Multi-Model Fusion Emotion Recognition Method of Speech Expression Based on Deep Learning

Mr. Bagal Purvesh, Mr. Auti Adarsh Mr. Pisal Shubham Mr. Pawar Shreyash
Under the Guidance of Prof. Palve P.B
Department of Computer Engineering, Bachelor of Engineering
Adsul's Technical Campus,
Chas,Ahmednagar

-

ABSTRACT

As the foundation of human–computer emotional interaction research, emotion recognition affects the development of artificial intelligence technology. At the same time, due to the integration of knowledge of multiple disciplines, the development of emotion recognition research has also led to the rapid development of other disciplines. At present, the research of emotion recognition is mainly concentrated in the field of mono modal emotion recognition such as text, speech, and image. Although unimodal emotion recognition has made many breakthrough achievements, with the passage of time, unimodal emotion recognition has also exposed some problems. It cannot fully describe a certain emotion of the user at the moment, and using multiple modal features to describe a certain emotion together will be more comprehensive and detailed. We humans are well trained because your reading experience recognizes different emotions which make us more reasonable and understandable. But only in the case of a machine, it can easily understand content- based information such as information in text, audio or video, but it is still far behind in accessing the depth of content.

CHAPTER 1

SYNOPSIS

Project Synopsis

Problem Statement: - Based Smart Alert for Drowsy Driver Detection computer system using CNN algorithm.

Abstract:-
In current years, drowsy driver detection is the most necessary procedure to prevent any road accidents, probably worldwide. The aim of this study was to construct a smart alert technique for building intelligent vehicles that can automatically avoid drowsy driver impairment. But drowsiness is a natural phenomenon in the human body that happens due to different factors. Hence, it is required to design a robust alert system to avoid the cause of the mishap. Drowsiness is identified by using vision-based techniques like eyes detection, yawning, and nodding. When it comes to yawning and nodding some people can sleep without yawning and nodding.

INTRODUCTION

Driver Drowsiness and sleep deprivation is one of the major causes for a lot of road accidents. Driver impairment caused by for example sleepiness, stress, visual inattention, workload etc. needs to be predicted or detected in order to avoid critical situations and crashes. Most of the accidents happen in India due to the lack of concentration of the driver. Driving ability of the driver deteriorates with time owing to drowsiness. To avoid these situations, we developed a system which will detect the drowsiness nature of the driver and will also alert him immediately.

The basic purpose of this system is to track the driver's eye movements using Eye blink Sensor and if the driver is feeling drowsy, then the system will trigger a warning message using a loud buzzer alert.

Lot of people drive on the highways all day and all night. This includes bus drivers, truck drivers, taxi men and people who are traveling long-distance; they suffer from lack of sleep. Because of sleep deprivation, it becomes very dangerous to drive when feeling fatigued.

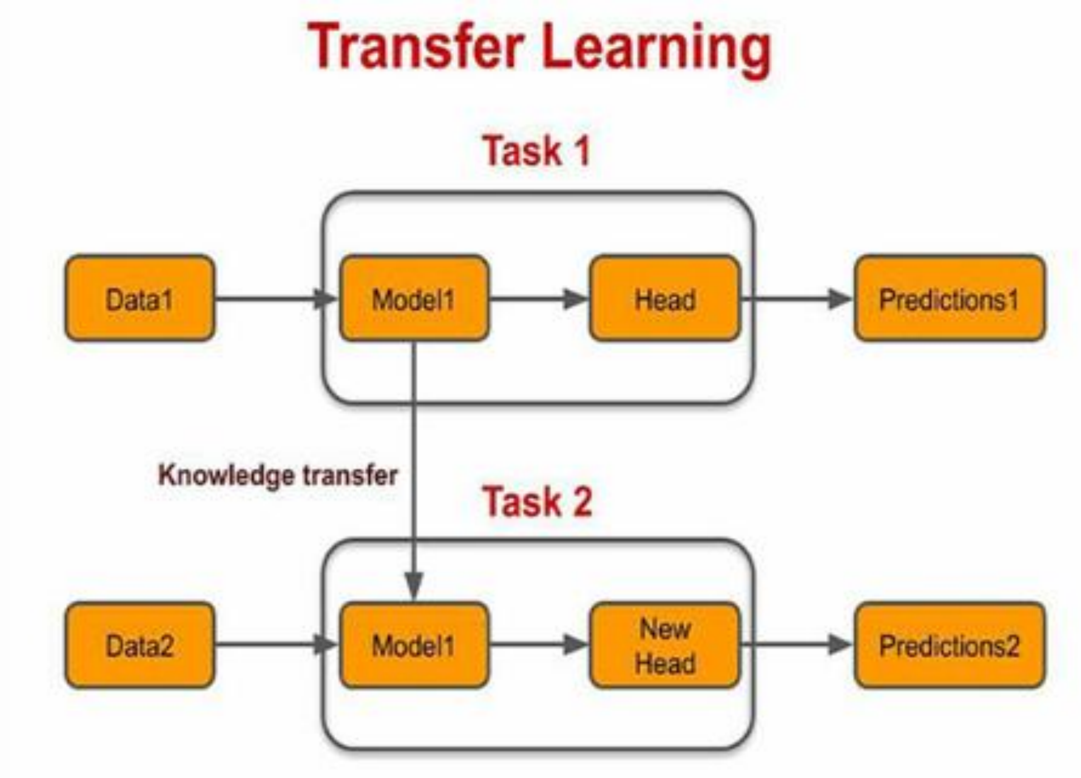
SYSTEM CONFIGURATION

Software Requirements

- Language: Python 3
- Operating system: Windows 10/8 (incl. 64-bit), Mac OS, Linux
- IDE: Visual Studio Code

Hardware Requirements

- Processor: 64-bit, quad-core, 2.5 GHz minimum per core
- RAM: 4 GB or more
- Display: 1024 x 768 or higher resolution monitors
- Camera: A webcam



CHAPTER 2 INTRODUCTION

2.1 INTRODUCTION

Speech emotion recognition is a very useful and important topic in today's world. A machine that detects the emotions of human speech can prove useful in various industries. A very basic use of speech recognition is in healthcare, where it can be used to detect depression, anxiety, stress, etc. in a patient. It can also be used in industries such as the crime sector, where emotions can be recognized from speech to distinguish between victims and criminals.

Emotions can be of different types like happy, sad, angry, hidden etc. depending on the feeling and mood of the person. In our study, we used different datasets with different emotions. We also combined the four data sets into one data set and then applied the model so that the efficiency of the model could be improvedand there could be diversity in the data points. This also led to the elimination of theoverfitting condition in our model. Speech Emotion Recognition (SER) is a systemthat can identify the emotion of different audio samples. From the description, this task is similar to text sentiment analysis, and both also share some applications sincethey differ only in the modality of the data – text versus audio. Like sentiment analysis, you can use speech emotion recognition to find the emotional range or sentimental value in various audio recordings such as job interviews, caller-agent calls, streaming videos, and songs. Moreover, even music recommendations or classification systems can cluster songs based on their mood and recommend curated playlists to the user. It is safe to assume that the complex algorithms of spotify and YouTube also have an SER component that helps in music recommendations. In this rapidly advancing AI world, human computer interactions(HCI) are of extreme importance. We live in a world where Siri and Alexa are physically closer to us than other humans. Soon the world will get more populated with physical and virtual service robots to accomplish tasks that range from caring for the elderly to assessing the effectiveness of your marketing campaign. Understanding human emotions paves the way to understanding people's needs better and, ultimately, providing better service.
A speech is a verbal action that includes expressing feelings through a person's words and sentences. People use different languages to express their emotions. A person has several emotions in his speech. He tries to tell them while speaking in the form of a speech. We took the emotions into the light: angry, sad, happy, disgust, neutral, surprise, angry, and fear. This paper is about recognizing the emotions of a person from a speech. To identify emotions, We used machine learning algorithms. We considered the classifiers to include random forest, extra trees, gradient boosting, decision tree, light gradient boosting classifiers. We took some datasets, trained them using the classifiers mentioned above, and got the results.

Speech Emotion Recognition, abbreviated as SER, is the act of attempting to recognize human emotion and affective states from speech. This is capitalizing on the fact that voice often reflects underlying emotion through tone and pitch. This is also the phenomenon that animals like dogs and horses employ to be able to understand human emotion. Human speech contains several features that the listener interprets to unpack the rich information transmitted by the speaker. The speaker also inadvertently shares tone, energy, speed, and other acoustic properties, which helps capture the subtext or intention and literal words.

2.2 Motivation

Emotions are fundamental for humans, impacting perception and everyday activities such as communication, learning and decision-making. They are expressed through speech, facial expressions, gestures and other non-verbal clues.

2.3 Problem Definition

To advance the performance of continuous emotion recognition from speech, we introduce a reconstruction-error-based learning framework with memory-enhanced Recurrent Neural Networks (RNN). Multi-Modal Fusion Emotion Recognition Method of Speech Expression Based on Deep Learning.

2.4 Purpose

The purpose of this article is to use machine learning algorithms for speech recog-nition. Specifically,this document will provide an overall description of our project including customer needs, product vision, content requirements, and general con straints. It will also provide specific requirements and capabilities required for the project, such as connectivity, functionality, and functionality.

2.5 Scope

The scope of this system is data persists for the life of the project. This help to identify emotions through Speech voice. This document specifies the final state of the software to be agreed upon by the customer and the developer. Finally, at the end of execution, all transactions can be traced from SRS to the product. This document describes the functionality, performance, limitactions, impacts, and reliability of the project throughout its lifecycle.

Machine Learning

Machine learning is a well-known process of predicting or classifying information to help people make important decisions. In order to learn from previous experiences and analyze verifiable data, ML calculations are prepared through cases or models. Structural models alone are not enough. The model should be advanced and tuned enough to give you accurate results. In order to achieve the best results, streamlining strategies require hyper-parameter tuning.

Types of Learning:

Supervised Learning

Supervised learning is type of artificial intelligence in which machines are trained carefully "marked" training data and based on this data the machine predict the outcome. Marked data indicates that some information is now marked with the correct output. In supervised learning, the training information provided to the machines acts as a boss to help the machinesaccurately predict the outcome. Supervised learning is most often used in pragmatic machine learning. When you use a calculation to get a planning capability from an input the proportion of the variable to the output, you control the learning. Input variable is (x) and the result variable is (Y).Y = f(x) (x).The goal is prepare accurately so that you can predict the resulting factors (Y). Information when you receive fresh information (x).

Unsupervised Learning

Unsupervised Learning is an ML method where you don't have to bother with managing the model. All things considered, you really want to allow the model to separate itself to find the data. Basically, it manages unlabeled information and searches for already undetected examples in the information register without prior labeling and at least with human supervision. Unlike administered discovery, which generally uses human-named information, it takes into account solo learning, otherwise called self-association. Account of the probability density display over the inputs. Computations based on unsupervised learning generate suggestions from a dataset without using named or known results. This is a machine setting that uses data that is neitherlabeled nor organized, and allows the computation on that input to run wild. In this case, the machine's job is to group unsorted data by analogies, comparisons, and contrasts with essentially no prior knowledge preparation. Unlike supervised learning, no educator is provided, meaning the machine receives no training. As a result, the machine is limited in its ability to find the hidden design hidden data.

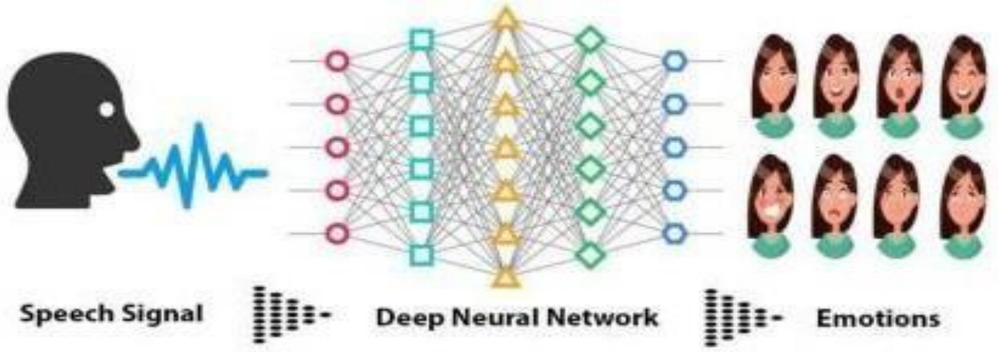


Fig 1. Speech Recognition system

2.6 Need Speech Emotion Recognition?

1. Emotion recognition is the part of speech recognition which is gaining more popularity and need for it increases enormously. Although there are methods to recognize emotion using machine learning techniques, this project attempts to use
deep learning to recognize the emotions from data.

2. SER (Speech Emotion Recognition) is used in call center for classifying calls according to emotions and can be used as the performance parameter for conversational analysis thus identifying the unsatisfied customer, customer satisfaction and so on. for helping companies improving their services

3. It can also be used in-car board system based on information of the mental state of the driver can be provided to the system to initiate his/her safety preventing accidents to happen.

CHAPTER 3 LITERATURE REVIEW

Paper 1: "Comparison of Glottal Closure Instants Detection Algorithms for Emotional Speech"
The performance of the algorithms was assessed using known evaluation metrics onspeech utterances of the EMO-DB database representing seven emotions. From theexperimental results, it was observed that the GCI detection performance in neutralemotion is nearly equal for all the algorithms (except DYPSA and MMF). Results on the remaining six other emotions indicate that the GCI detection performance degrades heavily especially in anger and joy. This is due to deterioration in the estimation of the average pitch period in these emotions compared to neutral [1].

Paper 2 : Speech Emotion Recognition using Machine Learning Algorithms T.Sai Samhith et al. Author explaining The TESS dataset that we considered is fine-tuned. Since it has noiseless data, it was easy for us to classify and feature the data.The classifier with utmost accuracy, AUC, F1 score, kappa, MCC is Random ForestClassifier. The classifier with the least accuracy and all the other terms is the decision tree classifier. We also mentioned why the decision tree classifier has lowprecision and the random forest classifier has high accuracy. The other classifiers that we analyzed, i.e., extra trees classifier, light gradient boosting machine, multi perceptron classifier, gradient boosting classifier, have the mid values in accuracy and other values. In conclusion, the random forest classifier is the most accurate algorithm and can be used in real-life scenarios to detect a person's emotions through his speech [3] at choosing a feature based on shared information can provide us with the best strategy.

CHAPTER 4 SOFTWARE SPECIFICATION

4.1 EXISTING SYSTEM

The speech emotion detection system is implemented as a Machine Learning(ML) model. The steps of implementation are comparable to any other ML project, with additional fine-tuning procedures to make the model function better. The flowchart represents a pictorial overview of the process (see Figure 1). The first step is data collection, which is of prime importance. The model being developed will learn from the data provided to it and all the decisions and results that a developed model will produce is guided by the data. The second step, called feature engineering, is a collection of several machine learning tasks that are executed overthe collected data. These procedures address the several data representation and dataquality issues. The third step is often considered the core of an ML project where an algorithmic based model is developed. This model uses an ML algorithm to learnabout the data and train itself to respond to any new data it is exposed to. The finalstep is to evaluate the functioning of the built model. Very often, developers repeat the steps of developing a model and evaluating it to compare the performance of different algorithms. Comparison results help to choose the appropriate ML algorithm most relevant to the problem.

4.1.1.1 PROPOSED SYSTEM:-

In this current study, we presented an automatic speech emotion 6 recognition (SER) system using machine learning algorithms to classify the emotions. The performance of the emotion detection system can greatly influence the overall performance of the application in many ways and can provide many advantages over the functionalities of these applications. This research presents a speech emotion detection system with improvements over an existing system in terms of data, feature selection, and methodology that aim human speech contains

Fig.2 Speech Recognition System

Several features that the listener. Interprets to unpack the rich information transmitted by the speaker. The speaker also inadvertently tone, energy, speed, and other acoustic properties, which helps capture the subtext or intention and literal words.ork in speech recognition started with converting speech to text (or creating a transcript). With that, the first level of information was captured (the words or the literal meaning of the speech).

In more advanced applications, the context and empathizing with the speakerbecomes vital for speech emotion recognition. This is also where text sentiment analysis differs from speech emotion recognition. In sentiment analysis, the emotionis conveyed literally in the text (using negative or positive words), making it easier to comprehend the intended meaning (positive or negative, angry or sad, for example). However, in SER, all this information is hidden under the first layer of information.

4.1.1.2 Datasets used in this project

Crowd-sourced Emotional Mutimodal Actors Dataset (Crema-D)

Ryerson Audio-Visual Database of Emotional Speech and Song (Ravdess) Surrey Audio-Visual Expressed Emotion (Savee)
Toronto emotional speech set (Tess) Audio MNIST
This dataset contains 30,000 audio clips of spoken digits (0–9) from 60different speakers.

4.4 ALGORITHMS USED

4.4.1 CLASSIFIERS

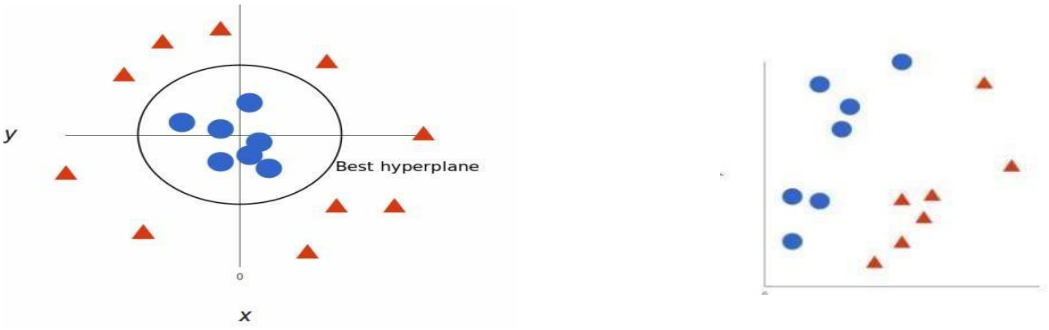
Classification is defined as the process of recognition, understanding, and groupingof objects and ideas into preset categories a.k.a ―sub populations. With the help ofthese pre-categorized training datasets, classification in machine learning programsleverage a wide range of algorithms to classify future datasets into respective and relevant categories. Classification algorithms used in machine learning utilize inputtraining data for the purpose of predicting the likelihood or probability that the datathat follows will fall into one of the predetermined categories. One of the most common applications of classification is for filtering emails into ―spam‖ or ―non- spam‖, as used by today‗s top email service providers. In short, classification is a form of ―pattern recognition,‖. Here, classification algorithms applied to thetraining data find the same pattern (similar number sequences, words or sentiments, and the like) in future data sets. We will explore classification algorithms in detail, and discover how a text analysis software can perform actions like sentiment analysis - used for categorizing unstructured text by opinion polarity (positive, negative, neutral, and the like).

4.4.2 SVM

SVM algorithms classify data and train models within Fig. 2 SVC Without Hyperplane super finite degrees of polarity,creating a 3-dimensional classification model that goes beyond just X/Y predictive axes. Take a look at this

visual representation to understand how SVM algorithms work. We have two tags: red and blue, with Best Hyperplanes two data features: X and Y, and we train our classifier to output an X/Y coordinate as either red or blue. SVM algorithms make excellent classifiers because, the more complex the data, the more accurate the prediction will be. Imagine the above as a 3-dimensional output, with a Z-axis added, so it becomes a circle. 8 Mapped back to 2D, with the best hyperplanes, it looks like above fig.

Fig 3. SVC



4.4.3. RANDOM FOREST CLASSIFIER

The term ―Random Forest Classifier‖ refers to the classification algorithm made up of several decision trees. The algorithm uses randomness to build each individual tree to promote uncorrelated forests, which then uses the forest_s predictive powers to make accurate decisions. Random forest classifiers fall under the broad umbrella of ensemble based learning methods. They are simple to implement, fast in operation, and have proven to be extremely successful in a variety of domains. The key principle underlying the random forest approach comprises the construction of many ―simple‖ decision trees in the training stage and the majority vote (mode) across them in the classification stage. Among other benefits, this voting strategy has the effect of correcting for the undesirable property of decision trees to overfit training data. In the training stage, random forests apply the general technique known as bagging to individual trees in the ensemble. Bagging repeatedly selects a random sample with replacement from the 9 training set and fits trees to these samples. Each tree is grown without any pruning. The number of trees in the ensemble is a free parameter which is readily learned automatically using the so-called out-of- bag error . Much like in the case of naïve Bayes– and k-nearest neighbor–based algorithms, random forests are popular in part due to their simplicity on the one hand, and generally good performance on the other. However, unlike the former two approaches, random forests exhibit a degree of unpredictability as regards the structure of the final trained model. This is an inherent consequence of the stochastic nature of tree building. As we will explore in more detail shortly, one of the key reasons why this characteristic of random forests can be a problem in regulatory reasons—clinical adoption often demands a high degree of repeatability not only in terms of the ultimate performance of an algorithm but also in terms of the mechanics as to how a specific decision is made.

4.4.4 K Neighbors Classifier

The concept of the k-nearest neighbor classifier can hardly be simpler described. This is an old saying, which can be found in many languages and many cultures. This means that the concept of the k-nearest neighbor classifier is part of our everyday life and judging: Imagine you meet a group of people; they are all very young, stylish and sportive. They talk about their friend Ben, who isn't with them. So, what is your imagination of Ben? Right, you imagine him as being young, stylish and sportive as well. If you learn that Ben lives in a neighborhood where people vote conservative and that the average income is above 200000 dollars a year? Both his neighbors make even more than 300,000 dollars per year? What do you think of Ben? Most probably, you do not consider him to be an underdog and you may suspect him to be a conservative as well? The principle behind nearest neighbor classification consists in finding a predefined number, i.e. the 'k' - of training samples closest in distance to a new sample, which has to be classified. The label of the new sample will be defined from these neighbors. K-nearest neighbor classifiers have a fixed user defined constant for the number of neighbors which have to be determined. There are also radius-based neighbor learning algorithms, which have a varying number of neighbors based on the local density of points, all the samples inside of a fixed radius. The distance can, in general, be any metric measure: standard Euclidean distance is the most common choice. Neighbors-based methods are known as non-generalizing machine learning methods, since they simply "remember" all of its training data. Classification can be computed by a majority vote of the nearest neighbors of the unknown sample. 10 The k-NN algorithm is among the simplest of all machine learning algorithms, but despite its simplicity, it has been quite successful in a large number of classification and regression problems, for example character recognition or image analysis. The algorithm for the k-nearest neighbor classifier is among the simplest of all machine learning algorithms.

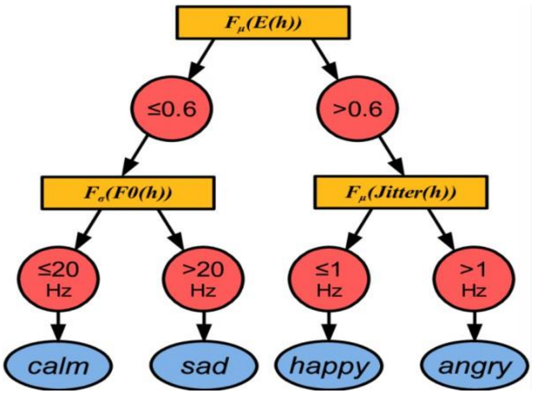
KNN is a type of instance-based learning, or lazy learning. In machine learning, lazy learning is understood to be a learning method in which generalization of the training data is delayed until a query is made to the system. On the other hand, we have eager learning, where the system usually generalizes the training data before receiving queries. In other words: The function is only approximated locally and all the computations are performed, when the actual classification is being performed.

4.4.5 MLP CLASSIFIER

MLPClassifier stands for Multi-layer Perceptron classifier which in the name itself connects to a Neural Network. Unlike other classification algorithms such as Support Vectors or Naive Bayes Classifier, MLPClassifier relies on an underlying Neural Network to perform the task of classification. A multi-layer rather than a single layer network is required since a single layer perceptron (SLP) can only compute a linear decision boundary, which is not flexible enough for most realistic learning problems. For problem that is linearly separable, (that is capable of being perfectly separated by linear decision boundary), the perceptron convergence theorem guarantees convergence. In its simplest form, SLP training is based on the simple idea of adding or subtracting a pattern from the current weights when the target and predicted class disagrees, otherwise the weights are unchanged. For a non-linearly separable problem, this simple algorithm can go on cycling indefinitely. The modification known as least mean square (LMS) algorithm uses amean squared error cost function to overcome this difficulty, but since there is only a single perceptron, the decision boundary is still linear. An MLP is a universal approximate that typically uses the same squared error function as LMS. However,the main difficulty with the MLP is that the learning algorithm has a complex error surface, which can become stuck in local minima. There does not exist any MLP learning algorithm that is guaranteed to converge, as with SLP. The popular MLP back propagation algorithm has two phases, the first being a forward pass, which isa forward simulation for the current training pattern and
enables the error to be calculated.

4.4.6 DECISION TREE CLASSIFIER

The decision tree acquires knowledge in the form of a tree, which can also be, rewritten as a set of discrete rules to make it easier to understand. The main advantage of the decision
tree classifier is its ability to using different feature subsets and decision rules at different stages of classification. General decision tree consists as shown in Figure above. number of internal and leaf nodes and branches. Leaf nodes indicate the class to be assigned to a sample. Each internal node of a tree corresponds to a feature, and branches represent conjunctions of features that lead to those classifications. For food quality evaluation using computer vision, the decision tree has been applied to the problem of meat quality grading (Song et al., 2002) and the classification of ―in the shell‖ pistachio nuts (Ghazanfari et al., 1998).



4.4 System Requirements

1. Database Requirements MySQL Database

2. Software Requirements (Platform Choice)

- Operating System :Windows and Linux
- Language: python3
- Documentation: Latex 2.9

3. Hardware Requirements

- Processor : CORE i3 and above
- RAM: 4GB RAM
- Hard Disk : Min 320 GB and above

4.5 Waterfall Model

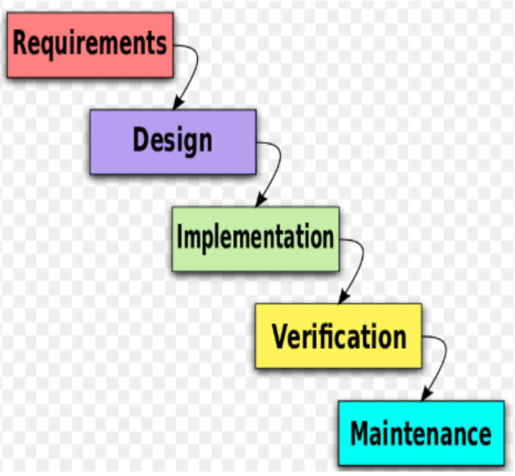
The Waterfall Model is sequential design process, often used in Software development processes; where progress is seen as flowing steadily download through the phase of conception, Analysis, Design, Construction, Testing, Production/Implementation and Maintenance, Initiation.

This Model is also calledas the classic Life cycle modelas it suggests a systematic sequential approach to software developments.

This is oldest model followed software engineering. The process begin with communication phase where the customer specifies the requirements and then progress through other phases like planning, modeling, Construction and deploymentof the software.

There are 5 Phase of waterfall Model.

The Waterfall Model is sequential design process, often used in Software development processes; where progress is seen as flowing steadily



.6.1 Waterfall model

Download through the phase of conception, Initiation, Analysis, Design, Construction, Testing, Production/Implementation and Maintenance Systematic sequential approach to software developments. This one of the oldest model Followed in software engineering. The process begins with the communication phase where the customer specifies the requirements and then progress through other phases like planning, modeling.

CHAPTER 5 SYSTEM ARCHITECTURE

5.1 ARCHITECTURE

Describing the overall features of the software is concerned with defining the requirements and establishing the high level of the system. During architectural design, the various web pages and their interconnections are identified and designed. The major software components are identified and decomposed into processing modules and conceptual data structures and the interconnections among the modules are identified. The following modules are identified in the proposed system.

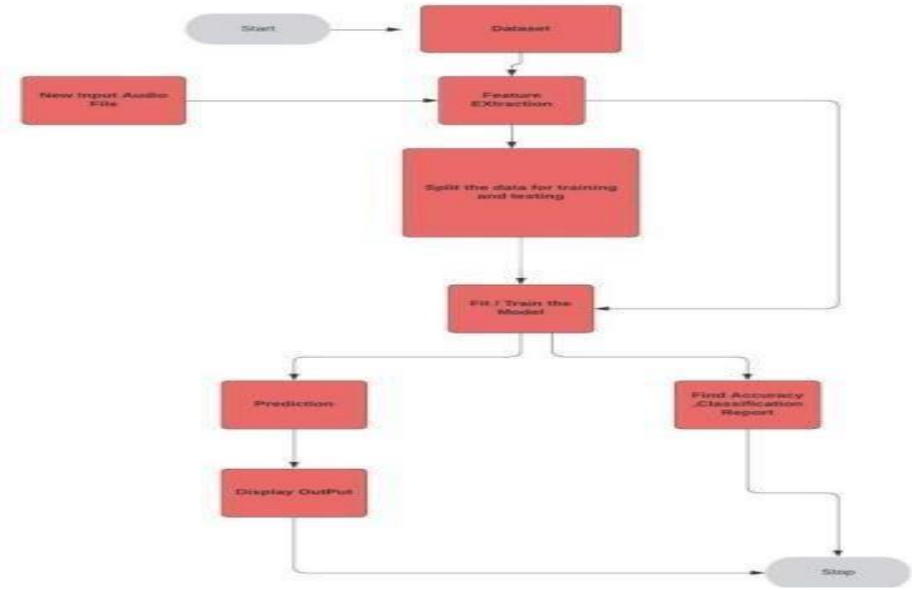


Fig 4.1 SYSTEM ARCHITECTURE

5.1 MODULES

Speech input Module
Feature extraction and selection Classification
Recognized emotional output

5.2 MODULE DESCRIPTION

1 Speech input Module

Input to the system is speech taken with the help of audio. Then equivalent digital representation of received audio is produced through sound file.

2 Feature extraction and selection

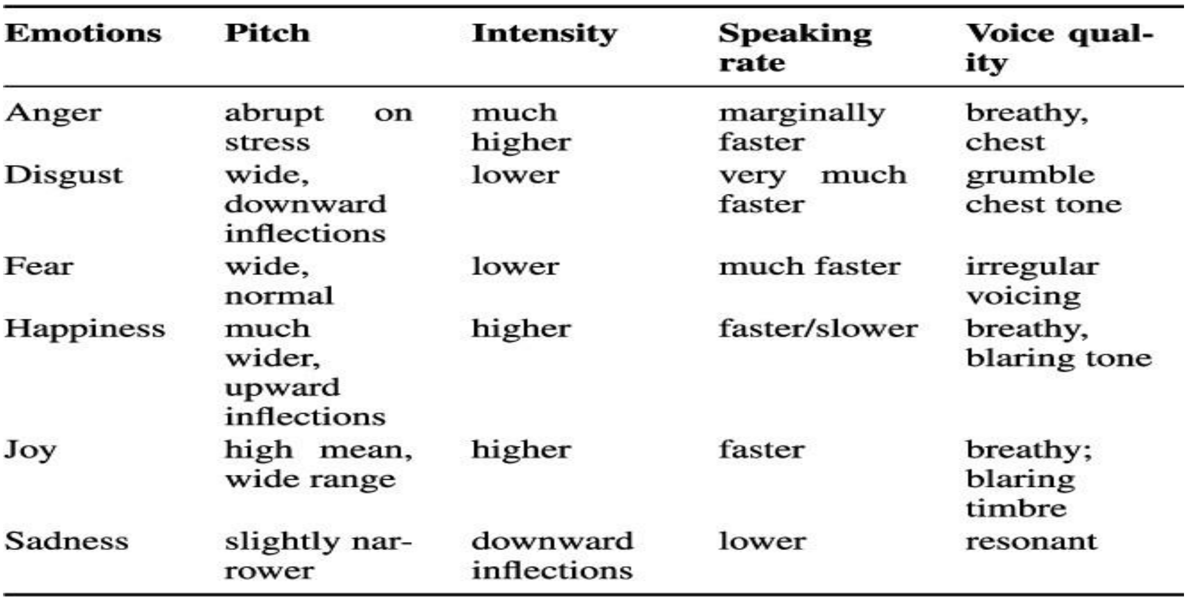
There are so many emotional states of emotion and emotion relevance is used toselect the extracted speech features. For speech feature 22 extraction to selection corresponding to emotions all procedure revolves around the speech signal.

3 Classification Module

Finding a set of significant emotions for classification is the main concern in speech emotion recognition system. There are various emotional states containsin a typical set of emotions that makes classification a complicated task.

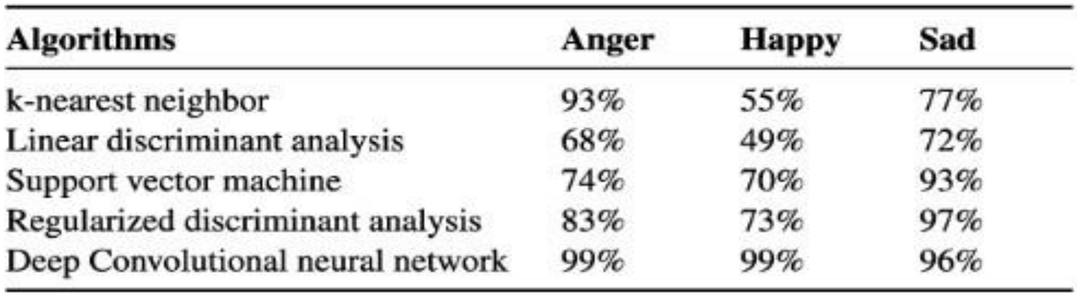
4 Recognized emotional output

Fear, surprise, anger, joy, disgust and sadness are primary emotions and naturalness of database level is the basis for speech emotion recognition systemevaluation.



Emotions	Pitch	Intensity	Speaking rate	Voice quality
Anger	abrupt on stress	much higher	marginally faster	breathy, chest
Disgust	wide, downward inflections	lower	very much faster	grumble chest tone
Fear	wide, normal	lower	much faster	irregular voicing
Happiness	much wider, upward inflections	higher	faster/slower	breathy, blaring tone
Joy	high mean, wide range	higher	faster	breathy; blaring timbre
Sadness	slightly narrower	downward inflections	lower	resonant

TABLE 2. Summarized form of some acoustic variations observed based on emotions.



Algorithms	Anger	Happy	Sad
k-nearest neighbor	93%	55%	77%
Linear discriminant analysis	68%	49%	72%
Support vector machine	74%	70%	93%
Regularized discriminant analysis	83%	73%	97%
Deep Convolutional neural network	99%	99%	96%

TABLE 3. Comparative analysis of different classifiers in SER.

a. DATA FLOW DIAGRAM

The DFD is also called as bubble chart. It is a simple graphical formalism that canbe used to represent a system in terms of input data to the system, various processing carried out

on this data, and the output data is generated by this system. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that areapplied as data moves from input to output. DFD is also known as bubble chart. ADFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

5.3 schematic flowcharts

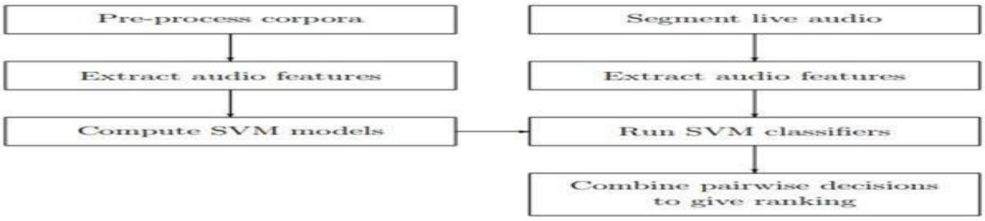


Fig.4.1 Systematic flowchart

5.1 TYPES OF SPEECH:

On the basis of ability they have to recognize a speech recognition systems can be separated in different classes.

Following are the classification:

Isolated words: In this type of recognizers sample window both sides contains lowpitch utterance. At a time only single word or utterance is accepted by it and there is need to wait between utterances by speaker as these systems have listen/non-listen states. For this class isolated utterance is a better name.

Connected words: In this separate utterance can run together with minimal pause between them otherwise it is similar to isolated words.

Continuous words: It allows users to speak naturally and content are determined by computer. Creation of recognizers that have continuous speech capabilities are difficult due to determination of utterance boundaries by utilizing a special method.

Spontaneous words: It can be thought of as speech at basic level that is natural sounding and not rehearsed. Variety of natural speech features are handle is the ability of spontaneous speech with ASR system.

Convolutional Neural Network?

A convolutional neural network is a feed-forward neural network that is generally used to analyze visual images by processing data with grid-like topology. It‘s also known as
a ConvNet. A convolutional neural network is used to detect and classify objects in an image.

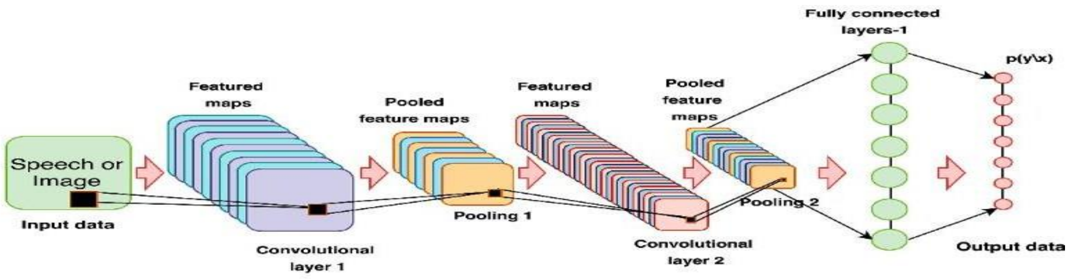


Fig .4.5.1 CNN

Convolution neural network has multiple hidden layers that help in extracting information from an image. The four important layers in CNN are:

1. Convolution layer
2. ReLU layer
3. Pooling layer
4. Fully connected layer Convolution Layer

This is the first step in the process of extracting valuable features from an image. A convolution layer has several filters that perform the convolution operation. Every image is considered as a matrix of pixel values.
ReLU stands for the rectified linear unit. Once the feature maps are extracted, the next step is to move them to a ReLU layer.
Pooling Layer

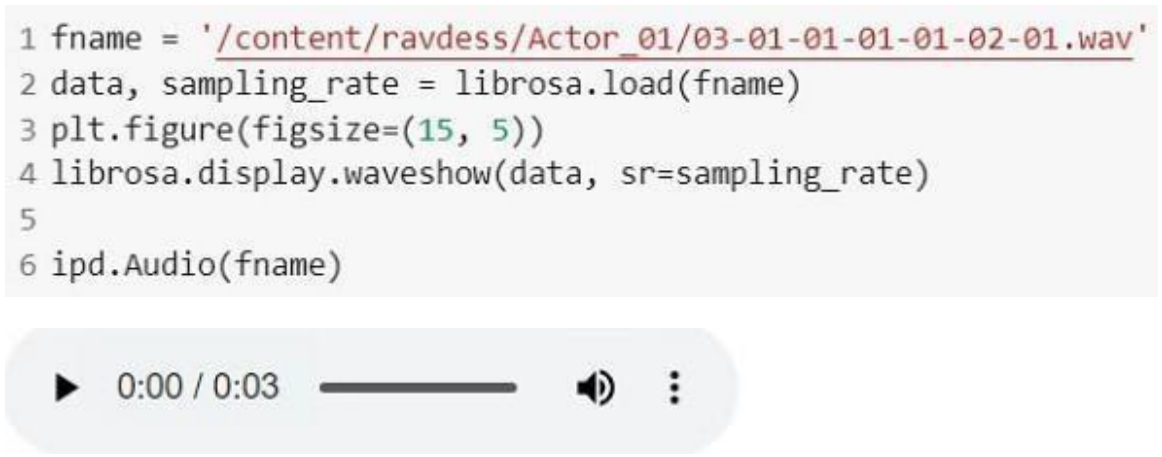
Pooling is a down-sampling operation that reduces the dimensionality of the feature. Flattening: The resulting feature maps are flattened into a one-dimensional vector after the convolution and pooling layers so they can be passed into a completely linked layer for categorization or regression.

Fully Connected Layers: It takes the input from the previous layer and computes the final classification or regression task.

CHAPTER 6 RESULTS AND DISCUSSION

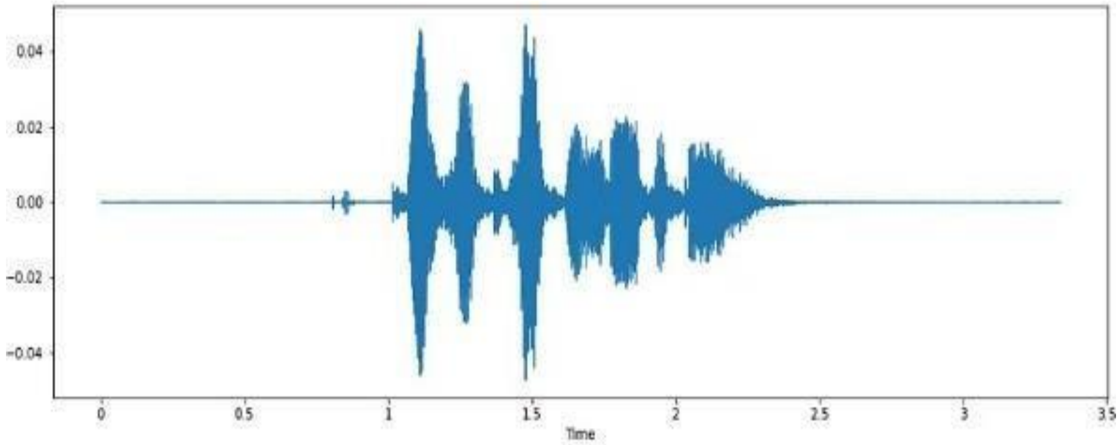
6.1 Result Analysis

Thus, in this model, we obtained a model accuracy of about 96% on the training data and 71% on the test data, and a value loss of about 15% after analyzing 48,000data samples using the MFCC feature extraction method and the CNN model for training. and testing purposes. By using additional feature extraction methods suchas ZCR and RMSE, as well as providing the model with more data - in our case weused 48,000 data samples, so the accuracy of the model would increase if the numberof samples used was increased. We can further increase the accuracy of the model by increasing the number of epochs.

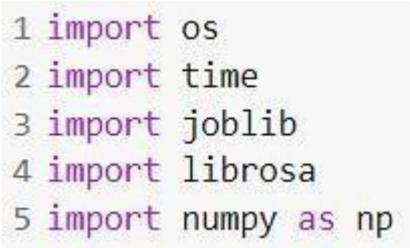


```
fname = '/content/ravdess/Actor_01/03-01-01-01-01-02-01.wav'
data, sampling_rate = librosa.load(fname)
plt.figure(figsize=(15, 5))
librosa.display.waveshow(data, sr=sampling_rate)

ipd.Audio(fname)
```

The waveshow function of librosa will assist in plotting the clip as below.

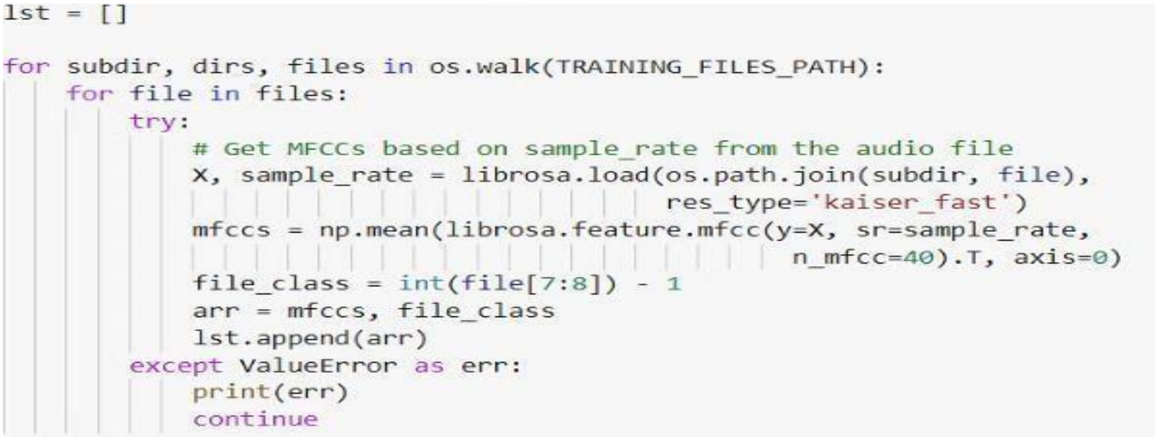


To import the entire dataset, we will use the following libraries in Python.



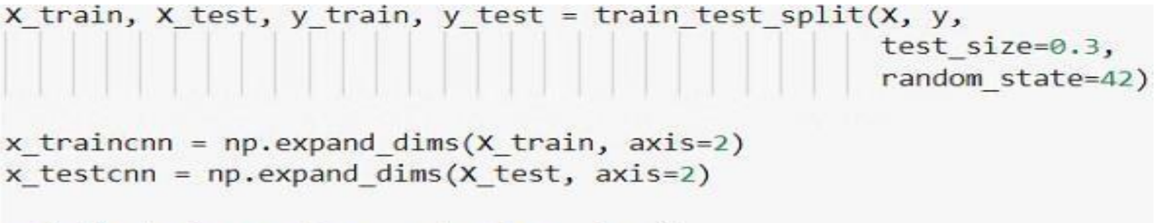
```
import os
import time
import joblib
import librosa
import numpy as np
```

The next step would be to use the MFCC function to extract thosefeatures.

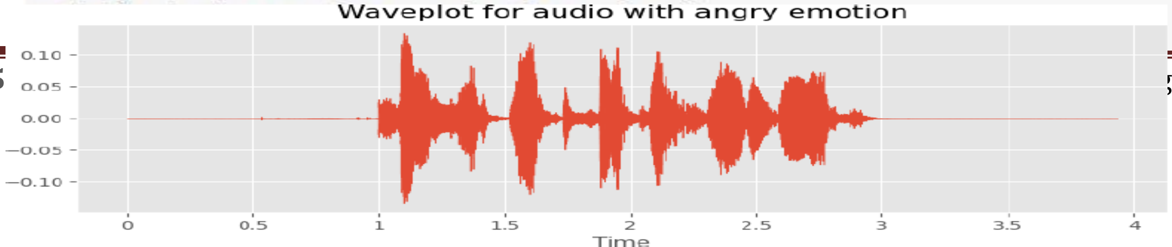


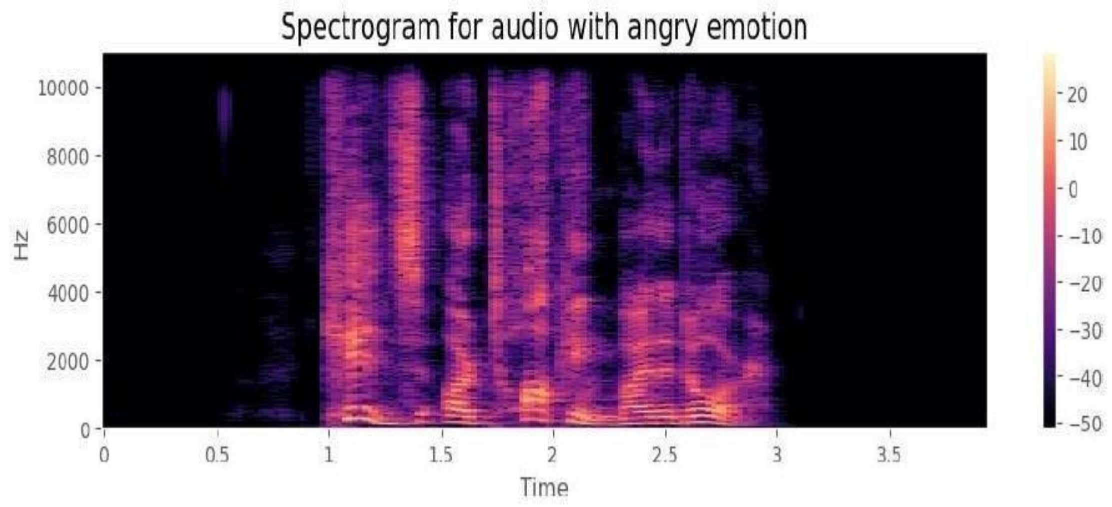
```
lst = []

for subdir, dirs, files in os.walk(TRAINING_FILES_PATH):
    for file in files:
        try:
            # Get MFCCs based on sample_rate from the audio file
            X, sample_rate = librosa.load(os.path.join(subdir, file),
                                          res_type='kaiser_fast')
            mfccs = np.mean(librosa.feature.mfcc(y=X, sr=sample_rate,
                                                  n_mfcc=40).T, axis=0)
            file_class = int(file[7:8]) - 1
            arr = mfccs, file_class
            lst.append(arr)
        except ValueError as err:
            print(err)
            continue
```

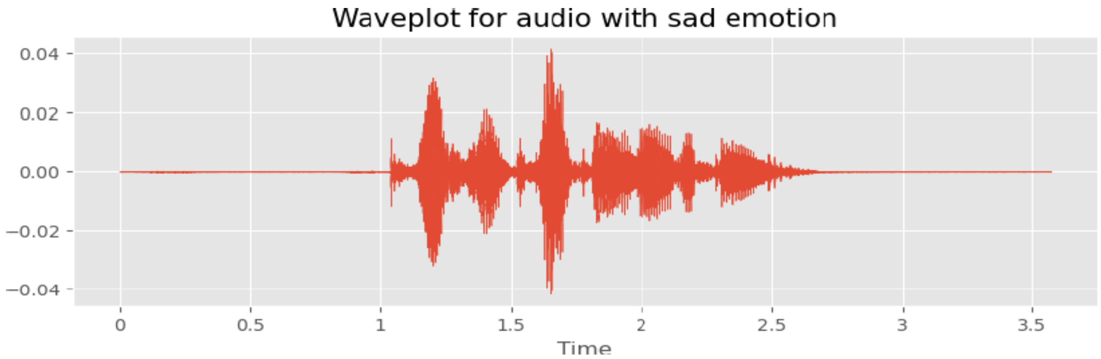
Using scikit-learn‘s train_test_split we prepare our training and testingsets

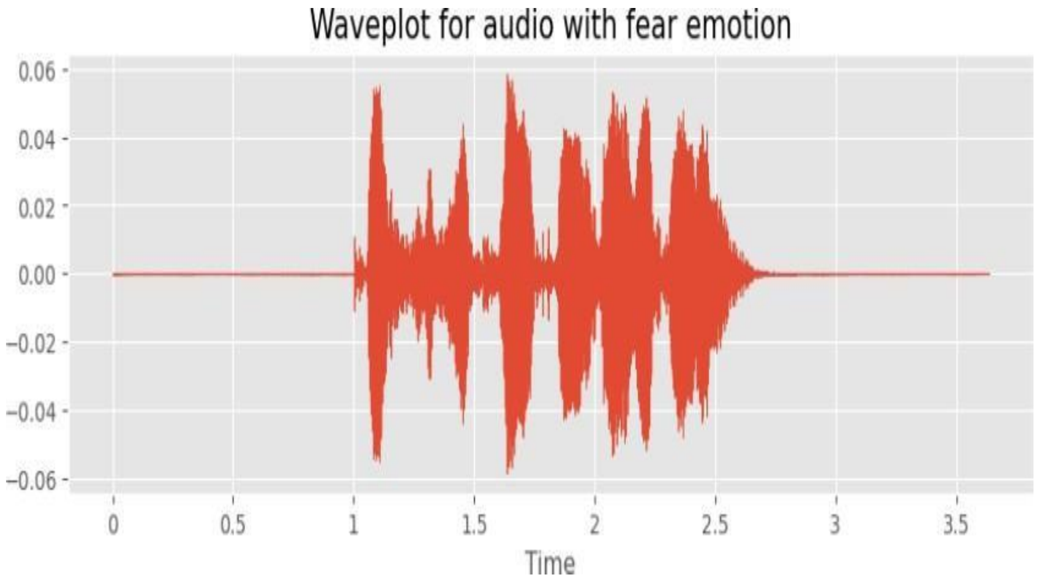


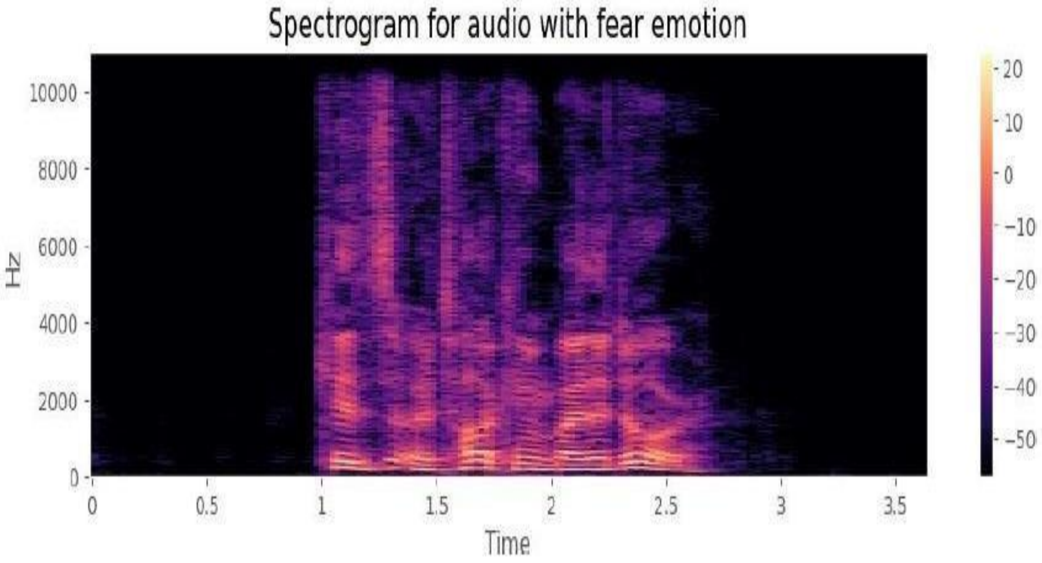
```
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                     test_size=0.3,
                                                     random_state=42)

x_traincnn = np.expand_dims(X_train, axis=2)
x_testcnn = np.expand_dims(X_test, axis=2)
```

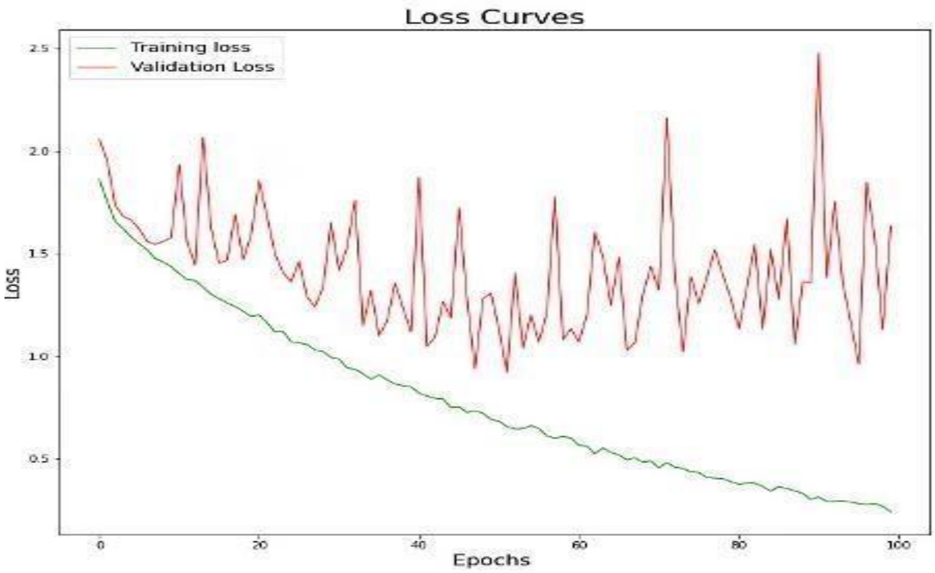


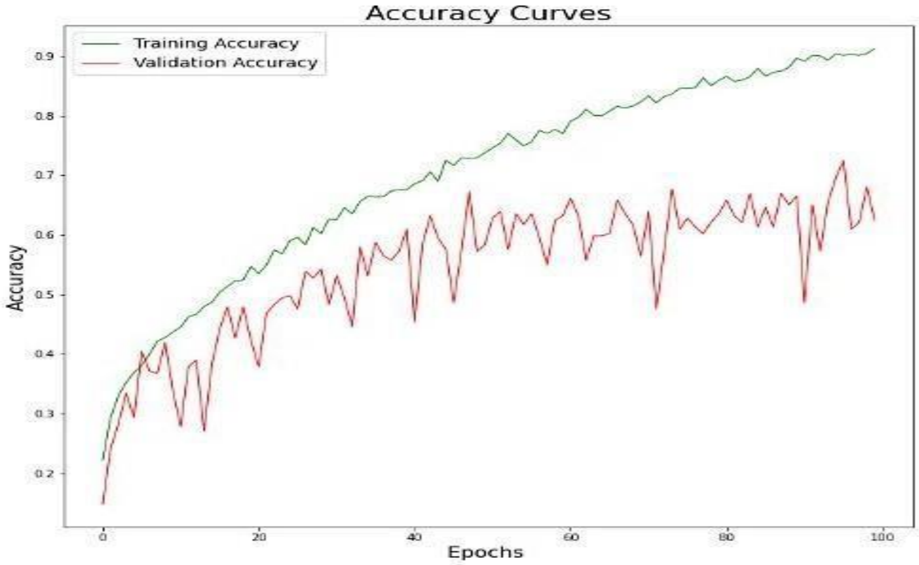

Spectrogram for audio with angry emotion
Hz
10000
8000
6000
4000
2000
0
0
0.5
1
1.5
2
2.5
3
3.5
Time
20
10
0
−10
−20
−30
−40
−50

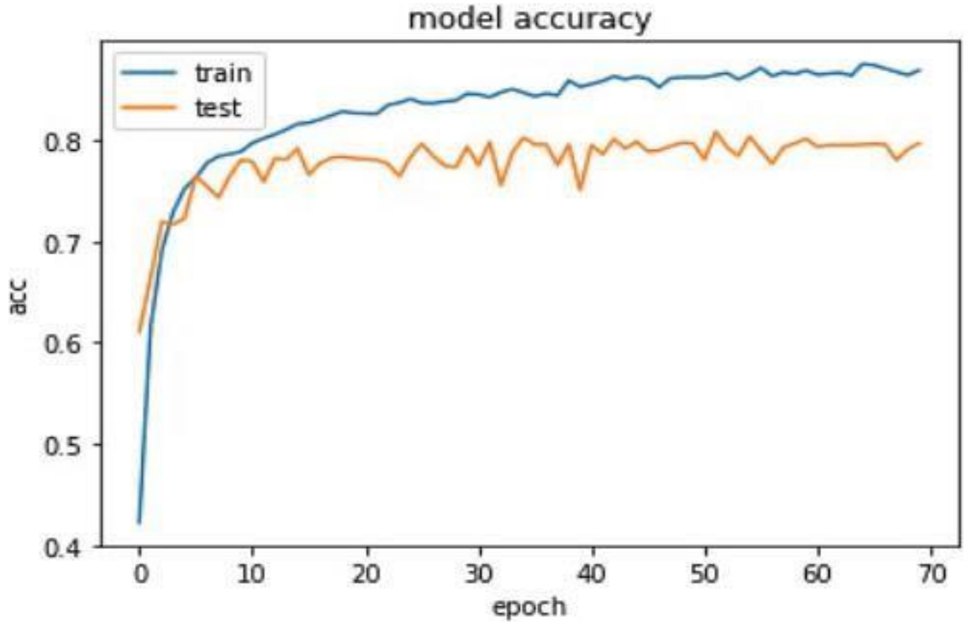

Waveplot for audio with sad emotion
0.04
0.02
0.00
−0.02
−0.04
0
0.5
1
1.5
2
2.5
3
3.5
Time


Waveplot for audio with fear emotion
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
0
0.5
1
1.5
2
2.5
3
3.5
Time


Spectrogram for audio with fear emotion
Hz
10000
8000
6000
4000
2000
0
0
0.5
1
1.5
2
2.5
3
3.5
Time
20
10
0
−10
−20
−30
−40
−50


Loss Curves
Training loss
Validation Loss
Loss
2.5
2.0
1.5
1.0
0.5
0
20
40
60
80
100
Epochs


Accuracy Curves
Training Accuracy
Validation Accuracy
Accuracy
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0
20
40
60
80
100
Epochs


model accuracy
train
test
acc
0.4
0.5
0.6
0.7
0.8
0
10
20
30
40
50
60
70
epoch

	Predicted_angry	Predicted_sad	Predicted_neutral	Predicted_ps	Predicted_happy
True_angry	92.307693	0.000000	1.282051	2.564103	3.846154
True_sad	12.820514	67.948715	3.846154	6.410257	8.974360
True_neutral	3.846154	8.974360	82.051285	2.564103	2.564103
True_ps	2.564103	0.000000	1.282051	83.333328	12.820514
True_happy	20.512821	2.564103	2.564103	2.564103	71.794876

Table 3 Emotion prediction

CHAPTER 7
PROJECT IMPLEMENTATIONS

In this project following steps are performed to achieve the objectives.

Step 1: Data collection

The first step in the machine learning process is data collection. Data is the lifeblood of machine learning - the quality and quantity of your data can directly impact your model's performance. Data can be collected from various sources such as databases, text files, images, audio files, or even scraped from the web.

Once collected, the data needs to be prepared for machine learning. This process involves organizing the data in a suitable format, such as a CSV file or a database, and ensuring that the data is relevant to the problem you're trying to solve.

Step 2: Data preprocessing

Data preprocessing is a crucial step in the machine learning process. It involves cleaning the data handling missing data, normalizing the data.Preprocessing improves the quality of your data and ensures that your machine learning model can interpret it correctly. This step can significantly improve the accuracy of your model. Our course, Preprocessing for Machine Learning in Python, explores how to get your cleaned data ready for modeling.

Step 3: Choosing the right model

Once the data is prepared, the next step is to choose a machine learning model. There are many types of models to choose from, including linear regression, decision trees, and neural networks. The choice of model depends on the nature of your data and the problem you're trying to solve. Factors to consider when choosing a model include the size and type of your data, the complexity of the problem, and the computational resources available.

Step 4: Training the model

After choosing a model, the next step is to train it using the prepared data. Training involves feeding the data into the model and allowing it to adjust its internal parameters to better predict the output.

During training, it's important to avoid over fitting (where the model performs well on the training data but poorly on new data) and under fitting (where the model performs poorly on both the training data and new data). You can learn more about the full machine learning process in our Machine Learning Fundamentals with Python skill track, which explores the essential concepts and how to apply them.

Step 5: Evaluating the model

Once the model is trained, it's important to evaluate its performance before deploying it. This involves testing the model on new data it hasn't seen during training.

Common metrics for evaluating a model's performance include accuracy (for classification problems), precision and recall (for binary classification problems), and mean squared error (for regression problems).

Step 6: Hyper parameter tuning and optimization

After evaluating the model, you may need to adjust its hyper parameters to improve its performance. This process is known as parameter tuning or hyper parameter optimization. In this hyper parameter tuning include grid search and cross validation.

Step 7: Predictions and deployment

Once the model is trained and optimized, it's ready to make predictions on new data. This process involves feeding new data into the model and using the model's output for decision-making or further analysis.

Deploying the model involves integrating it into a production environment where it can process real-world data and provide real-time insights.

Install required python libraries like pandas seaborn, matplotlib, pytorch, keras, and numpy. Also design speech input model and train that model according to requirement and predict the correct output in accordance with input data.

CHAPTER 8

SOFTWARE TESTING

8.1.1 INTRODUCTION

Software testing in the context of your multi-profile software application project plays a crucial role in verifying the functionality, quality, and reliability of the application. This testing process encompasses functional testing to ensure accurate performance, accessibility testing to adhere to standards for users with usability testing to guarantee user-friendliness, performance testing to optimize features, security testing to protect user data, compatibility testing for diverse platforms, and localization testing to support multiple languages. Ultimately, software testing is essential to ensure that the application delivers a high-quality, inclusive, and user- centric experience that meets the diverse needs of your target user classes.

8.1.2 TESTING PROCEDURE

The testing procedure for your multi-profile software application project entails a sys- tematic approach beginning with requirements analysis to identify test cases tailored to each user profile, followed by test design to define test scenarios, data, and environments. The actual testing involves executing test cases, recording results, and identifying and reporting defects. Multiple types of testing, including functional, accessibility, usability, performance, security, compatibility, and localization testing, will be conducted to en- sure the application meets the unique requirements of blind users, handicapped users, and multitasking users. Continuous feedback and iterative testing will be utilized to address issues and refine the application to guarantee a high- quality and accessible user experience for all user classes.

System is tested by following steps:

Analysis: Identify user class-specific requirements.

Test Planning: Define test scope, scenarios, data, and environments.

8.1.3 TEST STRATEGY

The test strategy for this project ensures thorough testing across user profiles, focus- ing on functionality, accessibility, usability, performance, security, compatibility, and localization. It emphasizes iterative testing, defect management, and continuous improvement, adhering to accessibility and quality standards.

8.1.4 Unit Testing:

Unit testing in the context of your multi-profile software application project involves testing individual components, modules, and functions of the software. Unit tests ensure that these components work as expected and are error-free, providing a strong foundation for the overall system‘s functionality. While unit tests for voice command recognition would verify its precision in understanding user commands, ensuring the reliability and performance of these critical features for multitasking users.

8.1.5 Integration Testing:

Integration testing in your multi-profile software application project involves validating the interactions and interfaces between different modules and components that constitute the complete system. This testing phase ensures that voice commands work harmoniously when integrated, confirming that the application performs seamlessly and accurately across user profiles.

8.1.6 Performance Testing:

Performance testing in your multi-profile software application project evaluates the ap- plication‘s responsiveness, speed, and efficiency under various conditions. Performance testing will determine how well the application handles user interactions, particularly in scenarios that demand rapid response times, making certain that the application delivers a seamless and efficient user experience across diverse user classes and usage conditions.

CHAPTER 9

CONCLUSION & FUTURE SCOPE

9.1 Conclusion

The project focuses on the important task of emotion recognition from speech signals. The goal is to extract emotion recognition features from speech signals, select appropriate features, and recognize emotions. Python is used for developmentand image, sound and text processing methods are used. Machine learning algorithms and artificial neural networks are used to train the model, and the LibrosaPython package is used for music and audio analysis. The project also includes text classification using BERT and artificial neural networks (ANN). The text data is pre-processed to remove noise and irrelevant information. The BERT model and a pre-trained tokenizer are used to transform the text into numerical embeddings, which are then processed by artificial neural networks to create a classification model. The project aims to effectively use deep learning methods in the field of speech emotion recognition and text classification. In this system shows Machine learning to obtain the underlying emotion from speech audio data and some insightson the human expression of emotion through voice.

This system can be employed in a variety of setups like Call Centre for complaints or marketing, in voice-based virtual assistants or chatbots, in linguistic research, etc. A few possible steps that can be implemented to make the models more robustand accurate are the following

An accurate implementation of the pace of the speaking can be explored to check ifit can resolve some of the deficiencies of the model.

- Figuring out a way to clear random silence from the audio clip.

- Exploring other acoustic features of sound data to check their applicability inthe domain of speech emotion recognition. These features could simply be some proposed extensions of MFCC like RAS-MFCC or they could be other features entirely like LPCC, PLP or Harmonic cepstrum.

- Following lexical features based approach towards SER and using an ensemble of the lexical and acoustic models. This will improve the accuracy of the system because in some cases the expression of emotion iscontextual rather than vocal.

- Adding more data volume either by other augmentation techniques like time- shifting or speeding up/slowing down the audio or simply finding more annotated audio clips.

Speech recognition application-

Applications of simple speech recognition are widespread –

1) YouTube auto-generated subtitles,
2) live speech transcripts
3) transcripts for online courses
4) Intelligent voice-assisted chatbots like Alexa and Siri.
5) Prevent from depression.

9.2 Future Scope

So the recognition of discourse sentiment is an extremely fascinating topic and there is more to be found in this area, in our model, future work will include improving the accuracy of the model that will yield better results, we can also prepare the model to provide discourse implications that are longer term, as in with this model we can perceive the feeling only for a short time. In the future, we will be ready to stack a larger sample dataset and the modelwill accommodate different sentiments in different time frames. His future work may also include recording time information through the receiver, with the goal of not needing to compile a data set; we just train the model and then the information can be recorded to give the feelings of the individual's voice.

CHAPTER 10 REFERENCES

REFERENCES

[1] B. Schuller, G. Rigoll and M. Lang, "Hidden Markov model-based speech emotion recognition," 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03)., 2003, pp. II-1, doi: 10.1109/ICASSP.2003.1202279.

[2] Moataz El Ayadi, Mohamed S. Kamel, Fakhri Karray, Survey on speechemotion recognition: Features, classification schemes, and databases, Pattern Recognition, Volume 44, Issue 3, 2011, Pages 572-587, ISSN 0031-3203,
https://doi.org/10.1016/j.patcog.2010.09.020..

[3] Koolagudi, S.G., Rao, K.S. Emotion recognition from speech: a review. Int J Speech Technol 15, 99–117 (2012).https://doi.org/10.1007/s10772-011-9125-1.

[4] Nicholson, J., Takahashi, K. &Nakatsu, R. Emotion Recognition in Speech Using Neural Networks. NCA 9, 290–296 (2000). https://doi.org/10.1007/s005210070006.

[5] Siqing Wu, Tiago H. Falk, Wai-Yip Chan, Automatic speech emotion recognition using modulation spectral features, Speech Communication,Volume 53, Issue 5, 2011, Pages 768-785, ISSN 01676393,https://doi.org/10.1016/j.specom.2010.08.013.