# **Study Analysis and Development of Algorithms to identify Human Emotion using Human Voice**

# **A project report submitted to**

# **Rajiv Gandhi University of knowledge technologies**

# **SRIKAKULAM**

# **In partial fulfilment of the requirements for the**

# **Award of the degree of**

# **BACHELOR OF TECHNOLOGY**

# **IN**

# **COMPUTER SCIENCE AND ENGINEERING**

# **Submitted by**

# **3rd year B. Tech 2nd semester**

# **Navyasri neelapu (S170336)**

# **Yalla. sandhya (S170185)**

# **S.Prema jyothi(s170567)**

# **Under the Esteemed Guidance of**

# **Mr. Sesha kumar Sir**



Rajiv Gandhi University of Knowledge Technologies – SKLM

# **CERTIFICATE**

This is to certify that the thesis work titled “**Study Analysis and Development of Algorithms to identify Human Emotion using Human Voice**” was successfully completed by Navyasri neelapu(s170336), Yalla.Sandhya(s170185), S.Prema Jyothi(s170567).In partial fulfilment of the requirements for the Mini Project in Computer Science and Engineering of **Rajiv Gandhi University of Knowledge Technologies** under my guidance and output of the work carried out is satisfactory.

**Mr. Sesha kumar sir,Asst Prof(CSE)** Mr. K.Dileep Kumar sir,Asst Prof(CSE)

**Project Guide Project Coordinator**

**DECLARATION**

I declared that this thesis work titled “**Study Analysis and Development of Algorithms to identify Human Emotion using Human Voice**” is carried out by our team during the year 2021-22 in partial fulfillment of the requirements for the Mini Project in **Computer Science and Engineering.**

I further declare that this dissertation and the matter embodied in this report has not been submitted elsewhere for any Degree. Furthermore, the technical details furnished in various chapters of this thesis are purely relevant to the above project and there is no deviation from the theoretical point of view for design, development and implementation.

NEELAPU NAVYASRI (S170336)

YALLA SANDHYA (S170185)

S.PREMA JYOTHI (S170567)

**ACKNOWLEDGEMENT**

I would like to articulate my profound gratitude and indebtedness to my project guide Mr.Sesha Kumar sir Assistant Professor who has always been a constant motivation and guiding factor throughout the project time. It has been a great pleasure for me to get an opportunity to work under his guidance and complete the thesis work successfully.

We would like to thank Mrs.S.Lakshmi Sri madam(Asst Prof), Head of the department for her cooperation in completing our project.

I wish to extend my sincere thanks to Mr. K.Dileep Kumar.sir, Project Coordinator of Computer Science and Engineering Department, for his constant encouragement throughout the project.

I am also grateful to other members of the department without their support my work would have been carried out so successfully.

I thank one and all who have rendered help to me directly or indirectly in the completion of my thesis work.

Project Members

Navyasri neelapu(s170336),

Yalla Sandhya(s170185),

S.Prema Jyothi(s170567).

ABSTRACT

We did a comparative study of speech emotion recognition (SER) systems. Emotion is a medium of expression of one’s perspective or one’s mental state to others. we are able to detect the mood of any person by just seating with him/her even if we don’t know him/ her first time by simply analyzing the frequency of the tone of a particular person. According to Past study of the voice analysis suggests that, analyzing the frequency of the voice model is able to predict the sentimental as well as emotion associated with the human voice. This act of attempting to recognize human emotion and effective states of speech. This capitalize on the fact that voice often reflects underlying emotions through tone, pitch and energy levels of human.

**INDEX**

CH.NO CONTENTS PG.NO

1 INTRODUCTION 1

1.1 Introduction 8

1.2 Problem of the Statement 9

1.3 Objectives 9

1.4 Goals 9

1.5 Scope 9

1.6 Applications 10

1.7 Limitations 10

2 LITERATURE SURVEY 11

2.1 Collecting Information 11

2.2 Study 12

2.3 Benefits 12

2.4 Summary 12

3 SYSTEM ANALYSIS 13

3.1 Existing System 14

3.2 Disadvantages 16

3.3 Proposed System 17

3.4 Advantages 18

3.5 System Requirements 19

4 SYSTEM DESIGN

Design of the System 20

4.1.1 Class Diagram 21

4.1.2 Use Case Diagram 22

4.1.3 Sequence Diagram 23

4.1.4 Data Flow Diagram 23

5 SOURCECODE 24

6. Test Code 32

7. SYSTEM TESTING 42

Testing Introduction 43

Levels Of Testing 44

8. CONCLUSION 45

9. SPEECH EMOTION RECOGNIZATION REFERENCES 46

**1. INTRODUCTION**

**1.1 Introduction**

Emotion plays a significant role in daily interpersonal human interactions. It helps us to match and understand the feelings of others by conveying our feelings and giving feedback to others. Human emotions are very difficult to comprehend from a quantitative perspective. Speech can be used as one of the best ways of guessing the emotional state of a person. Speech is a complex signal which contains information about the message, speaker, language and emotions. There are various kinds of emotions which can be articulated using speech. Emotional speech recognition is a system which basically identifies the emotional state of human being from his or her voice; speech is very misleading even for humans to judge the emotion of the speaker.

Research has revealed the powerful role that emotion play in shaping human social interaction. Emotional displays convey considerable information about the mental state of an individual. This has opened up a new research field called automatic emotion recognition, having basic goals to understand and retrieve desired emotions.

In prior studies, several modalities have been explored to recognize the emotional states such as facial expressions, speech, physiological signals, etc. Several inherent advantages make speech signals as a good source for affective computing. For example, compared to many other biological signals (e.g., electrocardiogram), speech signals usually can be acquired more readily and economically. This is why the majority of researchers are interested in speech emotion recognition (SER). The area has received increasing research interest all through current years.

**1.2 Statement of the problem**

1.3 OBJECTIVE

To build a model to recognize emotion from speech using the librosa and sklearn libraries and the RAVDESS, TESS and CREMA-D datasets. SER aims to recognize the underlying emotional state of a speaker from her voice. 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 finds its applications in medicine and forensics.

* 1. GOALS

The goal is to identify correct prediction values so that the classifier can accurately predict unseen testing data. Speech emotion recognition is a system through which various audio speech files are classified into different emotions such as happy, sad, anger and neutral.

1.5 SCOPE

* In the feature subsystems with further advancement and research can be used to detect severe mental illness like depression and others
* using the combination of ml approaches as this increase the accuracy and efficiency of the system

Emotion recognition plays a crucial role in the area of Artificial intelligence and Internet of things. It offers tremendous scope to human computer interaction, robotics, health care, biometric security and behavioral modeling. Emotion recognition systems recognize emotions from facial expressions, text data, body movements, voice, brain or heart signals. Along with basic emotions, attitude, control over emotions and power of activation of emotion can also be examined for analyzing sentiments. It identifies various supervised and unsupervised machine-learning techniques for feature extraction and emotion classification such as emotion recognition, compound emotions.

1.6 APPLICATIONS

There are many applications of detecting the emotion of the persons like in the interface with

* robots,
* audio surveillance,
* web-based E-learning,
* commercial applications,
* clinical studies,
* entertainment,
* banking,
* call centers,
* Music recommendations,
* computer games.

For classroom orchestration or E-learning, information about the emotional state of students can provide focus on the enhancement of teaching quality. For example, a teacher can use SER to decide what subjects can be taught and must be able to develop strategies for managing emotions within the learning environment. That is why learner’s emotional state should be considered in the classroom.

1.7 LIMITATIONS

* Each emotion may correspond to the different portions of the spoken utterance. The same utterance may show different emotions Therefore it is very difficult to differentiate these portions of utterances. Another problem is that Expression of emotion is depending on the speaker and their culture and environment.
* Difficult to build a perfect system. Speaking is loud and invites noise to others. Filtering background noise is a task and it is too much that can even be difficult for humans to accomplish. This biometric is sensitive to environmental conditions such as background noise.
* Speaking is loud and invites noise to others.
* Filtering background noise is a task and it is too much that can even be difficult for humans to accomplish.
* Error and misinterpretation of words.

Chapter 2

LITERATURE SURVEY

2.1 Collect Information

We have taken the information from the other sources like different research papers and other sites to check how they are categorized and organized and we proposed this ourselves.

* **Datasets used:**

1.Ryerson Audio-Visual Database of Emotional Speech and song(RAVDESS)

2.Toronto Emotional Speech Set(TESS)

* **Algorithms Studied:**

1.Convolutional Neural Network(CNN)

2.Multi-layer Perceptron Classifier(MLP)

3.Support Vector Machine(SVM)

4.Recurrent Neural Network(RNN)

2.2 Study

**SER key features:**

* Input speech signal
* Feature extraction using MFCC and LPCC
* Classifier based on SVM
* Training, and testing
* Output emotion tags.

2.3 Benefits

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.

* In psychiatric diagnosis, lie detection
* In Call center conversation may be used to analyze behavioral study of call attendants with the customers which helps to improve quality of service of a call attendant.
* In aircraft cockpits, speech recognition systems trained to recognize stressed speech are used for better performance.
* Emotion analysis of telephone conversation between criminals would help crime investigation department.
* It is Useful for enhancing the naturalness in speech based human machine interaction.
* Interactive movie, storytelling & E-tutoring applications would be more practical, if they can adapt themselves to listeners or Students emotional states.
* Conversation with robotic pets and humanoid partners would be more realistic and enjoyable, if they are able to understand and express emotions like humans.

**Chapter -3**

**ANALYSIS**

**3.1 Existing system**

(Social networking sites)

* The existing work in this area reveals that most of the present work relies on Lexical analysis is used 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 testing audio file and the training audio file is used as an integral parameter for identification of a particular emotion type. Apart from that in proposed system is used to recognize the emotions based on the selected features, and proved CNN(Conventional Neural Network) Produce more accurate results for emotion recognition than the Recurrent neural networks (RNN).
* The second technique is used with the feature extraction of discriminatory feature with the cubic SVM classifier for recognition of angry, happy and neutral emotion segments only.

*PROCEDURE OF EXISTING SYSTEM:*

Speech sample is first passed through a gender reference database which is maintained for recognition of gender before it gets into the process. Statistical approach is followed taking pitch as feature for gender recognition. A lower and upper bound pitch for both male and female samples could be found using the reference database. Input human voice sample was first broken down into frames of frame size 16ms each. This was done for frame level classification in further steps. For each frame MFCC (Mel Frequency Cepstrul Coefficient) was calculated as the main feature for emotion recognition. Reference database is maintained which contains the MFCCs of emotions i.e. of Sad, Anger, Neutral and Happy. MFCC of the frames were compared with the MFCCs stored in reference database and the distance was calculated between the comparable frames. Based on the distance of the analysis frame from the reference database, one can classify the frame as anger, happy or normal. The output is displayed in terms of emotional frame count.

Speech sample [2, 4, 6, 9] is first passed through a gender reference database which is maintained for

recognition of gender before it gets into the process. Statistical approach [5] is followed taking pitch as feature

for gender recognition [9]. A lower and upper bound pitch for both male and female samples could be found

using the reference database [14]. Input human voice sample was first broken down into frames of frame size

16 ms each. This was done for frame level classification in further steps.

For each frame MFCC(Mel Frequency Cepstral Coefficient)was calculated as the main feature for emotion

recognition. Reference database [14] is maintained which contains the MFCCs of emotions i.e. of Sad, Anger,

Neutral and Happy.

MFCC of the frames were compared with the MFCCs stored in reference database and the distance was

calculated between the comparable frames. Based on the distance of the analysis frame from the reference

database, one can classify the frame as anger, happy or normal. The output is displayed in terms of emotional

frame count

Speech sample [2, 4, 6, 9] is first passed through a gender reference database which is maintained for

recognition of gender before it gets into the process. Statistical approach [5] is followed taking pitch as feature

for gender recognition [9]. A lower and upper bound pitch for both male and female samples could be found

using the reference database [14]. Input human voice sample was first broken down into frames of frame size

16 ms each. This was done for frame level classification in further steps.

For each frame MFCC(Mel Frequency Cepstral Coefficient)was calculated as the main feature for emotion

recognition. Reference database [14] is maintained which contains the MFCCs of emotions i.e. of Sad, Anger,

Neutral and Happy.

MFCC of the frames were compared with the MFCCs stored in reference database and the distance was

calculated between the comparable frames. Based on the distance of the analysis frame from the reference

database, one can classify the frame as anger, happy or normal. The output is displayed in terms of emotional

frame count

Speech sample [2, 4, 6, 9] is first passed through a gender reference database which is maintained for

recognition of gender before it gets into the process. Statistical approach [5] is followed taking pitch as feature

for gender recognition [9]. A lower and upper bound pitch for both male and female samples could be found

using the reference database [14]. Input human voice sample was first broken down into frames of frame size

16 ms each. This was done for frame level classification in further steps.

For each frame MFCC(Mel Frequency Cepstral Coefficient)was calculated as the main feature for emotion

recognition. Reference database [14] is maintained which contains the MFCCs of emotions i.e. of Sad, Anger,

Neutral and Happy.

MFCC of the frames were compared with the MFCCs stored in reference database and the distance was

calculated between the comparable frames. Based on the distance of the analysis frame from the reference

database, one can classify the frame as anger, happy or normal. The output is displayed in terms of emotional

frame count

Mel Filter Bank

Fourier Transformation

Input

Framing

Inverse Fourier Transformation

MFCC

Logarithmic Conversion

**Disadvantages** :  
➨Validation of emotion dataset is a challenge in order to have accurate emotion recognition system. The system is very slow as to compare the correlations of the complete dataset with just one audio file.  
➨ Variable length audio files are not understandable. Finding or detecting emotion is haystack.  
➨It is a challenge to make emotion available in different languages.  
➨Long pre-processing steps are required for the model to understand the audio signals.

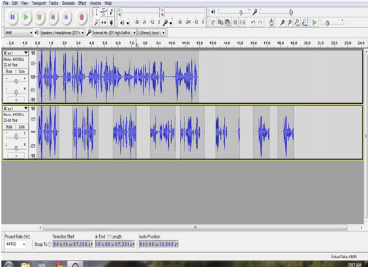
➨Performance and results of the emotion system depends on accuracy of the voice signals, voice recognition algorithm used and so on. Highly accurate system will be expensive due to use of costly components.

**3.3 Proposed System**

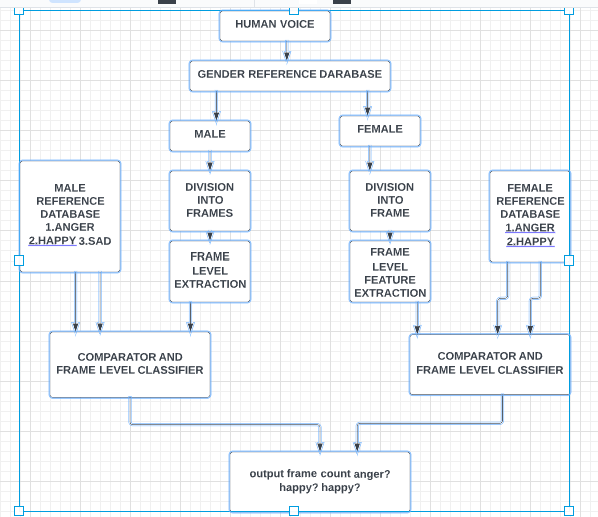
* In the project, MFCC has been used as the feature for classifying the speech data into various emotion categories employing artificial neural networks. The usage of the neural networks provides us the advantages of classifying many different types of emotions in a variable length of audio signal in a real time environment.
* This technique manages to establish a good balance between computational volume and performance accuracy of the real-time processes.

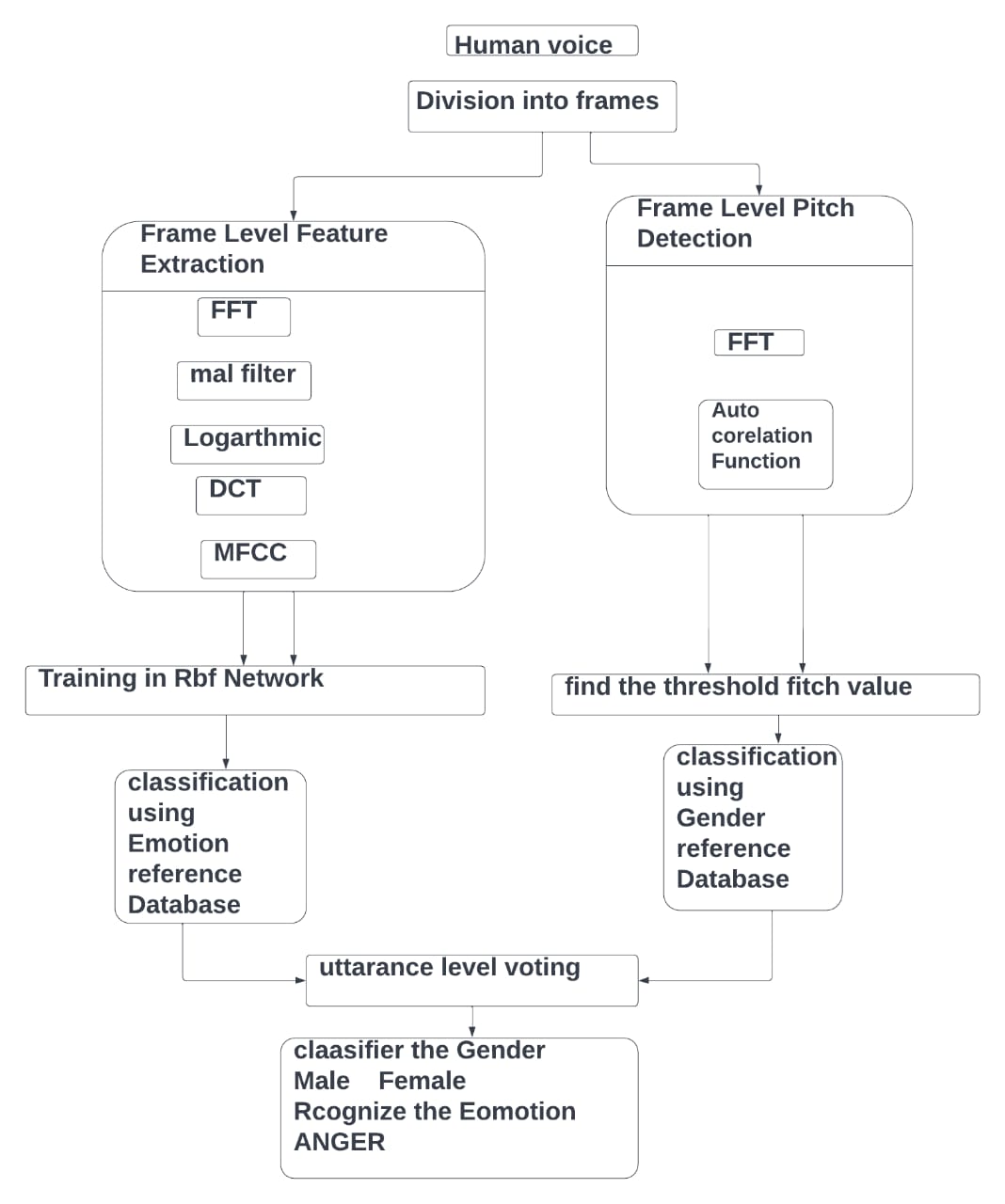
PROCEDURE OF PROPOSED SYSTEM:

* + In Proposed system, human voice is given as the input, then the output is converted into frames of frame size 60ms which means overlapping of data for 10ms. this is because for no missing of data. Fundamental frequencies are calculated based on pitch auto correlation. Using SVM reference database average pitch value is calculated based on which gender can be classified to human speech and classifying the emotion with more than 60% of accuracy.



For emotion recognition each frame can be entered into the proposed MFCC approach. Mel Frequency Cepstral Coefficient function contains group of four operations on human speech. Fast Fourier Transform will be applied to each for finding minimum and maximum frequencies. Later Mel filter bank can be applied to map the powers of spectrum obtained above using overlapped triangular windows, after which logarithmic conversion will be done for finding amplitude values. Finally discrete cosine transform will be applied to get the missing data while compressing the audio clip, finally the MFCC values for each frame will be calculated.





**ADVANTAGES OF PROPOSED SYSTEM:**

* Can be implemented in any hardware supporting the python language.
* Very fast in processing the audio and easy to use.
* Variable length audio files are understood by the system.

**SYSTEM SPECIFICATIONS**:

Hardware:

Processor: AMD A9-9420 RADEON R5, 5 COMPUTER CORES

2C+3G 3.00GHz

Hard disk: 60GB

RAM : 4GB

Software:

Operating system : WINDOWS 10 pro

Programming language: Python

**CHAPTER- 4**

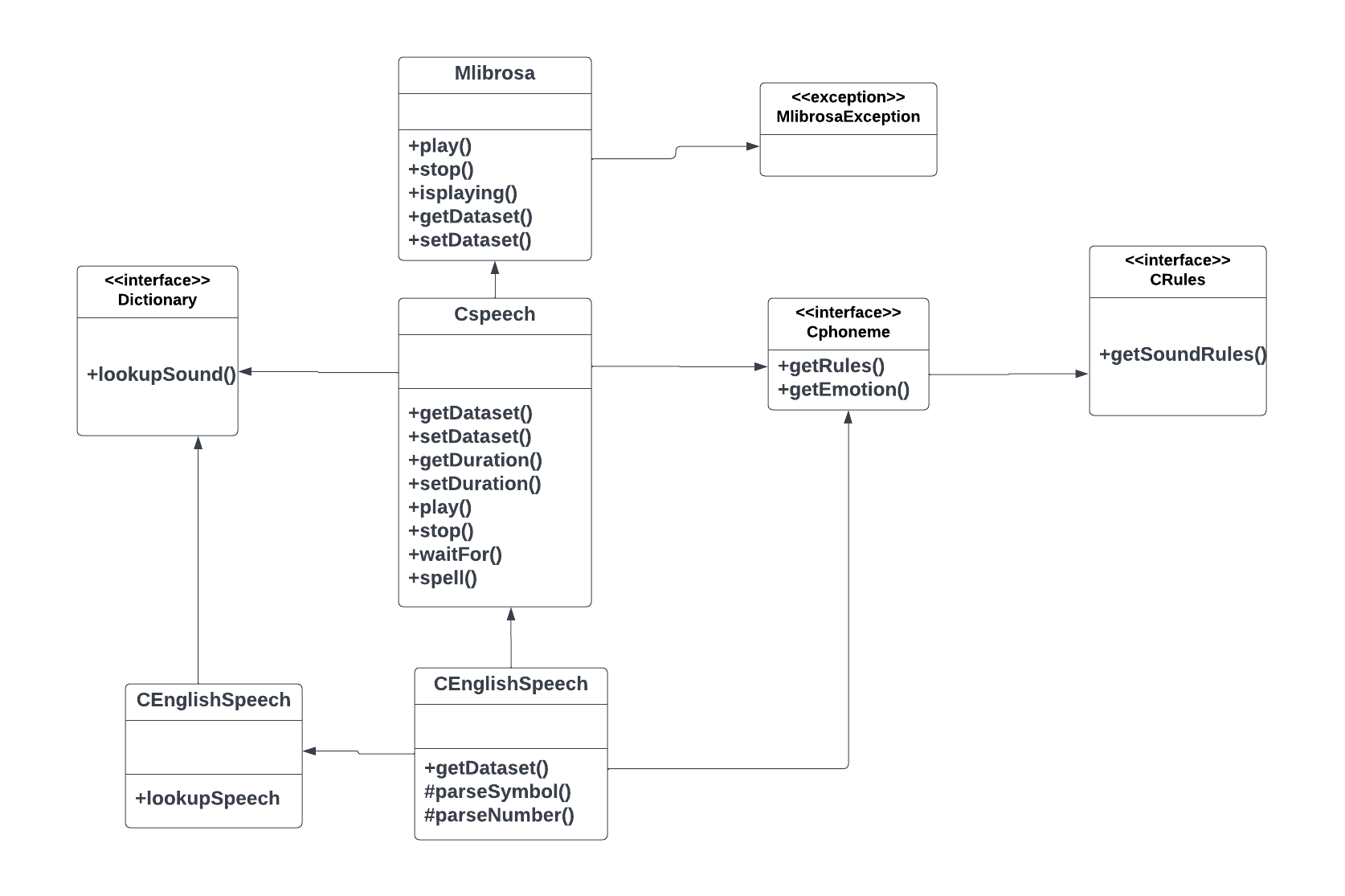
**SYSTEM DESIGN**

**4.1 DESIGN OF THE SYSTEM**

Speech emotion recognition system as a complex solution including a toronto dataset of emotion samples in a form of short sound records and the tool evaluating database samples by using subjective methods.  In order to create the database of emotion samples for learning and training of emotional classifier, it was necessary to extract short sound recordings. In the second step, all records in emotion database were evaluated using our designed evaluation tool and results were automatically evaluated how they are credible and reliable and how they represent different states of emotions.

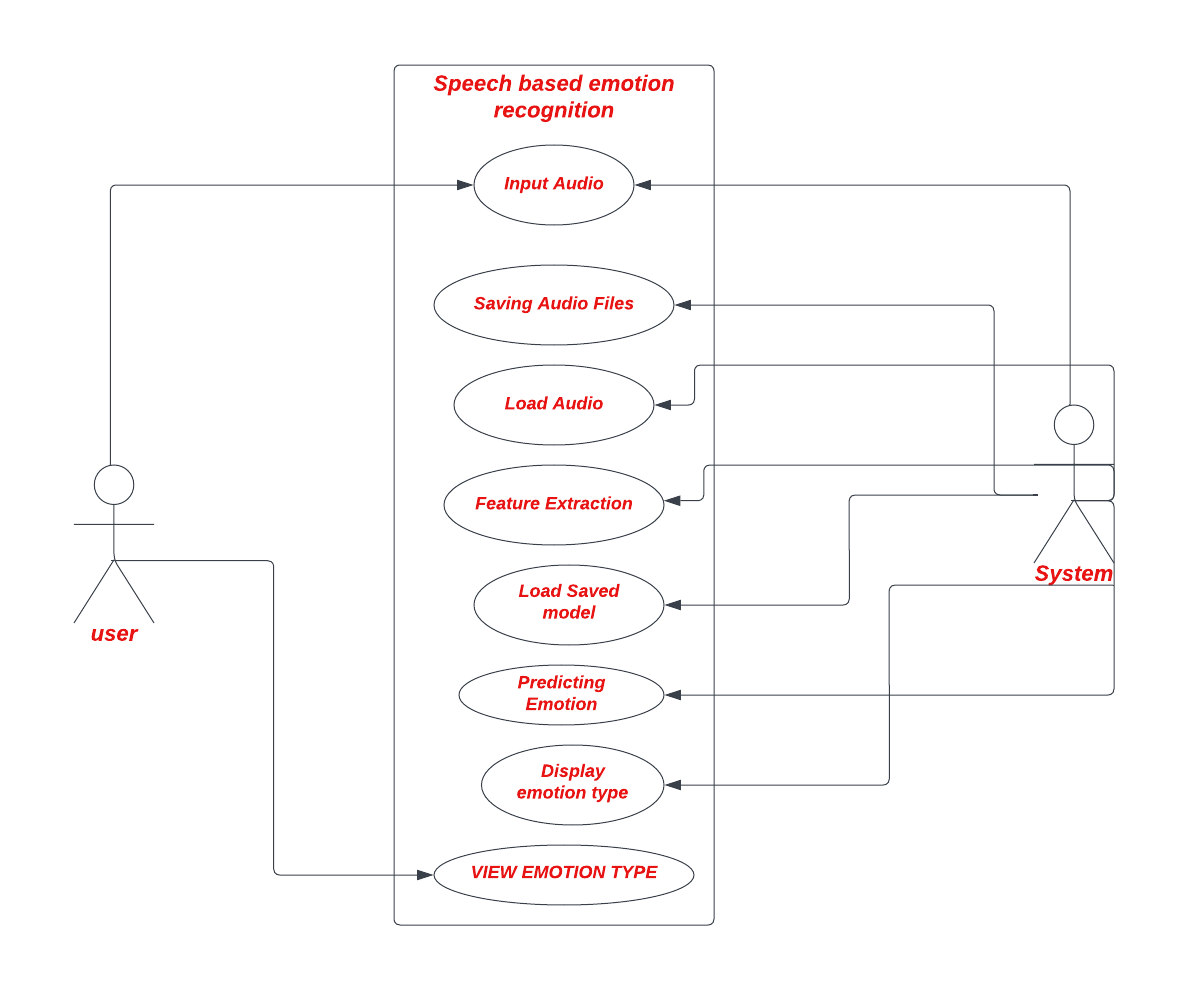
**4.1.1 Class diagram**:

Class diagram in the Unified Modelling Language (UML), is a kind of static structure diagram hat describes the constitution of a process through showing the system's classes, their attributes, and the relationships between the class. The motive of a class diagram is to depict the classes within a model. In an object-oriented software, classes have attributes (member variables), operations (member capabilities) and relation.



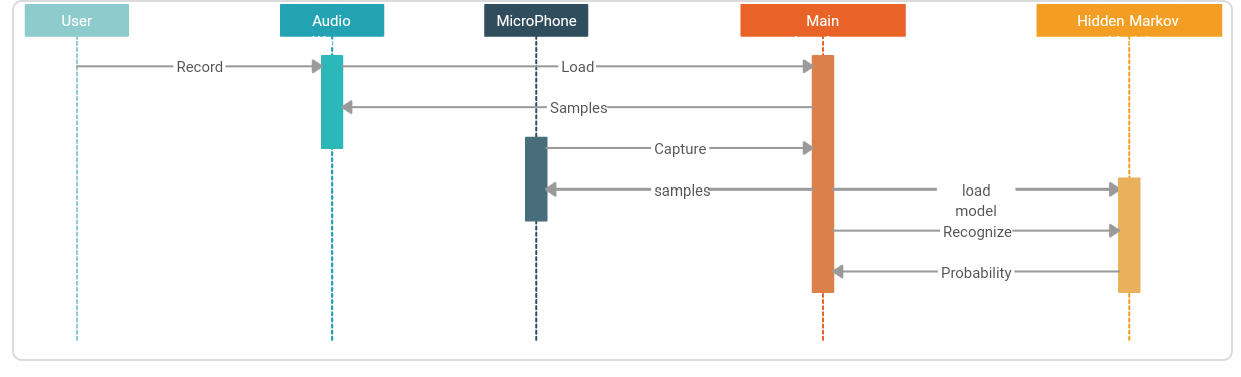
**4.1.2 UseCase Diagram:**

It is a visually representation what happens when actor interacts with system. A use case diagram captures the functional aspects of a system. The system is shown as a rectangle with name of the system inside, the actor are shown as stick figures, the use case are shown as solid bordered ovals labeled with name of the use case and relationships are lines or arrows between actor and use cases. Symbols used in Use case are as follows-



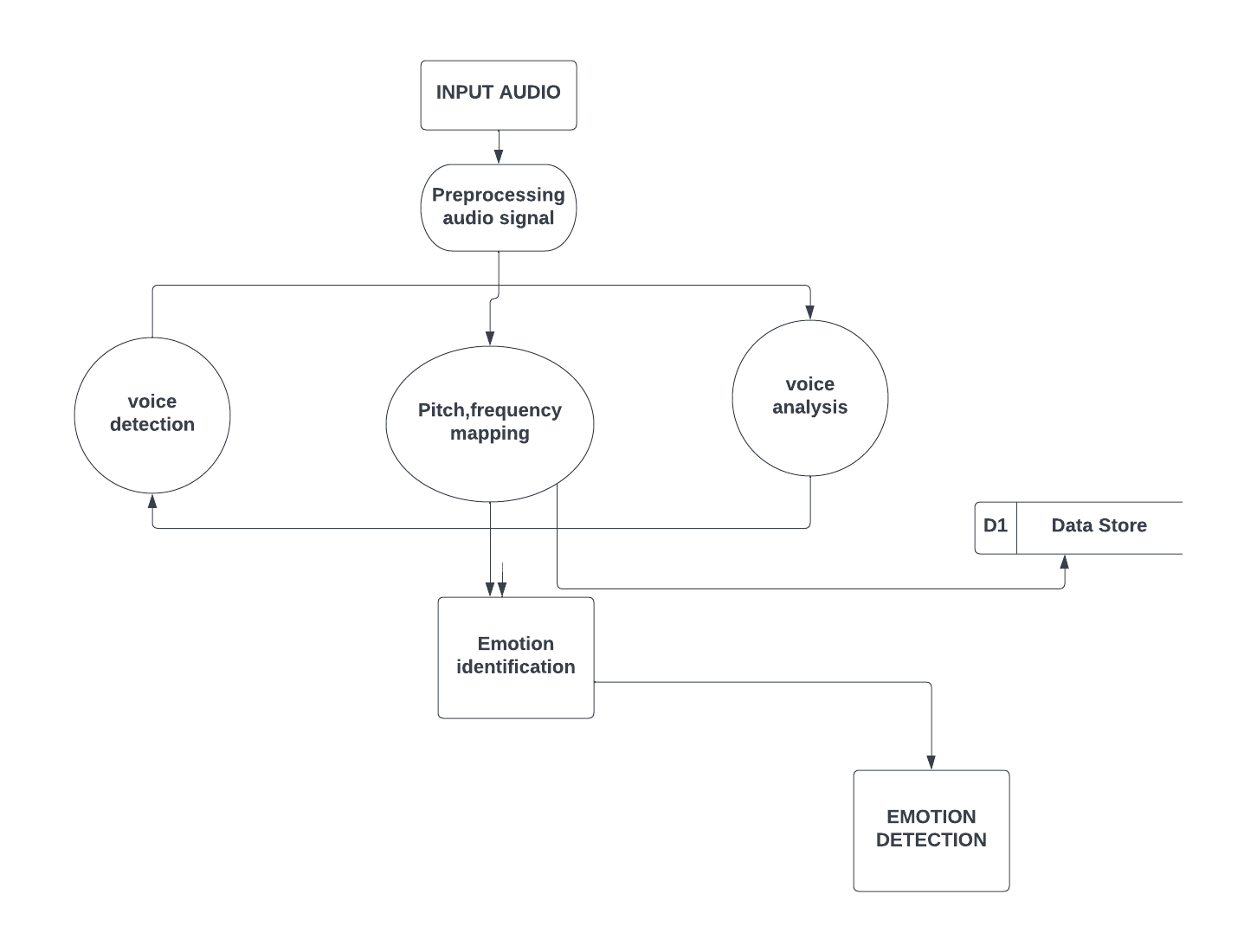
**4.1.3 SEQUENCE DIAGRAM**

A sequence diagram in Unified Modelling Language (UML) is one variety of interaction diagram that suggests how methods operate with one other and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are quite often referred to as event-hint diagrams, event situations, and timing diagrams. A sequence diagram suggests, as parallel vertical traces (lifelines), special systems or objects that are residing at the same time, and, as horizontal arrows, the messages exchanged between them, within the order the place they occur.



**4.1.4 DFD Diagram:**

A data flow diagram or bubble chart (DFD) is a graphical representation of the "flow" of data. through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design). A DFD shows what kinds of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel (which is shown on a flowchart).



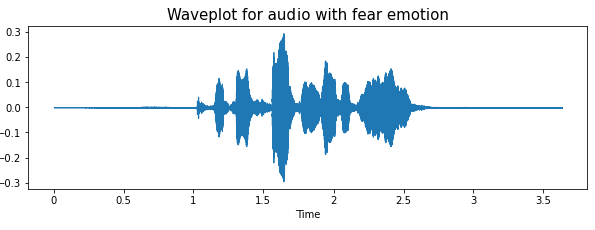
**CHAPTER -5**

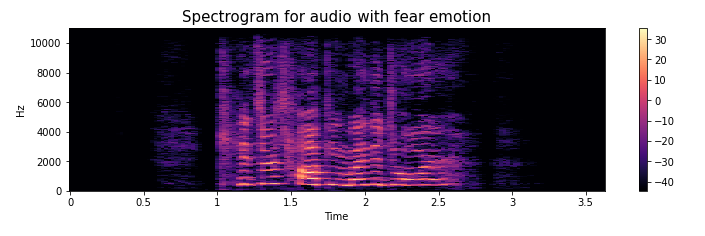
**SYSTEM IMPLEMENTATION**

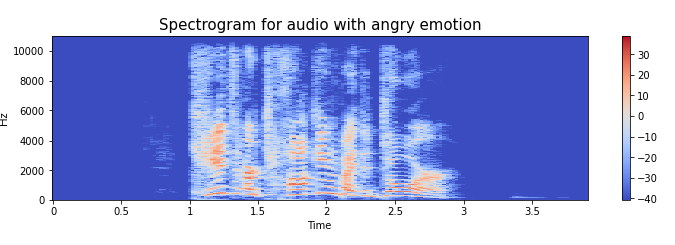
* 1. Speech Emotion Recognition

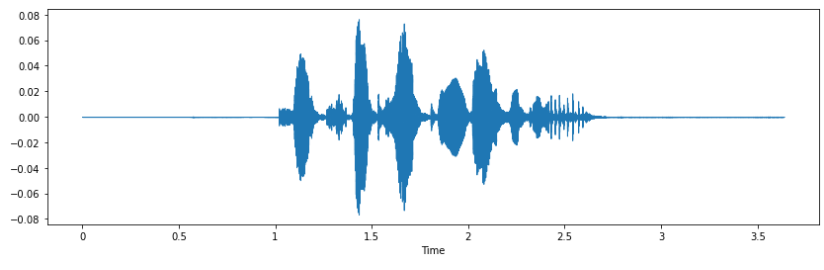
It is done by using Python programming language and run on the environment Kaggle notebook platform, we use the necessary libraries to implement this. To develop this Project we use Python programming language and to implement this project we use the models like SVM, CNN, RNN and LSTM and also collecting different types of datasets like toronto-emotional-speech-set-tess, ravdees-emotional-speech-audios.

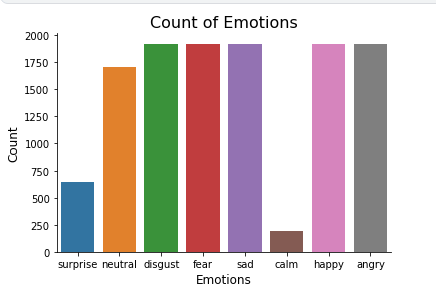
* + 1. Output

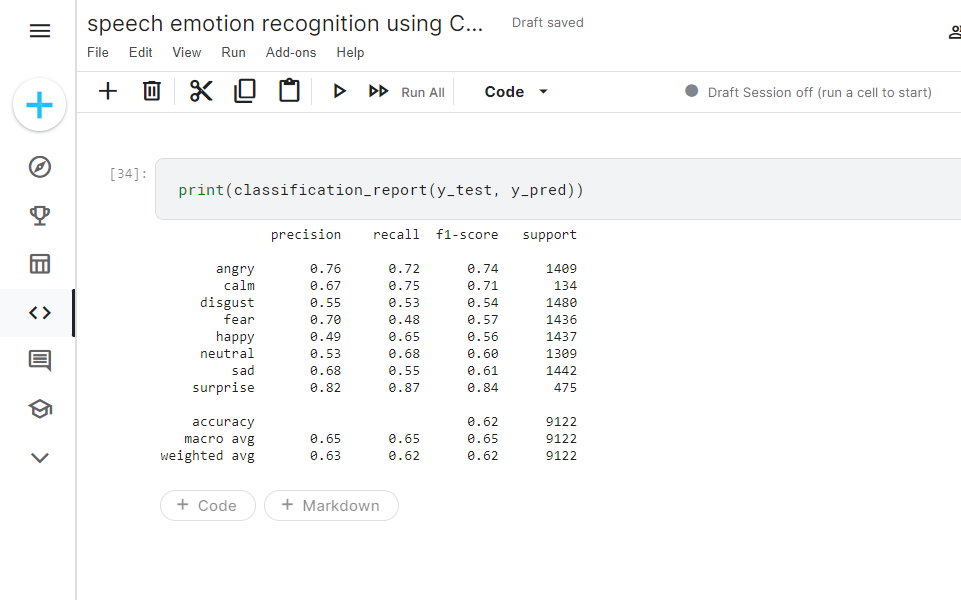


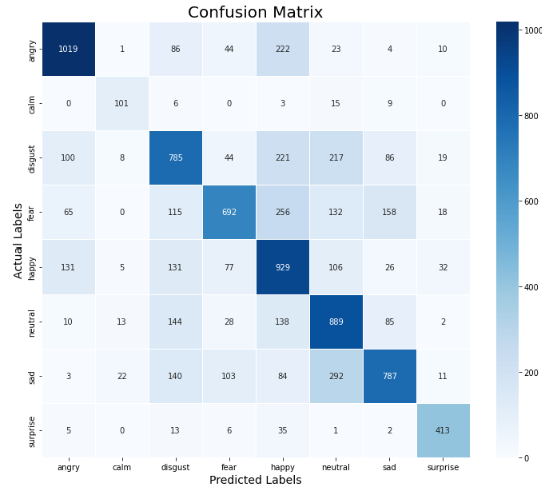


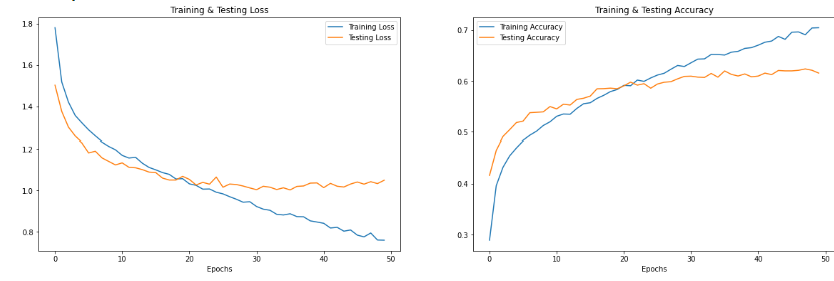












**CHAPTER 6**

**SOURCE CODE**

**Import Modules:**

**import numpy as np # linear algebra**

**import pandas as pd # data processing, CSV file I/O (e.g. pd.read\_csv)**

**# Input data files are available in the read-only "../input/" directory**

**# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory**

**import os**

**for dirname, \_, filenames in os.walk('/kaggle/input'):**

**for filename in filenames:**

**print(os.path.join(dirname, filename))**

import pandas as pd

import numpy as np

import os

import sys

# librosa is a Python library for analyzing audio and music. It can be used to extract the data from the audio files we will see it later.

import librosa

import librosa.display

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler, OneHotEncoder

from sklearn.metrics import confusion\_matrix, classification\_report

from sklearn.model\_selection import train\_test\_split

# to play the audio files

from IPython.display import Audio

import keras

from keras.callbacks import ReduceLROnPlateau

from keras.models import Sequential

from keras.layers import Dense, Conv1D, MaxPooling1D, Flatten, Dropout, BatchNormalization

#from keras.utils import np\_utils, to\_categorical

from tensorflow.keras.utils import to\_categorical

from keras.callbacks import ModelCheckpoint

import warnings

if not sys.warnoptions:

warnings.simplefilter("ignore")

warnings.filterwarnings("ignore", category=DeprecationWarning)

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 for each actor.

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', 7:'disgust', 8:'surprise'}, inplace=True)

Ravdess\_df.head()

crema\_directory\_list = os.listdir(Crema)

file\_emotion = []

file\_path = []

for file in crema\_directory\_list:

# storing file paths

file\_path.append(Crema + file)

# storing file emotions

part=file.split('\_')

if part[2] == 'SAD':

file\_emotion.append('sad')

elif part[2] == 'ANG':

file\_emotion.append('angry')

elif part[2] == 'DIS':

file\_emotion.append('disgust')

elif part[2] == 'FEA':

file\_emotion.append('fear')

elif part[2] == 'HAP':

file\_emotion.append('happy')

elif part[2] == 'NEU':

file\_emotion.append('neutral')

else:

file\_emotion.append('Unknown')

# 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'])

Crema\_df = pd.concat([emotion\_df, path\_df], axis=1)

Crema\_df.head()

tess\_directory\_list = os.listdir(Tess)

file\_emotion = []

file\_path = []

for dir in tess\_directory\_list:

directories = os.listdir(Tess + dir)

for file in directories:

part = file.split('.')[0]

part = part.split('\_')[2]

if part=='ps':

file\_emotion.append('surprise')

else:

file\_emotion.append(part)

file\_path.append(Tess + 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'])

Tess\_df = pd.concat([emotion\_df, path\_df], axis=1)

Tess\_df.head()

savee\_directory\_list = os.listdir(Savee)

file\_emotion = []

file\_path = []

for file in savee\_directory\_list:

file\_path.append(Savee + file)

part = file.split('\_')[1]

ele = part[:-6]

if ele=='a':

file\_emotion.append('angry')

elif ele=='d':

file\_emotion.append('disgust')

elif ele=='f':

file\_emotion.append('fear')

elif ele=='h':

file\_emotion.append('happy')

elif ele=='n':

file\_emotion.append('neutral')

elif ele=='sa':

file\_emotion.append('sad')

else:

file\_emotion.append('surprise')

# 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'])

Savee\_df = pd.concat([emotion\_df, path\_df], axis=1)

Savee\_df.head()

data\_path = pd.concat([Ravdess\_df, Crema\_df, Tess\_df, Savee\_df], axis = 0)

data\_path.to\_csv("data\_path.csv",index=False)

data\_path.head()

plt.title('Count of Emotions', size=16)

sns.countplot(data\_path.Emotions)

plt.ylabel('Count', size=12)

plt.xlabel('Emotions', size=12)

sns.despine(top=True, right=True, left=False, bottom=False)

plt.show()

def create\_waveshow(data, sr, e):

plt.figure(figsize=(10, 3))

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):

# stft function converts the data into short term fourier transform

X = librosa.stft(data)

Xdb = librosa.amplitude\_to\_db(abs(X))

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()

emotion='fear'

path = np.array(data\_path.Path[data\_path.Emotions==emotion])[1]

data, sampling\_rate = librosa.load(path)

create\_waveshow(data, sampling\_rate, emotion)

create\_spectrogram(data, sampling\_rate, emotion)

Audio(path)

emotion='angry'

path = np.array(data\_path.Path[data\_path.Emotions==emotion])[1]

data, sampling\_rate = librosa.load(path)

create\_waveshow(data, sampling\_rate, emotion)

create\_spectrogram(data, sampling\_rate, emotion)

Audio(path)

emotion='sad'

path = np.array(data\_path.Path[data\_path.Emotions==emotion])[1]

data, sampling\_rate = librosa.load(path)

create\_waveshow(data, sampling\_rate, emotion)

create\_spectrogram(data, sampling\_rate, emotion)

Audio(path)

emotion='happy'

path = np.array(data\_path.Path[data\_path.Emotions==emotion])[1]

data, sampling\_rate = librosa.load(path)

create\_waveshow(data, sampling\_rate, emotion)

create\_spectrogram(data, sampling\_rate, emotion)

Audio(path)

def noise(data):

noise\_amp = 0.035\*np.random.uniform()\*np.amax(data)

data = data + noise\_amp\*np.random.normal(size=data.shape[0])

return data

def stretch(data, rate=0.8):

return librosa.effects.time\_stretch(data, rate)

def shift(data):

shift\_range = int(np.random.uniform(low=-5, high = 5)\*1000)

return np.roll(data, shift\_range)

def pitch(data, sampling\_rate, pitch\_factor=0.7):

return librosa.effects.pitch\_shift(data, sampling\_rate, pitch\_factor)

# taking any example and checking for techniques.

path = np.array(data\_path.Path)[1]

data, sample\_rate = librosa.load(path)

plt.figure(figsize=(14,4))

librosa.display.waveshow(y=data, sr=sample\_rate)

Audio(path)

x = noise(data)

plt.figure(figsize=(14,4))

librosa.display.waveshow(y=x, sr=sample\_rate)

Audio(x, rate=sample\_rate)

x = stretch(data)

plt.figure(figsize=(14,4))

librosa.display.waveshow(y=x, sr=sample\_rate)

Audio(x, rate=sample\_rate)

x = shift(data)

plt.figure(figsize=(14,4))

librosa.display.waveshow(y=x, sr=sample\_rate)

Audio(x, rate=sample\_rate)

x = pitch(data, sample\_rate)

plt.figure(figsize=(14,4))

librosa.display.waveshow(y=x, sr=sample\_rate)

Audio(x, rate=sample\_rate)

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

# 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

def get\_features(path):

# duration and offset are used to take care of the no audio in start and the ending of each audio files as seen above.

data, sample\_rate = librosa.load(path, duration=2.5, offset=0.6)

# without augmentation

res1 = extract\_features(data)

result = np.array(res1)

# data with noise

noise\_data = noise(data)

res2 = extract\_features(noise\_data)

result = np.vstack((result, res2)) # stacking vertically

# data with stretching and pitching

new\_data = stretch(data)

data\_stretch\_pitch = pitch(new\_data, sample\_rate)

res3 = extract\_features(data\_stretch\_pitch)

result = np.vstack((result, res3)) # stacking vertically

return result

X, Y = [], []

for path, emotion in zip(data\_path.Path, data\_path.Emotions):

feature = get\_features(path)

for ele in feature:

X.append(ele)

# appending emotion 3 times as we have made 3 augmentation techniques on each audio file.

Y.append(emotion)

len(X), len(Y), data\_path.Path.shape

Features = pd.DataFrame(X)

Features['labels'] = Y

Features.to\_csv('features.csv', index=False)

Features.head()

X = Features.iloc[: ,:-1].values

Y = Features['labels'].values

# As this is a multiclass classification problem onehotencoding our Y.

encoder = OneHotEncoder()

Y = encoder.fit\_transform(np.array(Y).reshape(-1,1)).toarray()

# splitting data

x\_train, x\_test, y\_train, y\_test = train\_test\_split(X, Y, random\_state=0, shuffle=True)

x\_train.shape, y\_train.shape, x\_test.shape, y\_test.shape

# scaling our data with sklearn's Standard scaler

scaler = StandardScaler()

x\_train = scaler.fit\_transform(x\_train)

x\_test = scaler.transform(x\_test)

x\_train.shape, y\_train.shape, x\_test.shape, y\_test.shape

# making our data compatible to model.

x\_train = np.expand\_dims(x\_train, axis=2)

x\_test = np.expand\_dims(x\_test, axis=2)

x\_train.shape, y\_train.shape, x\_test.shape, y\_test.shape

allow\_soft\_placement=True

model=Sequential()

model.add(Conv1D(256, kernel\_size=5, strides=1, padding='same', activation='relu', input\_shape=(x\_train.shape[1], 1)))

model.add(MaxPooling1D(pool\_size=5, strides = 2, padding = 'same'))

model.add(Conv1D(256, kernel\_size=5, strides=1, padding='same', activation='relu'))

model.add(MaxPooling1D(pool\_size=5, strides = 2, padding = 'same'))

model.add(Conv1D(128, kernel\_size=5, strides=1, padding='same', activation='relu'))

model.add(MaxPooling1D(pool\_size=5, strides = 2, padding = 'same'))

model.add(Dropout(0.2))

model.add(Conv1D(64, kernel\_size=5, strides=1, padding='same', activation='relu'))

model.add(MaxPooling1D(pool\_size=5, strides = 2, padding = 'same'))

model.add(Flatten())

model.add(Dense(units=32, activation='relu'))

model.add(Dropout(0.3))

model.add(Dense(units=8, activation='softmax'))

model.compile(optimizer = 'adam' , loss = 'categorical\_crossentropy' , metrics = ['accuracy'])

model.summary()

from keras.callbacks import ReduceLROnPlateau

rlrp = ReduceLROnPlateau(monitor='val loss', factor=0.2, verbose=1, patience=2, min\_lr=0.0000001)

history=model.fit(x\_train, y\_train, batch\_size=162, epochs=50, validation\_data=(x\_test, y\_test), callbacks=[rlrp])

print("Accuracy of our model on test data : " , model.evaluate(x\_test,y\_test)[1]\*100 , "%")

epochs = [i for i in range(50)]

fig , ax = plt.subplots(1,2)

train\_acc = history.history['accuracy']

train\_loss = history.history['loss']

test\_acc = history.history['val\_accuracy']

test\_loss = history.history['val\_loss']

fig.set\_size\_inches(20,6)

ax[0].plot(epochs , train\_loss , label = 'Training Loss')

ax[0].plot(epochs , test\_loss , label = 'Testing Loss')

ax[0].set\_title('Training & Testing Loss')

ax[0].legend()

ax[0].set\_xlabel("Epochs")

ax[1].plot(epochs , train\_acc , label = 'Training Accuracy')

ax[1].plot(epochs , test\_acc , label = 'Testing Accuracy')

ax[1].set\_title('Training & Testing Accuracy')

ax[1].legend()

ax[1].set\_xlabel("Epochs")

plt.show()

pred\_test = model.predict(x\_test)

y\_pred = encoder.inverse\_transform(pred\_test)

y\_test = encoder.inverse\_transform(y\_test)

df = pd.DataFrame(columns=['Predicted Labels', 'Actual Labels'])

df['Predicted Labels'] = y\_pred.flatten()

df['Actual Labels'] = y\_test.flatten()

df.head(10)

cm = confusion\_matrix(y\_test, y\_pred)

plt.figure(figsize = (12, 10))

cm = pd.DataFrame(cm , index = [i for i in encoder.categories\_] , columns = [i for i in encoder.categories\_])

sns.heatmap(cm, linecolor='white', cmap='Blues', linewidth=1, annot=True, fmt='')

plt.title('Confusion Matrix', size=20)

plt.xlabel('Predicted Labels', size=14)

plt.ylabel('Actual Labels', size=14)

plt.show()

print(classification\_report(y\_test, y\_pred))

**CHAPTER 7**

**SYSTEM TESTING**

INTRODUCTION

The cause of testing is to detect mistakes. Making an attempt out is the technique of looking for to realize each viable fault or weakness in a piece product. It presents a method to determine the performance of add-ons, sub-assemblies, assemblies and/or a completed product. It is the method of ex excising g program with the intent of constructing certain that the application procedure meets its necessities and client expectations and does no longer fail in an unacceptable process. There are rather plenty of forms of scan. Each experiment sort addresses a special trying out requirement.

**Unit testing:**

Unit checking out involves the design of scan circumstances that validate that the Internal application good judgment is functioning safely, and that program inputs produce legitimate outputs. All decision branches and interior code float must be validated. It's the checking out of character application items of the application. It is achieved after the completion of a person unit earlier than integration. It is a structural checking out, that relies on competencies of its construction and is invasive. Unit exams participate in common exams at component level and scan a distinct business approach, utility, and/or process configuration. Unit assessments be certain that every specified course of an industry method performs appropriately to the documented requisites and involves clearly outlined inputs and anticipated results.

**Integration testing:**

Integration Testing are designed to scan built-in program accessories to determine within the occasion that they evidently run as one software. Trying out is occasion driven and is more concerned with the fundamental final result of screens or fields. Integration assessments reveal that despite the fact that the accessories had been for my part pleasure, as proven through effectively unit checking out, the combo of accessories is correct and regular. Integration checking out is chiefly aimed at exposing the issues that come up from the performance of different components.

**Functional testing:**

Functional Testing checks provide systematic demonstrations that capabilities established are to be had as particular by means of the business and technical specifications, method documentation, and consumer manuals. Functional testing is working on below mentioned data:

Legitimate input: identified lessons of legitimate input ought to be accredited.

Noisy voice : recognized lessons of unacceptable audio signals must be rejected.

Capabilities : recognized features ought to be exercised.

Output : recognize the audios and detect the human emotion

**Systems/Procedures**:

performance of the system here was invoked Individual and team work of useful checks is fascinated by specifications, key capabilities, or special scan instances. Moreover, systematic concerning method flows; data fields, predefined processes, and successive strategies have to be regarded for trying out. Before useful trying out is whole, extra checks are recognized and detect the human emotion

**System testing:**

This scheme is difficult to ensure so as to the whole included agenda process meets principles. It exams a pattern to make sure identified and predictable outcome. An illustration of procedure testing is the configuration oriented approach integration scan. System testing is based on approach descriptions and flows, emphasizing pre-driven system links and integration aspects.

**VIII. CONCLUSION**

In this Project, the concept implemented was emotion recognition using MFCC approach using Conventional Neural Network. Support vector machine is used to classify the female and male datasets based on pitch analysis. MFCC approach for emotion recognition from speech is a stand-alone approach which does not require calculation of any other acoustic features and produce more accurate results. Hence proved that the Conventional Neural Network recognize emotions more accurately than the Recurrent neural networks (RNN) and LSTM

The below mentioned table describes the difference in accuracy of emotion recognition using both the RNN, SVM, MLP and CNN Network. CNN classifies the Emotion more accurately than the RNN network.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | precision | | | recall | | |
| Class | RNN | LSTM | CNN | RNN | LSTM | CNN |
| anger | 0.69 | 0.70 | 0.87 | 0.43 | 0.67 | 0.62 |
| Fear | 0.52 | 0.67 | 1.00 | 0.33 | 0.67 | 0.67 |
| Happy | 0.18 | 0.50 | 0.60 | 0.10 | 0.10 | 0.30 |
| Neutral | 0.67 | 0.62 | 0.61 | 0.74 | 0.95 | 1.00 |
| Sad | 0.27 | 0.50 | 0.50 | 0.57 | 0.14 | 0.29 |
| Surprise | 0.11 | 0.20 | 0.22 | 0.17 | 0.33 | 0.33 |
| OVERAL ACCURACY | 50% 57% 62% | | | | | |

|  |  |  |
| --- | --- | --- |
| Dataset  (Female,Male) | Algorithm | Accuracy |
| TORONTO | RNN | 50% |
| RAVDESS,CREMAD,TORNOTO,SAVEE | LSTM | 57% |
| CREMAD | CNN | 62% |

IX- REFERENCES

1. Assistant Professor/Lecturer (on Deputation), Department of Computer Science and Engineering Annamalai University, Annamalainagar, Tamil Nadu, India
2. Computer Science Tripos Part II Gonville & Caius College
3. AasthaJoshi “Speech Emotion Recognition Using Combined Features & SVM Algorithm”, National Conference on August 2013.
4. AnkurSapra, Nikhil Panwar, SohanPanwar “Emotion Recognition from Speech”, International Journal of Emerging Technology and Advanced Engineering, Volume 3, Issue 2, pp. 341-345, February 2013.
5. BjörnSchuller, Manfred Lang, Gerhard Rigoll “Automatic Emotion Recognition by the Speech Signal”, National Journal on 2013, Volume 3, Issue 2, pp. 342-347.