working in the domain of linear algebra, matrices, etc.

```
In [1]:
import os
import sys
import pandas as pd
import numpy as np
import glob

import librosa              # librosa is a Python library for analyzing audio and music.
import librosa.display      #It can be used to extract the data from the audio files.
# to play the audio files
from IPython.display import Audio

import seaborn as sns
import matplotlib.pyplot as plt
import tensorflow as tf
from matplotlib.pyplot import specgram
%matplotlib inline
```

```
C:\Users\Acer\anaconda3\lib\site-packages\paramiko\transport.py:219: Cryptog
raphyDeprecationWarning: Blowfish has been deprecated
  "class": algorithms.Blowfish,
```

Figure 4.1: Importing all the library

Dataset:

In this portion, we import the dataset and assign it to the Ravdess variable for further processing.

Here we created two directories file_emotion and file_path. ravdess_directory_list loads all data files located in the path which we have provided. The 5 data frames are displayed with the file location and emotion label.

RAVDESS dataset contains 1440 files: 60 trials per actor x 24 actors = 1440. This dataset contains 24 professional actors (12 female, 12 male. Speech emotions include calm, happy, sad, angry, fearful, surprised, and disgusted expressions. Each expression is produced at two levels of emotional intensity (normal, strong), with an additional neutral expression.

In [5]:

```
# Paths for data.
Ravdess = "E:\BE Project\RADESS_DATASET//audio_speech_actors_01-24//"

ravdess_directory_list = os.listdir(Ravdess)

file_emotion = []
file_path = []
for dir in ravdess_directory_list:
    # as their are 20 different actors in our previous directory we need to extract files f
    actor = os.listdir(Ravdess + dir)
    for file in actor:
        part = file.split('.')[0]
        part = part.split('-')
        # third part in each file represents the emotion associated to that file.
        file_emotion.append(int(part[2]))
        file_path.append(Ravdess + dir + '/' + file)

# dataframe for emotion of files
emotion_df = pd.DataFrame(file_emotion, columns=['Emotions'])

# dataframe for path of files.
path_df = pd.DataFrame(file_path, columns=['Path'])
Ravdess_df = pd.concat([emotion_df, path_df], axis=1)

# changing integers to actual emotions.
Ravdess_df.Emotions.replace({1:'neutral', 2:'calm', 3:'happy', 4:'sad', 5:'angry', 6:'fear'
Ravdess_df.head()
```

Out[5]:

	Emotions	Path
0	neutral	E:\BE Project\RADESS_DATASET//audio_speech_act...
1	neutral	E:\BE Project\RADESS_DATASET//audio_speech_act...
2	neutral	E:\BE Project\RADESS_DATASET//audio_speech_act...
3	neutral	E:\BE Project\RADESS_DATASET//audio_speech_act...
4	calm	E:\BE Project\RADESS_DATASET//audio_speech_act...

Figure 4.2: Importing the Dataset

Plotting Emotions:

```
In [7]:
plt.title('Count of Emotions', size=16)
sns.countplot(data_path.Emotions)  # sns. countplot , Seaborn literally counts the number o
plt.ylabel('Count', size=12)
plt.xlabel('Emotions', size=12)
#sns.despine(top=True, right=True, left=False, bottom=False) #The despine() is a function t

plt.show()
```

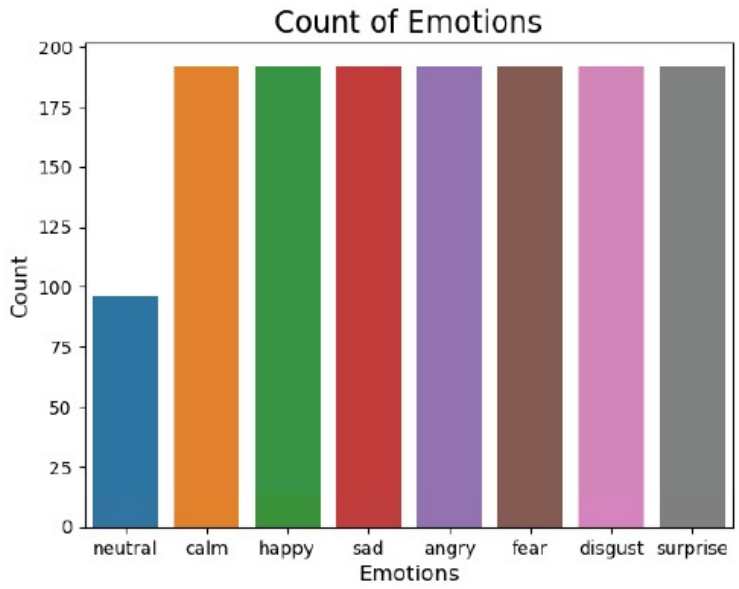


Figure 4.3: Plot containing Emotion Labels on the X-axis and Count in the Y-axis

Creating Function:

The waveplot method accepts data, sample rate and emotion label. It displays the provided data in the form of a waveplot using Librosa. The 'spectrogram' method accepts the same data and plot the spectrogram accordingly. It accepts the emotion for which the waveplot and spectrogram need to be plotted. The waveplot and spectrogram for each emotion is plotted using these two method, we will see below.

```
In [17]:
def create_waveplot(data, sr, e):  #waveplot is used to plot waveform of amplitude vs time
    plt.figure(figsize=(10, 3)) #creat_waveplot use for know the pitch for audio file at gi
    plt.title('Waveplot for audio with {} emotion'.format(e), size=15)
    librosa.display.waveshow(data, sr=sr)
    plt.show()

def create_spectrogram(data, sr, e):  #Spectogram. A spectrogram is a visual representation
                                      #use to display power spectrum movement of audio
    # stft function converts the data into short term fourier transform
    X = librosa.stft(data)
    Xdb = librosa.amplitude_to_db(abs(X))  #. The default for librosa.amplitude_to_db is to
                                           #The function also applies a threshold on the ra
    plt.figure(figsize=(12, 3))
    plt.title('Spectrogram for audio with {} emotion'.format(e), size=15)
    librosa.display.specshow(Xdb, sr=sr, x_axis='time', y_axis='hz')
    #librosa.display.specshow(Xdb, sr=sr, x_axis='time', y_axis='log')
    plt.colorbar()
```

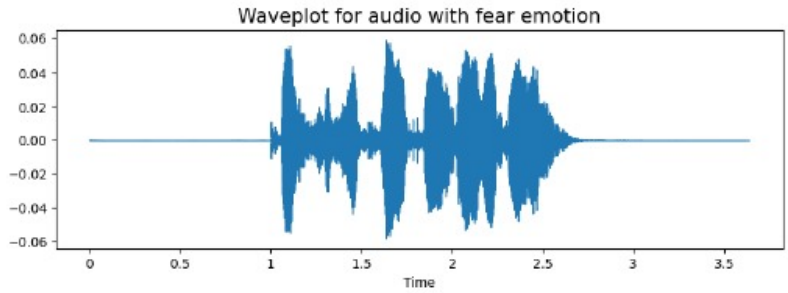
Figure 4.4: Code for Waveplot and Spectrogram method

Fear Emotion.

```
In [21]:
emotion='fear'
path = np.array(data_path.Path[data_path.Emotions==emotion])[1]
data, sampling_rate = librosa.load(path) #librosa.load--Load an audio file as a floating po
                                         #Audio will be automatically resampled to the given
                                         #To preserve the native sampling rate of the file,
                                         #create_waveplot(data, sampling_rate, emotion)
        #when we load file using Librosa it wil gives us the two things, 1) data (that is c
        # 2) sampling rate which by default converted to 22khz.

create_waveplot(data, sampling_rate, emotion)
create_spectrogram(data, sampling_rate, emotion)

data
librosa.util.normalize(data)
data
#create_waveplot(data, sampling_rate, emotion)
#create_spectrogram(data, sampling_rate, emotion)
#plt.title('Noramlise Audio')
#Audio(path)
```




```
Out[21]:
array([9.51426409e-05, 1.64977813e-04, 1.19230936e-04, ...,
       0.00000000e+00, 0.00000000e+00, 0.00000000e+00], dtype=float32)
```

Spectrogram for audio with fear emotion

Figure 4.5: Wave plot for Fear emotion

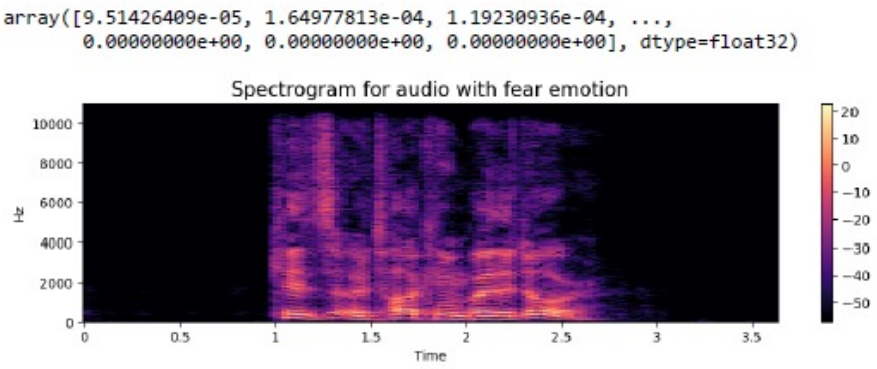


Figure 4.6: Waveplot and Spectogram for fear Emotion

Data Augmentation:

Data augmentation is the process by which we create new synthetic data samples by adding small perturbations to our initial training set. To generate syntactic data for audio, we can apply noise injection, shifting time, and changing pitch and speed. The objective is to make our model invariant to those perturbations and enhance its ability to generalize. In images data augmentation can be performed by shifting the image, zooming, and rotating.

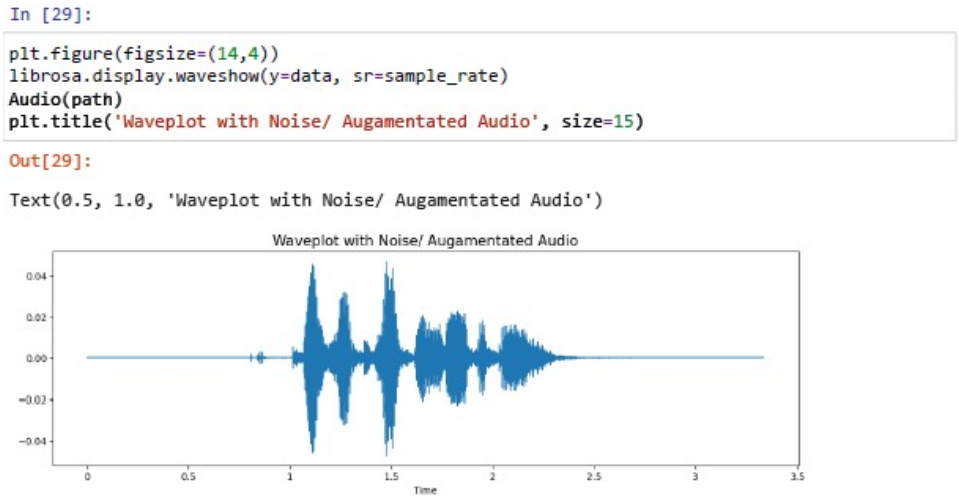


Figure 4.7: Data augmentation

Feature Extraction

Extraction of features is a very important part of analyzing and finding relations between different things. As we already know that the data provided by audio cannot be

understood by the models directly so we need to convert them into an understandable format for which feature extraction is used.

The audio signal is a three-dimensional signal in which three axes represent time, amplitude and frequency.

Zero Crossing Rate:

The rate of sign changes of the signal during the duration of a particular frame.

MFCC:

MFCCs Mel Frequency Cepstral Coefficients form a cepstral representation where the frequency bands are not linear but distributed according.

MFCC Convert audio into some kind of feature based on the frequency and time characteristics which help to do the classification.

```
In [35]:
def extract_features(data):
    # ZCR
    result = np.array([])
    zcr = np.mean(librosa.feature.zero_crossing_rate(y=data).T, axis=0)
    result=np.hstack((result, zcr)) # stacking horizontally
    #print(zcr)
    # Chroma_stft
    stft = np.abs(librosa.stft(data))
    chroma_stft = np.mean(librosa.feature.chroma_stft(S=stft, sr=sample_rate).T, axis=0)
    result = np.hstack((result, chroma_stft)) # stacking horizontally

    # MFCC
    mfcc = np.mean(librosa.feature.mfcc(y=data, sr=sample_rate).T, axis=0)
    result = np.hstack((result, mfcc)) # stacking horizontally

    # Root Mean Square Value
    rms = np.mean(librosa.feature.rms(y=data).T, axis=0)
    result = np.hstack((result, rms)) # stacking horizontally
     # MelSpectogram
    mel = np.mean(librosa.feature.melspectrogram(y=data, sr=sample_rate).T, axis=0)
    result = np.hstack((result, mel)) # stacking horizontally

    return result
```

Figure 4.8: Code for Feature Extraction

Chapter 5

Results and Discussion

We utilize a spectrogram to visualize signals in a picture, which depicts time on the x-axis, frequency on the y-axis, and amplitude on the x-axis for more comprehensive information. It can also be in different hues, with the color density indicating the signal's strength. Finally, it provides a summary of the signal, describing how the signal's strength is divided among frequencies. Amplitude can be defined as the greatest distance traveled by a moving body in a periodic motion in a single time unit or the highest distance of the wave on dips down or rising from its flat surface.
The spectrogram is computed using the Gabor transform. The Gabor transform is a subset of the short-time Fourier transform and is used to extract the sinusoidal frequency and phase content of a signal in a specific area. We transcribe some signals in space or time into their frequency components using the Fourier transform. We can extract the signal's frequency components to build the signal after executing a Fourier The Gaussian function is multiplied by our signal function in the Gabor transform.

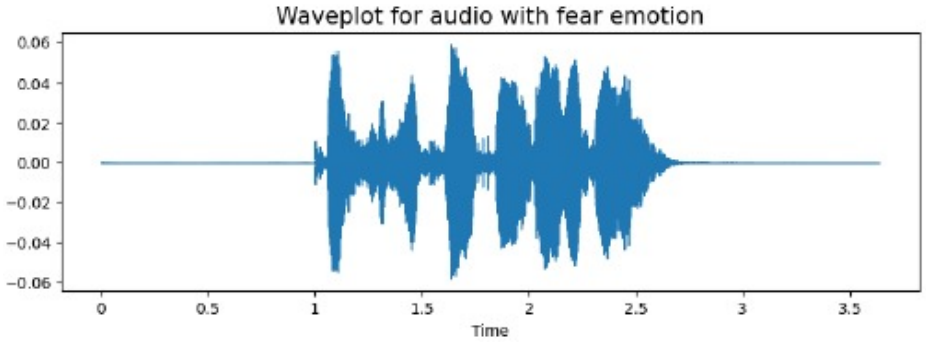
The function can be thought of as a window function, and the process resultant is then transformed with the Fourier transform to get a time-frequency analysis. The signal near the time we wish to study the signal and give it a larger weight is the window function.

$$G_x(\tau, \omega) = \int_{-\infty}^{\infty} x(t) e^{-\pi(t-\tau)^2} e^{-j\omega t} \, dt$$

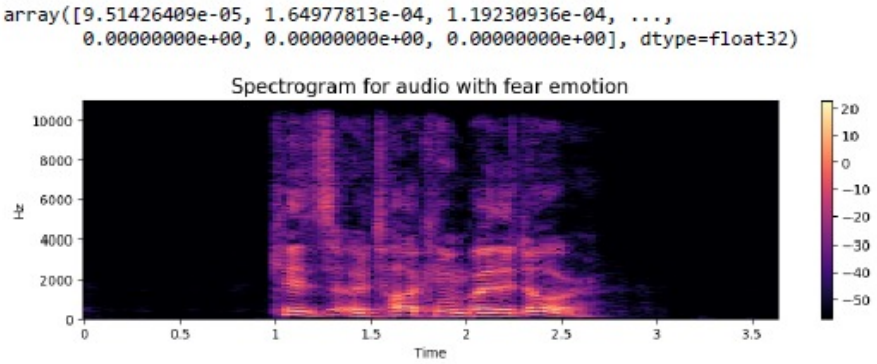
Figure 5.1: Equation for Gabor Transform of a signal x(t)

5.1 Simulation Results

The wave plot for fear emotion has been obtained.

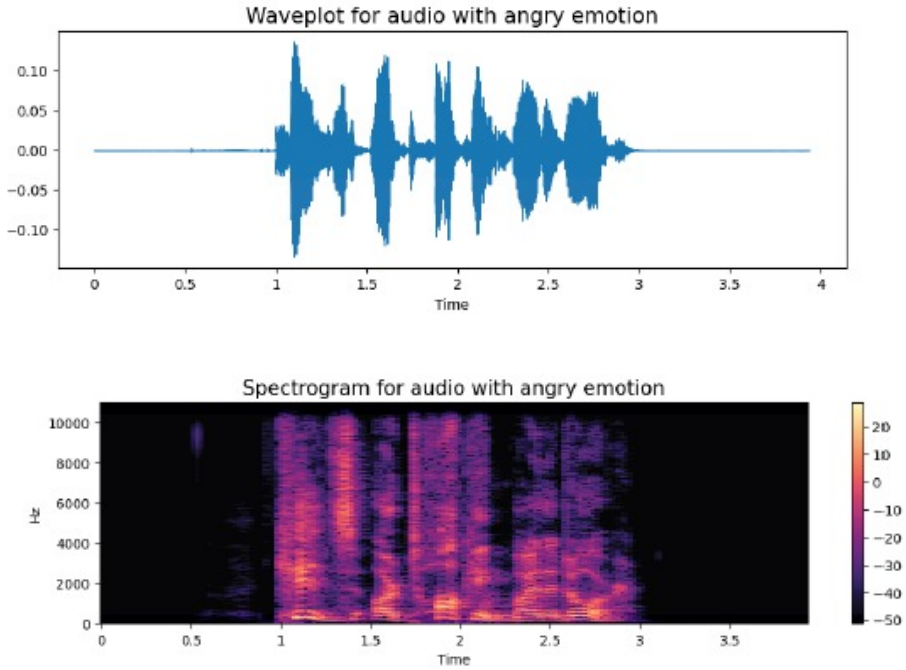


The Spectrogram for fear emotion has been obtained.

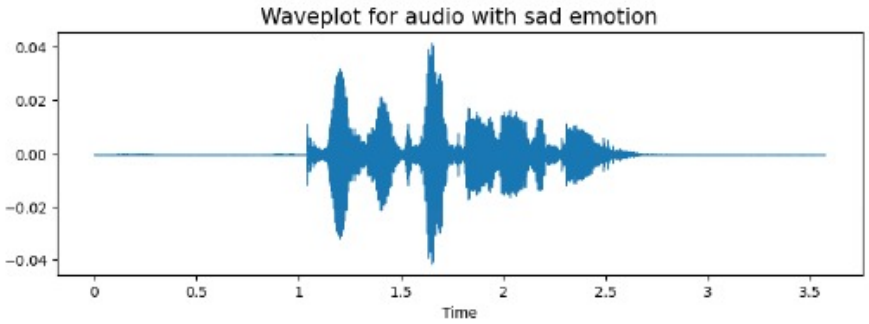


The wave plot for Angry emotion has been obtained.

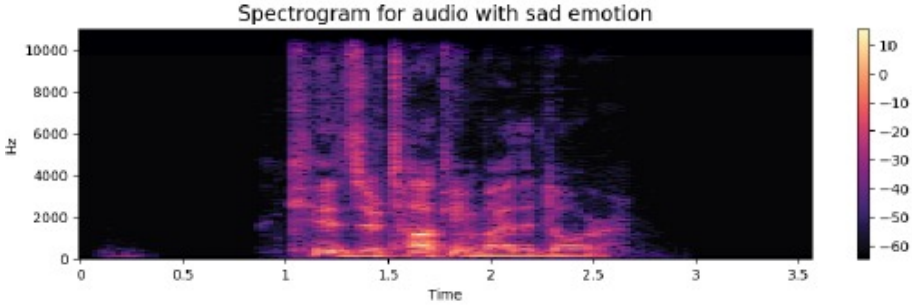
The Spectrogram for angry emotion has been obtained.



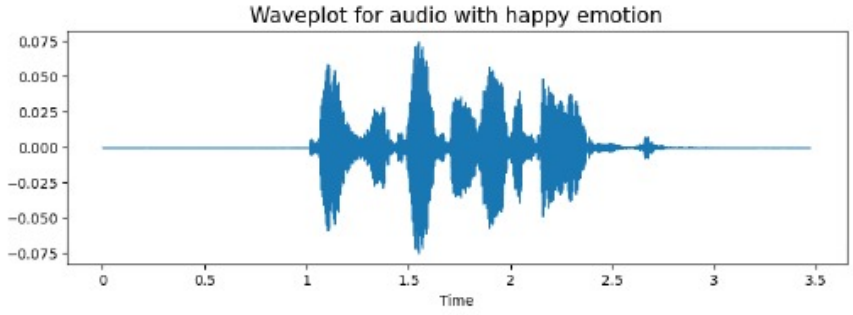
The wave plot for sad emotion has been obtained.



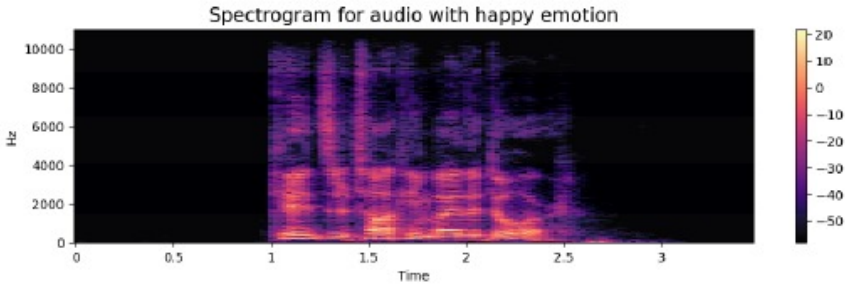
The Spectrogram for sad emotion has been obtained.



The wave plot for happy emotion has been obtained.



The Spectrogram for happy emotion has been obtained.



Chapter 6

Conclusions and Future Scope

6.1 Conclusions

The data collected during this process can be used for the good purpose of society such as can highlight which area needs to be developed more resulting in great help for the government and other NGO Organizations.
This project can be further exceeded to create a whole rehabilitation app where treatment can be provided to the subject.

6.2 Future Scope

The speech signal will be classified as emotions then the emotions with depression, sadness, anger, upset, and the other emotions which help in substance dependency will be further classified.

References

[1] A. M. Badshah, J. Ahmad, N. Rahim and S. W. Baik, "Speech Emotion Recognition from Spectrograms with Deep Convolutional Neural Network," 2017 International Conference on Platform Technology and Service (PlatCon), 2017, pp. 1-5, doi: 10.1109/PlatCon.2017.7883728.

[2] Dong , X., Ruan, A. (2021). Speech Emotion Recognition algorithm based on deep learning algorithm fusion of temporal and spatial features. Research gate. https://doi.org/Xu Dong AnZhou Ruan

[3] Short-time Fourier transform. (2022, October 11). In Wikipedia.

[4] H. Ibrahim, C. K. Loo and F. Alnajjar, "Speech Emotion Recognition by Late Fusion for Bidirectional Reservoir Computing With Random Projection," in IEEE Access, vol. 9, pp. 122855-122871, 2021, doi: 10.1109/ACCESS.2021.3107858.

[5] X. Cai, D. Dai, Z. Wu, X. Li, J. Li and H. Meng, "Emotion Controllable Speech Synthesis Using Emotion-Unlabeled Dataset with the Assistance of Cross-Domain Speech Emotion Recognition," ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, pp. 5734-5738, doi: 10.1109/ICASSP39728.2021.9413907.

[6] Z. Peng, X. Li, Z. Zhu, M. Unoki, J. Dang and M. Akagi, "Speech Emotion Recognition Using 3D Convolutions and Attention-Based Sliding Recurrent Networks With Auditory Front-Ends," in IEEE Access, vol. 8, pp. 16560-16572, 2020, doi: 10.1109/ACCESS.2020.2967791.

[7] Short-time Fourier transform. (2022, October 11). In Wikipedia.

[8] M. S. Likitha, S. R. R. Gupta, K. Hasitha and A. U. Raju, "Speech based human emotion recognition using MFCC," 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), 2017, pp. 2257-2260, doi: 10.1109/WiSPNET.2017.8300161.

[9] C. Jie, "Speech emotion recognition based on convolutional neural network," 2021 International Conference on Networking, Communications and Information Technology (NetCIT), 2021, pp. 106-109, doi: 10.1109/NetCIT54147.2021.00028.

[10] X. Ying and Z. Yizhe, "Design of Speech Emotion Recognition Algorithm Based on Deep Learning," 2021 IEEE 4th International Conference on Automation, Electronics and Electrical Engineering (AUTEEE), 2021, pp. 734-737, doi: 10.1109/AUTEEE52864.2021.9668689.

[11] Z. Han and J. Wang, "Speech emotion recognition based on Gaussian kernel nonlinear proximal support vector machine," 2017 Chinese Automation Congress (CAC), 2017, pp. 2513-2516, doi: 10.1109/CAC.2017.8243198.

[12] Ainurrochman, I. I. Febriansyah and U. L. Yuhana, "SER: Speech Emotion Recognition Application Based on Extreme Learning Machine," 2021 13th International Conference on Information Communication Technology and System (ICTS), 2021, pp. 179-183, doi: 10.1109/ICTS52701.2021.9609016.