DAYANANDA SAGAR UNIVERSITY

KUDLU GATE, BANGALORE – 560068

Bachelor of Technology in
COMPUTER SCIENCE AND ENGINEERING

Major Project Phase-II Report

Emotion Analysis Using Speech

By

M M Krupashree -ENG19CS0158
Naseeba Begum-ENG19CS0199
Nayana Priya A P-ENG19CS0202
Nithya S-ENG19CS0210

Batch no - **6**

Under the supervision of

Prof.Rashmi Mothkur
Assistant professor
Department of Computer Science

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING,
SCHOOL OF ENGINEERING
DAYANANDA SAGAR UNIVERSITY,
BANGALORE

(2022-2023)

DAYANANDA SAGAR UNIVERSITY

School of Engineering
Department of Computer Science & Engineering
Kudlu Gate, Bangalore – 560068
Karnataka, India

CERTIFICATE

This is to certify that the Phase-II project work titled **" EMOTION ANALYSIS USING SPEECH "** is carried out by **MM Krupashree (ENG19CS0158), Naseeba Begum (ENG19CS0199), Nayana Priya A P (ENG19CS0202), Nithya S(ENG19CS0210),** bonafide students of Bachelor of Technology in Computer Science and Engineering at the School of Engineering, Dayananda Sagar University, Bangalore in partial fulfillment for the award of degree in Bachelor of Technology in Computer Science and Engineering, during the year **2022-2023** .

Prof Rashmi Motkur	**Dr. Girisha G S**	**Dr. Udaya Kumar Reddy K R**
Assistant Professor Dept. of CS&E, School of Engineering Dayananda Sagar University	Chairman CSE School of Engineering Dayananda Sagar University	Dean School of Engineering Dayananda Sagar University
Date:	Date:	Date:

Name of the Examiner **Signature of Examiner**

1.

2.

DECLARATION

We, **MM Krupashree (ENG19CS0158), Naseeba Begum (ENG19CS0199), Nayana Priya A P (ENG19CS0202), Nithya S(ENG19CS0210),** are students of eighth semester B. Tech in **Computer Science and Engineering**, at School of Engineering, **Dayananda Sagar University**, hereby declare that the Major Project Stage-II titled **" EMOTION ANALYSIS USING SPEECH "** has been carried out by us and submitted in partial fulfilment for the award of degree in **Bachelor of Technology in Computer Science and Engineering** during the academic year **2022-2023** .

Student **Signature**

Name1:MM Krupashree

USN :ENG19CS0158

Name2:Naseeba Begum

USN :ENG19CS0199

Name3:Nayana Priya A P

USN :ENG19CS0202

Name4:Nithya S

USN :ENG19CS0210

Place : Bangalore

Date :

ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and support of many individuals who have been responsible for the successful completion of this project work.

First, we take this opportunity to express our sincere gratitude to School of Engineering & Technology, Dayananda Sagar University for providing us with a great opportunity to pursue our Bachelor's degree in this institution.

We would like to thank ***Dr. Udaya Kumar Reddy K R, Dean, School of Engineering & Technology, Dayananda Sagar University*** *for his constant encouragement and expert advice.*

It is a matter of immense pleasure to express our sincere thanks to ***Dr. Girisha G S, Department Chairman*** *,* ***Computer Science and Engineering*** *,* ***Dayananda Sagar University,*** *for providing right academic guidance that made our task possible.*

We would like to thank our guide **Prof.Rashmi Mothkur** *,* ***Assistant Professor*** *,* ***Dept. of Computer Science and Engineering*** *,* ***Dayananda Sagar University*** *, for sparing her valuable time to extend help in every step of our project work, which paved the way for smooth progress and fruitful culmination of the project.*

We would like to thank our ***Project Coordinator Dr. Meenakshi Malhotra*** *and* ***Dr. Pramod Kumar Naik*** *as well as all the staff members of Computer Science and Engineering for their support.*

We are also grateful to our family and friends who provided us with every requirement throughout the course.

We would like to thank one and all who directly or indirectly helped us in the Project work.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
DL	Deep Learning
GUI	Graphical User Interface
MySQL	My Structured Query Language
CNN	Convolution Neural Network
LSTM	Long Short Term Memory
SER	Speech Emotion Recognition
MFCC	Mel Frequency Cepstral Coefficient

LIST OF FIGURES

ABSTRACT

In human machine interface applications, emotion recognition from the speech signal has been a research topic for many years. Emotions play an extremely important role in human mental life. It is a medium of expression of one's perspective or one's mental state to others. Speech Emotion Recognition (SER) can be defined as extraction of the emotional state of the speaker from his or her speech signal. There are few universal emotions- including Neutral, Anger, Happiness, Sadness etc in which any intelligent system with finite computational resources can be trained to identify or synthesize as required. In this work, we are extracting Mel-frequency cepstral coefficients (MFCC), Chromogram, Mel scale spectrogram in conjunction with Spectral contrast and Tonal Centroid features. Deep Neural Network is used to classify the emotion in this work.

CHAPTER 1

INTRODUCTION

CHAPTER 1 INTRODUCTION

1.1. INTRODUCTION

There are many ways of communication but the speech signal is one of the fastest and most natural methods of communication between humans. Therefore, speech can be the fast and efficient method of interaction between humans and the machine as well. Through all the available senses people actually sense the emotional state of their communication partner. Emotional detection is natural for humans but it is a very difficult task for machines. Therefore the purpose of the emotion recognition system is to use emotion related knowledge in such a way that human machine communication will be improved.

Deep neural networks (DNN) has unprecedented success in the field of speech recognition and image recognition; however, so far no research on deep neural networks has been applied to speech emotion processing. We found that the DNN in speech emotion processing has a huge advantage. Therefore, this project proposed a method to realize the emotional features automatically extracted from the audio using the librosa package in python. We used CNN and LSTM network to extract speech emotion features. It incorporates the speech emotion features of more consecutive frames, to build a high latitude characteristic, and uses softmax classifier layer to classify the emotional speech. The speech emotion recognition test accuracy reached 82% which is a high value compared to the other models of this size.

Automatic emotion recognitions from human speech are increasing nowadays because it results in better interactions between human and machine.

1.2. OBJECTIVE

The main agenda of our project is to detect the emotions elicited by the speaker while talking. The main reason for choosing this project is speech emotion detection has become one of the biggest marketing strategies, in which the mood of the consumer plays an important role. We are employing deep learning models and various techniques to build a model which can detect human emotions through voice-pattern and speech pattern analysis. By the end of the project we would have a robust and proficient model with better accuracy

compared to existing ones to analyze emotions precisely.This model can be used by various apps, online shopping websites and so on to know about the user's emotions.

1.3. SCOPE

Emotion recognition has wide scope in many areas such as human computer interaction, biometric security etc.Emotion detection has become one of the biggest marketing strategies, in which the mood of the consumer plays an important role. So to detect the current emotion of the person and suggest to him the apt product or help them accordingly, will increase the demand of

the product or the company. Automatic emotion recognition using speech can help organizations to understand their customers better when in a call.Call centers can make separate strategies on dealing with people with different people. For E-Learning, schools can monitor the emotions of their students to better prepare their education system for the betterment of the students.Robotics has wide use of Emotion detection as a robot designed to interact with humans should understand the human's emotion.Emotion detection is the key to Human Computer Interaction (HCI).One among so many examples is that the AI virtual assistant product of Amazon can use this motion detection to help the customer to play music according to his mood or play what he likes based on his mood. So the vast area of practical application and importance in the real world, made us choose this topic.

CHAPTER 2
PROBLEM DEFINITION

CHAPTER 2 PROBLEM DEFINITION

Human speech is the most natural way to express ourselves. We use it everywhere from calls, emails, meetings, discussions etc. As emotions play a vital role in communication, the detection and analysis of the same is of vital importance in today's digital world of remote communication. This project can be defined as a collection of methodologies that process and classify speech signals to detect emotions in them. We will try to detect the emotions of a person or speaker. We will implement a Deep Neural Network (DNN) model to create the application.

CHAPTER 3

LITERATURE REVIEW

CHAPTER 3 LITERATURE REVIEW

Edward Jones et al [1] have presented a paper on Speech Emotion Recognition Using Deep Learning Techniques:A Review.These methods offer easy model training as well as the efficiency of shared weights. Limitations of deep learning techniques include their large layer-wise internal architecture, less efficiency for temporally-varying input data and over-learning during memorization of layer-wise information.This research work forms a base to evaluate the performance and limitations of current deep learning techniques.

Ron Hoory et al [2] have presented a paper on Speech Emotion Recognition Using Self-Supervised Features.They have clearly shown that well designed combinations of carefully fine-tuned and averaged Upstream models and averaged Downstream models can significantly improve the performance of E2E SER models.This research paper aims to introduce a modular End-to-End (E2E) SER system based on an Upstream + Downstream architecture model paradigm

Mira Kartiwi et al [3] have presented a paper on A Comprehensive Review of Speech Emotion Recognition Systems.The paper carefully identifies and synthesizes recent relevant literature related to the SER systems' varied design components/methodologies, thereby providing readers with a state-of-the-art understanding of the hot research topic.This paper points out that deep learning techniques are considered best suited for the SER system over traditional techniques because of their advantages like scalability, all-purpose parameter fitting, and infinitely flexible function.

Srinivasa Parthasarathy et al [4] have presented a paper on Semi-Supervised Speech Emotion Recognition With Ladder Networks.The results indicated significant gains when using the proposed models, underlying the generalization power of the ladder networks. The improvements were particularly high when using unlabeled data from the target domain, exploiting all the benefits of the proposed architecture.This paper pertains to one major drawback of the SER system, that is , is their lack of generalization across different conditions which can be resolved using ladder networks. . It combines the unsupervised

auxiliary task of reconstructing intermediate feature representations, with the primary task of predicting emotional attributes.

Aneesh Muppidi et al [5] have presented a paper on Speech Emotion Recognition Using Quaternion Convolutional Neural Networks.They have encoded the RGB domain of Mel Spectrogram features in a quaternion input and use custom quaternion convolutional layers to learn features in a quaternion space. These layers are implemented in a standard neural network structure to train on benchmark datasets such as RAVDESS, IEMOCAP, and EMO-DB. QCNN is reported to yield an accuracy of 77.87%, 70.46%, and 88.78% for the RAVDESS, IEMOCAP, and EMO-DB datasets, respectively.In comparison to other competitive methods, QCNN achieves state-of-the-art results on RAVDESS, and underperforms only one method in both IEMOCAP and EMO-DB. Additionally, QCNN is able to exploit its quaternion encoding for a reduced model size, confirming previous literature on the topic as well as providing an opportunity for deployment on lightweight machines such as voice-assistant devices.

Arya Aftab et al [6] have presented a paper on Light-Sernet: A Lightweight Fully Convolutional Neural Network For Speech Emotion Recognition.Deep convolutional blocks to extract high-level features, while ensuring sufficient separability. These features are finally used to classify the emotions of the speech signal segment. Comparing to the state-of-the-art models, the proposed model has smaller size to reach almost the same or higher recognition performance.Particularly, convolutional neural networks have achieved significant improvements as compared to conventional methods CNN is particularly powerful for disregarding the information conveyed by the input signal that could be irrelevant to the target task.

CHAPTER 4
PROJECT DESCRIPTION

CHAPTER 4 PROJECT DESCRIPTION

4.1. System Design:

Emotion recognition using speech data is a popular research topic in the field of natural language processing and machine learning. The goal of this task is to build a model that can accurately identify the emotional state of a speaker based on their speech. There are several approaches to building a model for emotion recognition using speech data. One common approach is to use a combination of acoustic and linguistic features. Acoustic features are extracted from the audio signal, such as pitch, intensity, and spectral features. Linguistic features, on the other hand, are extracted from the transcript of the speech, such as the use of certain words or phrases. Some popular machine learning algorithms used for emotion recognition include decision trees, CNN, KNN, and LSTM. These algorithms are trained on labeled speech data, where each recording is labeled with the corresponding emotional state of the speaker. The model can then use the learned patterns to predict the emotional state of a new recording.

4.1.1. SYSTEM ARCHITECTURE

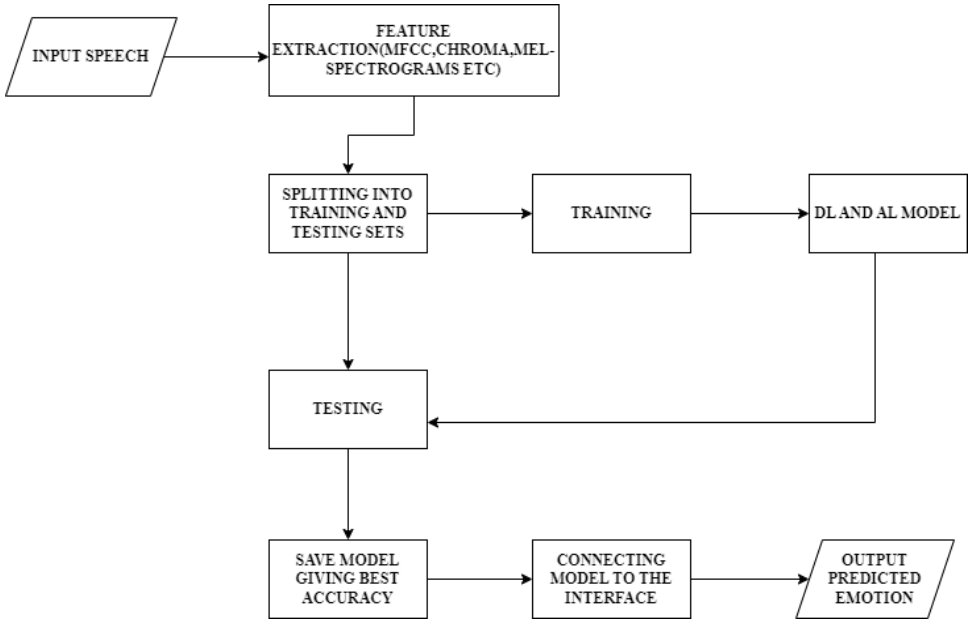


Figure 4.1.1 System Architecture Of Speech Emotion Recognition System

Figure.4.1.1. depicts the overall architecture of the project where input is taken in the form of speech or audio. Then we have to extract features from the input. Some of the main audio features extracted are Mel-frequency Cepstral Coefficient(MFCC), pitch, Mel-Spectrograms, chroma, zero-crossing rate, etc. Then the dataset is split into the training set and testing set. The very next step is training Deep-learning and Artificial Intelligence models using each feature extracted from the training dataset. The built model is tested using a testing dataset. Then we will save the model which gives the best accuracy. then we will be connecting the model to the interface and output the predicted emotion.

4.1.2. DATA FLOW DIAGRAM

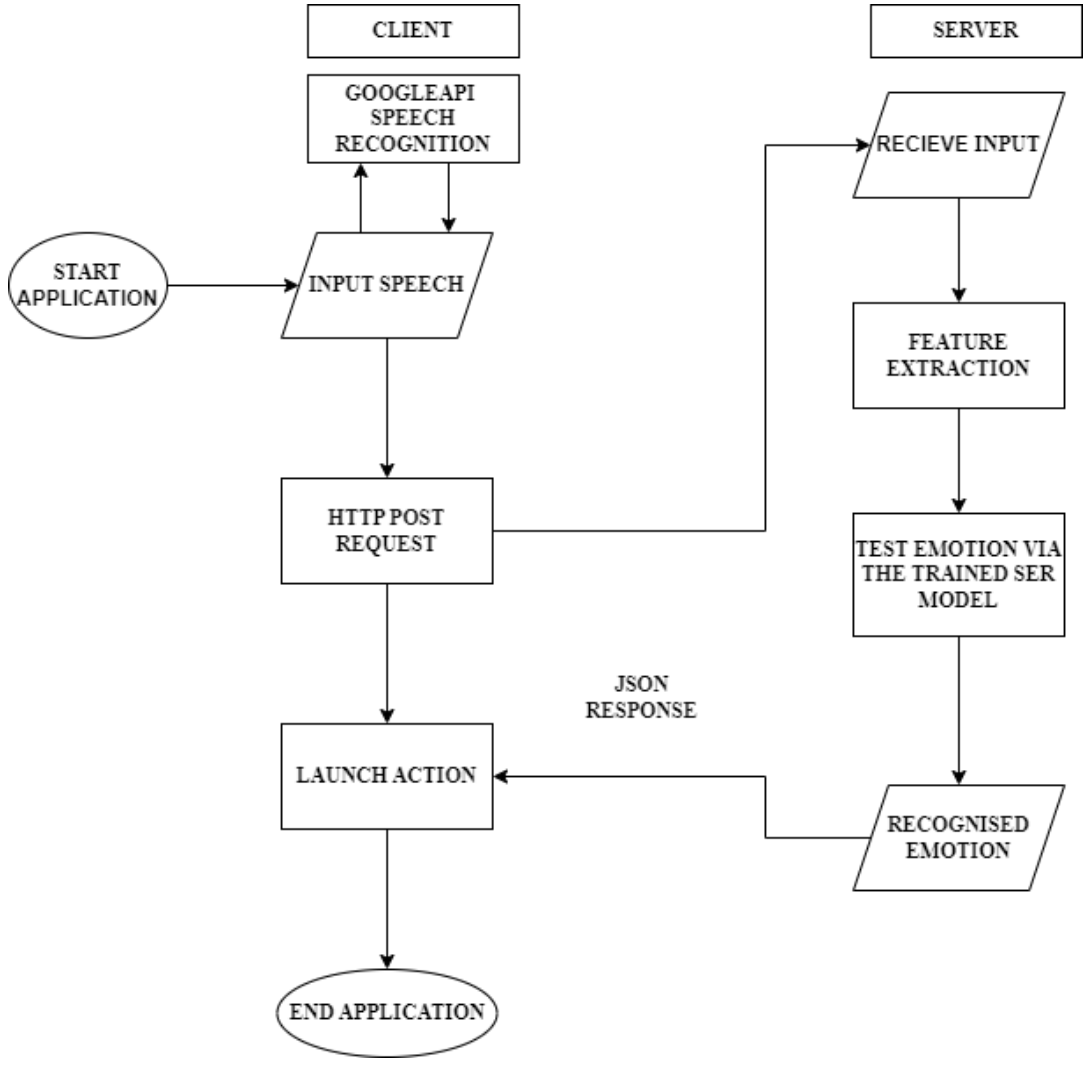


Figure 4.1.2. Data Flow Diagram Of Speech Emotion Recognition System

Figure 4.12. illustrates the data flow diagram of the project. Once the SER model is trained and tested, it is exported or embedded into an app. When you start the application it prompts the user to give input using Google Speech API. Input data is sent to the server via HTTP post request where it receives input and does feature extraction and tests extracted features using an already trained SER model/Then it predicts emotion and returns a JSON response.

4.1.3. BACKEND ARCHITECTURE

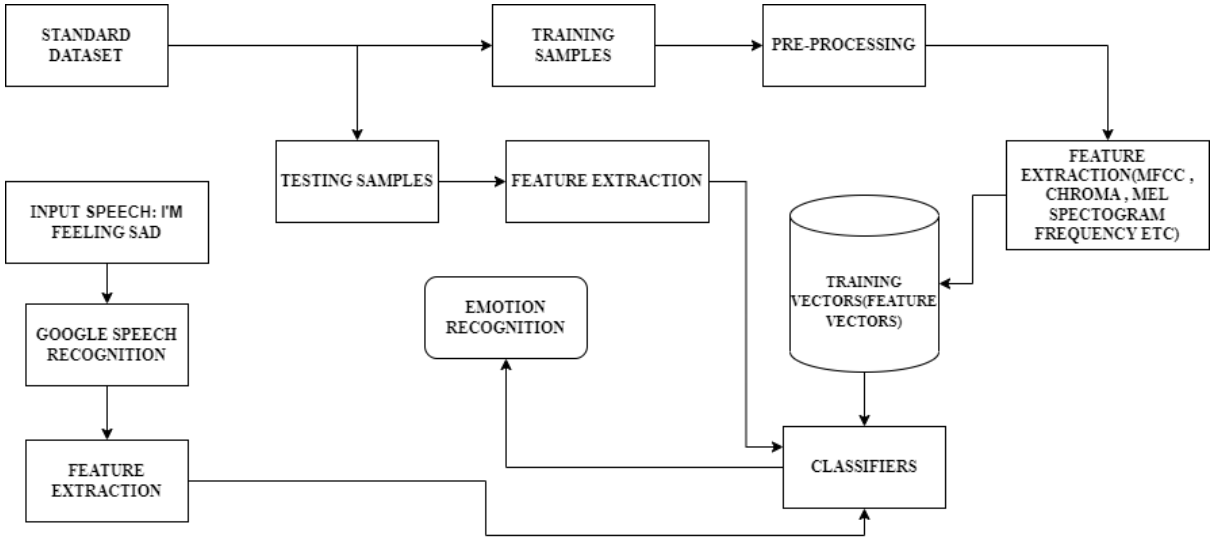


Figure 4.1.3. Backend Architecture Of Speech Emotion Recognition System

Figure 4.1.3. depicts the backend architecture of our project where standard datasets are taken as input and divided into training samples and testing samples. Training samples undergo pre-processing such as converting audio waves to mel spectrograms and data augmentation which increases the diversity of the dataset by using standard augmentation techniques such as changing pitch, injecting noise, etc. The next step is feature extraction which extracts features such as MFCC, chroma, and Mel-frequency spectrograms. These extracted features are then sent to classifiers for predicting emotion. To evaluate the model, we will be using testing samples that undergo feature extraction and are sent to classifiers for predicting emotion. We can also test by

providing live input via google speech recognition API which undergoes feature extraction and is sent to the model for predicting emotion.

4.1.4. USE CASE DIAGRAM

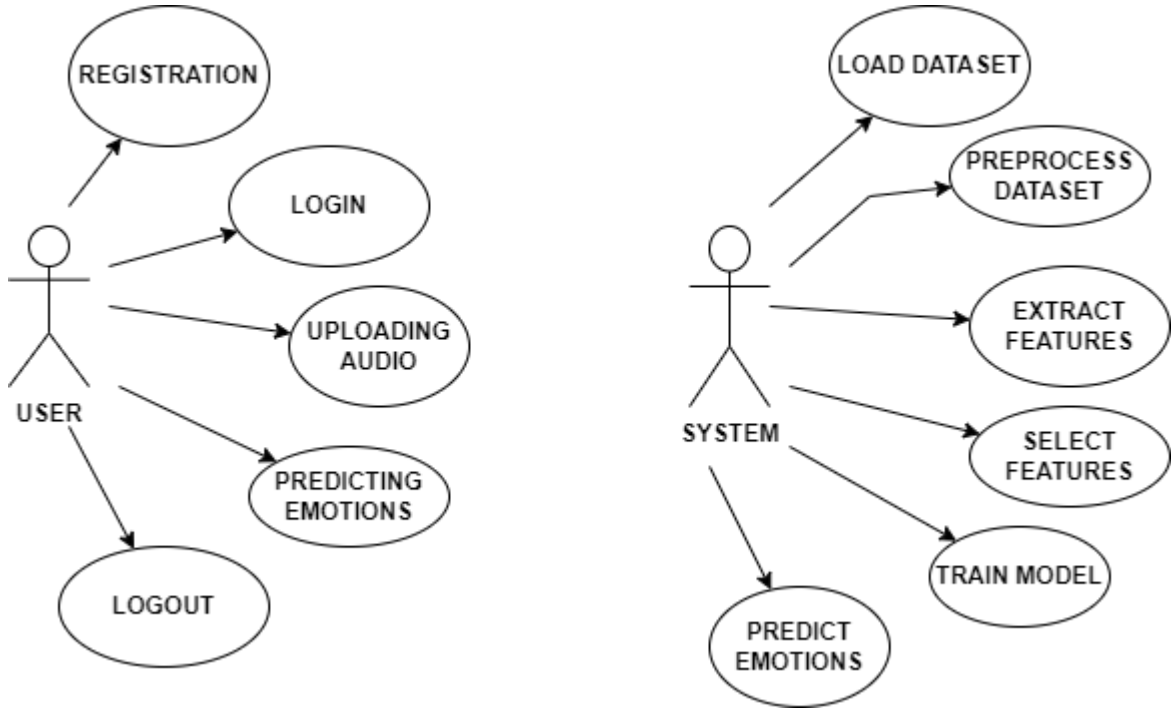


Figure 4.1.4. Use Case Diagram Of Speech Emotion Recognition System

Figure 4.1.4 depicts Use Case Diagram of Speech Emotion Recognition System. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. Here there are two actors mainly , User and System . Users perform certain functions such as registration , login , uploading audio , predicting emotions , and logout . Whereas Systems performs functions such as load dataset , preprocess dataset , extract features , select features , train model and predict emotions.

CHAPTER 5
REQUIREMENTS

CHAPTER 5 REQUIREMENTS

5.1 FUNCTIONAL REQUIREMENTS

- The main function of our speech emotion detector is to detect and classify the type of emotion from speech such as angry , calm , neutral , happy , fearful and sad .
- The speech emotion detector developed can analyze emotion in speech directly and it can also change speech to text and then analyze emotion.
- We are creating different deep learning models like LSTM. The main challenge was identification of different features in speech. For this we have used the Librosa library in Python.
- For changing Speech to text function , we will be using GoogleAPI for speech recognition and linear SVC for analyzing emotion via text.
- Speech Emotion detector model can be embedded in a software or an app so that it can work in real time.

5.2 NON-FUNCTIONAL REQUIREMENTS

- The application should be portable. So,moving from one OS to another OS does not create any problem.
- To access this application , one should have an internet connection.
- The application should be available and functional 24/7.

5.3 HARDWARE AND SOFTWARE REQUIREMENTS

5.3.1 Software Requirements:

- Operating System : Windows 7+
- Server side Script : Python 3.6+
- IDE : PyCharm
- Libraries Used : Pandas, Numpy, Keras, Tensorflow, Librosa,OpenCV,

- Dataset : RAVDESS speech dataset.

5.3.2 Hardware Requirements:

- Processors - Intel i3, i5, i7.
- Any OS - mac,windows
- RAM - 4GB(min)
- Hard disk - 128GB

CHAPTER 6
METHODOLOGY

CHAPTER 6 METHODOLOGY

Upload:

Upload the dataset of audio (.wav files) to be read using librosa library.

View:

Uploaded dataset can be viewed.

Preprocessing:

Data Preprocessing is a technique that is used to convert the raw data into a clean data set. Cleaning the data refers to removing the null values, filling the null values with meaningful values, removing duplicate values, removing outliers, removing unwanted attributes. If a dataset contains any categorical records, it means convert those categorical variables to numerical values.

Identifying Features:

The extracted features are Mel-frequency cepstral coefficients (MFCC), Chromogram, Mel scale spectrogram in conjunction with Spectral contrast and Tonal Centroid features.

Train and Test Split:

We split our dataset of 1440 audio files in 2 parts, training data with 1008 audio files and testing data with 432 audio files. Here 70% of the data is taken for the training dataset.

Building the model:

1. To understand the audio and predict emotions, we are proposing a Deep learning based method.
2. Deep learning can provide increased accuracy and decrease in computational power.
3. We will use Deep Neural Networks (DNN) to create the model.
4. Deep Neural Network (DNN) is widely used in deep learning to train models for tasks which traditional machine learning algorithms cannot do or are hard to do.

5. The model is created using 5 layers of neural networks.
6. We have used dropouts to minimize the problem of overfitting.
7. In order to classify the audio to the emotions, we are using softmax in the outermost layer of our DNN model.
8. Softmax takes in a vector of numbers and converts them to probabilities which are then used for image generating results.
9. Softmax converts logits into probabilities by taking the exponents from every output and then normalizing each of these numbers by the sum of such exponents, such that the entire output vector adds up to one.

Prediction:

An audio is uploaded by the user (which includes speech of a person), and the model is used to predict the emotion of the speaker in the audio.

User Interface:

A Flask architecture based web application is developed to use the model. It has 2 parts, the system and the user. There is a user registration and login management system in the UI.

Registration:

A new user first needs to register their details which includes name, email and the password. This user information is stored in the MySQL database.

User Login:

An already registered user, whose data is stored in the system's MySQL database can login to the web app using their valid credentials. Once they successfully log in, only then they are provided access to the application to predict the emotions.

CHAPTER 7
EXPERIMENTATION

CHAPTER 7 EXPERIMENTATION

7.1 Data Augmentation

Figure 7.1 shows wave plots of audio files after applying different data augmentation techniques such as injecting noise, stretching audio, and changing the pitch of audio in order to increase the diversity of the dataset and reduce the overfitting of the model.

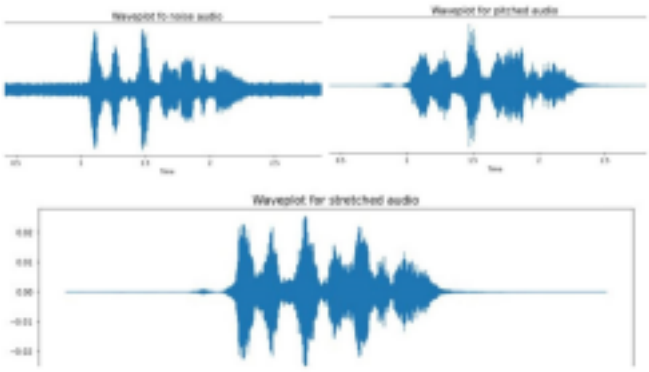


Figure 7.1. Waveplots of audio after applying different data augmentation techniques

7.2. FEATURE EXTRACTION

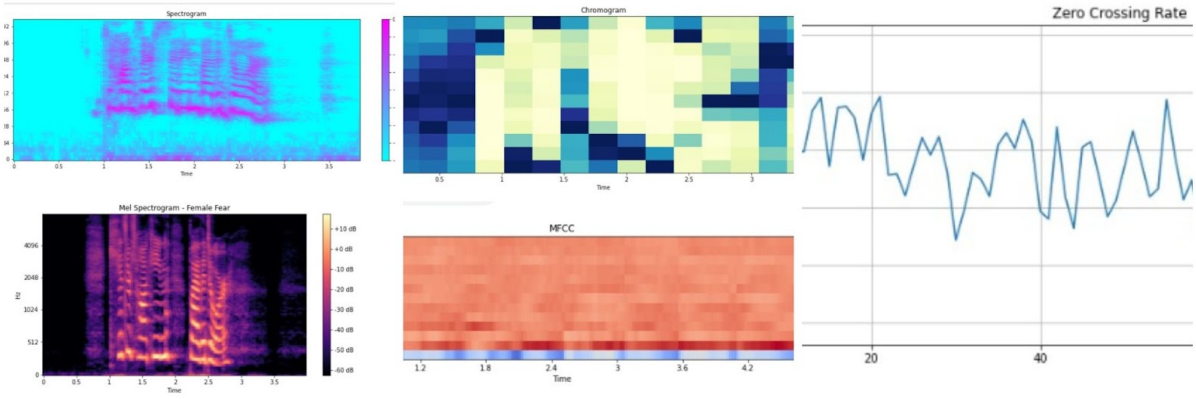


Figure 7.2. Different features extracted from audio

1. Mel-Spectrograms: These visualize audio or sound on a mel scale. The reason for using the mel scale is the way humans perceive sound is different from machines which have the same resolution in all frequencies unlike us who have higher resolution at lower frequencies. It is observed modeling human hearing property at feature extraction boosts the accuracy of the model so we convert our audio frequency to mel frequency.

2. Chroma: It is the standard representation of audio where a spectrogram is projected onto 12 bins indicating 12 distinct semitones. It indicates the energy of each pitch present in the signal on a standard chromatic scale.

3. Zero-Crossing Rate: It is the rate at which the signal changes from positive to negative and vice-versa. It can also be thought of as the number of times the signal crosses the horizontal axis.

4. MFCC: It represents the envelope of the short-time power spectrum which represents the shape of the vocal tract.

5. RMS value: It is one of the most important parameters as it indicates the strength or power of the signal.

7.3 Data Pre-Processing

In this Data-preprocessing, we will be loading features into the X variable and emotions into the Y variable. Since detecting the emotions of the speaker is a multiclass classification problem, we will be using a one-hot encoding technique by which categorical data are converted into binary features of data. Then we will be splitting the dataset into the training set and testing set. In our project, 75 percent is training data, and the rest 25 percent is testing data.

7.4 Imploying Models

1)CNN-LSTM: The CNN-LSTM model is well-suited for SER tasks as it can effectively capture both spatial and temporal features in speech signals. Here is a brief overview of how the model works:

1. Preprocessing: The speech signals are first preprocessed to extract relevant features such as Mel-frequency cepstral coefficients (MFCCs), which are commonly used in SER tasks.
2. Convolutional Neural Network: The CNN layer in the model is used to extract spatial features from the MFCCs. The CNN layer consists of multiple filters that convolve over the MFCCs to extract local features. The output of the CNN layer is a set of feature maps.
3. Long Short-Term Memory Network: The LSTM layer in the model is used to capture temporal dependencies in speech signals. The LSTM layer processes the feature maps generated by the CNN layer and outputs a sequence of hidden states that represent the temporal evolution of the speech signal.
4. Classification: The final output of the model is a probability distribution over the possible emotions. A fully connected layer with a softmax activation function is used to classify the input speech signal into one of the possible emotions.
5. The CNN-LSTM model has shown promising results in several SER benchmarks and can be further optimized by tuning hyperparameters and using more advanced architectures such as attention-based models

2)**ResNET-Transformer :** The ResNet-Transformer model for speech emotion recognition is a deep learning architecture that combines the residual network (ResNet) and transformer models.

The ResNet component is used for feature extraction, where the input speech signal is fed through a series of convolutional layers to extract high-level features. This helps to overcome the vanishing gradient problem that occurs in deep neural networks.

The transformer component is then used to capture the temporal dependencies between these features. The transformer is a self-attention mechanism that allows

the model to focus on important parts of the input sequence and can be used to model long-range dependencies.

The ResNet-Transformer model has shown to outperform traditional deep learning models for speech emotion recognition tasks. It can handle variable-length input sequences and is able to capture complex temporal patterns in speech signals.

7.5. Deploying Model

1. Prepare the deep learning model:
    a. Train and validate a Speech Emotion Recognition deep learning model using a dataset such as the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)
    b. Save the model's weights and architecture to a file using pickle library , which will be loaded later during the web app's runtime.

```
import pickle
# now you can save cnn-lstm model  to a file
with open('model_final.pkl', 'wb') as f:
    pickle.dump(model, f)
```

2. Install Flask and other dependencies:
    a. Install Flask, a Python web framework, using pip or another package manager.
    b. Install any additional dependencies required by your deep learning model, such as librosa or pydub.
3. Create a Flask app:
    a. Create a new Python file and import Flask and any other required libraries.
    b. Define a new Flask app instance using the Flask constructor.
    c. Define a new Flask route that will handle incoming requests to the app, such as '/predict'.
4. Load the deep learning model:
    a. Load the saved model's weights and architecture using your preferred deep learning framework's load_model() function.
    b. Save the loaded model as a global variable in your Flask app.
5. Process incoming requests:
    a. Define a function that will preprocess incoming requests to the Flask app, such as loading and processing an audio file.
    b. Use your deep learning model to predict the emotion of the audio, and save the predicted result in a variable.
6. Return the predicted result:

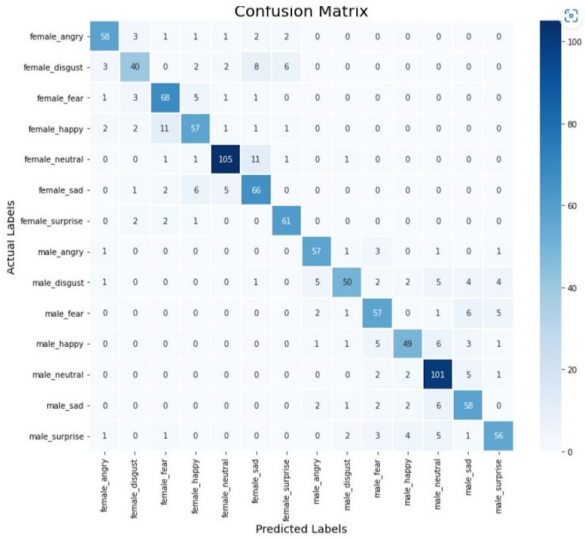
   a. Define a function that will return the predicted result to the user, such as in a JSON response.
   b. Return the predicted result to the user when they access the Flask app's specified route.

7. Run the Flask app:
   a. Run the Flask app on your local machine using the Flask run() function.

CHAPTER 8
TESTING AND RESULTS

CHAPTER 8 TESTING AND RESULTS

1) CNN-LSTM



	Predicted Labels	Actual Labels
0	female_fear	female_fear
1	male_angry	male_angry
2	female_fear	female_fear
3	female_neutral	female_neutral
4	female_angry	female_angry
5	female_surprise	female_surprise
6	male_sad	male_fear
7	male_happy	male_happy

Figure 8.1. Confusion Matrix And Output Table Of CNN-LSTM Model

Figure 8.1. illustrates the confusion matrix and the actual-predicted output of the CNN-LSTM model. It is observed that the accuracy of the model is around 82 percent from the classification report mentioned in Figure 8.8. clearly. This has overall good accuracy because the CNN-LSTM hybrid model was used for speech emotion detection where CNN extracts features and LSTM will handle sequential learning.

	precision	recall	f1-score	support
female_angry	0.87	0.85	0.86	68
female_disgust	0.78	0.66	0.71	61
female_fear	0.79	0.86	0.82	79
female_happy	0.78	0.76	0.77	75
female_neutral	0.91	0.88	0.89	120
female_sad	0.73	0.82	0.78	80
female_surprise	0.86	0.92	0.89	66
male_angry	0.85	0.89	0.87	64
male_disgust	0.88	0.68	0.76	74
male_fear	0.77	0.79	0.78	72
male_happy	0.83	0.74	0.78	66
male_neutral	0.81	0.91	0.86	111
male_sad	0.75	0.82	0.78	71
male_surprise	0.82	0.77	0.79	73
accuracy			0.82	1080
macro avg	0.82	0.81	0.81	1080
weighted avg	0.82	0.82	0.82	1080

Figure 8.2. Classification Report Of CNN-LSTM Model

2) RESNET-Transformer

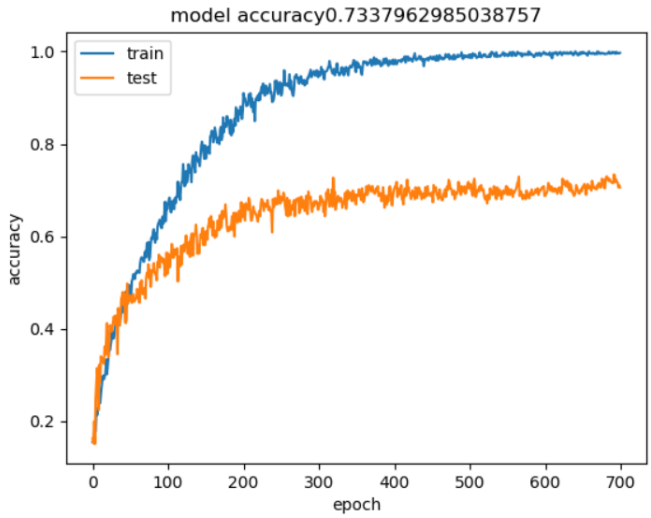


Figure 8.3. Model Accuracy Graph for Resnet-Transformer Model

Figure 8.4. illustrates the confusion matrix Resnet-Transformer model. It is observed that the accuracy of the model is around 73 percent from the model accuracy graph mentioned in figure 8.3. As you can see in the figure 8.3, the accuracy increases rapidly in the first two epochs, indicating that the network is learning fast. Afterwards, the curve flattens indicating that not too many epochs are required to train the model further.

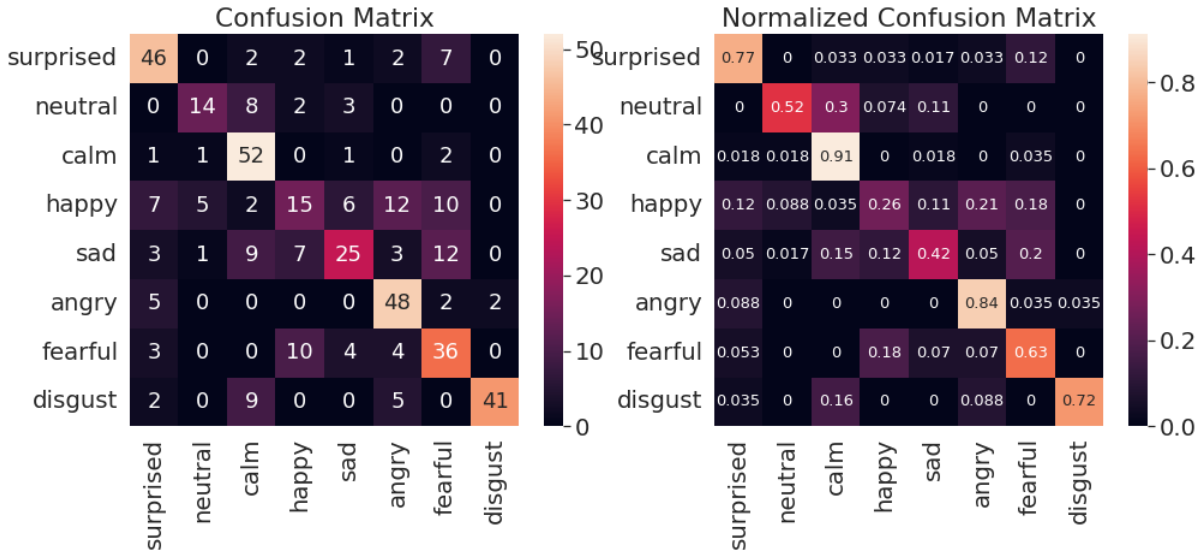


Figure 8.4.Confusion Matrix of Resnet-Transformer Model

3) Speech Emotion Recognition Website Home Page

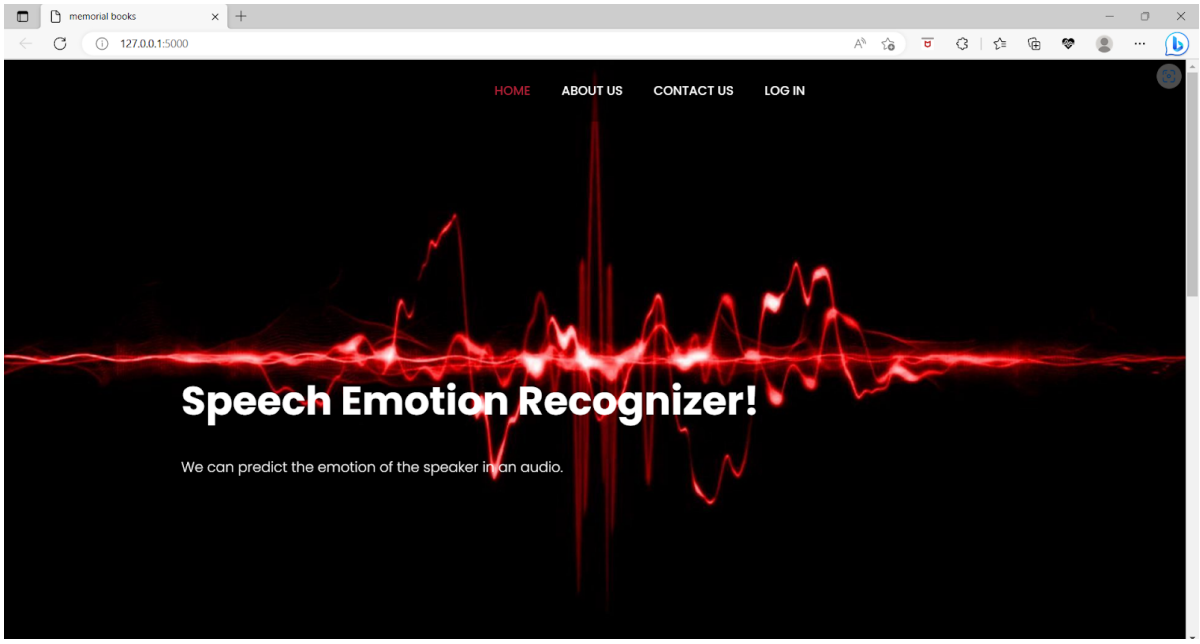


Figure 8.5. Home Page Of Speech Emotion Recognition Website

Figure 8.5. depicts the Home Page Of Speech Emotion Recognition website .The homepage typically includes a summary of the website's purpose, a brief introduction to the company or organization behind the website, and links to other important sections of the site, such as the "About" and "Contact Us" pages. The login button on the home page allows registered users to access their accounts, while new users can usually create an account from the login page. The "About" page provides more detailed information about the company or organization and its mission, while the "Contact Us" page typically includes a form or other means for visitors to get in touch with the website's owners or administrators.

4) Registration Page of Speech Emotion Recognition Website

Figure 8.6. depicts the registration page of Speech Emotion Recognition website.A registration page of a website that requires users to provide their username, password, and email address is a common type of user registration form. This form allows users to create an account on the website, which may give them access to certain features or content such as providing audio input to predict emotions.

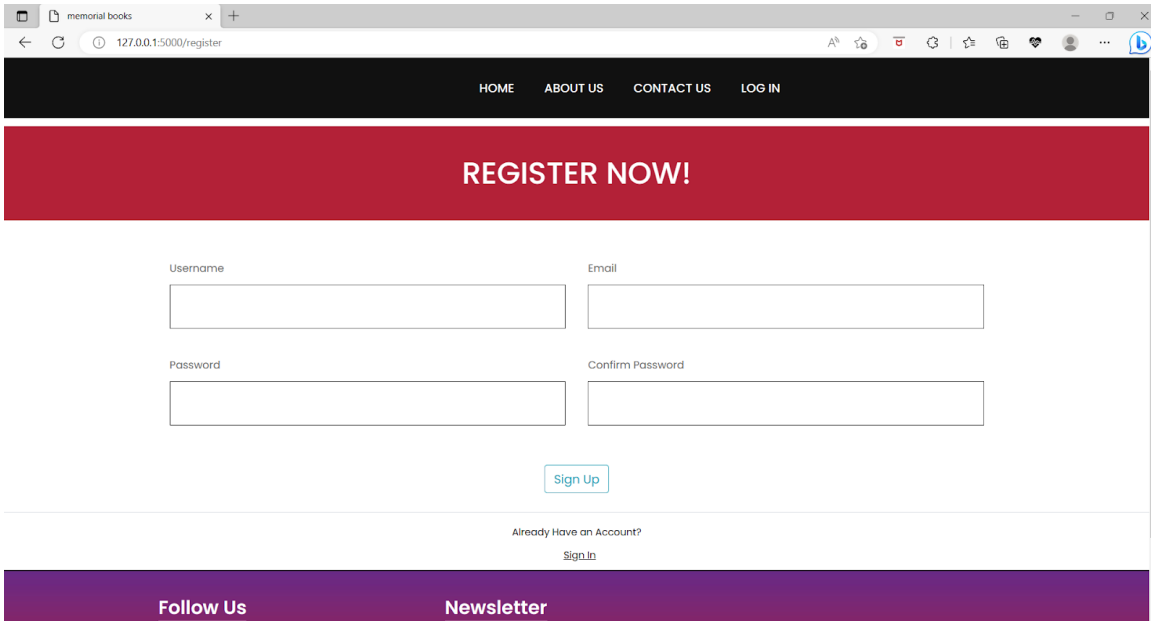


Figure 8.6. Registration Page of Speech Emotion Recognition Website

5) Login Page of Speech Emotion Recognition Website

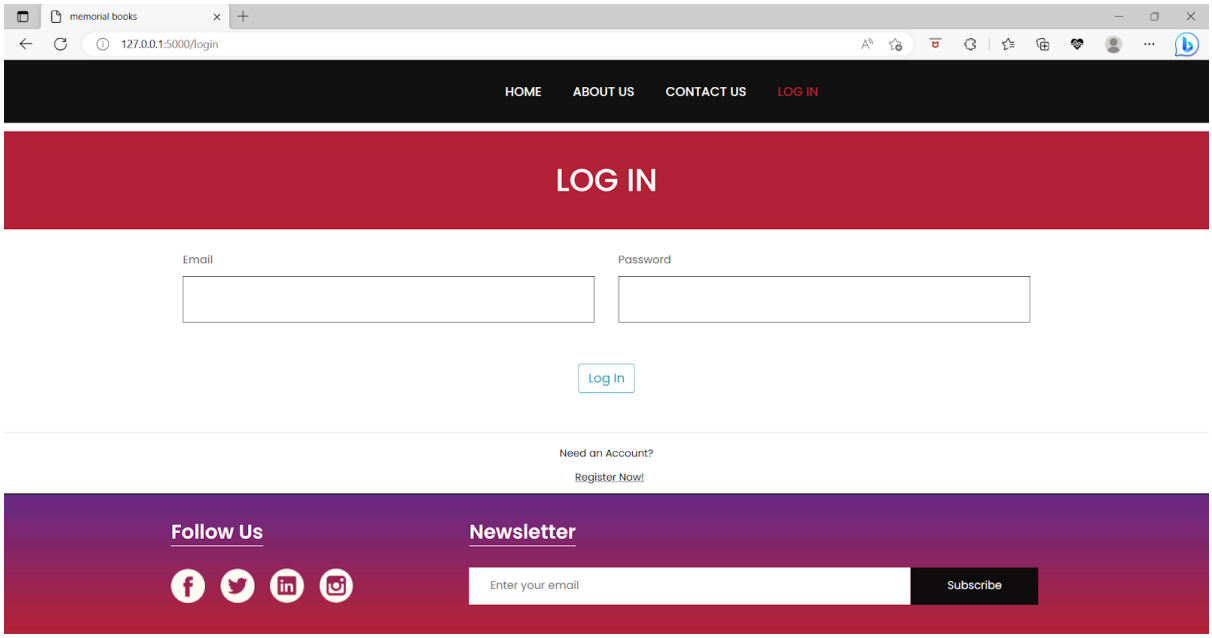
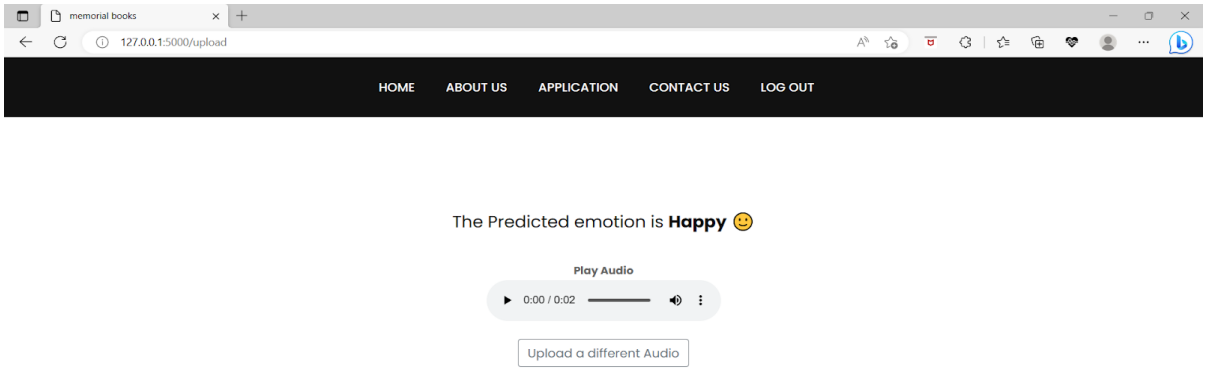


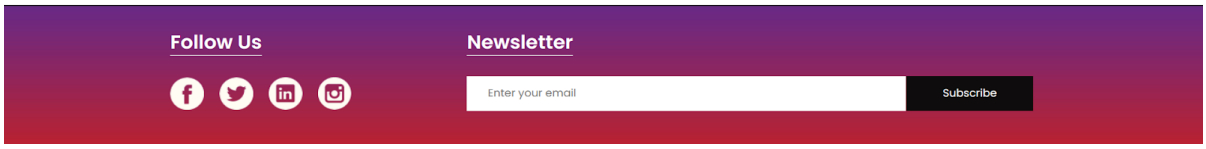
Figure 8.7. Login Page of Speech Emotion Recognition Website

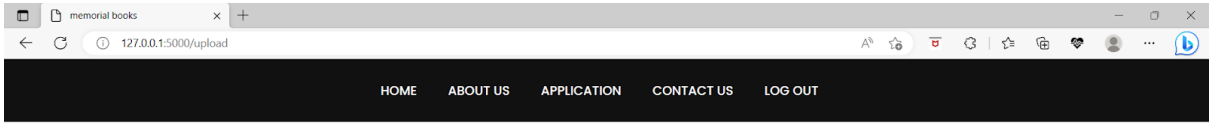
Figure 8.7. depicts the login page of Speech Emotion Recognition website where he has to log by providing email address and password in order to use speech emotion recognizer . If he has no account then he has to register by providing username , email

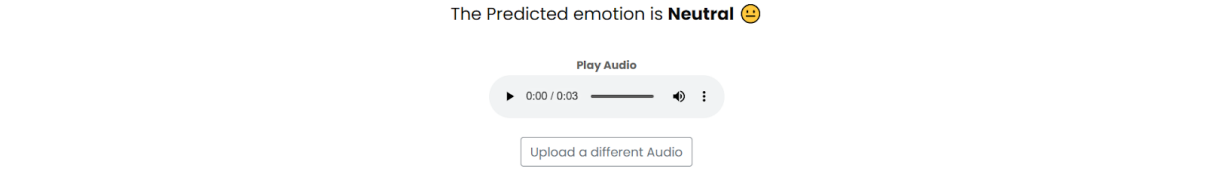
address and password.

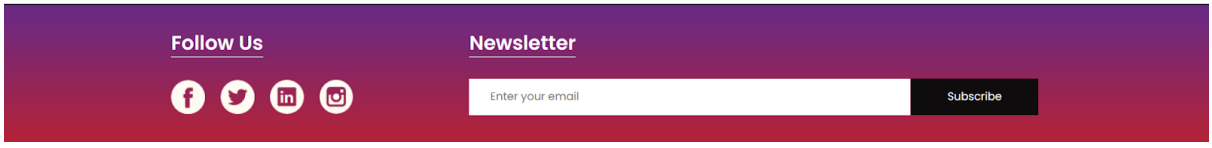
6) Final Output











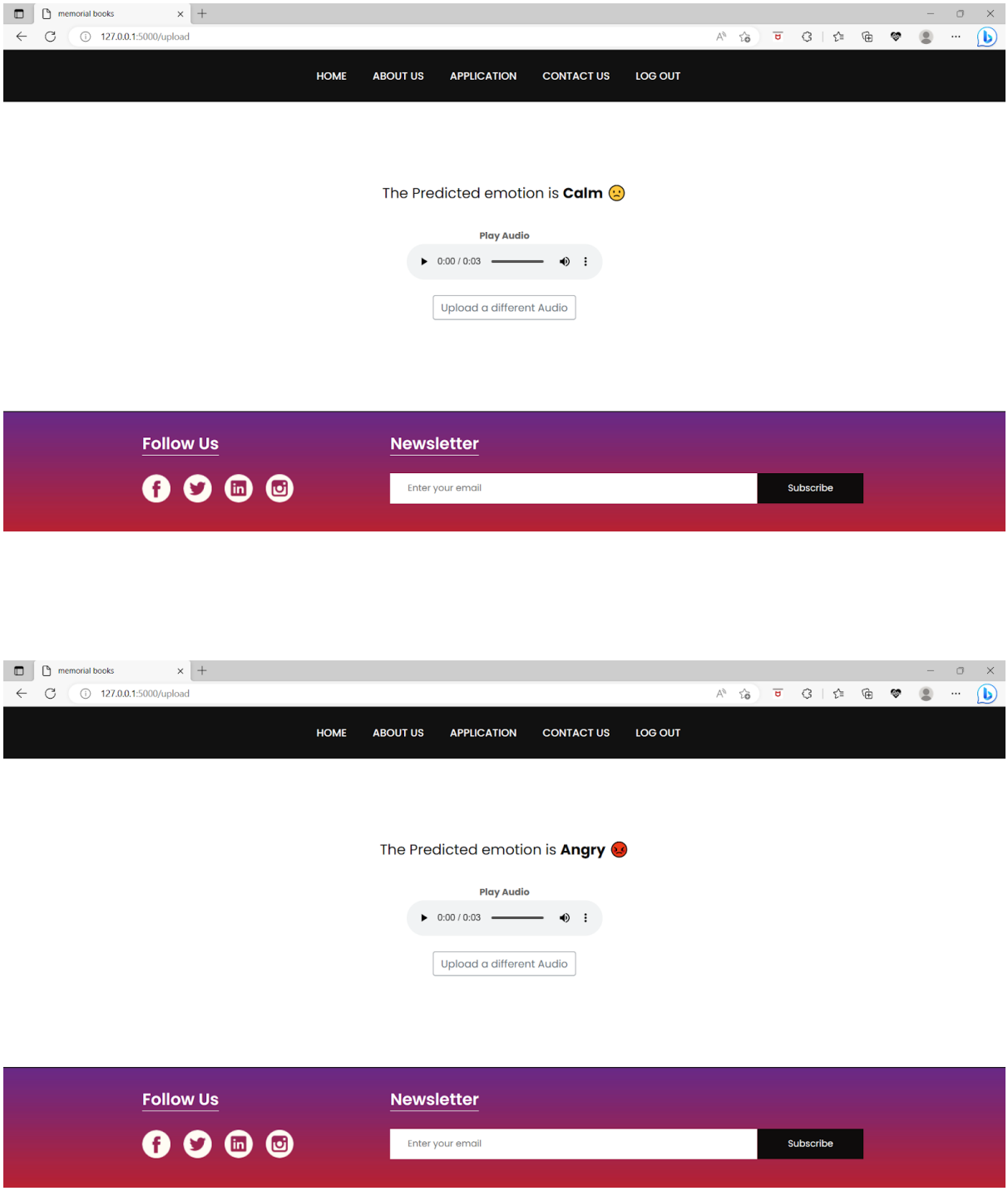


Figure 8.8. Final Output Of Speech Emotion Recognition Model.

Figure 8.8. depicts different final outputs of speech emotion recognition model . It displays different emotions such as happy , calm , neutral , angry etc, predicted for audio input given.

CHAPTER 9
CONCLUSION AND FUTURE WORK

CHAPTER 9 CONCLUSION AND FUTURE WORK

9.1. CONCLUSION

In conclusion, the development of speech-emotion recognition (SER) systems has gained significance as a research topic because of its potential applications in a range of fields, including human-computer interaction, medicine, and psychology.In this article, we covered the speech pre-processing, feature extraction, and classification processes that make up a typical SER system.

9.2. FUTURE WORK

- **Improving accuracy:** As speech emotion recognition technology continues to evolve, we can expect to see significant improvements in its accuracy. This could be achieved through more sophisticated algorithms, better training data, and more powerful computing resources.

- **Real-time emotion recognition:** One exciting application of speech emotion recognition technology is the ability to recognize emotions in real-time. This could be particularly useful in situations such as call centers, where agents could be alerted to a customer's emotional state and adjust their approach accordingly.

- **Multi-lingual support:** As speech emotion recognition systems become more advanced, they may be able to recognize emotions in multiple languages, making them even more useful for global communication.

- **Integration with other technologies:** Speech emotion recognition systems may also be integrated with other technologies such as virtual assistants, chatbots, and robotics to provide a more personalized and empathetic user experience.

REFERENCES

[1]Edward Jones et al(2019), "Speech Emotion Recognition Using Deep Learning Techniques : A Review," doi: 10.1109/ACCESS.2019.2936124

[2]Ron Hoory et al(2022),"Speech Emotion Recognition using Self Supervised Features,"https://www.researchgate.net/publication/358457970_Speech_Emotion_Recognition_using_Self-Supervised_Features.

[3]Mira Kartiwi et al(2021), "A Comprehensive Review of Speech Emotion Recognition Systems, doi: 10.1109/ACCESS.2021.3068045

[4]Srinivasa Parthasarathy et al (2019), "Semi-Supervised Speech Emotion Recognition with Ladder Networks," doi: https://doi.org/10.48550/arXiv.1905.02921.

[5]Aneesh Muppidi et al (2021), "Speech Emotion Recognition Using Quaternion Convolutional Neural,"ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021; doi: 10.1109/ICASSP39728.2021.9414248.

[6]Arya Aftab et al (2022), "Light-Sernet: A Lightweight Fully Convolutional Neural Network For Speech Emotion Recognition," ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022; doi:10.1109/ICASSP43922.2022.9746679

[7] Sathit Prasomphan et al (2015), "Detecting human emotion via speech recognition by using speech spectrogram" 2015 IEEE International conference on Data Science and Advance Analytics (DSAA) 2015; https://doi.org/10.1109/DSAA.2015.7344793

[8] Kotikalapudi Vamshi Krishna et al (2022), "Speech Emotion Recognition using Machine Learning" 2022 6th International Conference on Computing Methodologies and Communication (ICCMC) 2022; https://doi.org/10.1109/ICCMC53470.2022.9753976

[9] Apeksha Agarwal et al (2022), "Two-Way Feature Extraction for Speech Emotion Recognition Using Deep Learning" doi: https://doi.org/10.3390/s22062378

[10] Youddha Beer Singh et al (2022), "A systematic literature review of speech emotion recognition approaches" doi: https://doi.org/10.1016/j.neucom.2022.04.028

[11] Bagus Tris Atmaja et al (2022), "Survey on bimodal speech emotion recognition from acoustic and linguistic information fusion" doi: https://doi.org/10.1016/j.specom.2022.03.002

[12] N. Senthilkumar et al (2022), "Speech emotion recognition based on Bi-directional LSTM architecture and deep belief networks" doi: https://doi.org/10.1016/j.matpr.2021.12.246

[13] Sundararajan Srinivasan et al (2022), "Representation Learning Through Cross-Modal Conditional Teacher-Student Training For Speech Emotion Recognition" 2022 ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)

doi: https://doi.org/10.1109/ICASSP43922.2022.9747754

[14] V. M. Praseetha et al (2022), "Speech emotion recognition using data augmentation" doi: https://doi.org/10.1007/s10772-021-09883-3

[15] Kishor Bhangale et al (2022), "Speech Emotion Recognition Using Mel Frequency Log Spectrogram and Deep Convolutional Neural Network" doi: https://doi.org/10.1007/978-981-16-4625-6_24

SAMPLE CODE

SAMPLE CODE

A. BACKEND CODE

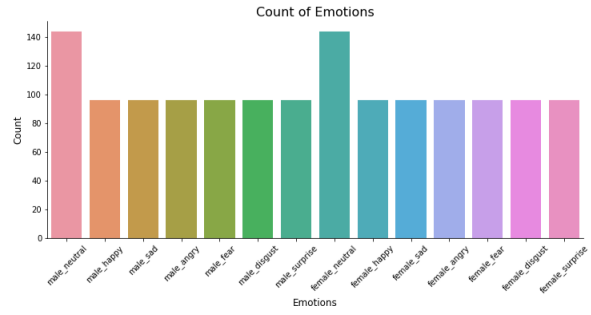
1. Importing libraries

```python
#IMPORT THE LIBRARIES
import pandas as pd
import numpy as np

import os
import sys
import glob

# librosa is a Python library for analyzing audio and music. It can be
import librosa
import librosa.display
import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import minmax_scale

# to play the audio files
import IPython.display as ipd
from IPython.display import Audio
from tqdm import tqdm
import nlpaug.augmenter.audio as naa
from pydub import AudioSegment
import keras
from keras.preprocessing import sequence
from keras.models import Sequential
from keras.layers import Dense, Embedding
from keras.layers import LSTM,BatchNormalization , GRU
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.utils import to_categorical
from keras.layers import Input, Flatten, Dropout, Activation
from keras.layers import Conv1D, MaxPooling1D, AveragePooling1D
from keras.models import Model
from keras.callbacks import ModelCheckpoint
from tensorflow.keras.optimizers import SGD
```

2.Data visualization

```
male_angry          96
male_fear           96
male_disgust        96
male_surprise       96
female_happy        96
female_sad          96
female_angry        96
female_fear         96
female_disgust      96
female_surprise     96
Name: labels, dtype: int64
```

```python
plt.figure(figsize=(12, 5))
plt.title('Count of Emotions', size=16)
sns.countplot(RAVD_df.labels)
plt.ylabel('Count', size=12)
plt.xlabel('Emotions', size=12)
plt.xticks(rotation=45)
sns.despine(top=True, right=True, left=False, bottom=False)
plt.show()
```



3. Data augmentation

```
# NOISE
def noise(data):
    noise_amp = 0.035*np.random.uniform()*np.amax(data)
    data = data + noise_amp*np.random.normal(size=data.shape[0])
    return data
# STRETCH
def stretch(data, rate=0.8):
    return librosa.effects.time_stretch(data, rate)
# SHIFT
def shift(data):
    shift_range = int(np.random.uniform(low=-5, high = 5)*1000)
    return np.roll(data, shift_range)
# PITCH
def pitch(data, sampling_rate, pitch_factor=0.7):
    return librosa.effects.pitch_shift(data, sampling_rate, pitch_factor)
```

4.Feature Extraction

FEATURE EXTRACTION

```
def feat_ext(data):
    mfcc = np.mean(librosa.feature.mfcc(y=data, sr=sample_rate).T, axis=0)
    return mfcc

def get_feat(path):
    data, sample_rate = librosa.load(path, duration=2.5, offset=0.6)
    # normal data
    res1 = feat_ext(data)
    result = np.array(res1)
    #data with noise
    noise_data = noise(data)
    res2 = feat_ext(noise_data)
    result = np.vstack((result, res2))
    #data with stretch and pitch
    new_data = stretch(data)
    data_stretch_pitch = pitch(new_data, sample_rate)
    res3 = feat_ext(data_stretch_pitch)
    result = np.vstack((result, res3))
    return result
```

5.Data preprocessing

```
X = Emotions.iloc[: ,:-1].values
Y = Emotions['labels'].values
```

```
# As this is a multiclass classification problem onehotencoding our Y
encoder = OneHotEncoder()
Y = encoder.fit_transform(np.array(Y).reshape(-1,1)).toarray()

```

```
# Train and Test Split
x_train, x_test, y_train, y_test = train_test_split(X, Y, random_state=0, shuffle=True)
x_train.shape, y_train.shape, x_test.shape, y_test.shape
```

```
((3240, 20), (3240, 14), (1080, 20), (1080, 14))
```

```
# Reshape for LSTM
X_train = x_train.reshape(x_train.shape[0] , x_train.shape[1] , 1)
X_test = x_test.reshape(x_test.shape[0] , x_test.shape[1] , 1)
```

```
# scaling our data with sklearn's Standard scaler
scaler = StandardScaler()
x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)
x_train.shape, y_train.shape, x_test.shape, y_test.shape

```

```
((3240, 20), (3240, 14), (1080, 20), (1080, 14))
```

6. Imploying Models

```
#CNN-Lstm
import tensorflow as tf
model = Sequential()
model.add(Conv1D(2048, kernel_size=5, strides=1, padding='same', activation='relu', input_shape=(x_train.shape[1], 1)))
model.add(MaxPooling1D(pool_size=2, strides = 2, padding = 'same'))
model.add(BatchNormalization())

model.add(Conv1D(1024, kernel_size=5, strides=1, padding='same', activation='relu'))
model.add(MaxPooling1D(pool_size=2, strides = 2, padding = 'same'))
model.add(BatchNormalization())

model.add(Conv1D(512, kernel_size=5, strides=1, padding='same', activation='relu'))
model.add(MaxPooling1D(pool_size=2, strides = 2, padding = 'same'))
model.add(BatchNormalization())

model.add(LSTM(256, return_sequences=True))

model.add(LSTM(128))

model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(64, activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(32, activation='relu'))
model.add(Dropout(0.2))

model.add(Dense(14, activation='softmax'))

optimiser = tf.keras.optimizers.Adam(learning_rate=0.0001)
model.compile(optimizer=optimiser,
                  loss='categorical_crossentropy',
                  metrics=['accuracy'])

model.summary()
```

7. .Evaluation Of Model

```
#CNN-Lstm
pred_test = model.predict(x_testcnn)
y_pred = encoder.inverse_transform(pred_test)

y_test = encoder.inverse_transform(y_test)

df = pd.DataFrame(columns=['Predicted Labels', 'Actual Labels'])
df['Predicted Labels'] = y_pred.flatten()
df['Actual Labels'] = y_test.flatten()

df.head(10)
```

```
#CNN-Lstm
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize = (12, 10))
cm = pd.DataFrame(cm , index = [i for i in encoder.categories_] , columns = [i for i in encoder.categories_])
sns.heatmap(cm, linecolor='white', cmap='Blues', linewidth=1, annot=True, fmt='')
plt.title('Confusion Matrix', size=20)
plt.xlabel('Predicted Labels', size=14)
plt.ylabel('Actual Labels', size=14)
plt.show()
```

B. FRONTEND CODE

1. Main.py

main.py 2 ×

CODE > main.py > home

```python
from flask import Flask, render_template, request, url_for, redirect, flash, send_from_directory, session
from forms import RegistrationForm, LoginForm
import pymysql
import pymysql.cursors
import pandas as pd
import os
from audio_wave import *

APP_ROOT=os.path.dirname(os.path.abspath(__file__))
app=Flask(__name__)
app.config['UPLOAD_FOLDER']=os.path.join(APP_ROOT, 'static/image/')
app.config['SECRET_KEY']='b0b4fbefdc48be27a6123605f02b6b86'

@app.before_first_request
def initialize():
    session['loggedin']=False

@app.route('/')
@app.route('/home')
def home():
    return render_template('index.html')

@app.route('/about')
def about():
    return render_template('about.html')

@app.route('/application')
def application():
    return render_template('application.html')

@app.route('/contact')
def contact():
    return render_template('contact.html')

@app.route('/library')
def library():
    return render_template('library.html')
```

2. Speech-extract.py

```python
#Import libraries
import glob
import os
import librosa
import numpy as np
from keras.models import Sequential
from keras.layers import Dense, Activation
from keras.layers import Dropout
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, accuracy_score
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from tensorflow.keras.callbacks import ReduceLROnPlateau, ModelCheckpoint


#Extract features
def extract_features(file_name):
    X, sample_rate = librosa.load(file_name)
    #Short time fourier transformation
    stft = np.abs(librosa.stft(X))
    #Mel Frequency Cepstra coeff (40 vectors)
    mfccs=np.mean(librosa.feature.mfcc(y=X, sr=sample_rate, n_mfcc=40).T, axis=0)
    #Chromogram or power spectrum (12 vectors)
    chroma=np.mean(librosa.feature.chroma_stft(S=stft, sr=sample_rate).T, axis=0)
    #mel scaled spectogram (128 vectors)
    mel=np.mean(librosa.feature.melspectrogram(y=X, sr=sample_rate).T, axis=0)
    # Spectral contrast (7 vectors)
    contrast=np.mean(librosa.feature.spectral_contrast(S=stft, sr=sample_rate).T, axis=0)
    #tonal centroid features (6 vectors)
    tonnetz=np.mean(librosa.feature.tonnetz(y=librosa.effects.harmonic(X),sr=sample_rate).T, axis=0)
    return mfccs, chroma, mel, contrast, tonnetz
```

3. Speech-train.py

```
#Loading features and labels
X=np.load('X.npy') #features
y=np.load('y.npy') #labels

#Splitting the dataset
train_X, test_X, train_y, test_y = train_test_split(X, y, test_size= 0.3,random_state=42)

def get_network():
    input_shape = (193,)
    num_classes = 8
    keras.backend.clear_session()

    model = keras.models.Sequential()
    model.add(keras.layers.Dense(1024, activation="relu", input_shape=input_shape))
    model.add(Dropout(0.5))
    model.add(keras.layers.Dense(512, activation="relu", input_shape=input_shape))
    model.add(keras.layers.Dense(256, activation="relu", input_shape=input_shape))
    model.add(keras.layers.Dense(128, activation="relu", input_shape=input_shape))
    model.add(keras.layers.Dense(num_classes, activation = "softmax"))
    model.compile(optimizer='adam',
        loss='categorical_crossentropy',
        metrics=["accuracy"])
    return model

model = get_network()

# Model Training
lr_reduce = ReduceLROnPlateau(monitor='val_accuracy', factor=0.9, patience=20, min_lr=0.000001)
# Please change the model name accordingly.
mcp_save = ModelCheckpoint('model/hello.h5', save_best_only=True, monitor='val_accuracy', mode='max')

#callbacks = [EarlyStopping(monitor='val_loss', mode='min', patience=20), mcp_save, lr_reduce]

history=model.fit(train_X, train_y, epochs = 700, batch_size = 24, validation_data=(test_X, test_y), callbacks=[mcp_save, lr_redu

#l, a = model.evaluate(x_test, y_test, verbose = 0)
```

4. forms.py

```
from flask_wtf import FlaskForm
from wtforms import StringField, PasswordField, SubmitField, BooleanField
from wtforms.validators import DataRequired, Length, Email, EqualTo

class RegistrationForm(FlaskForm):
    username=StringField('Username', validators=[DataRequired(), Length(min=3, max=20)])
    email=StringField('Email', validators=[DataRequired(), Email()])
    password=PasswordField('Password', validators=[DataRequired()])
    confirm_password=PasswordField('Confirm Password', validators=[DataRequired(), EqualTo('password')])

    submit=SubmitField('Sign Up')

class LoginForm(FlaskForm):
    email=StringField('Email', validators=[DataRequired(), Email()])
    password=PasswordField('Password', validators=[DataRequired()])
    remember=BooleanField('Remember Me')
    submit=SubmitField('Log In')
```

FUNDING AND PUBLISHED PAPER DETAILS

Emotion Analysis Using Speech

M M Krupashree[1], Naseeba Begum[2], Nayana Priya AP[3], Nithya S[4], Rashmi Motkur[5]*

[1]M M Krupashree, Department of Computer Science Engineering, Dayananda Sagar University, Bangalore, India

[2]Naseeba Begum, Department of Computer Science Engineering, Dayananda Sagar University, Bangalore, India

[3]Nayana Priya AP, Department of Computer Science Engineering, Dayananda Sagar University, Bangalore, India

[4]Nithya S, Department of Computer Science Engineering, Dayananda Sagar University, Bangalore, India

[5]Rashmi Motkur, Professor, Department of Computer Science Engineering, Dayananda Sagar University, Bangalore, India

***Abstract*—The main intention of our project is to detect the emotions elicited by the speaker while talking. For instance, speech generated in a state of fear, surprise, excitement, anger, or joy becomes loud and fast, with a higher and wider range in pitch, whereas emotions such as sadness or weariness produce slow and low-pitched speech. We are employing deep learning techniques to build a model which can identify human emotions through voice-pattern and speech-pattern analysis. The principal reason for selecting this project is speech emotion analysis has become one of the greatest commercialization strategies, in which the mood or temperament of the customer plays an immense role. So to detect the current emotion of the individual and recommending to him the appropriate product or helping him accordingly, will escalate the demand for the product or the company. It can also be used to monitor the neuropsychology state of an individual in lie detectors. In recent times, speech emotion recognition and analysis also finds applications in medical science and forensics.**

***Index Terms*—Deep learning, CNN, LSTM, MFCC, Mel spectrogram**

Keywords:deep learning;CNN;LSTM;MFCCS;mel-Spectrograms;

1. Introduction

Systems for recognizing speech emotions (SER) have developed from a specialized field to a crucial component of human-computer interaction (HCI). Instead of using conventional devices as input to understand rhetorical content and make it simple for human listeners to acknowledge, these HCI systems aim to speed up innate communication with machines through

explicit speech interaction. In some applications, dialogue systems for lingual languages are used for call center consultations, music recommendation systems are made based on the user's mood, and emotion analysis from the speech is used in medical and forensic applications. However, there are many difficulties with HCI systems, including noisy settings and different speaker accents that cause ambiguity that still needs to be properly resolved.

2. Literature Survey

Edward Jones et al [1] have presented a paper on Speech Emotion Recognition Using Deep Learning Techniques:A Review.These methods offer easy model training as well as the efficiency of shared weights. Limitations of deep learning techniques include their large layer-wise internal architecture, less efficiency for temporally-varying input data and over-learning during memorization of layer-wise information.This research work forms a base to evaluate the performance and limitations of current deep learning techniques.

Ron Hoory et al [2] have presented a paper on Speech Emotion Recognition Using Self-Supervised Features.They have clearly shown that well designed combinations of carefully fine-tuned and averaged Upstream models and averaged Downstream models can significantly improve the performance of E2E SER models.This research paper aims to introduce a modular End-to-End (E2E) SER system based on an Upstream + Downstream architecture model paradigm

Mira Kartiwi et al [3] have presented a paper on A Comprehensive Review of Speech Emotion Recognition Systems.The paper carefully identifies and synthesizes recent relevant literature related to the SER systems' varied design components/methodologies, thereby providing readers with a state-of-the-art understanding of the hot research topic.

This paper points out that deep learning techniques are considered best suited for the SER system over traditional techniques because of their advantages like scalability, all-purpose parameter fitting, and infinitely flexible function.

Srinivasa Parthasarathy et al [4] have presented a paper on Semi-Supervised Speech Emotion Recognition With Ladder Networks.The results indicated significant gains when using the proposed models, underlying the generalization power of the ladder networks. The improvements were particularly high when using unlabeled data from the target domain, exploiting all the benefits of the proposed architecture.This paper pertains to one major drawback of the SER system, that is , is their lack of generalization across different conditions which can be resolved using ladder networks. . It combines the unsupervised auxiliary task of reconstructing intermediate feature representations, with the primary task of predicting emotional attributes.

3. Design

3.1. System Architecture

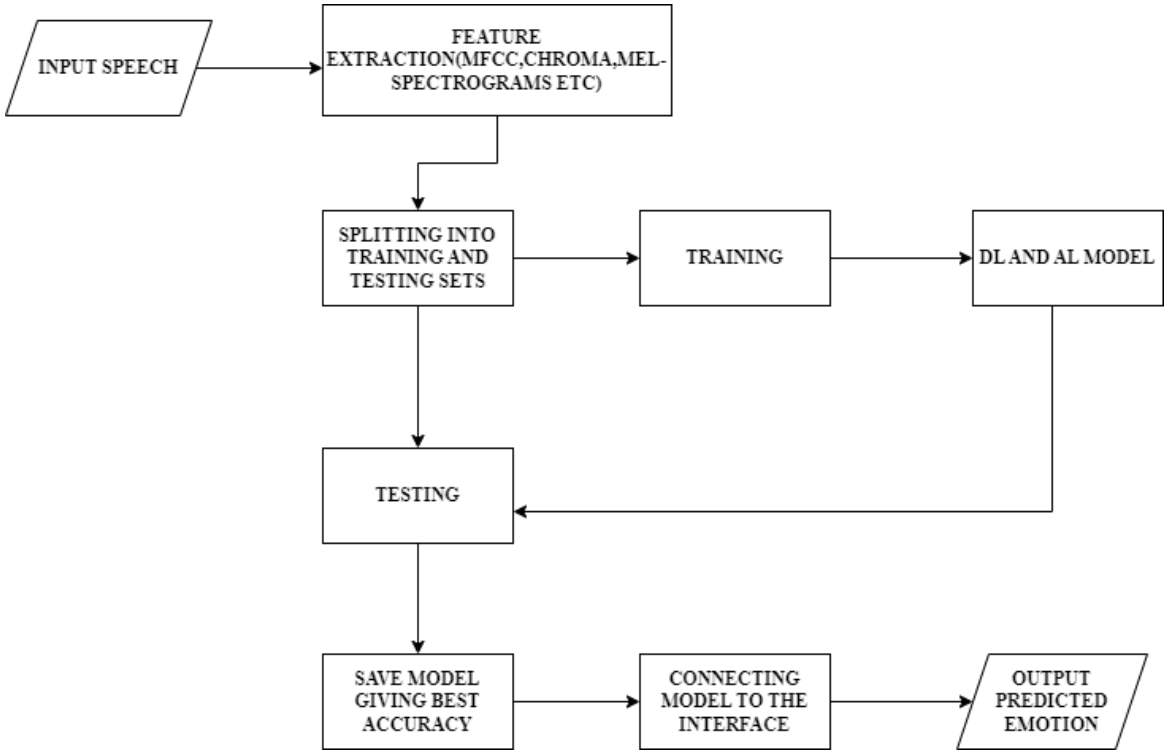


Figure 3.1.System Architecture of Speech Emotion Recognition System

Figure 3.1 depicts the overall architecture of the project where input is taken in the form of speech or audio. Then we have to extract features from the input. Some of the main audio features extracted are Mel-frequency Cepstral Coefficient(MFCC), pitch, Mel-Spectrograms, chroma, zero-crossing rate, etc. Then the dataset is split into the training set and testing set. The very next step is training Deep-learning and Artificial Intelligence models using each feature extracted from the training dataset.A testing dataset is used to evaluate the developed model. Then we will save the model which gives the best accuracy. then we will be connecting the model to the interface and output the predicted emotion.

3.2. Data Flow Diagram

Figure 3.2 illustrates the data flow diagram of the project. Once the SER model is trained and tested, it is exported or embedded into an app. When you start the application it prompts the user to give input using Google Speech API. Input data is sent to the server via HTTP post request where it receives input and does feature extraction and tests extracted features using an already trained SER model/Then it predicts emotion and returns a JSON response.

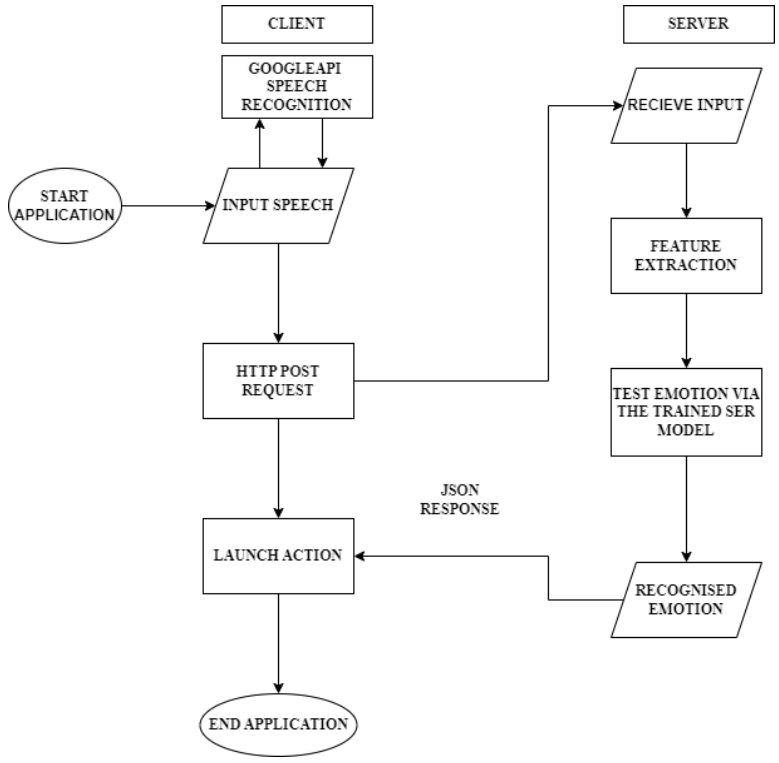


Figure 3.2.Data Flow Diagram of Speech Emotion Recognition System

3.3. Backend Architecture

Figure 3.3 depicts the backend architecture of our project where standard datasets are taken as input and divided into training samples and testing samples. Training samples undergo pre-processing such as converting audio waves to mel spectrogram and data augmentation which increases the diversity of the dataset by using standard augmentation techniques such as changing pitch, injecting noise, etc. The next step is feature extraction which extracts features such as MFCC, chroma, and Mel-frequency spectrograms. These extracted features are then sent to classifiers for predicting emotion. To evaluate the model, we will be using testing samples that undergo feature extraction and are sent to classifiers for predicting emotion. We can also test by providing live input via google speech recognition API which undergoes feature extraction and is sent to the model for predicting emotion.

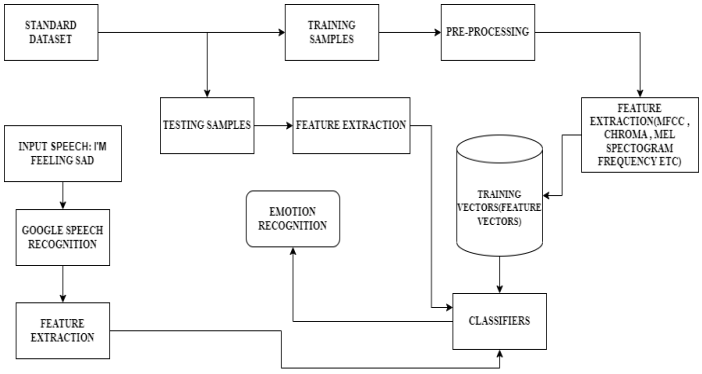


Figure 3.3. Backend Architecture of Speech Emotion Recognition System

4. Methodology

4.1 Existing system

The majority of the SER systems presently on the market employ traditional machine learning algorithms for emotion recognition, including Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Gaussian Mixture Model (GMM), etc.. The accuracies of these models are low and have high computational complexity. However, there are other deep learning models such as Convolutional Neural Network(CNN), Quaternion Convolutional Neural Network(CNN), and, Long-Short Term memory(LSTM), etc which give average accuracy of around 80 percent because of their computational complexity and many other reasons.

4.2 Proposed System

We propose an enhanced speech emotion recognition method that uses hybrid model of deepneural networks that is, CNN and LSTM to detect emotions elicited by the speaker.the method used Mel-frequency cepstral coefficients (MFCC), chromogram, Mel scale spectrogram in conjunction with spectral contrast to extract details about an audio file. These features are used to train our hybrid model which gives better accuracy compared to other existing models. The model classifies the speech audio in 8 different emotions such as neutral , calm , surprise , happy , anger , fearful , disgust, sad.

4.3.Proposed Methodology

The system's architecture makes it clear that we are using voice training. and it is then passed for preprocessing for the feature extraction of the sound which then gives the training arrays.These arrays are then used to form “classifiers “for making decisions about the emotion. So, a big data set of voices of different emotions is needed for the training sample. We searched on the web and found different sets of datasets some of which are mentioned below:

1. Crowd-sourced Emotional Multimodal Actors Dataset(Crema-D)
2. Ryerson Audio-Visual Database of Emotional Speech and Song (Ravdess)
3. Surrey Audio-Visual Expressed Emotion(Savee)
4. Toronto emotional speech set (Tess)

5. Experimentation

5.1 Data Augmentation

Figure 5.1 shows wave plots of audio files after applying different data augmentation techniques such as injecting noise, stretching audio, and changing the pitch of audio in order to increase the diversity of the dataset and lessen the model's overfitting.

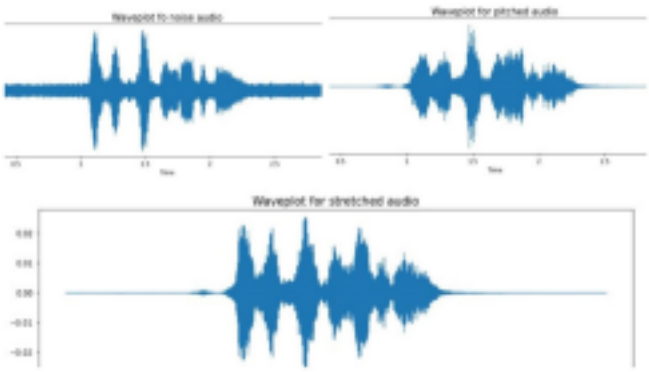


Figure 5.1.Wave plots of audio after applying different data augmentation techniques

5.2 Feature Extraction

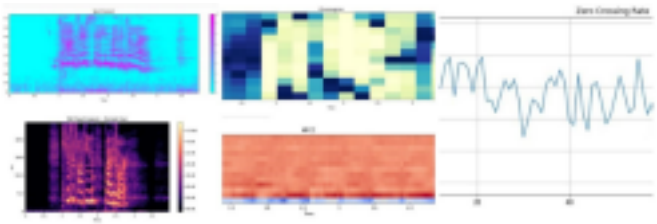


Figure 5.2.Different features extracted from audio

Figure 5.2 includes different features extracted from audio.

1. Mel-Spectrograms, which represent sound or audio on a mel scale. The mel scale is used because humans perceive sound differently from machines, which have a resolution that is the same across all frequencies as opposed to our higher resolution at lower frequencies. We convert our audio frequency to mel frequency because it has been found that simulating the human hearing characteristic during feature extraction improves the model's accuracy.

2. Chroma: A spectrogram is projected onto 12 bins to represent the 12 distinct semitones in the standard audio representation of audio. On a typical chromatic scale, it displays the energy of each pitch that is present in the signal.

3. The zero-crossing rate is the speed at which a positive signal turns negative and vice versa. The frequency of the signal crossing the horizontal axis is another way to conceptualize it.

4. MFCC: It represents the short-time power spectrum envelope, which represents the vocal tract's shape.

5. RMS value: One of the most crucial parameters, it shows the signal's strength or power.

5.3 Data Pre-Processing

In this Data-preprocessing, we will be loading features into the X variable and emotions into the Y variable. Since detecting the emotions of the speaker is a multiclass classification problem, we will be using a one-hot encoding technique by which categorical data are converted into binary features of data. Then we will be splitting the dataset into the training set and testing set. In our project, 75 percent is training data, and the rest 25 percent is testing data. then we will be standardizing data using StandardScaler to make sure all variables contribute equally.

6. Result And Analysis

6.1.Decision Tree

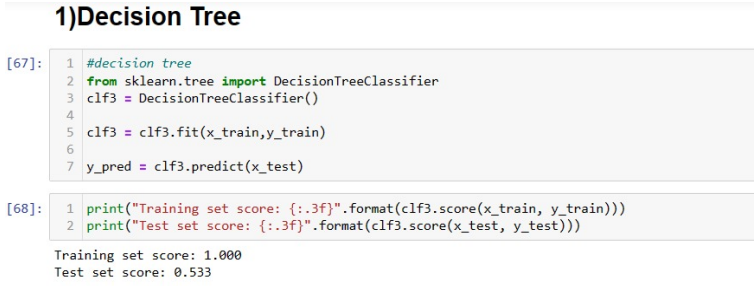


Figure 6.1.Training And Test Set Score Using Decision Tree

Figure 6.1 depicts training and a test score of the decision tree model. It is observed that the training score is 100 percent which is unusual whereas the test score is around 40 percent which indicates the model is overfitting. This is caused because the model is memorizing exact input and output pairs in training data instead of learning patterns. So, it underperforms when evaluated on test data.

6.2.KNN

Figure 6.2. illustrates training and test scores of the KNN model. It is observed that the training score is around 49 percent and the test score is around 37 percent. This accuracy is not good for deployment.

2)KNN

```
In [69]:
#knn
from sklearn.neighbors import KNeighborsClassifier
clf1=KNeighborsClassifier(n_neighbors=4)
clf1.fit(x_train,y_train)

Out[69]: KNeighborsClassifier(n_neighbors=4)

In [70]:
y_pred=clf1.predict(x_test)

In [71]:
print("Training set score: {:.3f}".format(clf1.score(x_train, y_train)))
print("Test set score: {:.3f}".format(clf1.score(x_test, y_test)))

Training set score: 0.492
Test set score: 0.372
```

Figure 6.2.Training And Test Set Score Using KNN

6.3. MLP Classifier

Figure 6.3. depicts the training and test scores of the MLP Classifier. It is observed that the training score is around 80 percent which is good but the test score is around 50 percent which makes the model not suitable for deployment.

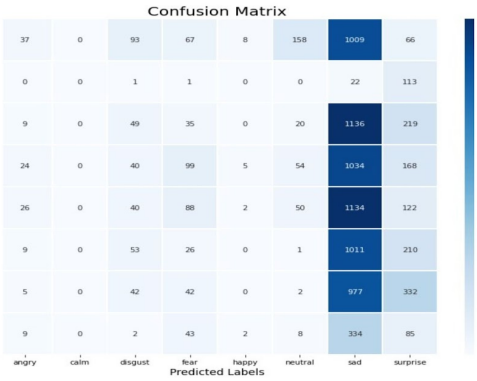
3)MLP Classifier

```
[72]:
#MLP Classifier
from sklearn.neural_network import MLPClassifier
clf2=MLPClassifier(alpha=0.01, batch_size=270, epsilon=1e-08, hidden_layer_sizes=(400,), learning_rate='adaptive', max_iter
clf2.fit(x_train,y_train)

[72]: MLPClassifier(alpha=0.01, batch_size=270, hidden_layer_sizes=(400,),
              learning_rate='adaptive', max_iter=400)

[73]:
print("Training set score: {:.3f}".format(clf2.score(x_train, y_train)))
print("Test set score: {:.3f}".format(clf2.score(x_test, y_test)))

Training set score: 0.804
Test set score: 0.535
```

Figure 6.3.Training And Test Set Score Using MLP Classifier

6.4. LSTM

Figure 6.4. illustrates the confusion matrix and the actual-predicted output of the LSTM model. It has been found that the model's accuracy is approximately 67%.Input data used in our project is sequential data and the LSTM model is predominantly used for this kind of data since it can remember long-term dependencies between time steps of data. Training accuracy was good but it underperformed on testing data. This is because LSTMs are easily overfitted and implementing dropouts is quite difficult.

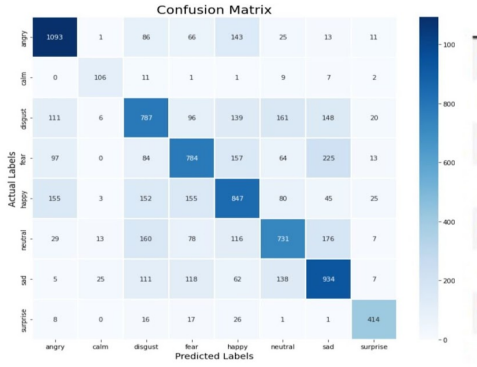


	Predicted Labels	Actual Labels
0	surprise	disgust
1	surprise	disgust
2	neutral	angry
3	sad	disgust
4	sad	fear
5	sad	fear
6	fear	happy
7	sad	happy
8	sad	sad
9	sad	sad

Figure 6.4.Confusion Matrix And Output Table Of LSTM Model

6.5. CNN

Figure 6.5.1 illustrates the confusion matrix and the actual-predicted output of the CNN model. It is observed that the accuracy of the model is around 62 percent from the classification report mentioned in Figure 6.5.2 clearly. We can see our model is more accurate in predicting surprise, and angry emotions and it makes sense also because audio files of these emotions differ from other audio files in a lot of ways like pitch, speed, etc.



	Predicted Labels	Actual Labels
0	sad	disgust
1	disgust	disgust
2	angry	angry
3	happy	disgust
4	sad	fear
5	disgust	fear
6	angry	happy
7	happy	happy

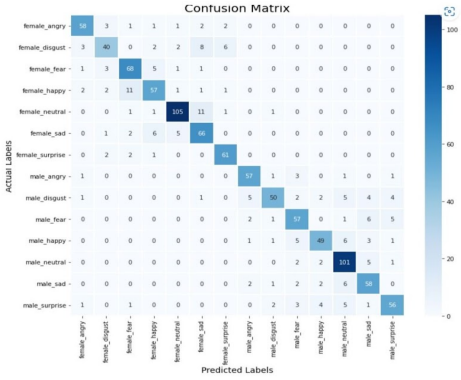
Figure 6.5.1.Confusion Matrix And Output Table Of CNN Model

	precision	recall	f1-score	support
angry	0.73	0.76	0.74	1438
calm	0.69	0.77	0.73	137
disgust	0.56	0.54	0.55	1468
fear	0.60	0.55	0.57	1424
happy	0.57	0.58	0.57	1462
neutral	0.60	0.56	0.58	1310
sad	0.60	0.67	0.63	1400
surprise	0.83	0.86	0.84	483
accuracy			0.62	9122
macro avg	0.65	0.66	0.65	9122
weighted avg	0.62	0.62	0.62	9122

Figure 6.5.2.Classification Report Of CNN Model

6.6. CNN-LSTM

Figure 6.6.1. illustrates the confusion matrix and the actual-predicted output of the CNN-LSTM model. It is observed that the accuracy of the model is around 82 percent from the classification report mentioned in Figure 6.6.2. clearly. This has overall good accuracy because the CNN-LSTM hybrid model was used for speech emotion detection where CNN extracts features and LSTM will handle sequential learning.



	Predicted Labels	Actual Labels
0	female_fear	female_fear
1	male_angry	male_angry
2	female_fear	female_fear
3	female_neutral	female_neutral
4	female_angry	female_angry
5	female_surprise	female_surprise
6	male_sad	male_fear
7	male_happy	male_happy

Figure 6.6.1.Confusion Matrix And Output Table Of CNN-LSTM Model

	precision	recall	f1-score	support
female_angry	0.87	0.85	0.86	68
female_disgust	0.78	0.66	0.71	61
female_fear	0.79	0.86	0.82	79
female_happy	0.78	0.76	0.77	75
female_neutral	0.91	0.88	0.89	120
female_sad	0.73	0.82	0.78	80
female_surprise	0.86	0.92	0.89	66
male_angry	0.85	0.89	0.87	64
male_disgust	0.88	0.68	0.76	74
male_fear	0.77	0.79	0.78	72
male_happy	0.83	0.74	0.78	66
male_neutral	0.81	0.91	0.86	111
male_sad	0.75	0.82	0.78	71
male_surprise	0.82	0.77	0.79	73
accuracy			0.82	1080
macro avg	0.82	0.81	0.81	1080
weighted avg	0.82	0.82	0.82	1080

Figure 6.6.2.Classification Report Of CNN-LSTM Model

7. Conclusion

In conclusion, because of its potential applications in a variety of domains, including human-computer interaction, healthcare, and psychology, the development of speech-emotion recognition (SER) systems has grown in importance as a research issue. The goal of SER is to automatically ascertain a speaker's emotional state from their speech signal. In this article, we covered the speech pre-processing, feature extraction, and classification processes that make up a typical SER system. We covered a number of methods for each of these elements, including feature extraction methods like Mel frequency cepstral coefficients (MFCCs), signal processing methods like filtering, and classification algorithms like support vector machines (SVMs) and deep learning models.Although SER has generally made tremendous progress in recent years, there is still much need for improvement.Overall, there is still much opportunity for improvement even though SER has made significant progress in recent years. To increase the

reliability and accuracy of SER systems and to make them viable for use in real-world circumstances, more research and development is required.

References

[1]Edward Jones et al(2019), "Speech Emotion Recognition Using Deep Learning Techniques : A Review," doi: 10.1109/ACCESS.2019.2936124

[2]Ron Hoory et al(2022),"Speech Emotion Recognition using Self Supervised Features,"https://www.researchgate.net/publication/358457970_Speech_Emotion_Recognition_using_Self-Supervised_Features.

[3]Mira Kartiwi et al(2021), "A Comprehensive Review of Speech Emotion Recognition Systems, doi: 10.1109/ACCESS.2021.3068045

[4]Srinivasa Parthasarathy et al (2019), "Semi-Supervised Speech Emotion Recognition with Ladder Networks," doi: https://doi.org/10.48550/arXiv.1905.02921.

[5]Aneesh Muppidi et al (2021), "Speech Emotion Recognition Using Quaternion Convolutional Neural,"ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021; doi: 10.1109/ICASSP39728.2021.9414248.

[6]Arya Aftab et al (2022), "Light-Sernet: A Lightweight Fully Convolutional Neural Network For Speech Emotion Recognition," ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022; doi:10.1109/ICASSP43922.2022.9746679

[7] Sathit Prasomphan et al (2015), "Detecting human emotion via speech recognition by using speech spectrogram" 2015 IEEE International conference on Data Science and Advance Analytics (DSAA) 2015; https://doi.org/10.1109/DSAA.2015.7344793

[8] Kotikalapudi Vamshi Krishna et al (2022), "Speech Emotion Recognition using Machine Learning" 2022 6th International Conference on Computing Methodologies and Communication (ICCMC) 2022; https://doi.org/10.1109/ICCMC53470.2022.9753976

[9] Apeksha Agarwal et al (2022), "Two-Way Feature Extraction for Speech Emotion Recognition Using Deep Learning" doi: https://doi.org/10.3390/s22062378

[10] Youddha Beer Singh et al (2022), "A systematic literature review of speech emotion recognition approaches" doi: https://doi.org/10.1016/j.neucom.2022.04.028

[11] Bagus Tris Atmaja et al (2022), "Survey on bimodal speech emotion recognition from acoustic and linguistic information fusion" doi: https://doi.org/10.1016/j.specom.2022.03.002

[12] N. Senthilkumar et al (2022), "Speech emotion recognition based on Bi-directional LSTM architecture and deep belief networks" doi: https://doi.org/10.1016/j.matpr.2021.12.246

[13] Sundararajan Srinivasan et al (2022), "Representation Learning Through Cross-Modal Conditional Teacher-Student Training For Speech Emotion Recognition" 2022 ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) doi: https://doi.org/10.1109/ICASSP43922.2022.9747754

[14] V. M. Praseetha et al (2022), "Speech emotion recognition using data augmentation" doi: https://doi.org/10.1007/s10772-021-09883-3

[15] Kishor Bhangale et al (2022), "Speech Emotion Recognition Using Mel Frequency Log Spectrogram and Deep Convolutional Neural Network" doi: https://doi.org/10.1007/978-981-16-4625-6_24

NATIONAL INSTITUTE OF TECHNOLOGY PUDUCHERRY

CERTIFICATE OF PRESENTATION

THIS IS TO CERTIFY THAT

Rashmi Mothkur, NASEEBA BEGUM, M M KRUPASHREE, NITHYA S, NAYANA PRIYA A P

PRESENTED THE PAPER TITLED

Emotion Analysis Using Speech

In the International Conference on intelligent COMPUting TEchnologies and Research (i-COMPUTER 2023) held on 24 -25 March 2023, Department of Computer Science and Engineering, NIT Puducherry, Karaikal.

03/04/2023

Dr. M Venkatesan
Chairperson