

INTERDISCIPLINARY PROJECT REPORT
at
Sathyabama Institute of Science and Technology
(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in Computer Science and Engineering

By
C VENKATA NAGA SRIHARSHA
(Reg.No 40731123)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC | 12 B Status
by UGC | Approved by AICTE
JEPPIAR NAGAR, RAJIV GANDHISALAI,
CHENNAI – 600119

APRIL 2023



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFAIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **C VENKATA NAGA SRIHARSHA (40731123)** who carried out the project entitled "**VOICE EMOTION DETECTOR USING DEEP LEARNING**" under my supervision from FEB 2023 to APRIL 2023.

Internal Guide

Mr.Murali

Head of the Department

Dr. S.Vigneshwari, M.E., Ph.D.

Submitted for Viva voce Examination held on 17-04-2023

Internal Examiner

External Examiner

DECLARATION

I **C VENKATA NAGA SRIHARSHA** hereby declare that the project report entitled **VOICE EMOTION DETECTOR USING DEEP LEARNING** done by me under the guidance of **Mr.Murali** at HCL Technologies submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

DATE: 17-04-2023



PLACE: Chennai

SIGNATURE OF THECANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr.T.Sasikala M.E., Ph.D., Dean**, School of Computing, **Dr.S. Vigneshwari, M.E., Ph.D. Head of the Department of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Mr.Murali** for his valuable guidance, suggestions, and constant encouragement that paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project

TRAINING CERTIFICATE

ABSTRACT

Communication is the key to express one's thoughts and ideas clearly. Amongst all forms of communication, speech is the most preferred and powerful form of communications in human. The era of the Internet of Things (IoT) is rapidly advancing in bringing more intelligent systems available for everyday use. These applications range from simple wearables and widgets to complex self-driving vehicles and automated systems employed in various fields. Intelligent applications are interactive and require minimum user effort to function, and mostly function on voice-based input. This creates the necessity for these computer applications to completely comprehend human voice. A voice percept can reveal information about the speaker including gender, age, language, and emotion. Several existing speech recognition systems used in IoT applications are integrated with an emotion detection system in order to analyze the emotional state of the speaker. The performance of the emotion detection system can greatly influence the overall performance of the IoT 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 aims at classifying speech percepts based on emotions, more accurately. While voice emotion recognition has been an active research field since the last three decades, the techniques that deal with the natural environment have only emerged in the last decade. These techniques have reduced the mismatch in the distribution of the training and testing data, which occurs due to the difference in speakers, texts, languages, and recording environments between the training and testing datasets. Although a few good surveys exist for VER, they either don't cover all aspects of VER in natural environments or don't discuss the specifics in detail. This survey focuses on VER in a natural environment, discussing VER techniques for natural environment along with their advantages and disadvantages in terms of speaker, text, language, and recording environments. In the recent past, the deep learning techniques have become very popular due to minimal speech processing and enhanced accuracy. Special attention has been given to deep-learning techniques and the related issues in this survey.

Table Of Contents

Chapter no	Title	Page no
	Abstract	i
	List of abbreviations	iii
	List of Tables	iv
1	Introduction	1
2	Aim & Scope of Project	2
	2.1 Aim	2
	2.2 Scope of the project	2
3	Materials & Methods Algorithms Used	3
	3.1Flow implementation	4
	3.2 Domain specification classification	5
	3.3 Data Set	7
	3.4 Ravdess emotional speech	7
	3.5 Data preparation	11
	3.6 Voice Emotion Detection	11
	3.7 MLP classifier algorithm	14
4	Result and Discussion	18
5	Conclusion	20
	References	22
	Source Code	25

LIST OF ABBREVIATIONS

SHORT FORM

FULL FORM

SER	Speech Emotion Recognition
ML	Machine Learning
PCA	Principal Component Analysis
ANN	Artificial Neural Network
SVM	Support Vector Machine
LFPC	Log Frequency Power Coefficients
HMM	Hidden Markov Mode
MFCC	Mel Frequency Cepstral Coefficients
KEEL	Knowledge Extraction based on Evolutionary Learning
TESS	Toronto Emotional Speech Set

List of Tables

- List of features present in an audio signal
- Pearson correlation coefficient guidelines
- Confusion Matrix
- Results of approach-1 implementation
- Results of approach-2 implementation
- Results of approach-3 implementation
- Comparison of the results
- Results of Ravdess dataset

CHAPTER 1

INTRODUCTION

For several years now, the growth in the field of Artificial Intelligence (AI) has been accelerated. AI, which was once a subject understood by computer scientists only, has now reached the house of a common man in the form of intelligent systems. The advancements of AI have engendered to several technologies involving Human-Computer Interaction (HCI) [1]. Aiming to develop and improve HCI methods is of paramount importance because HCI is the front-end of AI which millions of users experience. Some of the existing HCI methods involve communication through touch, movement, hand gestures, voice and facial gestures [1]. Among the different methods, the voice-based intelligent devices are gaining popularity in a wide range of applications. In a voice-based system, a computer agent is required to completely comprehend the human's speech percept in order to accurately pick up the commands given to it. This field of study is termed as Speech Processing and consists of three components:

- Speaker Identification
- Voice Recognition
- Voice Emotion Detection

Voice Emotion Detection is challenging to implement among the other components due to its complexity. Furthermore, the definition of an intelligent computer system requires the system to mimic human behavior. A striking nature unique to humans is the ability to alter conversations based on the emotional state of the speaker and the listener. Speech emotion detection can be built as a classification problem solved using several machine learning algorithms. This project discusses in detail the various methods and experiments carried out as part of implementing a Voice Emotion Detection system.

CHAPTER 2

AIM AND SCOPE OF THE PRESENT INVESTIGATION

2.1 AIM OF THE PROJECT

To create a machine learning algorithm for voice emotion recognition so that we can know the state of a person whether he is good, neutral or bad

2.2 SCOPE OF THE PROJECT

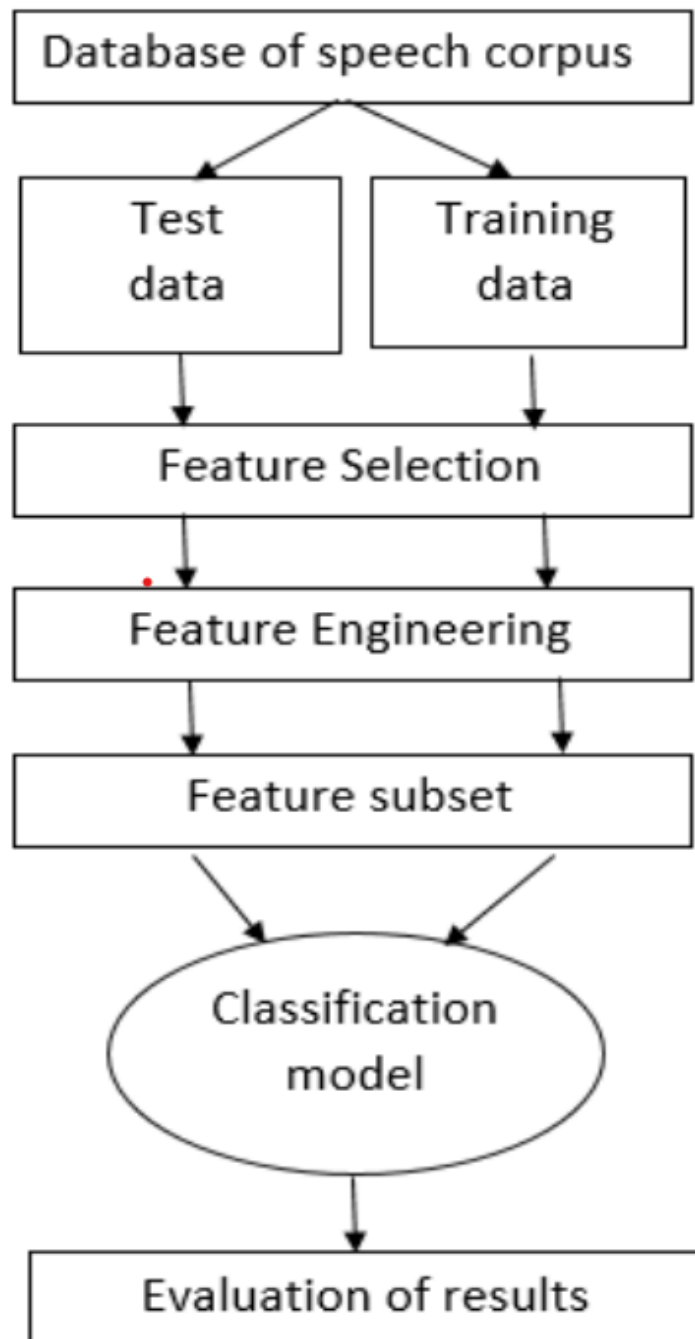
The primary objective of VER is to improve man-machine interface. It can also be used to monitor the psycho physiological state of a person in lie detectors. In recent time, speech emotion recognition also find its applications in medicine and forensics. Detecting emotions is one of the most important marketing strategies in today's world. You could personalize different things for an individual specifically to suit their interest. For this reason, we decided to do a project where we could detect a person's emotions just by their voice which will let us manage many AI-related applications. Some examples could be including call centres to play music when one is angry on the call. Another could be a smart car slowing down when one is angry or fearful. As a result, this type of application has much potential in the world that would benefit companies and even safety to consumers. It is used in forensic to check the person is speaking truth by detecting emotions of person in speech. The scope of this project involve with two different module, training module and testing module. For the training module, the system will be train by the known voice to get the data and store into the reference template. For the testing module, the system will be test with unknown voice and known voice.

CHAPTER 3

EXPERIMENTAL OR MATERIALS AND METHODS,ALGORITHMS USED

In our study, we will be studying about how to create a machine learning algo for Voice emotion recognition.

- The MFCC technique aims to develop the features from the audio signal which can be used for detecting the phones in the speech. But in the given audio signal there will be many phones, so we will break the audio signal into different segments with each segment having 25ms width and with the signal at 10ms.
- Artificial neural networks are one of the main tools used in machine learning. As the “neural” part of their name suggests, they are brain-inspired systems that are intended to replicate the way that we humans learn. Neural networks consist of input and output layers, as well as (in most cases) a hidden layer consisting of units



3.1 Flow of implementation

The Voice 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 over the collected data. These procedures address the several data representation and data quality 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 learn about the data and train itself to respond to any new data it is exposed to. The final step 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.

DIMENSIONALITY REDUCTION METHOD

Chen et al. [8] developed a system that had improvements in the pre-processing stage. Two preprocessing techniques, namely Fisher and Principle Component Analysis (PCA), were used in combination with two classifier algorithms, namely SVM and ANN. They carried out four experiments, each with a different combination of pre-processing and the classifier algorithm. The first experiment used Fisher method to select features for a multi-level SVM classifier (Fisher + SVM). The second experiment was to reduce feature dimensionality using Principle Component Analysis (PCA) for the SVM classifier (PCA + SVM). The third experiment used the Fisher technique over the ANN model (Fisher + ANN). Finally, PCA was applied before classification using ANN (PCA + ANN). From these experiments, two important conclusions were made. Firstly, dimensionality reduction improves the performance of the system. Secondly, SVM classifier algorithm classifies better than the ANN algorithm in the case of emotion detection. The winning experiment had an accuracy of 86.50% using Fisher for dimensionality reduction and SVM for classification.

LPC COEFFICIENT APPROACH

In the Nwe et al. [9] system, a subset of features, similar to the Mel Frequency Cepstral Coefficients (MFCC), was used. They used the Log Frequency Power Coefficients (LFPC) over a Hidden Markov Model (HMM) to classify emotions in speech. Their work is not publically available, as they used a dataset privately available to them. However, they claim that using the LFPC coefficients over the MFFCC coefficients shows a significant improvement in terms of the accuracy of the model. The average classification accuracy in their model is 78% and the best accuracy is even higher 96%.

EXTENDING THE FEATURE SPACE

Rong et al. [10] proposed an innovative way to improve the accuracy of existing models. Traditionally, computer scientists were using various pre-processing techniques to reduce the number of features. Contrastingly, this new system increased the number of features used for classification. They claimed to have performed classification over a small dataset containing audio percepts in the Chinese language, but do not disclose the features that they used. However, they also mentioned that none of their features are language-dependent. Using a high number of features over an ensemble random forest algorithm (ERFTrees), they achieved an accuracy of 82.54%.

3.2 DOMAIN SPECIFIC CLASSIFICATION

Narayanan [11], in his work, proposes a system that uses a more real-world dataset. For his work, data was collected from a call center and he performed a binary classification with only two distinct emotions, namely happy and angry. The research used numerous features including acoustic, lexical and other language-based features over the KNN algorithm. Moreover, this research was conducted specifically for the call center domain and was evaluated across male and female customers. The accuracy values showed improvements of 40.70% and 36.40% in male and female customers, respectively

3.3 DATASET

Two datasets created in the English language, namely the RavdassEmotional Speech Set contain a more diverse and realistic audio. The descriptions of the dataset are as follows

3.4 RAVDASS DATASET

Ravdass is an online dataset repository contributed by machine learning researchers worldwide [13]. The emotion for voice dataset contains 72 features extracted for each of the 593 sound files. The data are labeled across six emotions, namely: Happy, Sad, Angry, Surprise, Fear and Neutral. The repository also offers data to be downloaded in 10 or 5 folds for the purpose of training and testing. A few qualities of this dataset which makes it good for this project are:

- Data is represented as features directly, which saves conversion time and procedures.
 - All basic emotional categories of data are present. A combination of these emotions can be used for further research like Sarcasm and Depression detection
- VOICE EMOTION DETECTION

FEATURE EXTRACTION

THE ProcessVoice is a varying sound signal. Humans are capable of making modifications to the sound signal using their vocal tract, tongue, and teeth to pronounce the phoneme. The features are a way to quantify data. A better representation of the voice signals to get the most information from the speech is through extracting features common among speech signals. Some characteristics of good features include [14]:

- The features should be independent of each other. Most features in the feature vector are correlated to each other. Therefore it is crucial to select a subset of features that are individual and independent of each other.
- The features should be informative to the context. Only those features that are more descriptive about the emotional content are to be selected for further analysis.

- The features should be consistent across all data samples. Features that are unique and specific to certain data samples should be avoided.
- The values of the features should be processed. The initial feature selection process can result in a raw feature vector that is unmanageable. The process of Feature Engineering will remove any outliers, missing values, and null values. VOICE EMOTION DETECTION 23 The features in a speech percept that is relevant to the emotional content can be grouped into two main categories: 1. Prosodic features 2. Phonetic features. The prosodic features are the energy, pitch, tempo, loudness, formant, and intensity. The phonetic features are mostly related to the pronunciation of the words based on the language. Therefore for the purpose of emotion detection, the analysis is performed on the prosodic features or a combination of them. Mostly the pitch and loudness are the features that are very relevant to the emotional content.

Mel Frequency Cepstrum Coefficients (MFCC) FEATURES

A subset of features that are used for speech emotion detection is grouped under a category called the Mel Frequency Cepstrum Coefficients (MFCC) [16]. It can be explained as follows: • The word Mel represents the scale used in Frequency vs Pitch measurement (see Figure 2) [16]. The value measured in frequency scale can be converted into Mel scale using the formula $m = 2595 \log_{10} (1 + (f/700))$

- The word Cepstrum represents the Fourier Transform of the log spectrum of the speech signal.

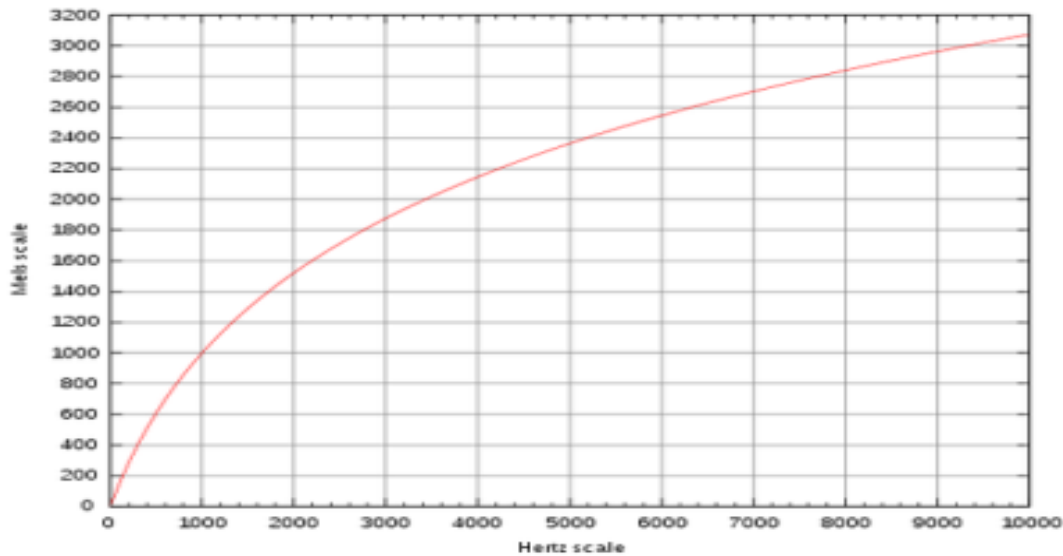


Fig.2 The Mel scale

COEFFICIENT COMPUTATION

Below is the mathematical approach to compute the MFCC features from a speech signal [16]:

1. The first step is to frame the audio signal. The method of frame blocking discussed earlier is used to split the audio signals into frames of an optimal length of 20ms to 30ms, with 50% overlap.
2. The next step is mathematical. In this step, for each frame of the signal, the power spectrum is computed. The power spectrum, also known as Periodogram, identifies the frequencies present in each frame (see Figure 3) [16]. In order to select a particular band of frequencies, the value at each frame is multiplied by a Hamming window value. Mathematically, the periodogram is the squared value of the modulus of the Discrete Fourier Transform (DFT).

2.The final step is to de-correlate the energies. For this purpose, the Discrete Cosine Transform (DCT) function is used. DCT outputs a list of coefficient values corresponding to the pitch and energy values obtained so far. The lower level coefficients (the first 12 to 13 coefficients) of each frame represents steady changes in the pitch and energy values, and therefore they are better for analysis. These lower level coefficients are called the Mel Frequency cepstral coefficients.

FRAME BLOCKING

The frame blocking method is used to analyze sound signals. It is the process of dividing the sound signal into blocks known as frames and performing analysis over each block, rather than the signal at large. It is preferred to analyze individual frames because audio signals are stable within short time intervals. Several acoustic features can be interpreted from a single frame. In order to ensure the time-varying characteristics of the signal are measured accurately, some part of the neighboring frames is also analyzed at every step to identify any subtle changes in the sound signal. This value is often termed as frameAccess Server”, the one with the “Free tier eligible” option and click Select.

overlap, indicating the amount of overlap to include from the neighboring frames. The steps of frame blocking are as follows [17]: 1. Set frame size value to an appropriate number. Each frame should not be too small or too large, as this would mislead the time-varying characteristics of the features. Standard framing window size is 20ms to 30ms for audio signal processing. 2. Set frame overlap value. If the overlap value is too large, more duration from the neighboring frames will need to be analyzed at each analysis step. This will increase the computation and hence is not recommended. Ideally, $\frac{1}{2}$ or $\frac{1}{3}$ of frame overlap is suggested. 3. Perform analysis on each frame Each frame is a unit of computation of the sound signal. Feature extraction of frames will quantify the acoustic features of the audio signal.

SILENCE REMOVAL

An audio signal at the time of recording can accommodate silent regions where no utterances had been made. Such silent regions of the audio signal do not provide any useful information regarding the emotion expressed, and can be removed. A semi-supervised learning approach is used for silence detection in audio signals. In this approach, a model is initially trained with sample audio signals in order to be able to distinguish between high and low energy features [15]. Later, a percentage of high and low energy frames are used as endpoints to detect the regions of actual audio in the signal. Finally, applying the trained model over the entire signal will provide silence-free audio segments [15]. Threshold values such as frame size, frame step, and sampling frequency are tunable in order to smooth the output signals.

3.5 DATA PREPARATION

DATA QUALITY ISSUES

Data must be cleaned to perform any meaningful analysis. As a next step, the dataset thus collected had to be inspected for its quality. Some of the data quality issues addressed for this experimentation includes: 1. Missing value analysis 2. Outlier identification 3. Null value handling 4. Invalid data 5. Duplicate data

5.1.1 MISSING VALUE ANALYSIS

Due to several influencing factors, a few or more data rows can contain no values for specific features. These values are termed 'missing' from the dataset. A large number of missing values can provide insights into the data. For example, if a particular feature has most of its values missing for all data rows, then it can be inferred that the feature is likely uncommon and can be removed from the dataset. Contrastingly, a small number of missing values can represent data entry error. Analyzing and amending individual features over missing values will improve and fix the quality of the dataset. Some of the methods to handle missing values include; if the number of missing values is large then the corresponding data rows or features can be removed, whereas if very few values are missing for a feature it can be imputed which means replacing with the mean or most frequent value of the feature.

3.6 VOICE EMOTION DETECTION

OUTLIER IDENTIFICATION Outlier values are also considered as data modifiers because often the prediction algorithm used will be misled by the outlier values. Outliers also alter the statistics of the overall data such as mean, variance and standard deviation. The proportion of the outliers amongst the whole dataset can be used to make decisions on how to handle them. If the outliers lie within a small range of difference and contribute to a very small proportion, then no fixes will be required. Some methods to handle large proportions of outliers include, replacing outlier values with boundary, or mean, or median, or mode values.

NULL VALUE HANDLING

A common error that can occur in a dataset is the null value error. It is when the words 'null' or 'NA' is used in place of missing values as fillers. Null values are mostly treated and handled in a fashion similar to missing values.

INVALID DATA

A dataset can have values irrelevant to the data type, such as symbols and special characters. These values, despite being meaningless, can cause errors during processing. Depending on the amount of invalid data present, it can either be removed or imputed.

DUPLICATE DATA

Few features might be a duplicate of each other, with different names or units of measurement. Such features increase the dimensionality of the data with no further significance. Removal of duplicate features is highly recommended.

VOICE EMOTION DETECTION 31

NORMALIZATION AND STANDARDIZATION

Different characteristics of the audio signal, represented by its features, are computed on different units or scales. Rescaling the values to a uniform range will ensure accurate

calculations are made. Many algorithms use distance metrics for their computation. Therefore it is necessary that all the values in the dataset are normalized. Two approaches are commonly used for the purpose of rescaling, namely Normalization and Standardization. Normalization alters all numeric values to lie in the range 0 to 1. For this purpose, all outliers in the data must be eliminated prior to normalizing the data. The formula for normalization is given as $x_{\text{new}} = (x - x_{\text{min}}) / (x_{\text{max}} - x_{\text{min}})$. In the formula, x represents the data [18]. Standardization transforms the data to have a mean value of zero and a variance of one. However, standardizing the data provides more insights into the data than normalization. The formula for standardization is given as $x_{\text{new}} = (x - \mu) / \sigma$. In the formula, x represents the data [18].

PRINCIPAL COMPONENT ANALYSIS

In some cases, most or all of the features might have an impact on the decision making. However, a high dimensional dataset with a large feature space could potentially slow the performance of the system in terms of space and time complexity. Choosing the right set of features for analysis can be challenging, as it requires high-level domain knowledge. A solution to this problem is the Principle Component Analysis (PCA) technique for dimensionality reduction. PCA is an approach to bring out the principal components or the important aspects of the data. By using this method, the original feature space of the data is transformed into a new set of features while retaining the variation present in the data [21]. The technique of PCA analyses the variance of the data by measuring the covariance between the features. This is done mathematically using the concept of eigenvalues and eigenvectors. Eigenvalues are 36 numbers denoting the value of variance in each dimension of the data, and the eigenvector is the dimension with the highest eigenvalue. This eigenvector is the principal component of the dataset [22]. Given below are the implementation steps of PCA [21].

DATA NORMALIZATION

PCA works with the numerical values of the dataset to compute the variance, hence it is necessary that the values are scaled and normalized. All normalized data variables

will have a mean value of 0. 5.5.2 COVARIANCE MATRIX COMPUTATION An $N \times N$ covariance matrix is computed, where N is the number of features in the dataset. The elements of the covariance matrix represent the variance between each of the features in the dataset.

CHOOSING COMPONENTS

The eigenvector component with the largest eigenvalue is the 1st Principal component, containing the most information about the dataset. Sorting the eigenvalues in decreasing order can give the list of principal components with the amount of variance needed. Depending on how much VOICE EMOTION DETECTION 37 information is needed, programmers can choose the top P number of components needed for further analysis.

FORMING PRINCIPLE COMPONENTS

A new dataset is created using the principal components selected for analysis. Mathematically, left multiplication of the transposed feature vector with the scaled original dataset will produce the new dataset. $\text{New Dataset} = (\text{Feature Vector})^T \times (\text{Scaled Data})^T$

SCREE PLOT

Scree plot is a way to select the optimal number of components to be selected such that enough information is being retained from the raw dataset. It is a curve plot having the information maintained and the number of components on the different axis. The elbow point of the curve indicates the optimal value of components to be used for further analysis.

3.7 ALGORITHMS

MLP CLASSIFIER

MLP classification

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.

The neurons in the MLP are trained with the **back propagation learning algorithm**. MLPs are designed to approximate any continuous function and can solve problems which are not linearly separable. The major use cases of MLP are pattern classification, recognition, prediction and approximation.

DECISION TREE

The Decision tree is an approach where the entire dataset is visualized as a tree, as the classification problem is loosely translated as finding the path from the root to leaf based on several decision conditions at each sub-tree level. Each feature in the dataset is treated as a decision node at its sub-tree level. The initial tree is designed using the values of the training sample. For a new data point, its values are tracked on the built tree from root to leaf where the leaf node represents the target or predicted class. There are two types of decision trees based on the type of the target class, namely [29]: 1. Binary variable decision tree – for binary class problems 2. Continuous variable decision tree – for continuous value problem The Decision tree is a very simple algorithm to implement, as it requires less data preparation. It is also widely used in data exploration. However, decision trees are very susceptible to noises in the dataset. This might lead to the problem of overfitting. The time complexity of decision tree depends on the height of the tree controlled by the number of data samples, and by the number of features used for the split. It can be represented as $O(n \text{ samples } n \text{ features } \log(n \text{ samples}))$. VOICE EMOTION DETECTION

Random forest

Random forest is a supervised classification similar to decision trees. While the root node and splitting features in the decision tree are based on the Gini and Information gain values, the random forest algorithm does it in a random fashion. The random forest is a collection of decision trees; therefore, a large number of trees gives better results. Overfitting is a potential drawback of random forests, but increasing the number of trees can reduce overfitting. Random forest also has several advantages like its capability to handle missing values and classify multi-class categorical variables. The steps of building a random forest classifier are as follows [30]: 1. Select a subset of features from the dataset. 2. From the selected subset of features, using the best split method, pick a node. 3. Continue the best split method to form child nodes from the subset of features. 4. Repeat the steps until all nodes are used as split. 5. Iteratively create n number of trees using steps 1 -4 to form a forest. The time complexity of Random Forest is higher by the factor of the number of trees used for building the forest.

IMPLEMENTATION

DATA COLLECTION

The first step in implementing the Voice Emotion Recognition system is to collect audio samples under different emotional categories which can be used to train the model. The audio samples are usually wav or mp3 files and publically available for download. The following steps are explained relative to the experiments performed on the RAVDASS dataset.

PYTHON LIBRARY

The next step after data collection was to represent these audio files numerically, in order to perform further analysis on them. This step is called feature extraction, where quantitative values for different features of the audio is obtained. The pyAudioAnalysis library was used for this purpose [15]. This python library provides functions for short-term feature extraction, with tunable windowing parameters such as frame size and frame step. At the end of this step, each audio file was represented as a row in a CSV file with 34 columns representing the different features. Each feature will have a range

of values for one audio file obtained over the various frames in that audio signal. The python library pyAudioAnalysis is an open Python library that provides a wide range of audio-related functionalities focusing on feature extraction, classification, segmentation, and visualization issues. The library depends on several other libraries which are: •Numpy• Matplotlib •ScipyVOICE EMOTION DETECTION 47 •Sklearn•Hmmlearn•Simplejson• eyeD3 •pydub

DATA VISUALIZATION

Visualizing the data gives more understanding of the problem and the type of solution to be built. The distribution of classes, the number of instances under each category, the spread of the data, the correlation between the features and clustering are a few methods to visualize the data. Python and R provide statistical functions for data visualization.

CHAPTER 4

RESULTA AND DISCUSSIONS

EVALUATION

4.1EVALUATION METRICS

The most important characteristic of machine learning models is its ability to improve. Once the model is built, even before testing the model on real data, machine learning experts evaluate the performance of the model. Evaluation metrics reveal important model parameters and provides numeric scores that will help judge the functioning of the model. The most important metric needed to evaluate the model is the confusion matrix (see Figure 8) [32]. The structure of a confusion matrix is against the actual and predicted positive and negative classes, and contains four values which are used to compute other metrics. The true positive represents the correct predictions made in the positive class, and the true negatives represent the correct predictions made in the negative class. The false positives and false negatives are the observations wrongly predicted for their respective classes.

Actual Class	Predicted class		
		Class = Yes	Class = No
	Class = Yes	True Positive	False Negative
	Class = No	False Positive	True Negative

Fig.8 Confusion Matrix

Four important metrics can be derived using the values in the confusion matrix, namely [31]:

4.2 ACCURACY

It is the ratio of the observations predicted correctly to the total number of observations. Accuracy works best for datasets with an equal class distribution, and hence it is not always a good measure to evaluate the model. Accuracy can be computed as $\text{Accuracy} = (\text{True Positives} + \text{True Negatives}) / (\text{True Positives} + \text{False positives} + \text{True Negatives} + \text{False Negatives})$

4.3 PRECISION

It is the ratio of the positive observations predicted correctly to the total positive observations predicted. Higher the value of precision, better and more accurate the model actually is. Precision can also work with an uneven class distribution. It can be computed as $\text{Precision} = \text{True Positives} / (\text{True positives} + \text{False positives})$

4.4 RECALL OR SENSITIVITY

It is the ratio of the positive observations predicted correctly to the total positive observations. A recall score of 50% and more reveals a good performing model. Recall can also work with an uneven class distribution. It can be computed as $\text{Recall} = \text{True positives} / (\text{True positives} + \text{False negatives})$

CHAPTER 5

CONCLUSION

We can work on different employing models like :

- 1. Decision tree**
- 2. GRU**
- 3. LSTM**
- 4. Loss accuracy plot using CNN**
- 5. Confusion matrix using CNN**

6. Classification report using CNN

Thus from the above project ie voice emotion detection the systems can adapt their responses and behavioral patterns according to the emotions of the humans and make the interaction more natural.

The emerging growth and development in the field of AI and machine learning have led to the new era of automation. Most of these automated devices work based on voice commands from the user. Many advantages can be built over the existing systems if besides recognizing the words, the machines could comprehend the emotion of the speaker (user). Some applications of a speech emotion detection system are computer-based tutorial applications, automated call center conversations, a diagnostic tool used for therapy and automatic translation system. In this thesis, the steps of building a speech emotion detection system were discussed in detail and some experiments were carried out to understand the impact of each step. Initially, the limited number of publically available speech database made it challenging to implement a well-trained model. Next, several novel approaches to feature extraction had been proposed in the earlier works, and selecting the best approach included performing many experiments.

Finally, the classifier selection involved learning about the strength and weakness of each classifying algorithm with respect to emotion recognition. At the end of the experimentation, it can be concluded that an integrated feature space will produce a better recognition rate when compared to a single feature. For future advancements, the proposed project can be further modeled in terms of efficiency, accuracy, and usability. Additional to the emotions, the model can be extended to recognize feelings such as depression and mood changes. Such systems can be used by therapists to monitor the mood swings of the patients. A challenging product of creating machines with emotion is to incorporate a sarcasm detection system. Sarcasm detection is a more complex problem of emotion detection since sarcasm cannot be easily identified using only the words or tone of the speaker. A sentiment detection using vocabulary, can be integrated with speech emotion detection to identify a possible sarcasm. Therefore, in the future, there would emerge many applications of a speech-based emotion recognition system.

REFERENCES

- [1] Soegaard, M. and Friis Dam, R. (2013). The Encyclopedia of Human-Computer Interaction. 2nd ed.
- [2] Developer.amazon.com. (2018). Amazon Alexa. [online] Available at: <https://developer.amazon.com/alexa>
- [3] Store.google.com. (2018). Google Home Tips & Tricks – Google Store. [online] Available at: https://store.google.com/product/google_home_learn
- [4] Apple. (2018). iOS - Siri. [online] Available at: <https://www.apple.com/ios/siri/>
- [5] The Official Samsung Galaxy Site. (2018). What is S Voice?. [online] Available at: <http://www.samsung.com/global/galaxy/what-is/s-voice/>
- [Accessed 2 May 2018].
- [6] Gartner.com. (2018). Gartner Says 8.4 Billion Connected. [online] Available at: <https://www.gartner.com/newsroom/id/3598917>
- [7] 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.
- [8] 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.
- [9] 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.
- [10] 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

- [11] S. S. Narayanan, "Toward detecting emotions in spoken dialogs," IEEE Trans. Speech Audio Process., vol. 13, no. 2, pp. 293–303, Mar. 2005.
- [12] Dupuis, K. and Pichora-Fuller, M. (2010). [Collection] University of Toronto, Psychology Department, Toronto emotional speech set (TESS). Toronto.
- [13] J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera. KEEL DataMining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. Journal of Multiple-Valued Logic and Soft Computing 17:2-3 (2011) 255-287.
- [14] S, Khalid, T, Khalil and S, Nasreen. (2014). 2014 Science and Information Conference, A survey of feature selection and feature extraction techniques in machine learning. PP.372-378.
- [15] Giannakopoulos, T. (2018). pyAudioAnalysis. [online] GitHub. Available at: <https://github.com/tyiannak/pyAudioAnalysis>
- [16] Practicalcryptography.com. (2018). Practical Cryptography. [online] Available at: <http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstralcoefficients-mfccs/>
- [17] Dsp.stackexchange.com. (2018). Framing an audio signal. [online] Available at: <https://dsp.stackexchange.com/questions/27243/framing-an-audio-signal>
- [18] Dataminingblog.com. (2018). Standardization vs. normalization | Data Mining Blog - www.dataminingblog.com. [online] Available at: <http://www.dataminingblog.com/standardization-vsnormalization/>
- [19] Statistics.laerd.com. (2018). Pearson Product-Moment Correlation - When you should run this test, the range of values the coefficient can take and how to measure strength of association.. [online] Available at: <https://statistics.laerd.com/statistical-guides/pearson-correlation-coefficient-statisticalguide.php>

[20] Trevino, A. (2018). Introduction to K-means Clustering. [online] Datascience.com. Available at: <https://www.datascience.com/blog/k-means-clustering>.

[21] DeZyre. (2018). Principal Component Analysis Tutorial. [online] Available at: <https://www.dezyre.com/data-science-in-python-tutorial/principal-component-analysis-tutorial>.

[22] George Dallas. (2018). Principal Component Analysis 4 Dummies: Eigenvectors, Eigenvalues and Dimension Reduction. [online] Available at: <https://georgemdallas.wordpress.com/2013/10/30/principalcomponent-analysis-4-dummies-eigenvectors-eigenvalues-and-dimension-reduction/>.

[23] Analytics Vidhya. (2018). Simple Guide to Logistic Regression in R. [online] Available at: <https://www.analyticsvidhya.com/blog/2015/11/beginners-guide-on-logistic-regression-in-r/>.

[24] En.wikipedia.org. (2018). Logistic regression. [online] Available at: https://en.wikipedia.org/wiki/Logistic_regression.

[25] Ray, S. (2018). 6 Easy Steps to Learn Naive Bayes Algorithm (with code in Python). [online] Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/>.

[26] En.wikipedia.org. (2018). Bayes' theorem. [online] Available at: https://en.wikipedia.org/wiki/Bayes%27_theorem

[27] Ray, S. (2018). Understanding Support Vector Machine algorithm from examples (along with code). [online] Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/>.

[28] Srivastava, T. (2018). Introduction to KNN, K-Nearest Neighbors : Simplified. [online] Analytics Vidhya. Available at:

<https://www.analyticsvidhya.com/blog/2014/10/introduction-k-neighboursalgorithm-clustering/>.

[29] Ray, S. (2018). Decision Tree | Predictive Analytics. [online] Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2015/01/decision-tree-simplified/2/>.

[30] (2018). How Random Forest Algorithm Works in Machine Learning. [online] Available at: <https://medium.com/@Synced/how-random-forest-algorithm-works-in-machine-learning3c0fe15b6674>.

[31] Brownlee, J. (2018). A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning - Machine Learning Mastery. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machinelearning/>.

[32] Exsilio Blog. (2018). Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures - Exsilio Blog. [online] Available at: <http://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/>

SOURCE CODE

```
pip install SoundFile
```

```
import pandas as pd
import numpy as nup
import glob
import soundfile
import os
import sys
```

```
!pip install librosa
```

```

from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score


from IPython.display import Audio


import warnings
if not sys.warnoptions:
    warnings.simplefilter("ignore")
warnings.filterwarnings("ignore", category=DeprecationWarning)


RavdessData= "C:\\Users\\HarshCVN\\Downloads\\HCL\\speech-emotion-recognition-
               ravdess-data\\"


import os
import pandas
ravdessDirectoryList = os.listdir(RavdessData)
fileEmotion = []
filePath = []
for dir in ravdessDirectoryList:
    actor = os.listdir(RavdessData + dir)
    for file in actor:
        part = file.split(".")[0]
        part = part.split("-")
        fileEmotion.append(int(part[2]))
        filePath.append(RavdessData + dir + '/' + file)
emotion_df = pandas.DataFrame(fileEmotion, columns=['Emotions'])
path_df = pandas.DataFrame(filePath, columns=['Path'])
Ravdess_df = pandas.concat([emotion_df, path_df], axis=1 )

```

```
Ravdess_df.Emotions.replace({1:'neutral', 2:'calm', 3:'happy', 4:'sad', 5:'angry', 6:'fear',  
7:'disgust', 8:'surprise'}, inplace=True)  
Ravdess_df.head(50)
```

```
data_path = pandas.concat([Ravdess_df], axis = 0)  
data_path.to_csv("data_path.csv",index=False)  
data_path.head()
```

```
pip install matplotlib
```

```
pip install seaborn
```

```
pip install seaborn[stats]
```

```
from matplotlib import pyplot as plt  
import seaborn as sbn  
import librosa.display  
sbn.countplot(x='Emotions', data=data_path)  
plt.title('Count of Emotions', size=16)  
plt.xlabel('Emotions', size=24)  
plt.ylabel('Count', size=24)  
sbn.despine(top=True, right=True, left=True, bottom=True)  
plt.show()
```

```
def create_waveplot(data, sr, e):  
    plt.figure(figsize=(10, 3))  
    plt.title('Waveplot for audio with {} emotion'.format(e), size=15)  
    librosa.display.waveshow(data, sr=sr)  
    plt.show()
```

```

def create_spectrogram(data, sr, e):
    # stft function converts the data into short term fourier transform
    X = librosa.stft(data)
    Xdb = librosa.amplitude_to_db(abs(X))
    plt.figure(figsize=(12, 3))
    plt.title('Spectrogram for audio with {} emotion'.format(e), size=15)
    librosa.display.specshow(Xdb, sr=sr, x_axis='time', y_axis='hz')
    #librosa.display.specshow(Xdb, sr=sr, x_axis='time', y_axis='log')
    plt.colorbar()

```

```

emotion='fear'
path = nup.array(data_path.Path[data_path.Emotions==emotion])[1]
data, sampling_rate = librosa.load(path)
librosa.display.waveshow(data, sr=sampling_rate)
create_waveplot(data, sampling_rate, emotion)
create_spectrogram(data, sampling_rate, emotion)
Audio(path)

```

```

emotion='angry'
path = nup.array(data_path.Path[data_path.Emotions==emotion])[1]
data, sampling_rate = librosa.load(path)
librosa.display.waveshow(data, sr=sampling_rate)
create_waveplot(data, sampling_rate, emotion)
create_spectrogram(data, sampling_rate, emotion)
Audio(path)

```

```

emotion='happy'
path = nup.array(data_path.Path[data_path.Emotions==emotion])[1]
data, sampling_rate = librosa.load(path)
librosa.display.waveshow(data, sr=sampling_rate)
create_waveplot(data, sampling_rate, emotion)

```

```
create_spectrogram(data, sampling_rate, emotion)
Audio(path)
```

```
def noise(data):
    noiseAmp = 0.035*nup.random.uniform()*nup.amax(data)
    data = data + noiseAmp*nup.random.normal(size=data.shape[0])
    return data
```

```
def stretch(data, rate=0.8):
    return librosa.effects.time_stretch(data, rate=0.8)
```

```
def shift(data):
    shiftRange = int(nup.random.uniform(low=-5, high = 5)*1000)
    return nup.roll(data, shiftRange)
```

```
def pitch(data, samplingRate, pitchFactor=0.7):
    return librosa.effects.pitch_shift(data, sr=sampling_rate, n_steps=0.7)
```

```
x = noise(data)
mpl.figure(figsize=(14,4))
librosa.display.waveshow(y=x, sr=sampling_rate)
Audio(x, rate=sampling_rate)
```

```
x = stretch(data)
mpl.figure(figsize=(14,4))
librosa.display.waveshow(y=x, sr=sampling_rate)
Audio(x, rate=sampling_rate)
```

```
x = shift(data)
mpl.figure(figsize=(14,4))
librosa.display.waveshow(y=x, sr=sampling_rate)
```

```
Audio(x, rate=sampling_rate)
```

```
x = pitch(data, sampling_rate)
mpl.figure(figsize=(14,4))
librosa.display.waveshow(y=x, sr=sampling_rate)
Audio(x, rate=sampling_rate)
```

```
def extractFeature(fileName, mfcc, chroma, mel):
    with soundfile.SoundFile(fileName) as soundFile:
        X = soundFile.read(dtype="float32")
        sampleRate=soundFile.samplerate
        if chroma:
            stft=nup.abs(librosa.stft(X))
            result=nup.array([])
            if mfcc:
                mfccs=nup.mean(librosa.feature.mfcc(y=X, sr=sampleRate, n_mfcc=40).T,
axis=0)
                result=nup.hstack((result, mfccs))
            if chroma:
                chroma=nup.mean(librosa.feature.chroma_stft(S=stft, sr=sampleRate).T,
axis=0)
                result=nup.hstack((result, chroma))
            if mel:
                mel=nup.mean(librosa.feature.melspectrogram(y=X, sr=sampleRate).T,
axis=0)
                result=nup.hstack((result, mel))
        return result
```

```
emotions={
    '01':'neutral',
    '02':'calm',
```

```

'03':'happy',
'04':'sad',
'05':'angry',
'06':'fearful',
'07':'disgust',
'08':'surprised'
}
observedEmotions=['calm', 'happy', 'fearful', 'disgust']

from sklearn.model_selection import train_test_split
def loadData(test_size=0.2):
    x,y=[],[]
    for file in glob.glob("C:\\Users\\HarshCVN\\Downloads\\HCL\\speech-emotion-
        recognition-ravdess-data \\Actor_.*\\*.wav"):
        fileName=os.path.basename(file)
        emotion1=emotions[fileName.split("-")[2]]
        if emotion1 not in observedEmotions:
            continue
        feature=extractFeature(file, mfcc=True, chroma=True, mel=True)
        x.append(feature)
        y.append(emotion1)
    return train_test_split(nup.array(x), y, test_size=test_size, random_state=9)

x_train,x_test,y_train,y_test=loadData(test_size=0.23)

print((x_train.shape[0], x_test.shape[0]))

print(f'Features extracted: {x_train.shape[1]}')

model=MLPClassifier(alpha=0.01,batch_size=256,epsilon=1e-08,
hidden_layer_sizes=(300,), learning_rate='adaptive', max_iter=500)

```

```
model.fit(x_train, y_train)
```

```
expected_Of_y = y_test
```

```
yPred=model.predict(x_test)
```

```
print(metrics.confusion_matrix(expected_Of_y,yPred))
```

```
print(classification_report(y_test,yPred))
```

```
accuracy=accuracy_score(y_true=y_test, y_pred=yPred)
```

```
print("Accuracy: {:.2f}%".format(accuracy*100))
```