

## Speech Emotion Recognition (SER) using Deep Neural Multilayer Perceptron (MLP)

### A PROJECT REPORT

***Submitted by***

## PRASHANT MISHRA [722817104113]

**NAFEES AHMAD [722817104091]**

**NARESH KUMAR [722817104093]**

***in partial fulfillment for the award of the degree of***

## BACHELOR OF ENGINEERING

***in***

COMPUTER SCIENCE AND ENGINEERING

### SRI ESHWAR COLLEGE OF ENGINEERING KINATHUKADAVU, COIMBATORE.

**ANNA UNIVERSITY:: CHENNAI 600 025**

**MARCH 2021**

**ANNA UNIVERSITY: CHENNAI 600 025**

**BONAFIDE CERTIFICATE**

Certified that this project report “**SPEECH EMOTION RECOGNITION (SER) USING DEEP NEURAL MULTILAYER PERCEPTRON”** is the bonafide work of

### PRASHANT MISHRA [722817104113]

**NAFEES AHMAD [722817104091]**

**NARESH KUMAR [722817104093]**

who carried out the project work under my supervision.

### SIGNATURE

……………………………..

### Dr. R. SUBHA, M.E., Ph.D.,

**HEAD OF THE DEPARTMENT**

Department of Computer Science & Engineering,

Sri Eshwar College Of Engineering, Kinathukadavu,

Coimbatore-641 202

### SIGNATURE

……………………………..

### MR. A ANANDARAJ, M.E. (Ph.D.)

Assistant Professor,

Department of Computer Science & Engineering,

Sri Eshwar College Of Engineering, Kinathukadavu,

Coimbatore-641 202

Submitted for the University Project Viva Voce conducted on 31-03-2021.

### Internal Examiner External Examiner

**DECLARATION**



**DECLARATION**

We,

**PRASHANT MISHRA [722817104013]**

**NAFEES AHMAD [722817104055]**

**NARESH KUMAR [722817104112]**

Declare that the project entitled “**Speech Emotion Recognition (SER) using Deep Neural Multilayer Perceptron (MLP)”,** submitted in partial fulfillment to Anna University as the project work of Bachelor of Engineering (Computer Science and Engineering) Degree, is a record of original work done by us under the supervision and guidance of Mr. A. Anandaraj, M.E.,(Ph.D.)., Assistant Professor, Sri Eshwar College of Engineering, Coimbatore.

Place: Coimbatore

Date:

[Prashant Mishra]

[Nafees Ahmad]

[Naresh Kumar]

Project Guided by

---------------------------

[Mr. A. Anandaraj, M.E.,(PhD).,AP/CSE]

**ACKNOWLEDGEMENT**



**ACKNOWLEDGEMENT**

The success of a work depends on a team and cooperation. We take this opportunity to express our gratitude and thanks to everyone who helped us in our project. We would like to thank the management for the constant support provided by them to complete this project.

It is indeed our great honor bounded duty to thank our beloved **Chairman Mr. R. Mohanram,** for his academic interest shown towards the students. We are indebted to our **Director Mr. R. Rajaram,** for motivating and providing us with all facilities.

We wish to express our sincere regards and a deep sense of gratitude to dear **Dr. Sudha Mohanram, M.E, Ph.D., Principal**, for the excellent facilities and encouragement provided during the study and project.

We are indebted to **Dr. R. Subha, M.E., Ph.D.,** Headof Computer Science and Engineering Department for having permitted us to carry out this project and giving us the complete freedom to utilize the resources of the department.

We also extend our heartfelt thanks to our internal project Guide, **Mr. A. ANANDARAJ M.E.,** Ph.D.Assistant Professor of Computer Science and Engineering Department for providing us her support and guidance which helped us.

We solemnly express our thanks to all the teaching and non-teaching staff of the Computer Science and Engineering Department, family, and friends for their valuable support which inspired us to work on this project.

# TABLE OF CONTENTS



## TABLE OF CONTENTS

### CHAPTER NO.

**ABSTRACT**

**TITLE**

**PAGE NO.**

**i**

**LIST OF FIGURES**

**LIST OF ABBREVIATIONS**

**ii iii**

### INTRODUCTION 1

* + 1. Objective **1**

### LITERATURE SURVEY 3

### SYSTEM STUDY 6

* + 1. Problem Definition **7**
    2. Existing System **7**
       1. Drawbacks of the Existing System **7**
    3. Proposed System **8**
       1. Advantages of the Proposed System **10**
    4. Feasibility Study **10**

### MODULES 12

* + 1. Pre-Processing **13**
    2. Feature Extraction **13**
    3. Emotions Recognition **13**
    4. Different Emotions **14**

### REQUIREMENT ANALYSIS 15

* + 1. Functional Requirements **16**
    2. Non-Functional Requirements **16**
    3. Pseudo Requirements **16**
    4. Software Specifications **16**
    5. Hardware Specifications **17**
  1. FUNCTIONING OF SER 22
  2. [TESTING 32](#_TOC_250013)
     1. [Black Box Testing **33**](#_TOC_250012)
     2. [White Box Testing **33**](#_TOC_250011)
     3. [Unit Testing **33**](#_TOC_250010)
     4. System Testing **34**
     5. [Integration Testing **34**](#_TOC_250009)
     6. Functional Testing **34**
  3. [RESULTS 37](#_TOC_250008)
     1. [Speech Recognition **38**](#_TOC_250007)
     2. [Emotion Testing **39**](#_TOC_250005)
     3. [Addition of voice notes to file **39**](#_TOC_250004)
     4. Viewing Content and Users **40**
     5. [**Judgment if mixed emotions 40**](#_TOC_250003)
     6. [Results **41**](#_TOC_250001)
  4. CONCLUSION 42
     1. Future Enhancement **43**

APPENDIX 44

[REFERENCE 49](#_TOC_250000)

**ABSTRACT**



**ABSTRACT**

This project presents an important and efficient method of Speech Emotion Recognition (SER) using Deep Neural Multilayer Perceptron (MLP). Speech Emotion Recognition has now become a necessity for human life. To achieve this study, many SER systems have been generated using various classifiers and different functions. Machine Learning has added efficient progress to this, by recognizing different speech graphs and similarities between them, and then Deep Learning is brought into consideration to study various graphs and perform feature extraction from the speech signals which is used for training different classifiers and retrieving data. Feature extraction was applied to get the most suitable dataset and use it for the most accurate results. The first recurrent neural network (RN) classifier is used to classify the seven emotions. Their performance is later compared with multivariate linear regression (MLR) and support vector machines (SVM) techniques, which are widely used in the field of emotion recognition for spoken audio signals. Berlin and Spanish databases are used as experimental data sets. This study shows that Berlin achieves 83% accuracy of all classifications for the database when a speaker normalization (SN) and a selection of features are applied to the features. For Spanish databases, optimal accuracy (% S%) is achieved by RNN classification without SN and by FS.

Librosa is a python package for music and audio analysis. It provides the building blocks necessary to create music information retrieval systems.

It has a flatter package layout, standardizes interfaces and names, backward compatibility, modular functions, and readable code. Further, in this python project, we demonstrate how to install it (pip).

JupyterLab s an open-source, web-based UI for project Jupyter and it has all the basic functionalities of the Jupyter Notebook, Like notebooks, terminals, text editors, file browsers, rich outputs, and more.

# LIST OF FIGURES



## LIST OF FIGURES

### FIGURE NO. TITLE PAGE NO.

1.1 Overview of the System 9

* 1. Screenshot 38
  2. Actors 38
  3. Voice notes 39
  4. Output 39
  5. General View 40



**LIST OF ABBREVIATIONS**



**LIST OF ABBREVATIONS**

|  |  |
| --- | --- |
| **DB** | - |
| **SE** | - |
| **JL** | - |
| **GUI** | - |
| **IP** | - |
| **MAC** | - |
| **DF** | - |
| **IM** | - |
| **VM** | **-** |
| **URL** | - |
| **JDBC** | - |
|  |  |

DataBase

Speech Emotions

JupyterLab

Manager Graphical User Interface Internet Protocol

Media Access Control Data Flair

Indexing Manager Virtual Machine

Unified Resource Locator

Java Database Connectivity

iii

# CHAPTER 1 INTRODUCTION

### CHAPTER 1 INTRODUCTION

### OBJECTIVE

Emotions – They play an important role in daily interpersonal human interaction. This is necessary for our rational as well as intelligent decisions. It helps to match and understand the feelings of others by expressing our feelings and reacting to others. Research has shown that emotions play a powerful role in shaping human social interactions. Emotional performance conveys significant information about a person's mental state. This has opened up a new field of research called Automatic Spirit Recognition, with the original goal of understanding and achieving the desired emotions.

In previous studies, many methods have been discovered to identify emotional states such as facial expressions, speech, physical gestures, etc. Some inherent advantages make speech cues a source for emotional computing. For example, compared to many other biological signals (e.g., electrocardiogram), speech signals can usually be obtained more easily and economically. That is why most researchers are interested in speech emotion recognition (SER).

The SER aims to identify the emotional state of the speaker from his voice. The area has received increasing interest in research in recent years. There are many applications to explore a person's emotions in interfaces such as robots, audio surveillance, web-based e-learning, commercial applications, clinical studies, entertainment, banking, call centre. cardboard systems, computer games, etc. Orchestration or e-learning in the classroom, information about the emotional state of students can focus on increasing the quality of teaching.

For example, the teacher may use SER to determine which subjects can be taught and should be able to develop strategies to manage emotions within the learning environment. That is why the emotional state of the learner in the classroom should be taken into consideration.

Deep neural networks (DNNs) are based on feedforward structures composed of one or more underlying hidden layers between inputs and outputs. Feed-forward architectures such as Deep Neural Network (DNN) and Convolutional Neural Network (CNN) provide efficient results for image and video processing. On the other hand, repetitive architectures such as recurrent neural networks (RNN) and long short-term memory (LSTM) are very effective in speech-based classifications such as natural language processing (NLP) and SER [1]. In addition to their effective method of classification, these models have some limitations.

For instance, the positive side of CNN is learning features from high-dimensional input data, but on the other hand, it also learns features from small changes and distortion events and, therefore, requires large storage capacity. Similarly, LSTM based RNN, variable input data, and model are capable of handling long-distance sequential text data.

Three key issues need to be addressed for a successful SER system, namely, (1) selecting a positive emotional speech database, (2) collecting effective features, and (3) creating reliable classifications using a machine learning algorithm. Emotional feature extraction is a major problem in the SER system. Research Power, Pitch, Format Frequency, Linear Prediction Spectrum Coefficients (LPCC), Mel-Frequency Spectrum Coefficients (MFCC), Modulation Spectral Features (MSF) [5]]. Therefore, many researchers prefer to use the integrated feature set, which has a wide variety of features that contain more emotional information [6]. However, the use of an integrated feature set can lead to high quality and repetition of speech features. Therefore for most machine learning algorithms it complicates the learning process and increases the risk of overfitting. Therefore, feature selection is inevitable to reduce the frequency of symptoms. Feature selection is presented in a review of models and methods [7]. Feature extraction and feature selection have the potential to improve learning performance, reduce computational complexity, create better generalization models, and reduce required storage. The final step in speech emotion recognition is classification. It involves classifying raw data into a specific type of emotion in the form of pronunciation or the framework of pronunciation based on the characteristics extracted from the data. In recent years of speech emotion recognition, researchers have developed the Gaussian hybrid model (GMM) [8], the latent Markov model (HMM) [9], the support vector machine (SVM) [10, 11, 12, 13, 14], and neural networks ( NN). [15], Repetitive Neural Networks (RNN) [16, 17, 18]. Some researchers have suggested other types of classifications, such as the modified Brain Emotional Learning Model (BEL) [19], which includes the Adaptive Neuro-Fuse Infection System (ANFIS) and Multilayer Perceptron (MLP) for speech-emotional recognition. Another specific strategy is the Multiple Kernel Process Process (GP) classification, which combines a linear kernel and a radial base function (RBF) kernel and provides two concepts similar to the learning algorithm. The Voice Segment Selection (VSS) algorithm proposed in [20] treats the voice signal segment as a different format image processing feature. It uses log-Gabor filters to extract sound and unknown properties from the spectrogram for classification.

There are three classes of features in a speech namely, the lexical features (the vocabulary used), the visual features (the expressions the speaker makes), and the acoustic features (sound properties like pitch, tone, jitter, etc.).

The problem of speech emotion recognition can be solved by analyzing one or more of these features. Choosing to follow the lexical features would require a transcript of the speech which would further require an additional step of text extraction from speech if one wants to predict emotions from real-time audio. Similarly, going forward with analyzing visual features would require the excess to the video of the conversations which might not be feasible in every case while the analysis on the acoustic features can be done in real-time while the conversation is taking place as we’d just need the audio data for accomplishing our task. Hence, we choose to analyze the acoustic features in this work.

Furthermore, the representation of emotions can be done in two ways:

* Discrete Classification: Classifying emotions in discrete labels like anger, happiness, boredom, etc.
* Dimensional Representation: Representing emotions with dimensions such as Valence (on a negative to positive scale), Activation of Energy (on a low to high scale), and Dominance (on an active to passive scale)

Both these approaches have their pros and cons. The dimensional approach is more elaborate and gives more context to prediction but it is harder to implement and there is a lack of annotated audio data in a dimensional format. The discrete classification is more straightforward to implement but it lacks the context of the prediction that dimensional representation provides. We have used the discrete classification approach in the current study for the lack of dimensionally annotated data in the public domain

# CHAPTER 2 LITERATURE SURVEY



### CHAPTER 2 LITERATURE SURVEY

**LITERATURE SURVEY**

Literature survey is the most important step in the software development process. Before developing the tool it is necessary to determine the time factor, economy, and company strength. Once these things are satisfied, the next steps are to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need a lot of external support.

Over the last years, an excessive investigation has been completed to recognize emotions by using speech statistics. Cao et al. [10] proposed a ranking SVM method for synthesizing information about emotion recognition to solve the problem of binary classification. This ranking method, instructs SVM algorithms for particular emotions, treating data from every utterer as a distinct query then mixed all predictions from rankers to apply multi-class prediction. Ranking SVM achieves two advantages, first, for training and testing steps in speaker- independent it obtains speaker specific data. Second, it considers the intuition that each speaker may express mixed of emotion to recognize the dominant emotion. Ranking approaches achieves substantial gain in terms of accuracy compare to conventional SVM in two public datasets of acted emotional speech, Berlin and LDC. In both acted data and the spontaneous data, which comprises neutral intense emotional utterances, ranking-based SVM achieved higher accuracy in recognizing emotional utterances than conventional SVM methods. Unweight average (UA) or Balance accuracy achieved 44.4%.

We found a investigation on a new spectral feature in order to determine emotions and to characterize groups. In this study based on acoustic features and a novel hierarchical classifier, emotions are grouped. Different classifier such as HMM, GMM and MLP have been evaluated with distinct configuration and input features to design a novel hierarchical techniques for classification of emotions. The innovation of the proposed method is two things, first the election of foremost performing features and second is employing of foremost class-wise classification performance of total features same as the classifier. Experimental result in Berlin dataset demonstrates the hierarchical approach achieves the better performance compare to best standard classifier, with decuple cross-validation. For example, performance of standard HMM method reached 68.57% and the hierarchical model reached 71.75%.

From books we found Narayanan proposed domain-specific emotion recognition by utilizing speech signals from call center application. Detecting negative and non-negative emotion (e.g. anger and happy) are the main focus of this research. Different types of information include acoustic, lexical, and discourse are used for emotion recognition. In addition, information-theoretic contents of emotional salience is presented to obtain data at emotion information at the language level. Both k-NN and linear discriminant classifier are used to work with different types of features. Experimental result confirms that the best results are achieved by combination of acoustic and language data. Outcomes demonstrates by combining three information source instead of one source, classification accuracy increases by 40.7% for males and 36.4% for females. Compare to pervious work improvement range in accuracy is from 1.4% to 6.75% for male and 0.75% to 3.96% for female.

We assured a multi-dimensional model by utilizing emotion primitives for speech emotion recognition. Three dimension were made by composing of three different value of emotion primitives, which is 020105-4 called valence, activation, and dominance. The value of these factors assumed to be in the range of [-1, +1]. A textfree, image-based method was introduced to assess the emotion primitives and achieves best inter-evaluator agreement. For extracting acoustic feature such as energy, pith and spectral specifications, both fuzzy logic and rulebased estimator are employed. The approached are validate by testing two EMA and VAM datasets, which are acted emotion and spontaneous speech emotion. Both dataset are recorded form talk-show in USA TV. Finally, for mapping the emotion primitives to certain emotion category, k-NN was employed as a classifier. K-NN achieves total recognition rate up to 83.5%.

# CHAPTER 3 SYSTEM STUDY



### CHAPTER 3 SYSTEM STUDY

### PROBLEM DEFINITION

Generally, analysis on speech recognition aims to increase the recognition rate as well as accuracy. From data which targeted the speech emotion recognition systems, and these are evaluated with its classifier, features set, and recognition rate and on different dataset levels. As it depicts that SVM has many motivating properties such as easy to fulfilment because of mathematical basis compare to simple classifier such as quadratic discriminant analysis (QDC). Normally, SVM is employed as alone and also with combination of other classifier such as ANN and RBF to reduce the dimensionality. It shows most accuracy on according to data-2, where it uses the log frequency power coefficient (LEPC) and MFCC features set and achieve more than 95% in best case.

### EXISTING SYSTEM

Existing system for speech recognition system records voice and tells the feeling/emotions of him/her at different tone level. It then announces the result telling current mood.

### DRAWBACKS OF THE EXISTING SYSTEM

Need to work more on dataset and voice recording quality, as it is not able to record proper voice and hence reducing the accuracy rate, also it needs to tell more about mixed emotions of a person and while showing the results, System can suggest ways in order to improve their current moods.

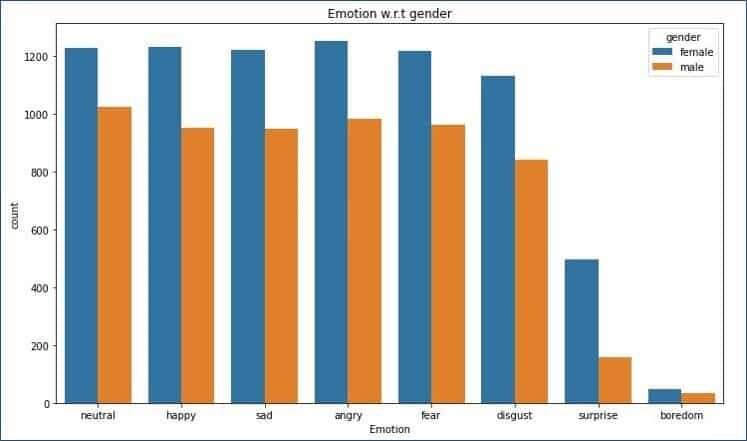
### PROPOSED SYSTEM

We present a secure and reliable system with great accuracy which provides results including all possible mixed emotions as of that time, as well as providing ways to improve those moods.

In speech database voice notes are utilized to validate the proposed methods in speech emotion recognition. Among all dataset Berlin and AIBO are most common used. Burkhardt et al. was recorded by actors in USA language. The place of record was Department of Technical Acoustics of Technical University Berlin. 5 male and 5 female USA actor have participated in providing the dataset by reading one of the chosen sentences. Different recorded emotion are anger, fear, neutral, disgust, happiness and sadness. Another emotional database names Aibo was collected in the real conditions by interacting and playing of fifty-one children with the Sony’s robot Aibo that govern by human operator to extract the children’s spoken speech. In AIBO five collected emotions are positive, neutral, angry, rest and emphatic.

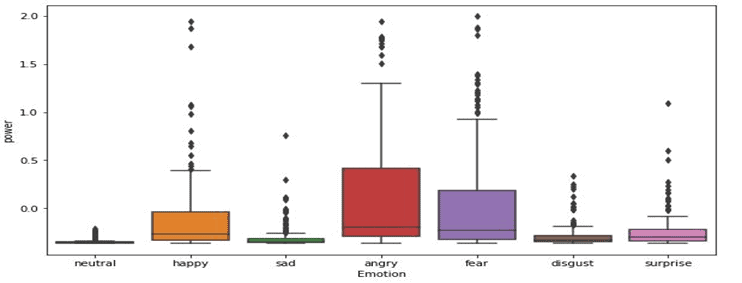
For modelling the emotional states, there are different classification methods utilized to create proper classifier such as support vector machine (SVM), hidden Markov models (HMM) , neural network, Knearest neighbor and Gaussian mixture model (GMM). Conversely, a standard level of classifier may not achieve on very emotional statuses. For example ranking SVM approach cannot leads to considerable improvements in recognition of emotion compare to combination of SVM with radial basis function (RBF). Some hybrid/fusion-based methods achieve high recognition rate compare to individual approaches.

This native sampling functionally keeps the original sonic features in the output array instead of mashing features together. The size of the array gets smaller with a smaller sample rate. The longer array (more detailed audio) can add to processing time later on in workflow. For example computing a mel spectrogram the largest array took almost 4 seconds versus about 1.6 for the resampled smaller array. This additional time is much smaller than the margin(15s to 1.4) from load time, but if your running multiple operations it may add up. If you’re running 100 operations on the array the size of the array the may run up a bigger tab than the load time and you may want to resample to a smaller array.



*Figure 1.1: Overview of the system*

* + - Here we can see difference in count on emotions with respect to gender.



### ADVANTAGES OF PROPOSED SYSTEM

We present a reliable and accurate system with feasible solution of overcoming a bad mood. In our system we distinguish the level of different moods in mixed emotions, and try to detect the emotions causes.

### FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. Three key considerations involved in the feasibility analysis are

ECONOMICAL FEASIBILITY TECHNICAL FEASIBILITY SOCIAL FEASIBILITY

### ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of funds that the company can pour into their search and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

### SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

# CHAPTER 4 MODULES



### CHAPTER 4 MODULES

### PRE-PROCESSING

The emotional states of a speaker are implicitly expressed in human speech together with linguistic information. To realize natural communication between human and computers, the expression of emotions is incorporated into synthesized speech so that the computer is able to convey emotions in speech. This has attracted a lot of interest over the recent years [1-8]. Voice conversion technology aims to automatically transform the voice with a source speaking style into that with a specified target speaking style. This has been used to generate emotional speech by means of conversion from neutral speech. This is desirable as it allows us to generate emotional speech from many existing Text-to-Speech (TTS) systems.

The system for converting neutral speech to emotional speech can be broken down into two stages, namely, training and transformation stages. In the training stage, the transformation function for voice conversion is formulated based on the information from the source and target voices. This function is then employed in the transformation stage to modify the source voice so that the converted speech can match the characteristics of the target voice and portray the targeted speech emotion. The transformation function is derived by analyzing the voice samples of source and target speech. How well the emotional characteristics of the target voice may be portrayed by the converted speech is largely dependent on the quality of the training data used for deriving the transformation model. Quality here refers not only to voice quality but also to the degree of emotion expressed in the training speech.

Another problem encountered in speech emotion research is that there still lacks a precise method of defining and quantifying each emotional state. Consequently, there is no explicit way to express emotions in speech. Methods of expression of different emotions may be widely varied over different cultures, gender and ages. When the training data is recorded, even though the professional actors are employed to deliver utterances conveying different specific emotions, how much of the intended emotion may be accurately perceived by each listener is yet uncertain. To address this, a data pre-processing method is proposed. Firstly, an emotion-detection system is build by combining four classification methods, namely, Probabilistic Neural Network (PNN), Support Vector Machines (SVMs), kNearest Neighbor (KNN), and Linear Discriminant Analysis (LDA). This is to predict the emotional states of the speech samples in the training database.

A selection criterion is used to choose better speech samples based on the accuracy of emotion prediction. A refined and reduced database is then obtained using these selected samples for training purposes. After processing the database, the Gaussian Mixture Model (GMM) linear transformation is utilized to model the mapping function between neutral and emotional speech. With the help of data-preprocessing, we can choose to only use the speech samples that are predicted to better portray the target emotion in training. This may not only improve the naturalness of the generated speech, but also reduce the computational cost incurred at the training stage. The remaining part of this paper is organized as follows. Section II elaborates on the data pre-processing method. Section III presents the GMM-based voice conversion methods. Section IV follows with the experiment results. Section V ends off with the concluding remarks.

### FEATURE EXTRACTION

Emotion can be expressed by speech because speech contains the characteristic parameters that can reflect emotion information [5]. We can extract and observe the change of characteristic parameters to measure the corresponding speech emotional changes. The key above is extracting characteristic parameters of speech emotion from speech signals. The quality of feature extraction directly affects the accuracy of speech emotion recognition. Meanwhile, speech signals contain not only emotional feature information but also the speaker’s own important information, therefore research on how to extract and which speech emotion characteristic parameters to extract are of great importance [6].

### EMOTIONS RECOGNITION

Modulation spectral features (MSFs) are extracted from an auditory-inspired long-term spectro-temporal representation. These features are obtained by emulating the Spectro-temporal (ST) processing performed in the human auditory system and considers regular acoustic frequency jointly with modulation frequency. The steps for computing the ST representation are illustrated in figure 2.In order to obtain the ST representation, the speech signal is first decomposed by an auditory filterbank. The Hilbert envelopes of the critical-band outputs are computed to form the modulation signals. A modulation filterbank is further applied to the Hilbert envelopes to perform frequency analysis. The spectral contents of the modulation signals are referred to as modulation spectra, and the proposed features are thereby named modulation spectral features (MSFs) (Wua et al., 2011). Lastly, the ST representation is formed by measuring the energy of the decomposed envelope signals, as a function of regular acoustic frequency and modulation frequency. The mean of energy, taken over all frames in every spectral band provides a feature. In total, 95 MSFs are calculated in this work from the ST representation.

### DATASET

 We’ll use the RAVDESS dataset; this is the Ryerson Audio-Visual Database of Emotional Speech and Song dataset, and is free to download. This dataset has 7356 files rated by 247 individuals 10 times on emotional validity, intensity, and genuineness. The entire dataset is 24.8GB from 24 actors, but we’ve lowered the sample rate on all the files. The performance and robustness of the recognition systems will be easily affected if it is not well-trained with suitable database. Therefore, it is essential to have sufficient and suitable phrases in the database to train the emotion recognition system and subsequently evaluate its performance. In this section, we detail the two emotional speech databases used in our experiments: Berlin Database and Spanish Database.

# CHAPTER 5 REQUIREMENT ANALYSIS



### CHAPTER 5 REQUIREMENT ANALYSIS

### FUNCTIONAL REQUIREMENTS

Functional requirements specify which output file should be produced from the given file; they describe the relationship between the input and output of the system, for each functional requirement a detailed description of all data inputs and their source and the range of valid inputs must be specified.

### NON FUNCTIONAL REQUIREMENTS

Describe user-visible aspects of the system that are not directly related with the functional behavior of the system. Non-Functional requirements include quantitative constraints, such as response time (i.e. how fast the system reacts to user commands.) or accuracy ((.e. how precise are the system's numerical answers.)

### PSEUDO REQUIREMENTS

The client that restricts the implementation of the system imposes these requirements. Typical pseudo requirements are the implementation language and the platform on which the system is to be implemented. These have usually no direct effect on the users view of the system**.**

### HARDWARE REQUIREMENTS

* System : Intel(R) core(TM) i7 Processsor.
* Hard Disk : 1 TB.
* Monitor : Acer EB321HQ.
* Ram : 8gb.

### SOFTWARE REQUIREMENTS

* Operating system : Windows 10.
* Coding Language : Python
* Tools : VS Code

# CHAPTER 6 Functioning of SER



### CHAPTER 6 Functioning of SER

### EXTRACTION PROCESS

**FEATURE EXTRACTION**

The present acoustic features typically utilized in speech emotion recognition are time-domain features, frequency-domain features, statistical features, deep features and hybrid features. In general, feature extraction depends on the frame involved. A speech signal with a certain duration is divided into frames with an interval ranging from10 to 30ms, while features are then extracted from every frame. Time-domain feature is the direct process of the time domain waveform. This is the most prevalent feature extraction method since it is relatively easy to think of. It can extract features such as short-time zero crossing rate, short-time energy and pitch frequency [8]. Frequency domain feature involves (short-time) Fourier transform wavelet transform and others. Initially, the time-domain signal is transformed into the frequency-domain, from which the feature is extracted. Such features are highly associated with the human perception of speech.

Hence, they have apparent acoustic characteristics. These features are usually comprised of formant frequency, linear prediction cepstral coefficient (LPCC), and Mel frequency cepstral coefficients (MFCC) [9]. They can aptly signify the channel feature [10] [11], indicating noise resistance and good recognition performance [12]. The statistics extracted through a centralized instantaneous data processing refer to statistical features. These features are of an utterance level, while the previous two are frame level. Utterance level feature scan relatively replicate the emotional attributes of speech more deeply [13] [14]. Mean value, extreme value, variance, center moment of each order, and origin moment of each order are some of the common statistical features. Deep features are those extracted by the deep neural network. This generally depicts taking the original speech signal or its spectrum as an input for the deep neural network. RNN [3], CNN [4] and CNN + RNN [5] are some of the common DNNs.

These features can automatically extract features while alleviating the complexity of manual design features. Hybrid features involve all the aforementioned features to form a feature set. For instance, the GeMAPS [15] feature set has 62 statistical features, calculated from18 time-domain and frequency domain features. Meanwhile, the eGeMAPS is an extension of GeMAPS, with 88 features including 18 time-domain and frequency domain features and 5 spectrum features.

**EMOTIONAL CORPUS**

Current models on speech emotion vary depending on the involved datasets, which generally produce a low overall rate of accuracy rate. This is because such data sets are not quite sufficient in scale. Most of the mare comprised of only a few to a dozen hours, which is not a sufficient duration in speech emotion recognition. Indeed, some scholars have put forward transfer learning or multi-task learning technology to resolve lack of data. The rationale involves the use of data among several datasets to enhance the model’s generalization, and to somehow lessen problems resulting from lack of data. Huang et al.[6] initiated a shared-hidden-layer multilingual DNN (SHLMDNN), in which different languages are sharing the hidden layer while the output layer's softmax corresponds to those. This model alleviates the error rate by 3%-5% when compared to the single language DNN that is trained only with a certain language data. It is verified that knowledge sharing among languages can enhance speech emotion recognition accuracy using single language. Later, Zhang et al. [7] employed the multi-task learning method to assess the effect of corpus, domain and gender on speech emotion recognition. It revealed that by increasing the number of corpora, better performance is expected. The reason for the efficacy of this method is that information involving emotions is typical regardless of the language. A Chinese who lacks comprehension of a foreign language may assess whether a non-Chinese expresses happiness or sadness through the language being spoke.

Receiving an accurate emotional response from robots has been a challenging tasks for researchers for the past few years. With the advancements in technology, robots like service robots interact with users of different cultural and lingual backgrounds. The traditional approach towards speech recognition can not be utilized to enable the robot and give an efficient and emotional recognition cannot be utilized to enable the robot and give an efficient and emotional response. The conventional approach towards sufficient duration in speech emotion recognition. Indeed, some scholars have put forward transfer learning or multi-task learning technology to resolve lack of data. The rationale involves the use of data among several datasets to enhance the model’s generalization, and to somehow lessen problems resulting from lack of data. Huang et al.[6] initiated a shared-hidden-layer multilingual DNN (SHLMDNN), in which different languages are sharing the hidden layer while the output layer's softmax corresponds to those. This model alleviates the error rate by 3%-5% when compared to the single language DNN that is trained only with a certain language data.

### 6.1 TECHNOLOGY DESCRIPTION

### The Python Module

The Platform module is used to retrieve as much possible information about the platform on which the program is being currently executed. Now by platform info, it means information about the device, it’s OS, node, OS version, python version, etc. This module plays a crucial role when you want to check whether your program is compatible with the python version installed on a particular system or whether the hardware specifications meet the requirements of your program.

This module already exists in the python library and does not trequire any installation using pip.

A platform is the hardware or software environment in which a program runs. We’ve already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and Mac OS. Most platforms can be described as a combination of the operating system and hardware. The Python platform differs from most other platforms in that it’s a software-only platform that runs on top of other hardware-based platforms.

**MACHINE LEARNING**

Machine learning is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention.

Because of new computing technologies, machine learning today is not like machine learning of the past. It was born from pattern recognition and the theory that computers can learn without being programmed to perform specific tasks; researchers interested in artificial intelligence wanted to see if computers could learn from data. The iterative aspect of machine learning is important because as models are exposed to new data, they are able to independently adapt. They learn from previous computations to produce reliable, repeatable decisions and results. It’s a science that’s not new – but one that has gained fresh momentum.

# CHAPTER 7 TESTING

### CHAPTER 7 TESTING

* 1. **BLACK BOX TESTING**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

### WHITE BOX TESTING

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purposeful. It is used to test areas that cannot be reached from a black box level.

### UNIT TESTING

Unit testing is usually conducted as part of a combined code and unit test phase of the software life cycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach Field testing will be performed manually and functional tests will be written in detail.

Test objectives

All field entries must work properly.

Pages must be activated from the identified link.

The entry screen, messages and responses must not be delayed. Features to be tested

Verify that the entries are of the correct format No duplicate entries should be allowed

All links should take the user to the correct page.

### SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

### INTEGRATION TESTING

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or one step up software applications at the company level interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

### FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system

documentation, and user manuals.Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted. Invalid Input : identified classes of invalid input must be rejected. Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised. Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**TEST CASE TABLE**

A database is a collection of data about a specific topic.

**VIEWS OF TABLE**

We can work with a table in two types,

* + 1. Design View
    2. Data sheet View Design View

To build or modify the structure of a table we work in the table design view. We can specify what kind of data will be held.

Data sheet View

To add, edit or analyse the data itself we work in tables data sheet view

mode.

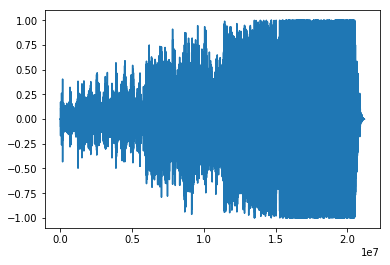
**QUERY**

A query is a question that has to be asked in the data. Access gathers data that answers the question from one or more tables. The data that make up the answer is either a dynaset (if you edit it) or a snapshot (it cannot be edited). Each time we run a query, we get the latest information in the dynaset. Access either displays the dynaset or snapshot for us to view or perform an action on it, such as deleting or update

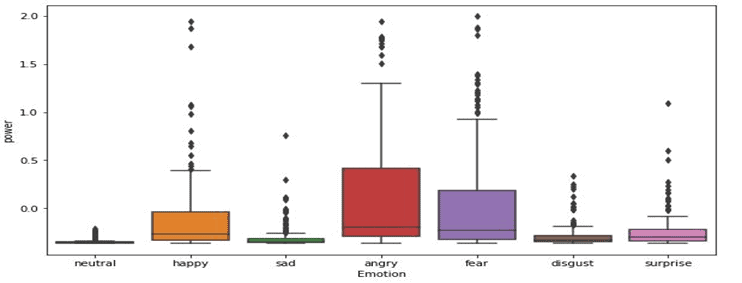
# CHAPTER 8 RESULTS



### CHAPTER 8 RESULTS

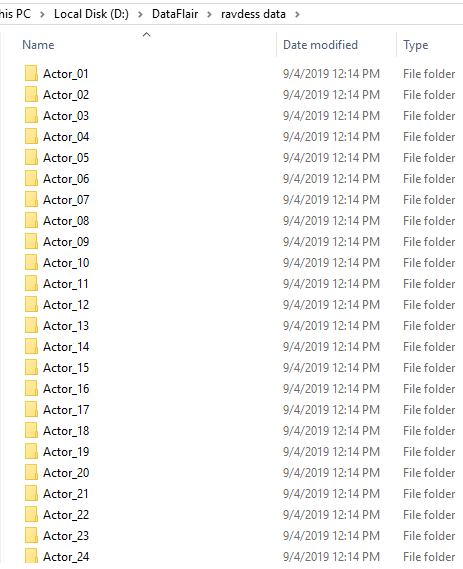
* 1. **SEECH RECOGNITION**
  2. **EMOTION TESTING**

*Figure 8.1:SPEECH RECOGNITION*



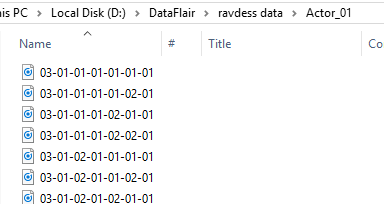
*Figure 8.2: EMOTION TESTING*

### ADDITION OF VOICE NOTES & USERS



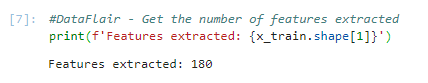
*Figure 8.3: ADDITION OF VOICE NOTES & USERS*

### CONTENT OF VOICE NOTES



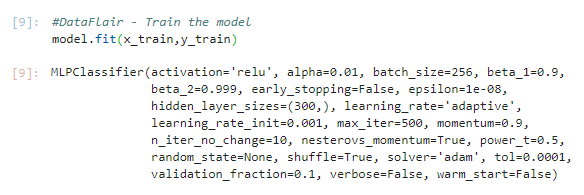
*Figure 8.4: VOICE NOTES*

### JUDGEMENT OF MIXED EMOTION



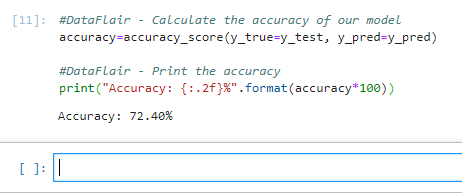
*Figure 8.5: JUDGEMENT OF MIXED EMOTIONS.*

### RESULTS



*Figure 8.6: RESULTS*

### ACCURACY



*Figure 8.7: ACCURACY*

# CHAPTER 9 CONCLUSION



### CHAPTER 9 CONCLUSION

Through this project, we showed how we can leverage Machine learning to obtain the underlying emotion from speech audio data and some insights on 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.

### FUTURE ENHANCEMENT

A few possible steps that can be implemented to make the models more robust and accurate are the following

* Adding facial recognition using AI in order to judge moods and emotions in a better way.
* An accurate implementation of the pace of the speaking can be explored to check if it can resolve some of the deficiencies of the model.
* Figuring out a way to clear random silence from the audio clip.
* Adding up of data using machine learning & deep learning in order to speak results more accurately.

# APPENDIX



### APPENDIX

**SAMPLE CODE**

#Install all the Reqiuired Libraries and Packages

import os

import glob

from tqdm import tqdm

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from scipy.io import wavfile

from python\_speech\_features import mfcc , logfbank

import librosa as lr

import pickle

from scipy import signal

import noisereduce as nr

from glob import glob

#get\_ipython().magic('matplotlib inline')

#All the Required Packages and Libraies are installed.

import soundfile

from tensorflow.keras.layers import Conv2D,MaxPool2D, Flatten, LSTM

from keras.layers import Dropout,Dense,TimeDistributed

from keras.models import Sequential

from keras.utils import to\_categorical

from sklearn.utils.class\_weight import compute\_class\_weight

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

from sklearn.neural\_network import MLPClassifier

from sklearn.metrics import accuracy\_score

#Loading the required RAVDESS DataSet with length of 1439 Audio Files

os.listdir(path='.\speech-emotion-recognition-ravdess-data')

def getListOfFiles(dirName):

listOfFile=os.listdir(dirName)

allFiles=list()

for entry in listOfFile:

fullPath=os.path.join(dirName, entry)

if os.path.isdir(fullPath):

allFiles=allFiles + getListOfFiles(fullPath)

else:

allFiles.append(fullPath)

return allFiles

dirName = './speech-emotion-recognition-ravdess-data'

listOfFiles = getListOfFiles(dirName)

len(listOfFiles)

#Use the Speech-Recognition API to get the Raw Text from Audio Files, Though Speech Recognition

#is less strong for large chunk of files , so used Error Handling , where when it is not be able to

#produce the text of a particular Audio File it prints the statement 'error'.Just for understanding Audio

import speech\_recognition as sr

r=sr.Recognizer()

for file in range(0 , len(listOfFiles) , 1):

with sr.AudioFile(listOfFiles[file]) as source:

audio = r.listen(source)

try:

text = r.recognize\_google(audio)

print(text)

except:

print('error')

#Now Cleaning Step is Performed where:

#DOWN SAMPLING OF AUDIO FILES IS DONE AND PUT MASK OVER IT AND DIRECT INTO CLEAN FOLDER

#MASK IS TO REMOVE UNNECESSARY EMPTY VOIVES AROUND THE MAIN AUDIO VOICE

def envelope(y , rate, threshold):

mask=[]

y=pd.Series(y).apply(np.abs)

y\_mean = y.rolling(window=int(rate/10) , min\_periods=1 , center = True).mean()

for mean in y\_mean:

if mean>threshold:

mask.append(True)

else:

mask.append(False)

return mask

#Plotting the Basic Graphs for understanding of Audio Files :

for file in range(0 , len(listOfFiles) , 1):

audio , sfreq = lr.load(listOfFiles[file])

time = np.arange(0 , len(audio)) / sfreq

fig ,ax = plt.subplots()

ax.plot(time , audio)

ax.set(xlabel = 'Time (s)' , ylabel = 'Sound Amplitude')

plt.show()

#PLOT THE SEPCTOGRAM

for file in range(0 , len(listOfFiles) , 1):

sample\_rate , samples = wavfile.read(listOfFiles[file])

frequencies , times, spectrogram = signal.spectrogram(samples, sample\_rate)

plt.pcolormesh(times, frequencies, spectrogram)

plt.imshow(spectrogram)

plt.ylabel('Frequency [Hz]')

plt.xlabel('Time [sec]')

plt.show()

#Next Step is In-Depth Visualisation of Audio Fiels and its certain features to plot for.

#They are the Plotting Functions to be called later.

def plot\_signals(signals):

fig , axes = plt.subplots(nrows=2, ncols=5,sharex =False , sharey=True, figsize=(20,5))

fig.suptitle('Time Series' , size=16)

i=0

for x in range(2):

for y in range(5):

axes[x,y].set\_title(list(signals.keys())[i])

axes[x,y].plot(list(signals.values())[i])

axes[x,y].get\_xaxis().set\_visible(False)

axes[x,y].get\_yaxis().set\_visible(False)

i +=1

def plot\_fft(fft):

fig , axes = plt.subplots(nrows=2, ncols=5,sharex =False , sharey=True, figsize=(20,5))

fig.suptitle('Fourier Transform' , size=16)

i=0

for x in range(2):

for y in range(5):

data = list(fft.values())[i]

Y,freq = data[0] , data[1]

axes[x,y].set\_title(list(fft.keys())[i])

axes[x,y].plot(freq , Y)

axes[x,y].get\_xaxis().set\_visible(False)

axes[x,y].get\_yaxis().set\_visible(False)

i +=1

def plot\_fbank(fbank):

fig , axes = plt.subplots(nrows=2, ncols=5,sharex =False , sharey=True, figsize=(20,5))

fig.suptitle('Filter Bank Coefficients' , size=16)

i=0

for x in range(2):

for y in range(5):

axes[x,y].set\_title(list(fbank.keys())[i])

axes[x,y].imshow(list(fbank.values())[i],cmap='hot', interpolation = 'nearest')

axes[x,y].get\_xaxis().set\_visible(False)

axes[x,y].get\_yaxis().set\_visible(False)

i +=1

def plot\_mfccs(mfccs):

fig , axes = plt.subplots(nrows=2, ncols=5,sharex =False , sharey=True, figsize=(20,5))

fig.suptitle('Mel Frequency Capstrum Coefficients' , size=16)

i=0

for x in range(2):

for y in range(5):

axes[x,y].set\_title(list(mfccs.keys())[i])

axes[x,y].imshow(list(mfccs.values())[i],

cmap='hot', interpolation = 'nearest')

axes[x,y].get\_xaxis().set\_visible(False)

axes[x,y].get\_yaxis().set\_visible(False)

i +=1

def calc\_fft(y,rate):

n = len(y)

freq = np.fft.rfftfreq(n , d= 1/rate)

Y= abs(np.fft.rfft(y)/n)

return(Y,freq)

# Here The Data Set is loaded and plots are Visualised by Calling the Plotting Functions .

import matplotlib.pyplot as plt

from scipy.io import wavfile as wav

from scipy.fftpack import fft

import numpy as np

for file in range(0 , len(listOfFiles) , 1):

rate, data = wav.read(listOfFiles[file])

fft\_out = fft(data)

get\_ipython().run\_line\_magic('matplotlib', 'inline')

plt.plot(data, np.abs(fft\_out))

plt.show()

signals={}

fft={}

fbank={}

mfccs={}

# load data

for file in range(0 , len(listOfFiles) , 1):

# rate, data = wavfile.read(listOfFiles[file])

signal,rate =librosa.load(listOfFiles[file] , sr=44100)

mask = envelope(signal , rate , 0.0005)

signals[file] = signal

fft[file] = calc\_fft(signal , rate)

bank = logfbank(signal[:rate] , rate , nfilt = 26, nfft = 1103).T

fbank[file] = bank

mel = mfcc(signal[:rate] , rate , numcep =13 , nfilt = 26 , nfft=1103).T

mfccs[file]=mel

plot\_signals(signals)

plt.show()

plot\_fft(fft)

plt.show()

plot\_fbank(fbank)

plt.show()

plot\_mfccs(mfccs)

plt.show()

#Now Cleaning Step is Performed where:

#DOWN SAMPLING OF AUDIO FILES IS DONE AND PUT MASK OVER IT AND DIRECT INTO CLEAN FOLDER

#MASK IS TO REMOVE UNNECESSARY EMPTY VOIVES AROUND THE MAIN AUDIO VOICE

def envelope(y , rate, threshold):

mask=[]

y=pd.Series(y).apply(np.abs)

y\_mean = y.rolling(window=int(rate/10) , min\_periods=1 , center = True).mean()

for mean in y\_mean:

if mean>threshold:

mask.append(True)

else:

mask.append(False)

return mask

#The clean Audio Files are redirected to Clean Audio Folder Directory

import glob,pickle

for file in tqdm(glob.glob(r'C:\Users\Sakshi jain\speech-emotion-recognition-ravdess-data\\\*\*\\\*.wav')):

file\_name = os.path.basename(file)

signal , rate = librosa.load(file, sr=16000)

mask = envelope(signal,rate, 0.0005)

wavfile.write(filename= r'C:\Users\Sakshi jain\clean\_speech\\'+str(file\_name), rate=rate,data=signal[mask])

#Feature Extraction of Audio Files Function

#Extract features (mfcc, chroma, mel) from a sound file

def extract\_feature(file\_name, mfcc, chroma, mel):

with soundfile.SoundFile(file\_name) as sound\_file:

X = sound\_file.read(dtype="float32")

sample\_rate=sound\_file.samplerate

if chroma:

stft=np.abs(librosa.stft(X))

result=np.array([])

if mfcc:

mfccs=np.mean(librosa.feature.mfcc(y=X, sr=sample\_rate, n\_mfcc=40).T, axis=0)

result=np.hstack((result, mfccs))

if chroma:

chroma=np.mean(librosa.feature.chroma\_stft(S=stft, sr=sample\_rate).T,axis=0)

result=np.hstack((result, chroma))

if mel:

mel=np.mean(librosa.feature.melspectrogram(X, sr=sample\_rate).T,axis=0)

result=np.hstack((result, mel))

return result

#Emotions in the RAVDESS dataset to be classified Audio Files based on .

emotions={

'01':'neutral',

'02':'calm',

'03':'happy',

'04':'sad',

'05':'angry',

'06':'fearful',

'07':'disgust',

'08':'surprised'

}

#These are the emotions User wants to observe more :

observed\_emotions=['calm', 'happy', 'fearful', 'disgust']

#Load the data and extract features for each sound file

from glob import glob

import os

import glob

def load\_data(test\_size=0.33):

x,y=[],[]

answer = 0

for file in glob.glob(r'C:\Users\Sakshi jain\clean\_speech\\\*.wav'):

file\_name=os.path.basename(file)

emotion=emotions[file\_name.split("-")[2]]

if emotion not in observed\_emotions:

answer += 1

continue

feature=extract\_feature(file, mfcc=True, chroma=True, mel=True)

x.append(feature)

y.append([emotion,file\_name])

return train\_test\_split(np.array(x), y, test\_size=test\_size, random\_state=9)

#Split the dataset

import librosa

import numpy as np

x\_train,x\_test,y\_trai,y\_tes=load\_data(test\_size=0.25)

print(np.shape(x\_train),np.shape(x\_test), np.shape(y\_trai),np.shape(y\_tes))

y\_test\_map = np.array(y\_tes).T

y\_test = y\_test\_map[0]

test\_filename = y\_test\_map[1]

y\_train\_map = np.array(y\_trai).T

y\_train = y\_train\_map[0]

train\_filename = y\_train\_map[1]

print(np.shape(y\_train),np.shape(y\_test))

print(\*test\_filename,sep="\n")

#Get the shape of the training and testing datasets

# print((x\_train.shape[0], x\_test.shape[0]))

print((x\_train[0], x\_test[0]))

#Get the number of features extracted

print(f'Features extracted: {x\_train.shape[1]}')

# Initialize the Multi Layer Perceptron Classifier

model=MLPClassifier(alpha=0.01, batch\_size=256, epsilon=1e-08, hidden\_layer\_sizes=(300,), learning\_rate='adaptive', max\_iter=500)

#Train the model

model.fit(x\_train,y\_train)

#SAVING THE MODEL

import pickle

# Save the Modle to file in the current working directory

#For any new testing data other than the data in dataset

Pkl\_Filename = "Emotion\_Voice\_Detection\_Model.pkl"

with open(Pkl\_Filename, 'wb') as file:

pickle.dump(model, file)

# Load the Model back from file

with open(Pkl\_Filename, 'rb') as file:

Emotion\_Voice\_Detection\_Model = pickle.load(file)

Emotion\_Voice\_Detection\_Model

#predicting :

y\_pred=Emotion\_Voice\_Detection\_Model.predict(x\_test)

y\_pred

#Store the Prediction probabilities into CSV file

import numpy as np

import pandas as pd

y\_pred1 = pd.DataFrame(y\_pred, columns=['predictions'])

y\_pred1['file\_names'] = test\_filename

print(y\_pred1)

y\_pred1.to\_csv('predictionfinal.csv')

#RECORDED USING MICROPHONE:

import pyaudio

import wave

CHUNK = 1024

FORMAT = pyaudio.paInt16 #paInt8

CHANNELS = 2

RATE = 44100 #sample rate

RECORD\_SECONDS = 4

WAVE\_OUTPUT\_FILENAME = "output10.wav"

p = pyaudio.PyAudio()

stream = p.open(format=FORMAT,

channels=CHANNELS,

rate=RATE,

input=True,

frames\_per\_buffer=CHUNK) #buffer

print("\* recording")

frames = []

for i in range(0, int(RATE / CHUNK \* RECORD\_SECONDS)):

data = stream.read(CHUNK)

frames.append(data) # 2 bytes(16 bits) per channel

print("\* done recording")

stream.stop\_stream()

stream.close()

p.terminate()

wf = wave.open(WAVE\_OUTPUT\_FILENAME, 'wb')

wf.setnchannels(CHANNELS)

wf.setsampwidth(p.get\_sample\_size(FORMAT))

wf.setframerate(RATE)

wf.writeframes(b''.join(frames))

wf.close()

#The file 'output10.wav' in the next cell is the file that was recorded live using the code :

data, sampling\_rate = librosa.load('output10.wav')

get\_ipython().run\_line\_magic('matplotlib', 'inline')

import os

import pandas as pd

import librosa.display

import glob

plt.figure(figsize=(15, 5))

librosa.display.waveplot(data, sr=sampling\_rate)

## Appying extract\_feature function on random file and then loading model to predict the result

file = 'output10.wav'

# data , sr = librosa.load(file)

# data = np.array(data)

ans =[]

new\_feature ,labels = extract\_feature(file, mfcc=True, chroma=True, mel=True)

ans.append(new\_feature)

ans = np.array(ans)

# data.shape

Emotion\_Voice\_Detection\_Model.predict([ans])

# REFERENCES



### REFERENCES

1. M. El Ayadi, M. S. Kamel, and F. Karray, “Survey on speech emotion recognition: Features, classification schemes, and databases,” Pattern Recognit., vol. 44, no. 3, pp. 572–587, Mar. 2011..
2. R. Banse and K. R. Scherer, “Acoustic profiles in vocal emotion expression.,” Pers. Soc. Psychol, vol. 70, no. 3, pp. 572–587, 1996.
3. V. Hozjan and Z. Kačič, “Context-Independent Multilingual Emotion Recognition from Speech Signals,” Int. J. Speech Technol., vol. 6, no. 3, pp. 311–320, 2003.
4. 4. F. Burkhardt, A. Paeschke, M. Rolfes, W. F. Sendlmeier, and B. Weiss, “A database of German emotional speech.,” in Interspeech, 2005, vol.
5. 5, pp. 1517–1520. 5. A. Schuller, B. , Steid, S. l, and Batliner, “The interspeech 2009emotion challengee,” Interspeech, pp. 312– 315, 2009.
6. M. M. H. El Ayadi, M. S. Kamel, and F. Karray, “Speech Emotion Recognition using Gaussian Mixture Vector Autoregressive Models,” in 2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP ’07, 2007, vol. 4, pp. IV–957–IV–960.
7. W. Dai, D. Han, Y. Dai, and D. Xu, “Emotion Recognition and Affective Computing on Vocal Social Media,” Inf. Manag., Feb. 2015.
8. B. S. Atal, “Effectiveness of liner prediction characteristics of the speech wave for automatic speaker speech wave for automatic speaker identification and verification,” Acoust. Soc. Am., vol. 55, no. 6, pp. 1304–1312, 2005.
9. S. Wu, T. H. Falk, and W.-Y. Chan, “Automatic speech emotion recognition using modulation spectral features,” Speech Commun., vol. 53, no. 5, pp. 768–785, May 2011.
10. H. Cao, R. Verma, and A. Nenkova, “Speaker-sensitive emotion recognition via ranking: Studies on acted and spontaneous speech,” Comput. Speech Lang., vol. 28, no. 1, pp. 186–202, Jan. 2015.
11. C.-C. Lee, E. Mower, C. Busso, S. Lee, and S. Narayanan, “Emotion recognition using a hierarchical binary decision tree approach,” Speech Commun., vol. 53, no. 9–10, pp. 1162–1171, Nov. 2011.
12. T. L. Nwe, S. W. Foo, and L. C. De Silva, “Speech emotion recognition using hidden Markov models,” Speech Commun., vol. 41, no. 4, pp. 603–623, Nov. 2003.
13. L. Chen, X. Mao, Y. Xue, and L. L. Cheng, “Speech emotion recognition: Features and classification models,” Digit. Signal Process., vol. 22, no. 6, pp. 1154–1160, Dec. 2012.
14. J.-H. Yeh, T.-L. Pao, C.-Y. Lin, Y.-W. Tsai, and Y.-T. Chen, “Segment-based emotion recognition from continuous Mandarin Chinese speech,” Comput. Human Behav., vol. 27, no. 5, pp. 1545–1552, Sep. 2011.
15. J. Rong, G. Li, and Y.-P. P. Chen, “Acoustic feature selection for automatic emotion recognition from speech,” Inf. Process. Manag., vol. 45, no. 3, pp. 315–328, May 2009.
16. E. M. Albornoz, D. H. Milone, and H. L. Rufiner, “Spoken emotion recognition using hierarchical classifiers,” Comput. Speech Lang., vol. 25, no. 3, pp. 556–570, Jul. 2011.
17. C.-H. Wu and W.-B. Liang, “Emotion Recognition of Affective Speech Based on Multiple Classifiers Using Acoustic-Prosodic Information and Semantic Labels,” IEEE Trans. Affect. Comput., vol. 2, no. 1, pp. 10–21, Jan. 2011



