Applying Machine Learning Techniques for Speech Emotion Recognition

[1]Pratik Game, [2]Rahul Kokate, [3]Apeksha Shinde, [4]Mahesh Wandhekar

Department of Computer Engineering,
Sanjivani College of Engineering, Kopargaon, Maharashtra, India

***Abstract*: Emotion recognition is a rapidly growing research domain in recent years. Unlike humans, machines lack the abilities to perceive and show emotions. But human-computer interaction can be improved by automated emotions recognition, thereby reducing the need of human intervention. In the past decade a lot of research has gone into Automatic Speech Emotion Recognition (SER). The primary objective of SER is to improve man-machine interface. It can also be used to monitor the psycho physiological state of a person in lie detectors. In recent time, speech emotion recognition also find its applications in medicine and forensics. Speech emotion recognition is a system that measures the user's emotion through speech. It uses Machine Learning and Deep Learning algorithms and techniques to extract and capture user's emotion through speech. In the field of speech emotion recognition there are many algorithms to identify user's emotion. Data used in this system is user's speech. Emotion recognition system from speech is one of the advance topics in the electronics media. The insights gained may be helpful in range of applications.**

Introduction

Speech emotion recognition is a system that measures the user's emotion through speech. It uses Machine Learning and Deep Learning algorithms and techniques to extract and capture user's emotion through speech. In the field of speech emotion recognition there are many algorithms to identify user's emotion. Data used in this system is user's speech. Emotion recognition system from speech is one of the advance topics in the electronics media. This project is based mainly on Machine Learning. Machine learning is an algorithm category that allows software applications to be more accurate than estimating results without being explicitly programmed.

Literature Review

The challenging module in CAS (computer-aided services) has recognized the emotion from the signals of speech. In SER (speech emotion recognition), several schemes have used for extracting emotions from the signals, comprising various classification speech analysis methods. This manuscript represents an outline of methods explores some contemporary literature where the existing models have used for emotion recognition based on speech. This literature review presents contributions that made towards emotion recognition of speech and extracted the features for determining emotions [1].

Modulation spectral features (MSFs) are proposed for the automatic recognition of human affective information from speech. The features are extracted from an auditory-inspired longterm spectro-temporal representation. Obtained using an auditory filter-bank and a modulation filterbank for speech analysis, the representation captures both acoustic frequency and temporal modulation frequency components, thereby conveying information that is important for human speech perception but missing from conventional short-term spectral features. On an experiment assessing classification of discrete emotion categories, the MSFs show promising performance in comparison with features that are based on mel-frequency cepstral coefficients and perceptual linear prediction coefficients, two commonly used short-term spectral representations. The MSFs further render a substantial improvement in recognition performance when used to augment prosodic features, which have been extensively used for emotion recognition [2].

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 speech. A speech 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 proposed system 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 [3].

Speech is a complex signal consisting of various information, such as information about the message to be communicated, speaker, language, region, emotions etc. Speech Processing is one of the important branches of digital signal processing and finds applications in Human computer interfaces, Telecommunication, Assistive technologies, Audio mining, Security and so on. Speech emotion recognition is important to have a natural interaction between human being and machine. In speech emotion recognition, emotional state of a speaker is extracted from his or her speech. The acoustic characteristic of the speech signal is Feature. [4]

Emotion recognition from speech signals is an important but challenging component of Human-Computer Interaction (HCI). In the literature of speech emotion recognition (SER), many techniques have been utilized to extract emotions from signals, including

many well-established speech analysis and classification techniques. Deep Learning techniques have been recently proposed as an alternative to traditional techniques in SER. This proposed system presents an overview of Deep Learning techniques and discusses some recent literature where these methods are utilized for speech-based emotion recognition. The review covers databases used, emotions extracted, contributions made toward speech emotion recognition and limitations related to it [5].

A combination of three sources of information - acoustic, lexical, and discourse - is used for emotion recognition. To capture emotion information at the language level, an informationtheoretic notion of emotional salience is introduced. Optimization of the acoustic correlates of emotion with respect to classification error was accomplished by investigating different feature sets obtained from feature selection, followed by principal component analysis. Experimental results on our call center data show that the best results are obtained when acoustic and language information are combined [6].

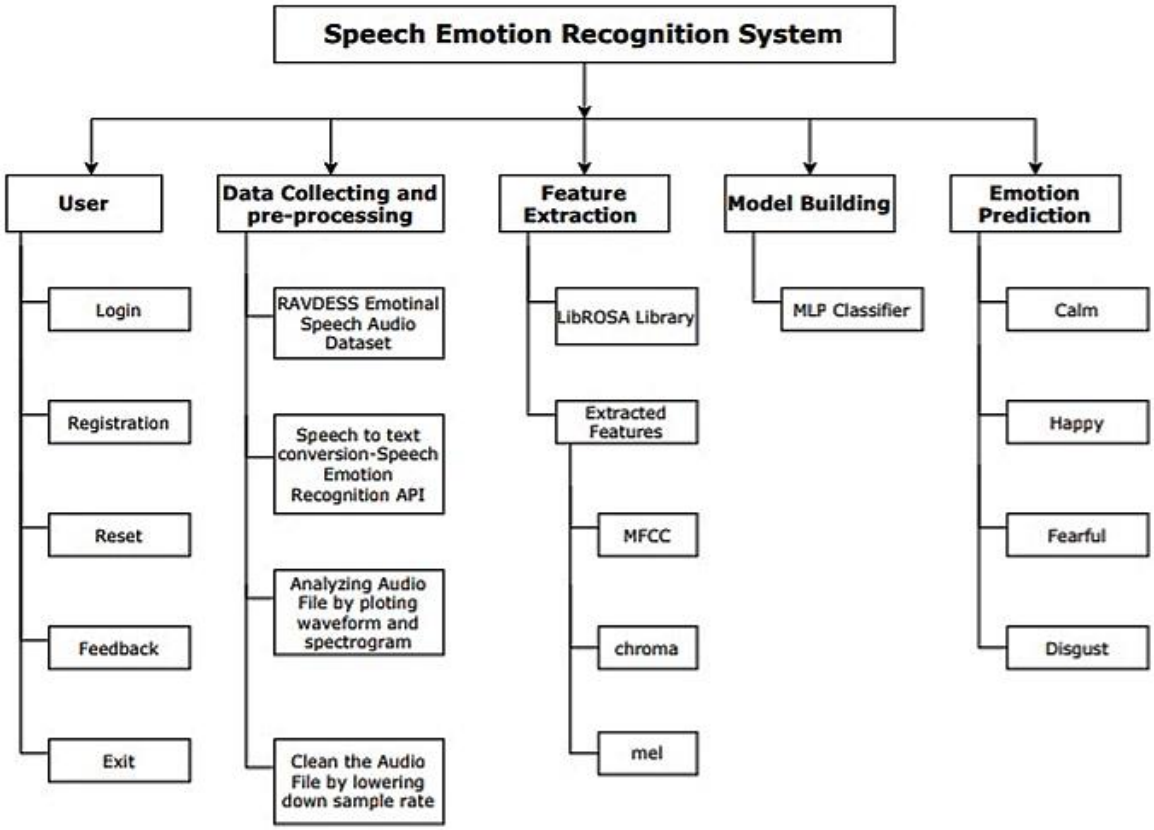
The recognition of emotional states is a relatively new technique in the field of Machine Learning. The proposed system presents the study and the performance results of a system for emotion classification using the architecture of a Distributed Speech Recognition System (DSR). The features used were extracted by the front-end ETSI Aurora eXtended of a mobile terminal in compliance with the ETSI ES 202 211 V1.1.1 standard [7].

The importance of automatically recognizing emotions in human speech has grown with increasing role of spoken language interfaces in the field of human machine interaction to make the human machine interface more efficient. It can also be used for in-car board system where information of the mental state of the driver maybe provided to initiate his/her safety. In automatic remote call center, it is used to timely detect customers dissatisfaction. In e-learning field, identifying students emotion timely and making appropriate treatment can enhance the quality of teaching. Both spectral and prosodic features can be used for speech emotion recognition because both of these features contain the emotional information. Linear Predictive Cepstrum Coefficients (LPCC) and Mel-Frequency Cepstrum Coefficients (MFCC) are some of the spectral features [8].

We propose to utilize deep neural networks (DNNs) to extract high level features from raw data and show that they are effective for speech emotion recognition. We first produce an emotion state probability distribution for each speech segment using DNNs. We then construct utterance-level features from segment-level probability distributions. These utterance-level features are then fed into an extreme learning machine (ELM), a special simple and efficient singlehidden- layer neural network, to identify utterance-level emotions [9].

The concept of speech recognition with deep learning methods. Introduction of speech recognition, deep learning and deep learning methods is discussed in this review paper. Models of deep learning that are used in speech recognition is also described. This proposed system defines the related work on speech recognition using deep learning methods and about the sphinx, software allow the implementation of speech recognition in java language. The main motive of this review is to define the use of sphinx and eclipse to recognize speech [10].

System Architecture



User

User interacts with the system through web application.

1. Registration
   The user will register to our system with his name, mobile no, email, age, username and select the preferred password.
2. Login
   The registered user can login with username and password .After login the page will be directed to the UI page of system.
3. Reset
   The user can reset the system by using this button.
4. Feedback
   The feedback button can collect the feedback from user.
5. Exit
   The exit button can close the system.

Data Collection and Pre-processing

1. Data Collection

Data collection is one of the most important parts of building machine learning models. Because no matter how well designed our model is, it won't learn anything useful if the training data is invalid.

2. RAVDESS Emotional Speech Audio Dataset

The dataset includes around 150 audio file input from 24 actors. 12 men and 12 females where these actors record short audios in 4 different emotions. Each audio file is named in such a way that the 7 th character is consistent with the different emotions that they represent.

3. Data Pre-processing

Data pre-processing in machine learning is crucial step that helps enhance the quality of data to promote the extraction of meaningful insights from the data. Data pre-processing refers to the technique of cleaning and organizing the raw data to make it suitable for a building and training machine learning model.

4. Speech Emotion Recognition API

Initially we tested the audio by translating it back into the text mode using Speech Emotion API to know what the audio is all about. Applying Machine Learning techniques for speech emotion recognition.

5. Analysing Audio File
Audio signals can be analysed in several different way, depending on the kind of information desired from the signal. So that second our step is to test out the audio files by plotting out the waveform and spectrogram to see the sample audio files.

6. Clean the Audio File
Next step is to clean the audio files by lowering down the sample rate and removing the unwanted noise around the raw audio via masking.

Feature Extraction
The next step involves extracting the features from the audio files which will help our model learn between these audio files. For feature extraction we make use of librosa library of python which is one of the libraries used for audio analysis. Also there are labels of emotions defines, when the clean dataset is loaded with the calling of feature extraction process, every audio is classified into the labels defined. Voice frequently reflects hidden feeling through tone and pitch. The objective of feature extraction is to reveal applicable feature from discourse signals given as information.

1. librosa Library
It is a python package for music and audio analysis. It provides the building blocks necessary to create audio information retrieval system. The librosa package is structured as collection of submodules like beat, core, decompose, display, feature, sequence, etc.

2. Extracted Features
The first step in any automatic speech recognition system is to extract feature i.e. identify the components of the audio signal that are good for identifying the linguistic content and discarding all the other stuff which carries information like background noise, emotion, etc. So the features which are extracted can be:

1. MFCC: Mel Frequency Cepstral Coefficients (MFCC) are a feature widely used in automatic speech and speaker recognition. They were introduce by Davis and Mermelstein in the 1980 and have been state-of-the-art ever since.

2. Chroma: The chroma feature is a descriptor, which represents the tonal content of a musical audio signal in a condensed form. Therefore chroma feature can be considered as important prerequisite for high-level semantic analysis like chord recognition or harmonic similarity estimation. A better quality of the extracted chroma feature enables much better results in these high-level tasks.

3. Mel: The mel scale relates perceived frequency, or pitch, of a pure tone to its actual measured frequency. Humans are much better at discerning small changes in pitch at low frequencies than they are at high frequencies. Incorporating this scale make our features match more closely what humans hear.

Model Building
Since the project is a classification problem, MultiLayer Perceptron seems the obvious choice. We chose this model to predict the right emotions. Theclassifier connects to a Neural Network. Unlike Applying Machine Learning techniques for speech emotion recognition others classification algorithms such as Support Vectors or naïve Bayes Classifier, MLP Classifier relies on an underlying Neural Network to perform the task of classification.

MLP Classifier
An MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. In MLP classifier, an input layer is pass to some hidden layer which are used for the implementation of abstraction and then result is process and you can see the predicted emotion. MLP classifier relies on an underlying Neural Network to perform classification. It can implement a MLP algorithm and trains the neural network using Back propogation. Building the MLP Classifier involves the following steps:

1. Initialize the MLP classifier by defining and initiating the required parameters.
2. Data is given to the Neural Network to train it.
3. The trained network is used to predict the output.
4. Calculate the accuracy of the prediction.

Emotion Prediction
Emotion recognition is the process of identifying the human emotion. This system aims at detecting emotion states from continuous and spontaneous speech. After tuning the model, tested it out by predicting the emotions for the test data. Following the splitting of training and testing data to saving the model. Model is loaded again to predict the test data store its result in .csv file along with its labels for mapping individual result to its wav file name.

Experimental Result

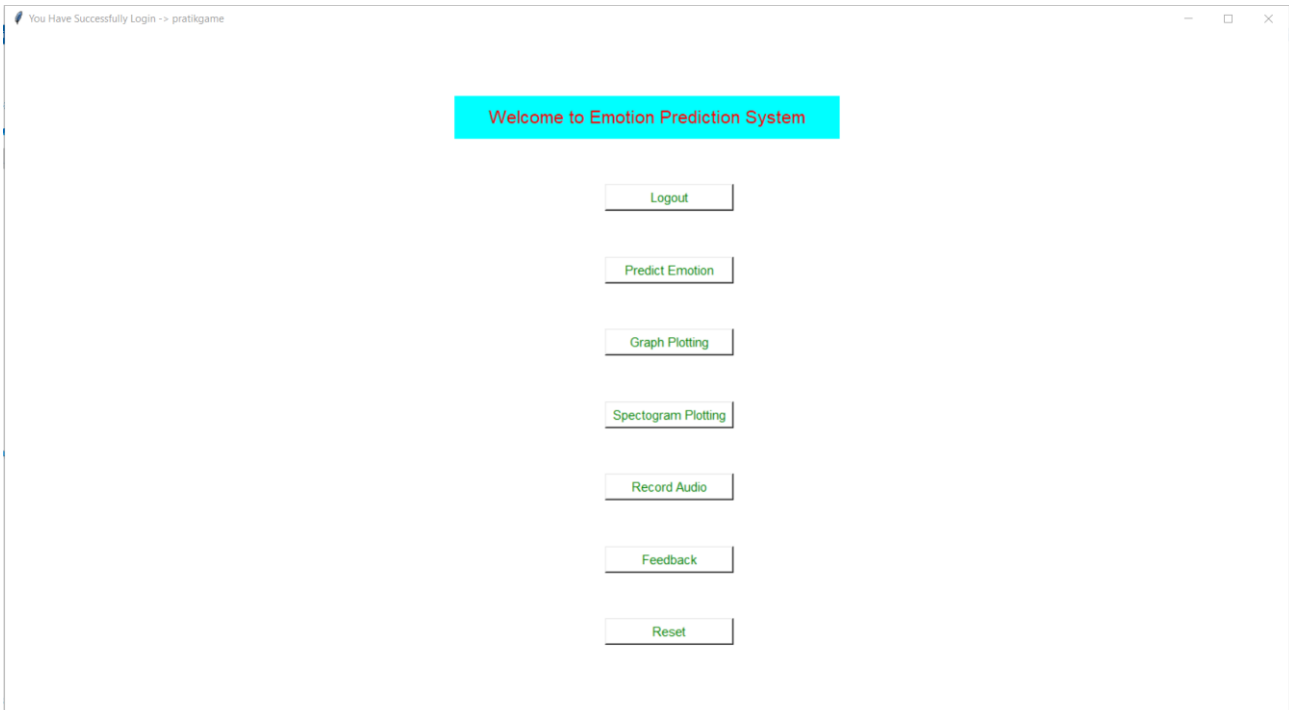


Fig. 1: First Screen

Fig. 2: Welcome Screen

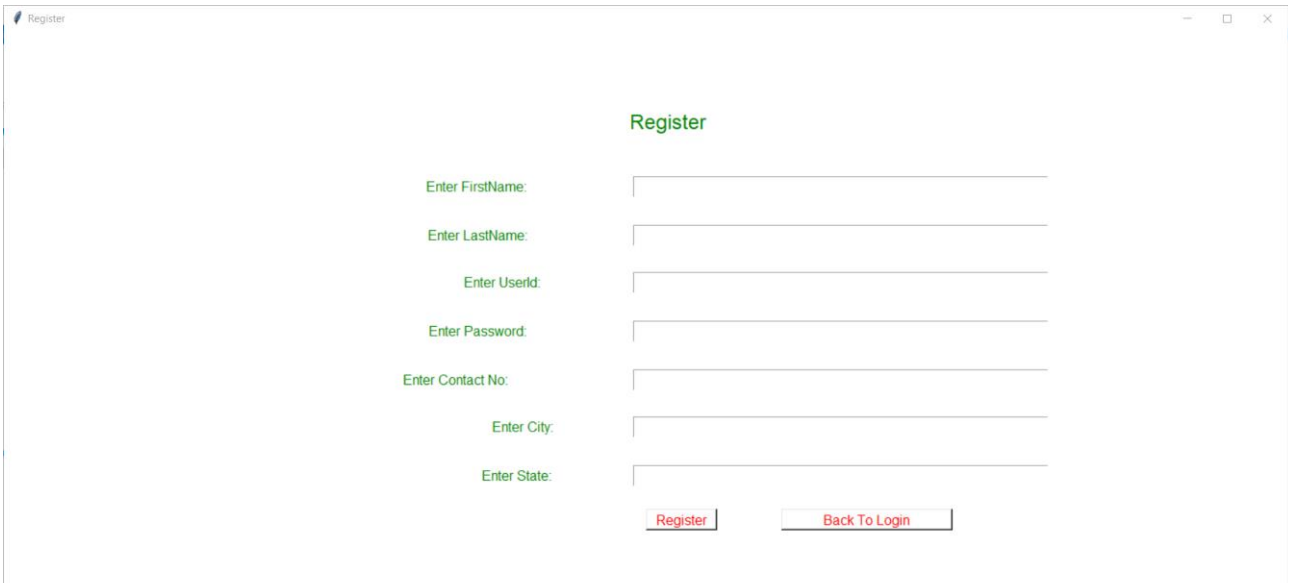


Fig. 3: Registration Screen

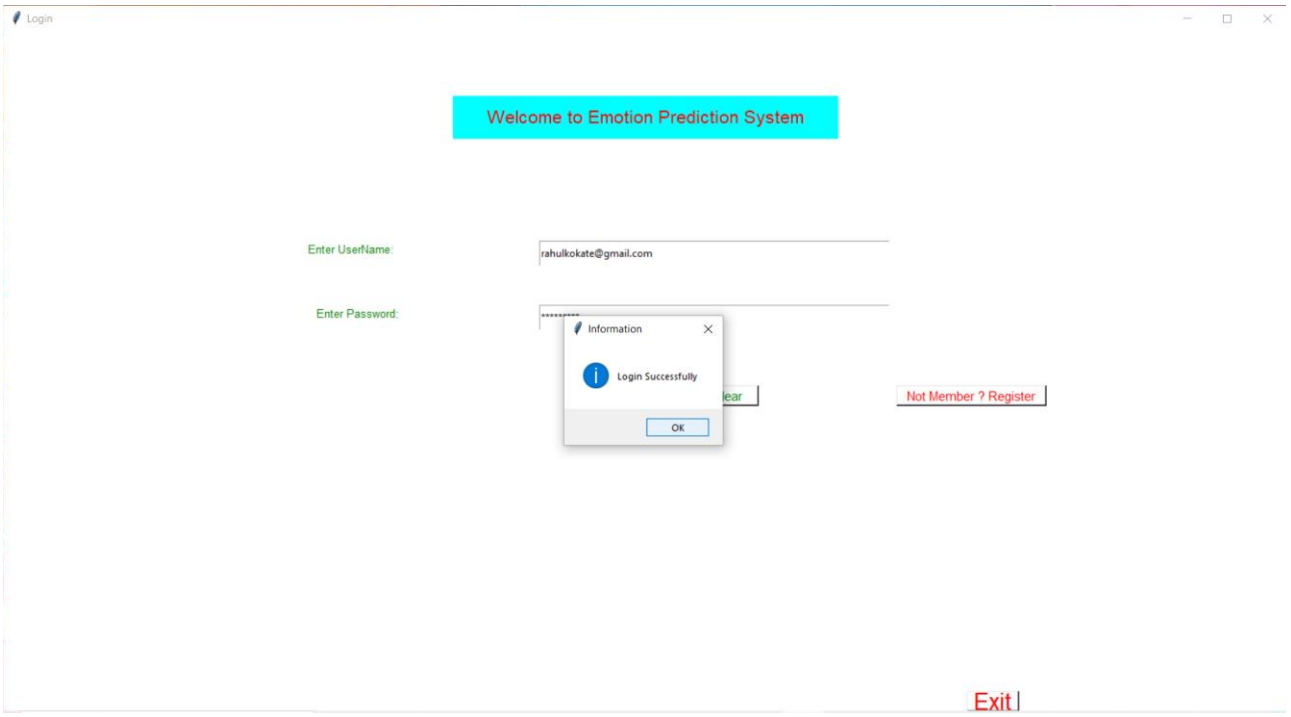


Fig. 4: Login Screeen

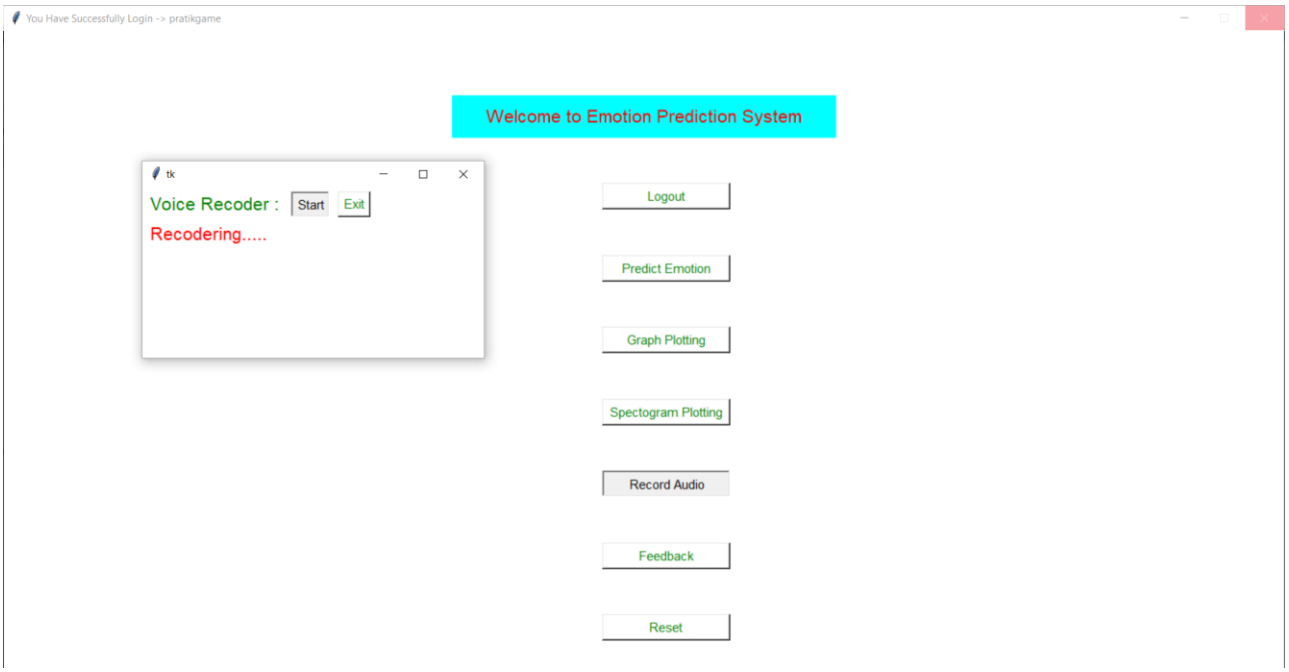


Fig. 5: Audio Recording

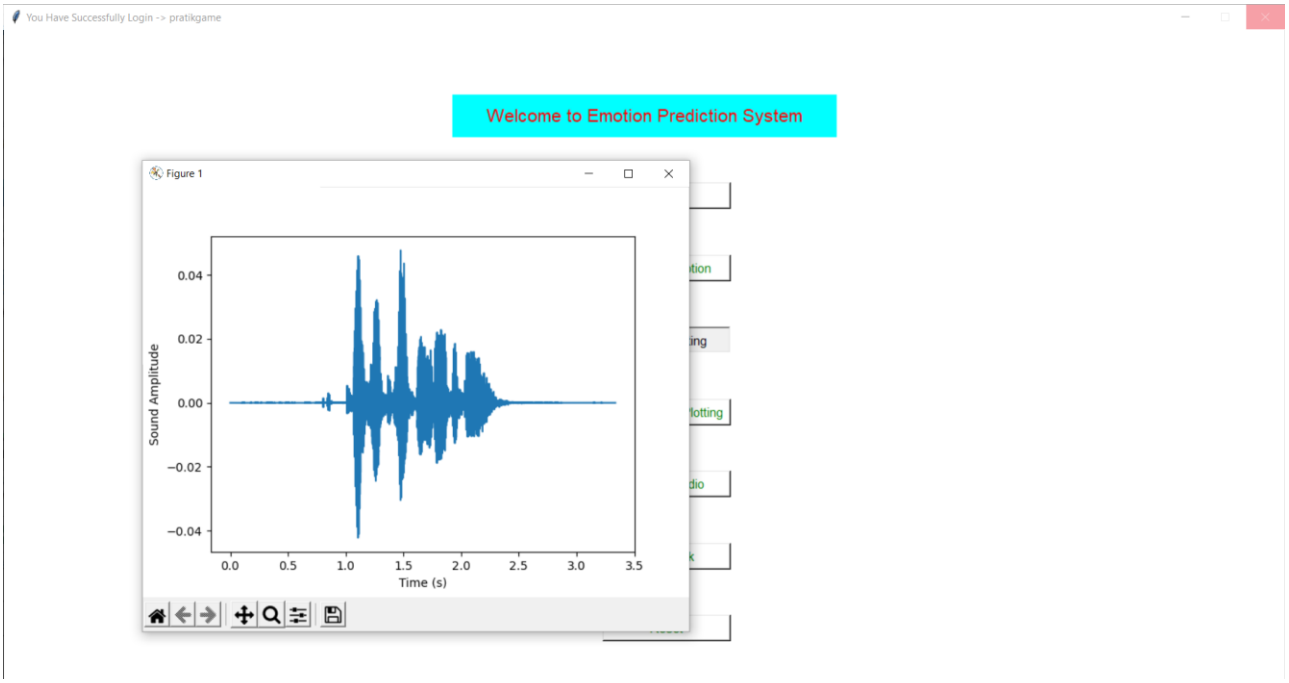


Fig. 6: Graph Plotting

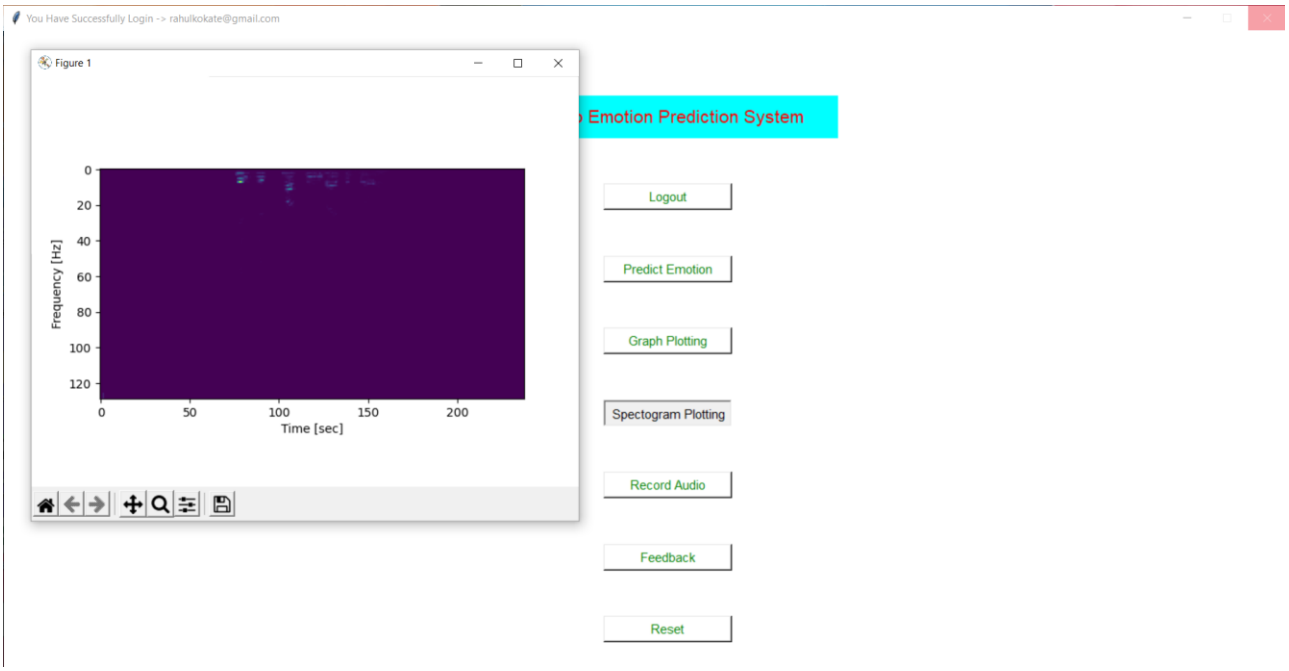


Fig. 7: Spectogram

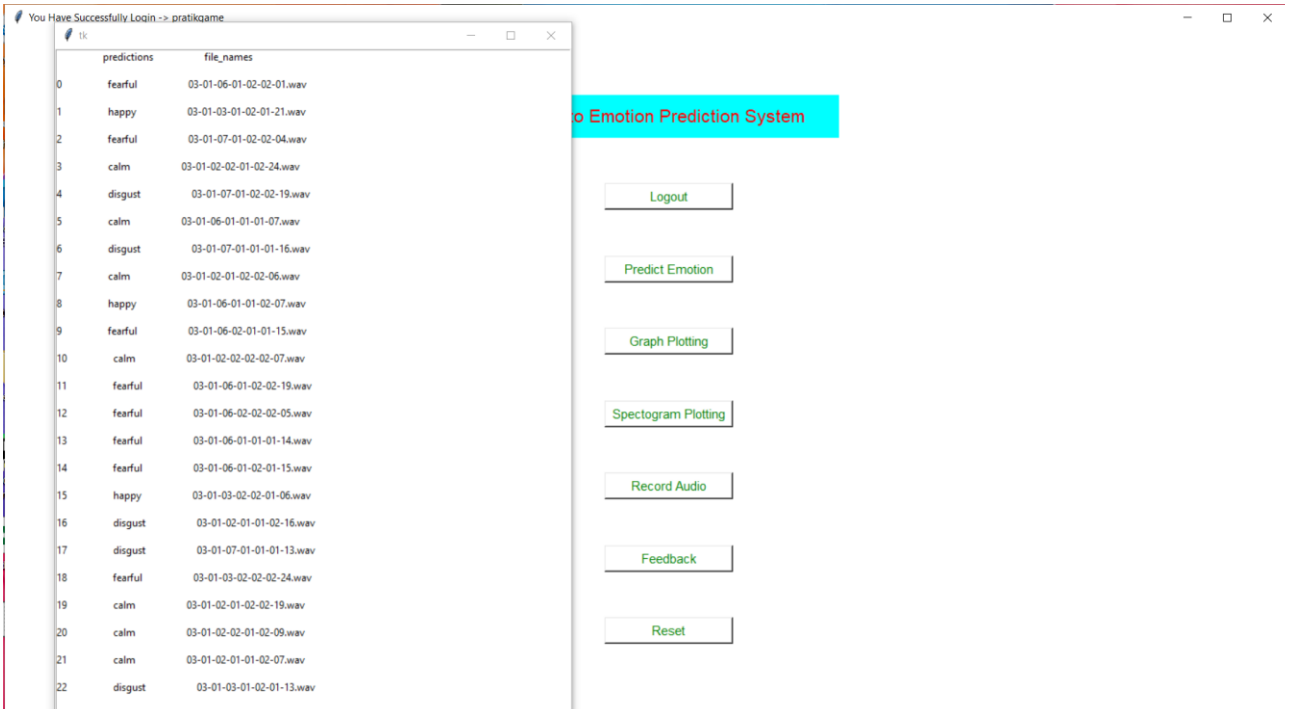


Fig. 8: Emotion Prediction

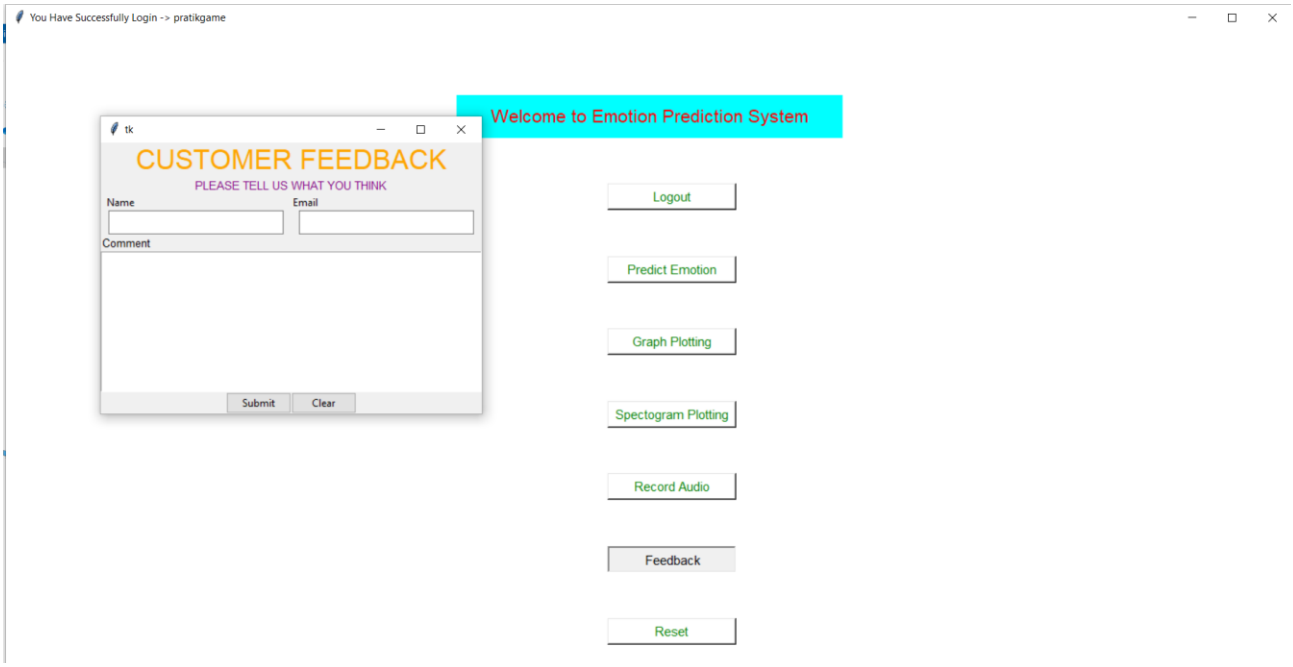


Fig. 9: Feedback Screen

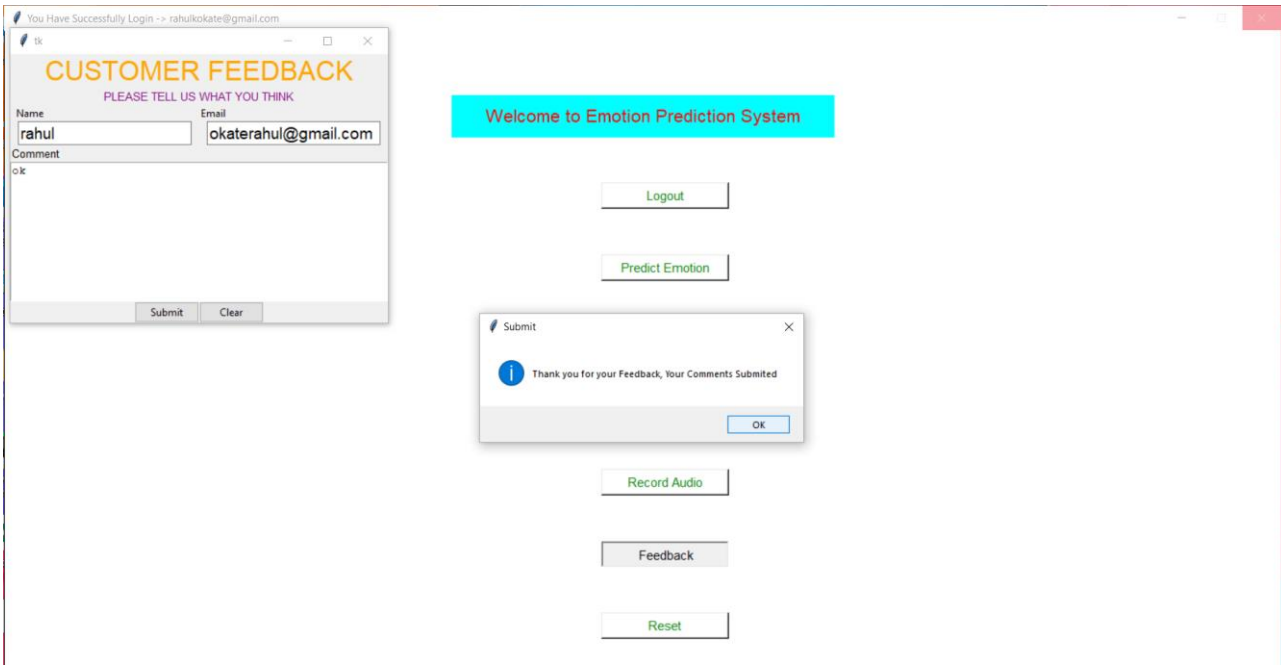


Fig. 10: Feedback Submitted

Testing

Introduction

In general, testing is finding out how well something works. In terms of human beings, testing tells what level of knowledge or skill has been acquired. In computer hardware and software development, testing is used at key checkpoints in the overall process to determine whether objectives are being met. Software testing, depending on the testing method employed, can be implemented at any time in the development process. Software testing can be stated as the process of validating and verifying that a software program/application/product.

Unit Testing

Unit testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are proper. Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. Unit testing is often automated but it can also be done manually. Unit testing involves only those characteristics that are vital to the performance of the unit under test. This encourages developers to modify the source code without immediate concerns about how such changes might affect the functioning of other units or the program as a whole. Once all of the

units in a program have been found to be working in the most efficient and error-free manner possible, larger components of the program can be evaluated by means of integration testing. Unit testing can be time consuming and tedious.

Integration Testing
Integration Testing is a level of software testing where individual modules are combined and tested as a group. Testing of integrated modules to verify combined functionality after integration. Modules are typically code modules, individual applications, client and server applications on a network etc. This type of testing is especially relevant to clientserver and distributed system. Integration testing occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing. The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items. There are two major ways of carrying out an integration test, called the bottom-up method and the top down method. Bottom-up integration testing begins with unit testing, followed by tests of of progressively higher-level combinations of units called modules or builds. In top-down integration testing, the highestlevel modules are tested first and progressively lower-level modules are tested after that.

Manual Testing
Manual testing is a software testing process in which test cases are executed manually without using any automated tool. All test cases executed by the tester manually according to the end user's perspective. It ensures whether the application is working, as mentioned in the requirement document or not. Test cases are planned and implemented to complete almost 100 percent of the software application. Test case reports are also generated manually. Manual Testing is one of the most fundamental testing processes as it can find both visible and hidden defects of the software. The difference between expected output and output, given by the software, is defined as a defect. The developer fixed the defects and handed it to the tester for retesting. Manual testing is mandatory for every newly developed software before automated testing. This testing requires great efforts and time, but it gives the surety of bug-free software. Manual Testing requires knowledge of manual testing techniques but not of any automated testing tool. Manual testing is essential because one of the software testing fundamentals is "100% automation is not possible". There are various methods used for manual testing. Each technique is used according to its testing criteria. Types of manual testing are given below:

- **White Box Testing**

The white box testing is done by Developer, where they check every line of a code before giving it to the Test Engineer. Since the code is visible for the Developer during the testing, that's why it is also known as White box testing.

- **Black Box Testing**

The black box testing is done by the Test Engineer, where they can check the functionality of an application or the software according to the customer / client's needs. In this, the code is not visible while performing the testing; that's why it is known as black-box testing.

- **Gray Box Testing**

Gray box testing is a combination of white box and Black box testing. It can be performed by a person who knew both coding and testing. And if the single person performs white box, as well as black-box testing for the application, is known as Gray box testing.

Automation Testing
Automation Testing means using an automation tool to execute your test case suite. On the contrary, Manual Testing is performed by a human sitting in front of a computer carefully executing the test steps. The automation software can also enter test data into the System Under Test, compare expected and actual results and generate detailed test reports. Test Automation demands considerable investments of money and resources. Software Test automation makes use of specialized tools to control the execution of tests and compares the actual results against the expected result. Usually, regression tests, which are repetitive actions, are automated. Successive development cycles will require execution of same test suite repeatedly. Using a test automation tool, it's possible to record this test suite and re-play it as required. Once the test suite is automated, no human intervention is required. This improved ROI of Test Automation. The goal of Automation is to reduce the number of test cases to be run manually and not to eliminate Manual Testing altogether. Testing Tools not only helps us to perform regression tests but also helps us to automate data set up generation, product installation, GUI interaction, defect logging, etc. Automation tools are used for both Functional and Non-Functional testing. There are many approaches to test automation, however below are the general approaches used widely:

- **GUI Testing**

A testing framework that generates user interface events such as keystrokes and mouse clicks, and observes the changes that result in the user interface, to validate that the observable behavior of the program is correct.

- **API driven Testing**

A testing framework that uses a programming interface to the application to validate the behaviour under test. Typically API driven testing bypasses application user interface altogether. It can also be testing public (usually) interfaces to classes, modules or libraries are tested with a variety of input arguments to validate that the results that are returned are correct.

- **Reliability Testing**

Realiability Testing is a software testing type that checks whether the software can perform a failure-free operation for a specified period of time in a particular environment. Reliability means ”yielding the same,” in other terms, the word ”reliable” means something is dependable and that it will give the same outcome every time. The same is true for Reliability testing. Reliability testing in software assures that the product is fault free and is reliable for its intended purpose. Reliability Testing can be categorized into three segments:

- Modeling
- Measurement
- Improvement

The following formula is for calculating the probability of failure:
Probability = Number of Failing Cases / Total Number of Cases under Consideration

Test Tool Selection

Pytest

The pytest framework makes it easy to write small tests, yet scales to support complex functional testing for applications and libraries. Pytest is a testing framework which allows us to write test codes using python. You can write code to test anything like database , API, even UI if you want. But pytest is mainly being used in industry to write tests for APIs. Various features of pytest are given below:

- Detailed info on failing assert statements (no need to remember self.assert* names)
- Auto-discovery of test modules and functions
- Modular fixtures for managing small or parametrized long-lived test resources
- Can run unittest (including trial) and nose test suites out of the box
- Python 3.5+ and PyPy 3
- Rich plugin architecture, with over 315+ external plugins and thriving community

Conclusion

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 will comprehend the emotion of the speaker (user). Some applications of a speech emotion detection system are computer-based tutorial applications, automatic call center conversations, a diagnostic tool used for therapy and automatic translation system.

References

[1] S. Casale, A. Russo, G. Scebba, “Speech Emotion Classification using Machine Learning Algorithms”, 2008, IEEE International Conference on Semantic Computing
[2] Aastha Joshi, Rajneet Kaur, “A Study of Speech Emotion Recognition Methods”, Vol. 2, Issue. 4, April 2013, pg. 28-31
[3] Vladimir Chernkh, Grigoriy Sreling, Pavel Prihodko, “Emotion Recognition From Speech With Recurrent Neural Networks”, 2017
[4] Rahul B. Lanjewar, Swarup Mathurkar, Nileah Patel, “Implementation and Comparison of Speech Emotion Recognition System Using Gaussian Mixture Model (GMM) and K-Nearest Neighbor (KNN) Techniques”. Procedia Computer Science, Vol. 49, 2015
[5] Chul Min Lee, Shrikanth S. Narayanan, “Toward detecting emotions in spoken dialogs”, IEEE Transaction on Speech and Audio Processing, Vol. 13, No. 2, pp. 293-303, Mar. 2005
[6] Rubi, Chhavi Rana, “A Review: Speech Recognition with Deep Learning Methods”, International Journal of Computer Science and Mobile Computing, Vol. 4, Issue. 5, May 2015, pg. 1017-1024
[7] K. V. Krishna Kishore, P. Krishna Satish, “Emotion Recognition in speech Using MFCC and Wavelet Features”, 3rd IEEE International Advance Computing Conference (IACC), 2013
[8] Jashmin K Shan, Brett Smolenki , Robert E Yantorno , Ananth N Iyer, “Sequential k-nearest neighbor pattern recognition for usable speech classification”, 2004, 12th European Signal Processing Conference, IEEE Proceedings
[9] R. B. Pradeeba, K. Tarunika, Dr. P. Aruna, “Accuracy of speech emotion recognition through deep neural network and k-nearest”, International Journal of Engineering Research in Computer Science and Engineering, Vol 5, Issue 2, February 2018
[10] Kun Han, Dong Yu, Ivan Tashev, “Speech Emotion Recognition Using Deep Neural Network and Extreme Learning Machine”, INTERSPEECH 2014