CHAPTER 1

INTRODUCTION

1.1 PROBLEM DEFINITION:

Speech emotion recognition (SER) is the task of recognition of various emotions linked with humans, it comes under machine learning project. speech recognition is one of the tremendously growing system in which it works how to recognize speech signals. When a user gives an input as speech, that is done by speech to text conversion, this is also well known as speech recognition it is nothing but identify the speech given by user into written texts. Now let's talk about emotions in speech, it may be regarded as communication system featuring in several parts like expressions or portrayal of the emotion by the speaker. Facial expressions are the main way to convey the emotions. Various techniques have been developed to discover the emotions such as signal processing, machine learning, neural networks, computer vision. Speech emotion recognition is now a hot topic in the world which makes life easier and hence making a world better place to live, as it has huge applications in all the fields like in BPO center and call center to ascertain the emotion which is used to identify that how much the customer is satisfied by the product. At present, speech emotion recognition was an emerging crossing field of artificial intelligence and artificial psychology. Speech emotion processing and recognition system is generally composed of three parts, which are speech signal acquisition, feature extraction, and emotion recognition. There are seven emotions spotted through few datasets: anger, disgust, fear, happiness, sad, surprise, and contempt.

1.2 OBJECTIVES:

- The main objective of speech emotion recognition is to enhance the human-machine interface, to adapt the system response upon detecting annoyance or frustration in the speaker's voice. The main objective of speech emotion recognition is to enhance the human-machine interface
- to adapt the system response upon detecting annoyance or frustration in the speaker's voice.
- Create a tool to help people learn to speak English correctly in an effective way.
- Simple and intuitive to use

1.3 LITERATURE SURVEY

- The idea of choosing this project speech emotion recognition was carried out with the help of few articles and videos mentioned below:

1. Babak Basharirad, and Mohammadreza Moradhaseli.

   "Speech emotion recognition methods"

2. Mr. Bhola Kumar, Mr. Pappu Sahar.

   "Speech emotion recognition and its techniques"

3. A Pramod Reddy and V. Vijayarajan.

   "Extraction of emotions from speech"

1.4 PROPOSED SYSTEM

*Emotion sensing systems depends on accuracy of the sensors in this we are going to improve the accuracy.

* We will modify the distractions in the background

when his speech is recognized.

*There are many words that sounds similar like There,Their etc. but they mean in a different way.

* Usually the speech pace varies, this program recognizes it and modifies it.

1.5 EXISTING SYSTEM

Speech sample is first passed through a gender reference database which maintain for recognition of gender before it's getting into the process, Followed by pitch and voice as a feature,human voice sample was broken into frames. For each farme MFCC(Mel Frequency Cepstral Coefficient) it is the main feature for emotion recognition, The database contains of emotionsthat are Sad, Anger, Happy…etc.

CHAPTER 2

FUNDAMENTALS OF PYTHON

2.1 INTRODUCTION TO PYTHON:

Python is one of the well-knownfast-growing programming language.it was created by Guido van Rossum in the year 1991.it is high -level object -oriented programming language and also called as general-purpose programming language.it is almost used in most of the fields few are web development, software development, Game development, Artificial intelligence and Machine learning.

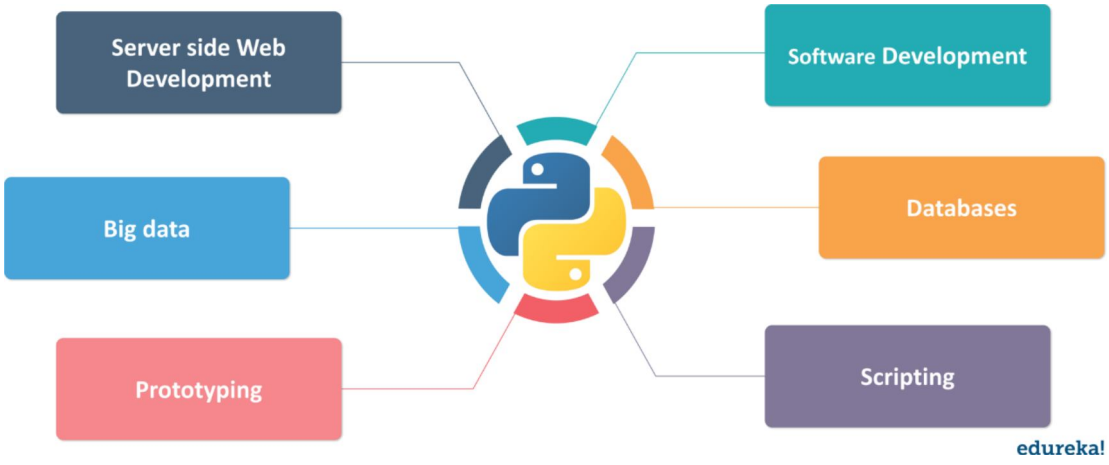


Figure 2.1.1:Python used in various stream

The above figure (2.1) specifies that in how many various ways python programming language is used.

2.2 ADVANTAGES OF PYTHON:

Few advantages of python programming language are listed

→Data science Library

→Embeddable

→Extensible

→Improved Productivity

→Portable

→Free and Open Source

→Dynamically Typed

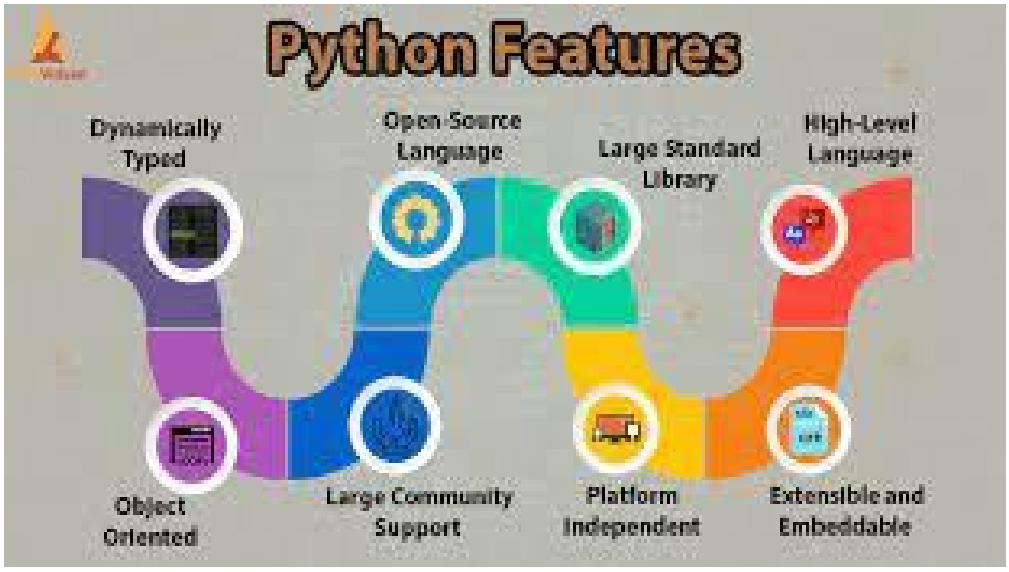


FIG NO: 2.2.2 PYTHON FEATURES

2.3 CHARECTERISTICS OF PYTHON:

→It supports both functional and structural programming methods as well as OOP.

→It supports Automatic garbage collection.

→It can be easily integrated with other programming language like C,C++, COM, ActiveX, COBRA and Java.

2.4 LIBRERIES USED IN PYTHON:

NumPy: NumPy is a python library, which adds the support for the huge, multi-dimensional arrays and matrices and huge collection of high – level mathematical functions to operate on these arrays.

Pandas: pandas is a python library, it is widely used in the field of data science and machine learning. It is built upon top of another library NumPy, which provides support for multi - dimensional arrays.

Matplotlib: Matplotlib is a python library its numerical mathematics extension of NumPy.it is used to plot the different kind of graphs. it provides an object -oriented API for embedding plots into applications using GUI.

Librosa: Librosa is a python library used for audio and video visualizations.

2.5 HARDWARE REQUIREMENTS:

For Python

RAM: 8GB and Above

Hard-disk drive:250GB

Hard-disk drive:500GB

Processor:Core-i5,

Core-i7.

CHAPTER 3

DATASETS

3.1 RAVDEES:

Ryerson Audio-Visual Database of Emotional Speech and Song(RAVDEES) contains 7,356 files. It is the database of 24 professional actors it includes both male and female voices in a neutral North American accent. Speech includes 7 different Emotions Calm, happy, Sad, angry, fearful, surprise and disgust expressions, and song contains of 5 emotions like calm, happy, sad, angry and fearful. Each emotion is identified at two levels of emotional intensity normal and strong, with an additional neutral emotion.

3.2 TESS:

Toronto emotional speech set (TESS) contains 2800 files it is created in 2010.it is created in English language.it identifies seven emotions like happy, sad, angry, surprise, fear, disgust and neutral.it has recorded voice of two level of actors are age groups of old(64- year old) and young(24- year old).

3.3 CREMA-D:

CREMA-D is an audio -visual dataset for speech emotion recognition. This dataset consists of vocal emotional expressions. In this dataset there are six different emotions like happy, sad, anger, neutral, fear and disgust. These consists of 7,442 audio clips of various 91 actors of two genders. In these 91 audio clips, 48 clips are from male actors and remaining 43 clips are from female actress. And this are further classified into age groups of actors like two age groups old and young, where old are from the age around 74 years and young are from the age around 20 years. They noticed that there are four different levels of emotions like Low, Medium, High, and unspecified.

3.4 ALGORITHM:

STEP 1: CREATE A DATASET

STEP 2: IMPORT THE LIBRARIES (like LIBROSA,SKLearn..etc)

STEP 3:IMPORT THE DATASET FILES TO THE CODE BY GIVING THE PATH TO IT

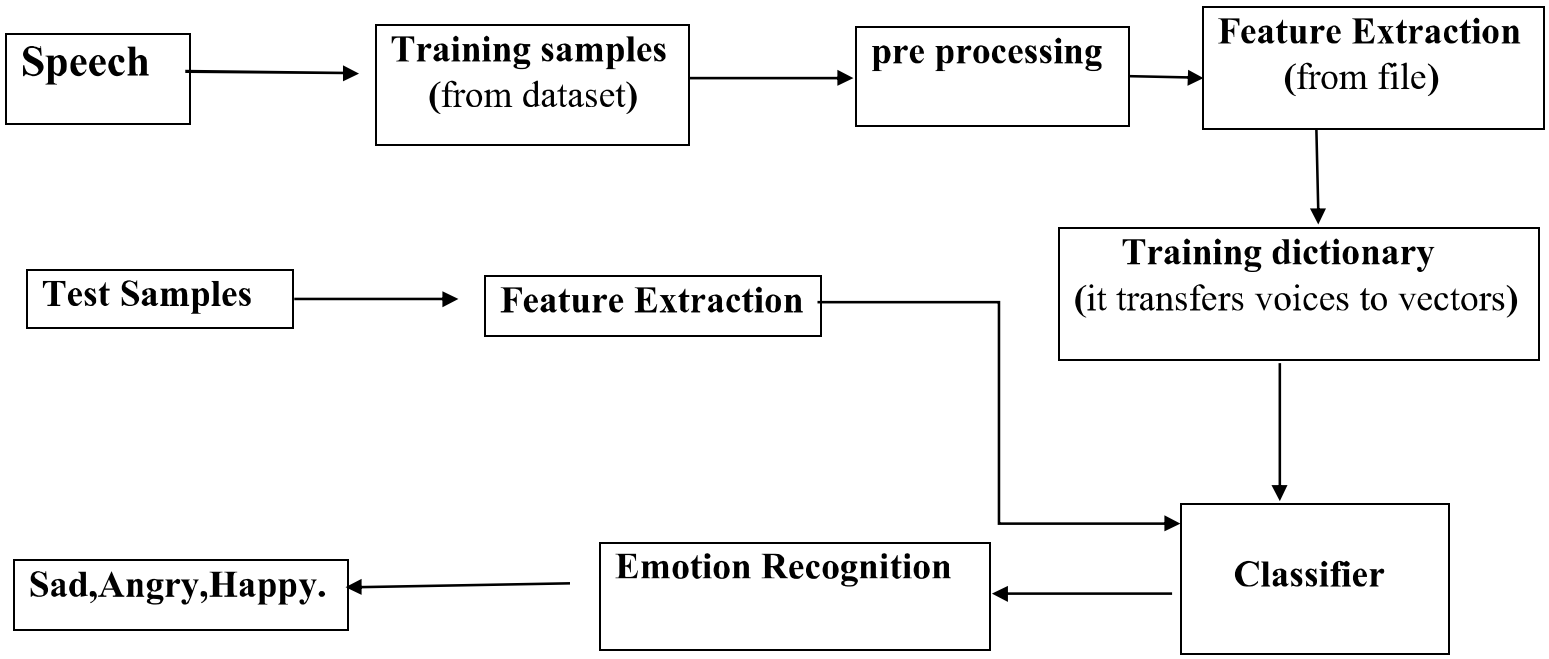
STEP 4:BY THE PATH DIVIDE THE AUDIO FILES WITH DATAFRAMES

STEP 5:COUNT THE EMOTIONS BY PLOTTING THE GRAPH

STEP 6:COVERT AUDIOFILE WAVEFORM TO THE NUMBERS WHICH SYSTEM UNDERSTANDS

STEP 7:TRAIN THE MODEL AND FIND THE ACCURACY

3.5 SYSTEM ARCHITECTURE:



CHAPTER -4

Implementations and results

4.1 PSUEDOCODE:

```
import pandas as pd
import numpy as np

import os
import sys

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

from IPython.display import Audio

import tensorflow as tf
from tensorflow.keras.callbacks import ReduceLROnPlateau
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, LSTM, Flatten, Dropout, BatchNormalization
from tensorflow.keras.callbacks import ModelCheckpoint
```

```
import warnings
if not sys.warnoptions:
warnings.simplefilter("ignore")
warnings.filterwarnings("ignore", category=DeprecationWarning)
Crema = "./audio_files/AudioWAV/"
import os
import pandas as pd

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')
elifpart[2] == 'ANG':
file_emotion.append('angry')
elifpart[2] == 'DIS':
file_emotion.append('disgust')
elifpart[2] == 'FEA':
file_emotion.append('fear')
elifpart[2] == 'HAP':
file_emotion.append('happy')
elifpart[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()
plt.title('Count of Emotions', size=16)
sns.countplot(Crema_df.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_waveplot(data, sr, e):
plt.figure(figsize=(10, 3))
plt.title('Waveplot for {} emotion'.format(e), size=15)
librosa.display.waveplot(data, sr=sr)
plt.show()

def create_spectrogram(data, sr, e):
    X = librosa.stft(data)
Xdb = librosa.amplitude_to_db(abs(X))
plt.figure(figsize=(12, 3))
plt.title('Spectrogram for {} emotion'.format(e), size=15)
librosa.display.specshow(Xdb, sr=sr, x_axis='time', y_axis='hz')
plt.colorbar()
emotion='disgust'
path = np.array(Crema_df.Path[Crema_df.Emotions==emotion])[0]
```

```
print(path)
data, sampling_rate = librosa.load(path)
create_waveplot(data, sampling_rate, emotion)
create_spectrogram(data, sampling_rate, emotion)
Audio(path)
labels = {'disgust':0,'happy':1,'sad':2,'neutral':3,'fear':4,'angry':5}
Crema_df.replace({'Emotions':labels},inplace=True)
num_mfcc=15#10 to 20
n_fft=2048
hop_length=550#450 to 650
SAMPLE_RATE = 20000#upto 22000
data = {
        "labels": [],
        "mfcc": []
    }
for i in range(7442):
    data['labels'].append(Crema_df.iloc[i,0])
    signal, sample_rate = librosa.load(Crema_df.iloc[i,1], sr=SAMPLE_RATE)
mfcc = librosa.feature.mfcc(signal, sample_rate, n_mfcc=13, n_fft=2048, hop_length=512)
mfcc = mfcc.T
    data["mfcc"].append(np.asarray(mfcc))
    if i%500==0:
        print(i)
X = np.asarray(data['mfcc'])
y = np.asarray(data["labels"])
X = tf.keras.preprocessing.sequence.pad_sequences(X)
X.shape
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1)
X_train, X_validation, y_train, y_validation = train_test_split(X_train, y_train, test_size=0.2)
print(X_train.shape,y_train.shape,X_validation.shape,y_validation.shape,X_test.shape,y_test.
shape)
```

Model

```
def build_model(input_shape):
    model = tf.keras.Sequential()

model.add(LSTM(120, input_shape=input_shape, return_sequences=True))
model.add(LSTM(64))

model.add(Dense(64, activation='sigmoid'))
model.add(Dropout(0.3))

model.add(Dense(6, activation='softmax'))

    return model
```

create network

```
input_shape = (None,13)
model = build_model(input_shape)

# compile model
optimiser = tf.keras.optimizers.Adam(learning_rate=0.001)
model.compile(optimizer=optimiser,
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

model.summary()
```

[3:23 PM, 1/31/2022] Dinesh Nhce: history = model.fit(X_train, y_train, validation_data=(X_validation, y_validation), batch_size=32, epochs=18)

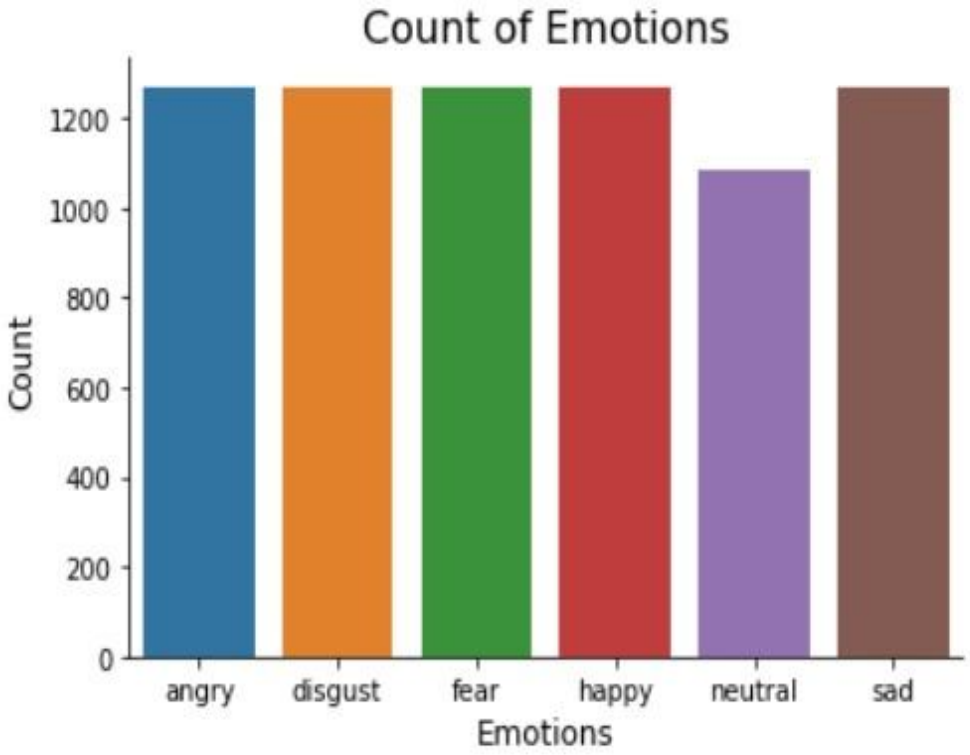
[3:23 PM, 1/31/2022] Dinesh Nhce: test_loss, test_acc = model.evaluate(X_test, y_test, verbose=0)

```
print("Test Accuracy: ",test_acc*100)
```

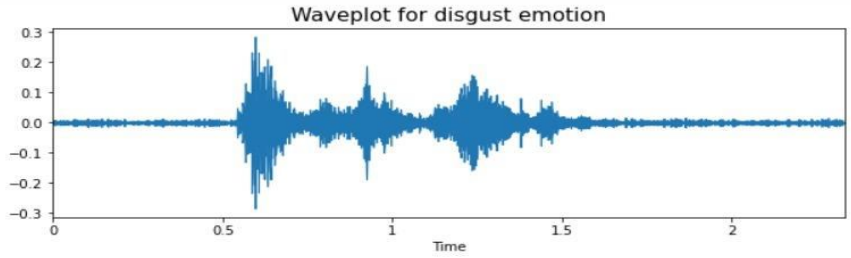
4.2 Results:

	Emotions	Path
0	angry	E:/Mini Project/archive/AudioWAV/1001_DFA_ANG_...
1	disgust	E:/Mini Project/archive/AudioWAV/1001_DFA_DIS_...
2	fear	E:/Mini Project/archive/AudioWAV/1001_DFA_FEA_...
3	happy	E:/Mini Project/archive/AudioWAV/1001_DFA_HAP_...
4	neutral	E:/Mini Project/archive/AudioWAV/1001_DFA_NEU_...

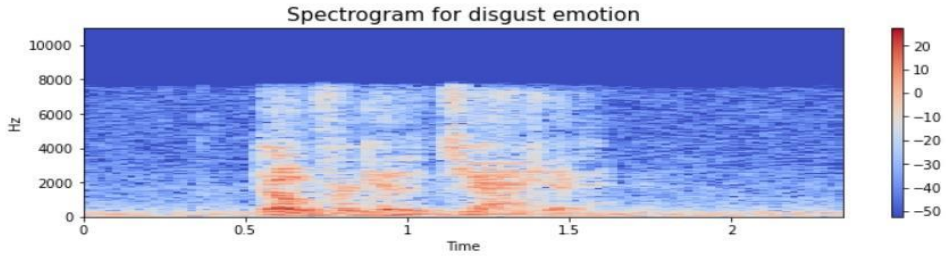
4.2.1 SELECTED AUDIO FILES



4.2.2 COUNT OF EMOTIONS



Out[6]:




```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_6 (LSTM)                (None, None, 120)         64320
_________________________________________________________________
lstm_7 (LSTM)                (None, 64)                47360
_________________________________________________________________
dense_5 (Dense)              (None, 64)                4160
_________________________________________________________________
dropout_2 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_6 (Dense)              (None, 6)                 390
=================================================================
Total params: 116,230
Trainable params: 116,230
Non-trainable params: 0
_________________________________________________________________
```

FIG NO:4.2.3 TRANSFOMING THE WAVE FORM TO VECTOR

```
168/168 [==============================] - 78s 434ms/step - loss: 1.6360 - accuracy: 0.3102 - val_loss: 1.4959 - val_accuracy:
0.3746
Epoch 2/18
168/168 [==============================] - 68s 407ms/step - loss: 1.5307 - accuracy: 0.3588 - val_loss: 1.4884 - val_accuracy:
0.3701
Epoch 3/18
168/168 [==============================] - 70s 414ms/step - loss: 1.4964 - accuracy: 0.3737 - val_loss: 1.4707 - val_accuracy:
0.3791
Epoch 4/18
168/168 [==============================] - 71s 421ms/step - loss: 1.4710 - accuracy: 0.3831 - val_loss: 1.4431 - val_accuracy:
0.3933
Epoch 5/18
168/168 [==============================] - 70s 415ms/step - loss: 1.4435 - accuracy: 0.4023 - val_loss: 1.4179 - val_accuracy:
0.4015
Epoch 6/18
168/168 [==============================] - 71s 422ms/step - loss: 1.4267 - accuracy: 0.4083 - val_loss: 1.4379 - val_accuracy:
0.4007
Epoch 7/18
168/168 [==============================] - 72s 430ms/step - loss: 1.4107 - accuracy: 0.4221 - val_loss: 1.4119 - val_accuracy:
0.4269
Epoch 8/18
168/168 [==============================] - 73s 434ms/step - loss: 1.4073 - accuracy: 0.4271 - val_loss: 1.3749 - val_accuracy:
0.4224
Epoch 9/18
168/168 [==============================] - 73s 437ms/step - loss: 1.3885 - accuracy: 0.4333 - val_loss: 1.3916 - val_accuracy:
0.4194
Epoch 10/18
168/168 [==============================] - 73s 434ms/step - loss: 1.3715 - accuracy: 0.4374 - val_loss: 1.3585 - val_accuracy:
0.4194
Epoch 11/18
168/168 [==============================] - 73s 438ms/step - loss: 1.3620 - accuracy: 0.4476 - val_loss: 1.3392 - val_accuracy:
0.4590
Epoch 12/18
168/168 [==============================] - 75s 444ms/step - loss: 1.3444 - accuracy: 0.4555 - val_loss: 1.3218 - val_accuracy:
0.4485
Epoch 13/18
```

FIG NO:4.2.4 TRANSFORMED VECTORS

```
test_loss, test_acc = model.evaluate(X_test, y_test, verbose=0)
print("Test Accuracy: ",test_acc*100)
```

```
Test Accuracy:  54.09395694732666
```

FIG NO: 4.2.5 TEST ACCURACY

CHAPTER-5

CONCLUSION AND FUTURE ENHANCEMENT

The emerging growth and enlargement of artificial intelligence and machine learning has created a new technology i.e, automation. Most of the automated devices works on voice produced by user. Some of the applications of speech emotion recognition are computer aided applications, BPO call centers, used in diagnostic centers.

For future furtherance,proposed system can be modified like how by this project we find emotions through voice commands similarly feelings can also be detected like depression, anxiety and mood swings. This really helps for the therapists if they want to notice the patients.

Therefore, in the future, there would emerge many applications of a speech-based emotion recognition system.

REFERENCES

From Internet:

1) https://data-flair.training/blogs/python-mini-project-speech-emotion-recognition/

2) https://github.com/abhay8463/Speech-Emotion-Recognition-using-ML-and-DL

3) https://www.kaggle.com/ashishsingh226/speech-emotion-recognition-using-lstm

From youtube:

1) https://youtu.be/eHJrZa2LtKw

2) https://youtu.be/p5glHn5Nlo0