SPEECH EMOTION RECOGNITION

## A PROJECT REPORT

***Submitted by***

# ABSARKHAN.K (Reg.No.1920106006) AJITH.V (Reg.No.1920106008) GURUSURYA.M (Reg.No.1920106024) HARIHARAN.A (Reg.No.1920106026)

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

# BACHELOR OF TECHNOLOGY

## IN INFORMATION TECHNOLOGY

**SONA COLLEGE OF TECHNOLOGY**

# ANNA UNIVERSITY: CHENNAI 600 025

## NOVEMBER 2023

**BONAFIDE CERTIFICATE**

Certified that this project report **“SPEECH EMOTION RECOGNITION”** is the bonafide work of **“ABSARKHAN.K (1920106006), AJITH.V (1920106008), GURUSURYA.M (1920106024), HARIHARAN.A**

**(1920106026)”** who carried out the project work under my supervision.

|  |  |
| --- | --- |
| **SIGNATURE** | **SIGNATURE** |
| (Dr.J.Akilandeswari) | (Mr.R.Krishna Prakash) |
| Professor | Assistant Professor |
| **HEAD OF THE DEPARTMENT** | **SUPERVISOR** |
| Department of Information Technology | Department of Information Technology |
| Sona College of Technology | Sona College of Technology |
| Salem-636 005 | Salem-636 005 |

Submitted for Mini Project viva voice examination held on

……………………………..

|  |  |
| --- | --- |
| **INTERNAL EXAMINER** | **EXTERNAL EXAMINER** |

# ABSTRACT

Speech Emotion Recognition (SER) is an evolving field that involves the identification and interpretation of emotions expressed in spoken language. This project focuses on developing an effective SER system by employing advanced signal processing techniques and machine learning models. Leveraging a diverse dataset, we extract key features from speech signals, including Mel-Frequency Cepstral Coefficients (MFCCs) and prosodic features. Our investigation encompasses both traditional machine learning algorithms and deep learning architectures for emotion classification. Through meticulous training, validation, and performance evaluation, we aim to achieve a robust SER model capable of accurately recognizing a spectrum of emotions. The outcomes of this project contribute to the refinement of SER methodologies and the responsible integration of emotion recognition technology into real-world applications.

## ACKNOWLEDGEMENT

First and foremost, we thank to **power of almighty** for showing us inner peace and for all blessings. Special gratitude to our parents, for showing their support and love always.

We express our sincere thanks to chairman **Sri.C.VALLIAPPA** and principal **Dr.S.R.R.SENTHILKUMAR** for providing adequate facilities to complete the project.

We express our sincere thanks to the Head of Information Technology, **Dr.J.Akilandeswari,M.E,Ph.D,** for providing adequate laboratory facilities to complete this thesis.

We feel elated to keep on record our heartfelt thanks and gratitude to our project guide **Mr.R.Krishna Prakash** our steadfast inspiration, for his valuable guidance, untiring patience and diligent encouragement during the entire span of this project.

We feel proud in sharing this success with my staff members, non- teaching staffs and friends who helped directly or indirectly in completing this project successfully.

## TABLE OF CONTENTS

**CHAPTER NO. TITLE PAGE NO.**

|  |  |
| --- | --- |
| **ABSTRACT**  **LIST OF FIGURES**  **LIST OF SYMBOLS AND ABBREVIATIONS**  1. **INTRODUCTION** | **i ii**  **iii**  **1** |
| 1.1 Speech Emotion Recognition | 1 |
| 1.2 Convolutional Neural Network | 2 |
| 1.3 Characteristics of SER | 3 |
| 1.4 Features of SER | 4 |
| 1.5 Components of SER | 5 |
| 1.6 Applications | 9 |
| 2. **LITERATURE SURVEY** | **12** |
| 3. **HARDWARE AND SOFTWARE** | **15** |
| **REQUIREMENTS** |  |
| 4. **PROJECT DESCRIPTION** | **16** |
| 4.1 Objective | 16 |
| 4.2 MODULES |  |
| 4.2.1 Data Collection | 16 |
| 4.2.2 Feature Extraction | 17 |
| 4.2.3 Test Train Split | 18 |
| 4.2.4 Classification | 18 |

4.2.5 Load Model 19

[4.3 Workflow 19](#_TOC_250006)

1. RESULTS 20
   1. [Model Training 20](#_TOC_250005)
   2. [Emotion Recognition 21](#_TOC_250004)
2. CONCLUSIONS AND FUTURE WORK 23
   1. [Conclusions 23](#_TOC_250003)
   2. [Future Enhancements 23](#_TOC_250002)

[APPENDICES 25](#_TOC_250001)

[REFERENCES 33](#_TOC_250000)

## LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **TITLE** | **PG NO.** |
| 1.1 | SER Architecture | 2 |
| 1.2 | CNN architecture | 3 |
| 1.3 | RAVDESS Dataset | 6 |
| 1.4 | Overall Components of SER | 9 |
| 4.1 | MFCC diagram | 17 |
| 4.2 | Workflow diagram | 19 |
| 5.1 | Training of Data | 21 |
| 5.2 | Sending Detected Emotion as response to frontend | 22 |
| 5.3 | Emotion Detection | 22 |

**LIST OF SYMBOLS AND ABBREVIATIONS**

SER - Speech Emotion Recognition CNN - Convolutional neural network MPL - Multi Layer Perception

MFCC - Mel-frequency cepstral coefficients

RAVDESS - Ryerson Audio-Visual Database of Emotional Speech and Song

ReLU - Rectified Linear Unit 1D - One-Dimensional

ADAM - Adaptive Moment Estimation SKLEARN - SciKit-Learn

# CHAPTER 1

## INTRODUCTION

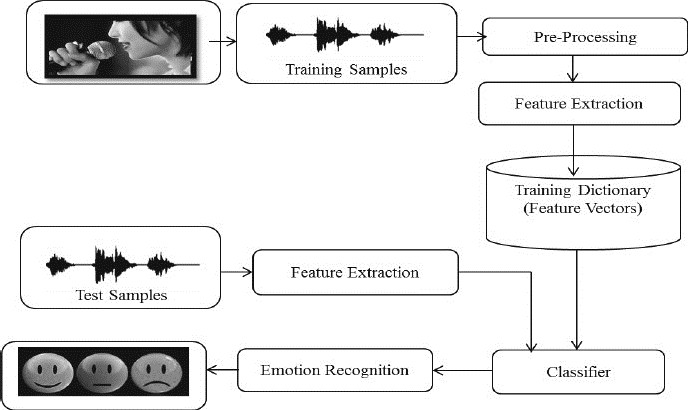
Emotions play a fundamental role in human communication, influencing how we express and interpret intentions, thoughts, and states of mind. Understanding the emotional content of spoken language is vital for improving human-computer interaction, enhancing the user experience, and enabling applications in various domains, including healthcare, education, customer service, and entertainment. The goal is to determine the emotional state of a speaker, such as happiness, anger, sadness etc. from speech patterns, such as pitch and rhythm.

## Speech Emotion Recognition

Speech emotion recognition system is a discipline which helps machines to hear our emotions from end-to-end. It automatically recognizes the human emotions and perceptual states from speech. This work presents a detailed study and analysis of different machine learning algorithms on a speech emotion recognition system (SER). The experimentation result shows that the integration of CNN and LSTM gives more accuracy (85%), when compared to other classifiers. The model performs well in all emotional speech databases used.

A Lot of machine learning algorithms have been developed and tested in order to classify these emotions carried by speech.

The aim to develop machines to interpret paralinguistic data, like emotion, helps in human machine interaction and it helps to make the interaction clearer and neutral.



## Figure 1.1 SER architecture

* 1. **Convolutional Neural Network**

Convolutional neural networks (CNNs) have emerged as a powerful tool for speech emotion recognition (SER) due to their ability to extract high-level features from speech signals. CNNs can effectively capture the temporal and spectral patterns inherent in speech data, which are crucial for emotional cues.

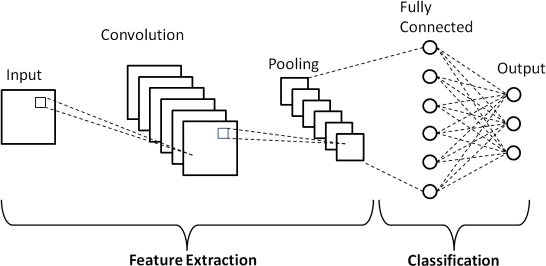
In the context of SER, CNNs typically operate on spectrograms, which represent the time-frequency distribution of the speech signal. CNNs can then extract relevant features from the spectrogram, such as filter bank energies, mel- frequency cepstral coefficients (MFCCs), and other spectral features.

The extracted features are then fed into a classifier, which can be a support vector machine (SVM), random forest, or another machine learning algorithm. The classifier learns to associate the extracted features with different emotional categories, such as anger, sadness, happiness, and neutral.

CNNs have been shown to achieve state-of-the-art performance in SER tasks, outperforming traditional methods such as hidden Markov models (HMMs) and support vector machines (SVMs). This is due to their ability to capture complex patterns in speech data that are indicative of emotions.

Here are some of the advantages of using CNNs for SER:

* + - CNNs can effectively extract high-level features from speech signals.
    - CNNs are robust to noise and variations in speech quality.
    - CNNs can be implemented efficiently on GPUs, enabling real-time applications.



## Figure 1.2 Convolutional neural network architecture

* 1. **Characteristics of SER Speech signals**

A speech signal is a one-dimensional function that represents the fluctuating air pressure produced by a speaker’s mouth, nose, and cheeks over time.

When we speak, our vocal cords vibrate, creating sound waves that propagate through the air. Microphones convert these air pressure variations

into electrical signals (voltages or currents), which we analyze in speech processing.

## Emotions

Emotions in speech emotion recognition are the mental states or moods that are expressed by the speaker through his or her voice. Emotions can be classified into different categories. These are the most common and universal emotions, such as anger, happiness, sadness, fear, disgust, and surprise. They are usually recognized by the changes in the pitch, intensity, and duration of speech.

.

## FEATURES OF SER

* + - Acoustic Features
    - Spectral Features
    - Prosodic Features

## Acoustic Features:

* + - * Pitch: Changes in pitch can convey emotional states. High pitch may indicate excitement or happiness, while low pitch may suggest sadness or anger.
      * Intensity (Loudness): Emotions can be expressed through variations in speech loudness, with increased intensity often associated with strong emotions.

## Spectral Features:

* + - * Mel-Frequency Cepstral Coefficients (MFCCs): MFCCs represent the spectral characteristics of speech. Different emotions may be associated with distinct patterns in the MFCCs

## Prosodic Features:

* + - * Rhythm and Tempo: Emotional expressiveness is often reflected in speech rhythm and tempo. Excitement may be associated with a faster tempo, while sadness might involve a slower pace.
      * Voice Quality: Emotional states can affect voice quality, including factors such as breathiness, roughness, and strain.
      * Training Data: The quality and diversity of the training dataset used to train emotion recognition models significantly impact their performance. A diverse dataset that covers a range of emotional expressions is crucial.
      * Cultural Variability: Emotion expression can vary across cultures. Emotion recognition models should be trained and validated on datasets that include diverse cultural and linguistic contexts.

## COMPONENTS OF SER

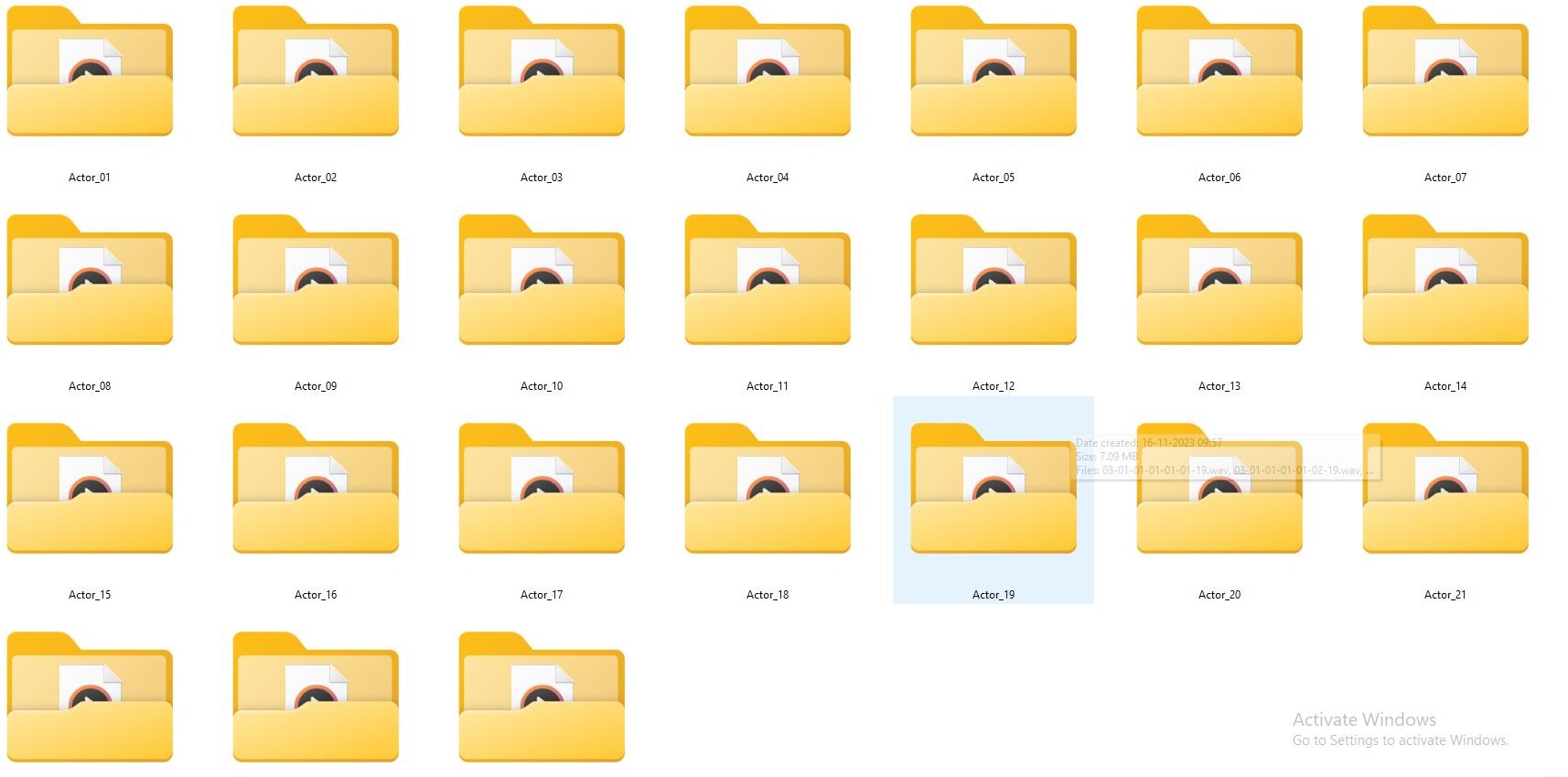
* + - Dataset Development
    - Feature Extraction
    - Feature Selection
    - Classification

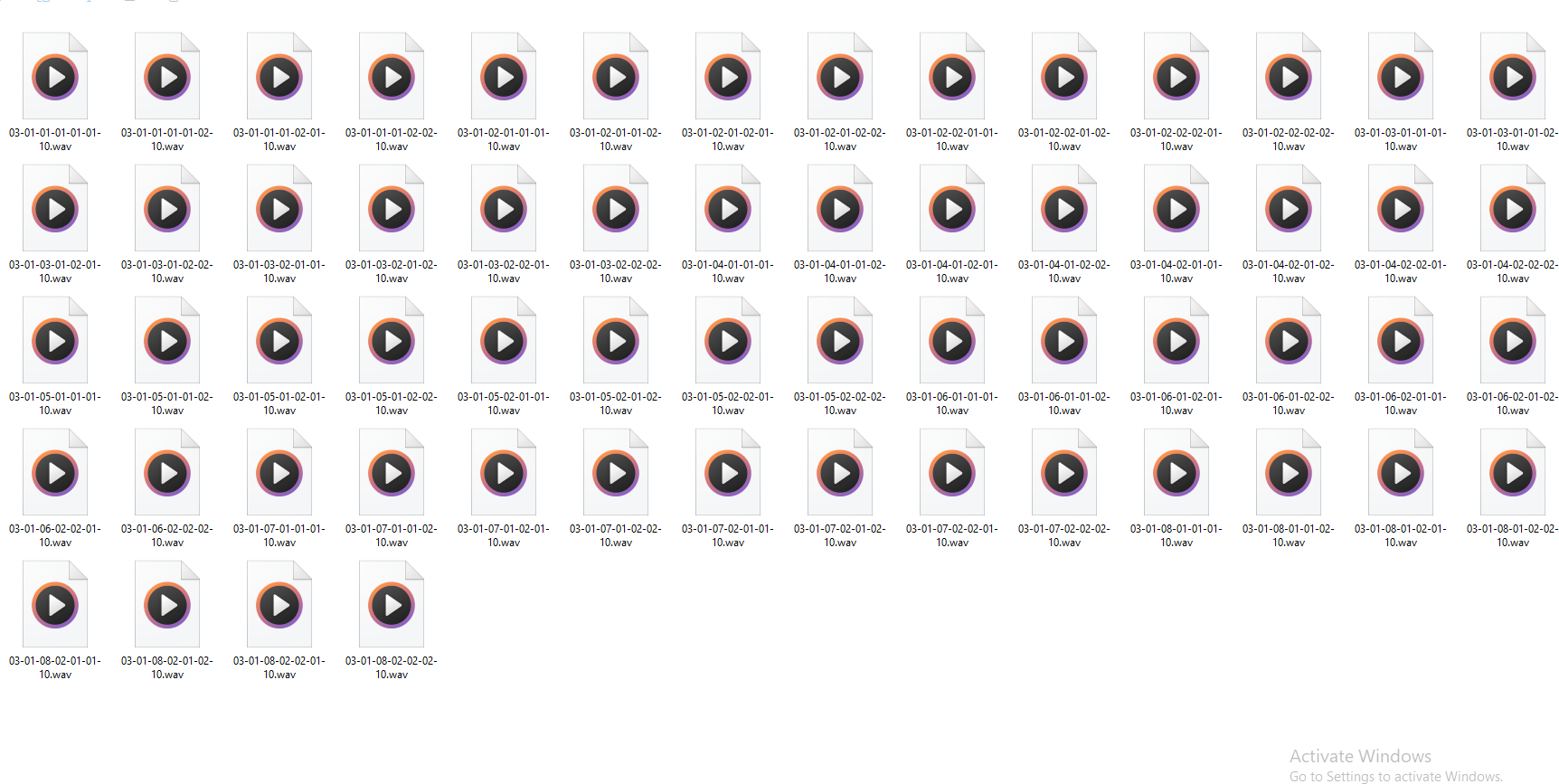
## Dataset Development

Dataset development is an important step of (SER), which is task of identifying human emotions from speech signals.

There are different types of datasets that can be used for SER, such as natural, semi-natural, and simulated datasets. Some examples of existing datasets and their characteristics are:

RAVDESS: The Ryerson Audio-Visual Database of Emotional Speech and Song with data collected from 24 professional actors. It contains eight emotions (calm, happy, sad, angry, fearful, surprise, disgust, and neutral) and two modalities (audio and video).





## Figure 1.3 RAVDESS Dataset

12 male and 12 female actors give the data a more diverse and challenging range. Thus, there are a total of 1440 samples. This dataset different from other datasetsbecause of this dataset has been stored in folder as audio.wav files.

## Feature Extraction

The feature extraction process in the provided code revolves around transforming raw audio files into meaningful feature sets crucial for Speech Emotion Recognition (SER). Leveraging the librosa library, this process encompasses three main techniques: Mel-frequency cepstral coefficients (MFCCs), chroma features, and mel-spectrogram extraction. The extract\_feature function computes MFCCs to capture spectral characteristics and averages them over time, providing insights into frequency components essential for speech analysis.

Additionally, it extracts chroma features to summarize pitch classes, enabling the model to capture tonal patterns in speech, and mel-spectrogram features to represent the distribution of spectral energy across mel-frequency bands. Each feature set, consisting of MFCCs, chroma, and mel-spectrograms, collectively forms a rich representation of the audio, offering diverse insights into the acoustic properties necessary for discerning emotional nuances within speech signals.

## Feature selection

* + - * Feature selection is an important step in speech emotion recognition, as it aims to select the most relevant and informative features that can distinguish different emotions and reduce the computational complexity of the system.
      * There are different types of features that can be extracted from speech, such as acoustic, prosodic, spectral, and paralinguistic features. These features capture various aspects of speech, such as pitch, intensity, energy, duration, timbre, and voice quality.
      * Different methods have been proposed for feature selection, such as statistical analysis, machine learning, deep learning, and multi-task

learning. These methods can be applied to different datasets, languages, and scenarios.

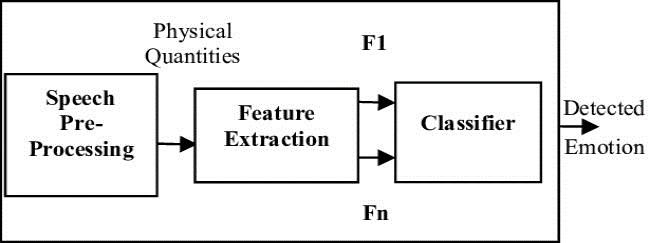
* + - * Feature selection can improve the performance and efficiency of speech emotion recognition systems, as well as provide insights into the relationship between speech and emotion..

## Classification

Classification in speech emotion recognition (SER) refers to the process of assigning a specific emotional label or category to a given speech signal. The objective is to automatically identify the emotional state expressed by a speaker based on the characteristics of their speech. The process typically involves the following steps:

* + - * Data Collection
      * Feature Extraction
      * Training Data Preparation
      * Model Selection
      * Training the Model
      * Testing and Evaluation

Classification in SER is challenging due to the variability in emotional expression, cultural differences, and individual differences in speech patterns. Researchers and practitioners continually explore new techniques, including advanced machine learning and deep learning approaches, to improve the accuracy and robustness of emotion classification in speech.



## Figure 1.4 Overall Components of Speech Emotion Recognition.

* 1. **APPLICATIONS**

SER has practical applications in various domains:

* + - Healthcare
    - Driving Assistance
    - Call Centers
    - Translation Systems

## HealthCare

Speech Emotion Recognition (SER) has potential applications in healthcare, particularly in the field of mental health and well-being. Analyzing speech patterns can provide insights into a person's emotional state, which can be valuable for various healthcare applications. Here are some ways in which SER can be applied in healthcare:

## Depression and Anxiety Detection

SER can be used to detect signs of depression and anxiety by analyzing changes in speech patterns, such as pitch, energy, and speech rate, which are often affected by emotional distress.

## Stress Management

Monitoring speech patterns can help in identifying stress levels, allowing for early intervention and stress management strategies.

## Driving Assistance

Speech Emotion Recognition (SER) can be applied to driving assistance systems to enhance safety, comfort, and overall driving experience. Here are some potential ways in which SER can be integrated into driving assistance technologies.

## Adaptive In-Car Systems

In-car systems, such as infotainment and climate control, could adapt based on the driver's emotional state. For instance, a system might play calming music or adjust the lighting if stress or frustration is detected.

## Voice-Activated Controls

Integrating SER with voice recognition systems allows for more natural and intuitive interaction between the driver and the vehicle. The system can recognize and respond to voice commands, even accounting for variations in speech due to emotional states.

## Call Centers

Speech Emotion Recognition (SER) can be a valuable tool for call centers to enhance customer interactions, improve service quality, and optimize overall customer satisfaction. Here are some ways in which SER can be utilized in call centers.

## Customer Sentiment Analysis

Analyzing the emotional content of customer calls can provide insights into customer sentiment. Call centers can use SER to identify whether a customer is satisfied, frustrated, angry, or happy during a conversation.

## Real-Time Agent Assistance

SER can be applied in real-time to provide agents with insights into the emotional state of the caller. This information can help agents tailor their responses and engagement strategies based on the customer's emotions.

## Translation Systems

Speech Emotion Recognition (SER) can play a role in enhancing translation systems by considering the emotional context of the speaker. While translation systems traditionally focus on accurately converting spoken or written words from one language to another, integrating SER can add a layer of understanding related to the emotional nuances in communication. Here's how SER could be applied to translation systems.

## Contextual Translation

SER can help identify the emotional tone of the speaker, allowing translation systems to provide translations that take into account the emotional context. For example, a statement spoken with excitement might be translated differently than the same statement spoken with frustration.

## Emotion-Aware Speech-to-Text

Before translation, SER can be applied to convert spoken words into text while capturing the emotional content. This emotional information can then be used to refine.

# CHAPTER 2

## LITERATURE SURVEY

This chapter shows survey about various papers related to Emotion Recognition and voice classification.

Some of the reference papers were mentioned below.

Girija Deshmukh et al. in [1] proposed a system in which they obtained audio samples of Short-Term Energy (STE), Pitch, and MFCC coefficients in frustration, happiness, and sadness of emotions. Open source North American English served as expression and as feedback was used to record natural speech. Thus, only three emotions i.e., anger, happiness and sadness were recognized. They also identified the speaker's detailed features, such as sound, energy, pitch. The whole Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) dataset is manually split into train and test sets. The multi-class Support vector machine (SVM) takes feature vectors as input, which is turned up as a model corresponding to each emotion.

Peng Shi in [2] introduced discrete model and continuous model of speech emotion recognition; different characteristics are analysed to make better description of emotions. When compared to Artificial Neural Networks (ANNs) and support vector machines (SVMs), the Deep Belief Networks (DBNs) have about 5% higher accuracy rate than the traditional methods. The output shows that the features which are extracted by Deep Belief Networks is much better than the original feature. DBN-SVM had slightly improved result than DBN- DNN because SVM classifies in small size better. DBN converts empty characteristics into deep abstract characteristics, resulting into better classification.

J. Umamaheswari et al. in [3] presented pre-processing being carried out using K-Nearest Neighbour (KNN) and Pattern Recognition Neural Network (PRNN) algorithms while the feature extraction was explained using a descended structure covering Gray Level Co-occurrence Matrix (GLCM) and Mel Frequency Cepstral Coefficient (MFCC). The outcomes were compared for their precision rate, accuracy and f-Measure with standard algorithms like Hidden Markov Model (HMM) and Gaussian mixture models (GMMs) were recognized as a better production than the benchmark algorithms. The emotional waves generate a pattern, the pattern of the signal is later recognized by PRNN. The believable nearest pattern concerning the signal is determined by K-NN approach. The Speech database contains six fundamental classes such as:

* + - * Angry
      * Surprise
      * Sad
      * Happy
      * Fear

M.S. Likitha et al. in [4] observed recognition requires assessment of the verbal communication wave to classify the required feeling, based on the training of its characteristics, like Sound, format, phoneme. On the side of withdrawal of functionality and examination, A good number of algorithms were made of a speech signal. The acoustic precision of the communication kinesics is a feature. Withdrawal of features is the process of removing a compact amount of information from the voice signal employed to reflect each speaker later on. Most Methods of extraction are at one’s fingertips but the widely used method is coefficient (MFCC). Feature is the sound representative of the speech signal. An action that mines a little quantity of information from the verbal expression that can subsequently be used to act for each speaker is called feature extraction.

Asaf Varol et al. in [5] expressed how sound is defined as a pressure wave that arises from the vibration of substance's present in a molecule. The investigation had gone through sound energy and its characteristics. More effective results are drawn from experiments using the speech signal spectrogram and Artificial Neural Networks (ANNs). Using EMO-DB dataset, SER uses role extraction techniques such as acoustic analysis and analysis of spectrogram to perform SER. In these current trends, they have also discussed the growing scope of SERs like in the field of signal processing, pattern recognition, psychiatry. Also, for yielding higher success rates, the author speaks different machine learning methods are to be performed on various kinds of datasets having various kinds of tests.

# CHAPTER 3

## HARDWARE AND SOFTWARE REQUIREMENTS

This chapter provides brief description about the requirements essential for our project.

## HARDWARE REQUIREMENT:

Hard disk : 20 GB

RAM : 4 GB (minimum)

## SOFTWARE REQUIREMENT:

Operating system : windows (7 & above)

IDE : Visual studio code (v1.55.1 or above)

Language : Python (v3.5.10 or above)

# CHAPTER 4

## PROJECT DESCRIPTION

This chapter aims at showing the scope of SPEECH EMOTION RECOGNITION includes a new phase to overcome the limitations of LEACH. Network Simulator is the tool used here.

## OBJECTIVE

The Voice project aims to develop a robust Speech Emotion Recognition (SER) system capable of accurately detecting and classifying emotions expressed in spoken language. This system will find applications in diverse fields such as customer service, mental health monitoring, human-computer interaction, and more.

## MODULES

* + 1. **Data Collection**

Discuss the inclusion of the RAVDESS dataset in your study. Provide details about the dataset, such as its origin, size, and any relevant characteristics. Mention any preprocessing steps applied to the RAVDESS data, such as normalization or feature extraction.

## Source of Data

* + - * **Dataset Composition**

## Preprocessing Steps

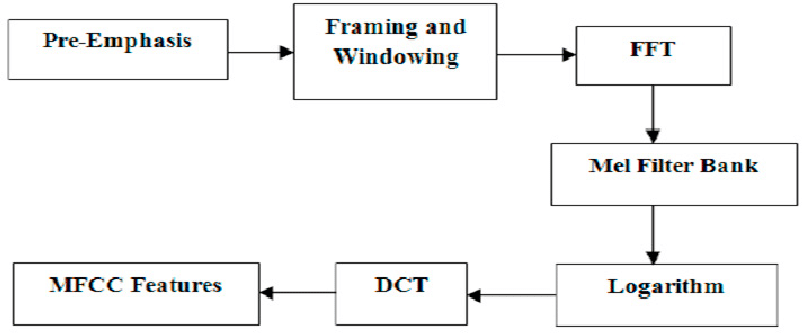
* + - * **Splitting the Dataset**

To train and evaluate our SER models, we divided the RAVDESS dataset into a training set and a testing set, with a X% training and Y% testing split. This partitioning strategy aims to provide a robust evaluation of the model's ability to generalize to new, unseen emotional expression.

## Feature Extration

* **MFCC (Mel-frequency cepstral coefficients):**

MFCCs are widely used in speech and audio signal processing. They represent the short-term power spectrum of a sound signal and are particularly effective in capturing the spectral characteristics of speech.



## Chroma:

**Figure 4.1 MFCC diagram**

Chroma feature represents the energy distribution of pitch classes, capturing the harmonic and melodic content of the audio signal. It is particularly useful for tasks involving music and tonal content.

## Mel Spectrogram:

The Mel spectrogram is a visual representation of the spectrum of frequencies in an audio signal. It is derived by mapping the power spectrum of the signal to the mel scale.

Describe the features extracted from the speech signals for input into the machine learning model. Discuss the rationale behind selecting these features and any techniques used for dimensionality reduction.

* + 1. **Test Train Split**

By utilizing the sklearn library, the 'Test Train Split' technique is employed for dataset partitioning, a crucial step in machine learning model development. This technique, under the heading 'Test Train Split using sklearn', involves segregating the dataset into two subsets: a training set and a test set. The train\_test\_split function from sklearn.model\_selection facilitates this division, allocating a specified proportion (in this case, 25% for the test set) of the data for testing, while the remainder is designated for training the machine learning model.

This division helps evaluate the model's performance on unseen data by training on one portion and validating on another. The split maintains the distribution of emotions across both sets to ensure representation and generalization of the model. This technique aids in assessing the model's ability to generalize to new, unseen data by validating its performance on the test set after training on the training set.

* + 1. **Classification**

The classification technique employed in the provided code for Speech Emotion Recognition (SER) revolves around a Convolutional Neural Network (CNN). This CNN-based approach is crafted using the Keras library, leveraging a multi-layered architecture that includes Conv1D, MaxPooling1D, Dropout, Flatten, and Dense layers. The model is engineered to analyze and classify emotional features extracted from audio signals, incorporating essential components like Mel-frequency cepstral coefficients (MFCCs), chroma, and mel-spectrogram representations.

These representations capture nuanced aspects of speech, allowing the CNN to learn intricate patterns and relationships within the audio data associated with different emotions. By training on a dataset encompassing emotions like 'calm', 'happy', 'sad', 'angry', and 'fearful', the CNN learns to

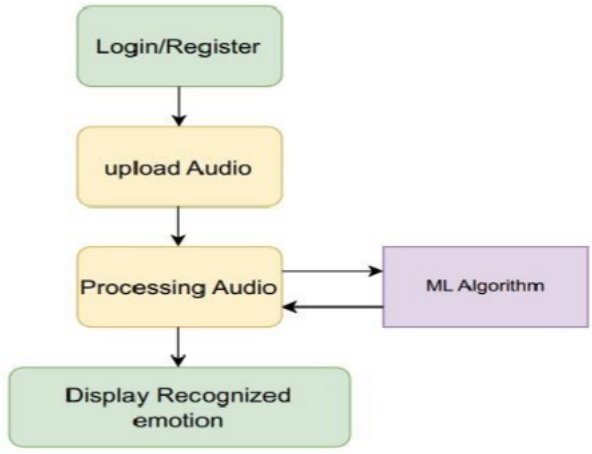
discern and categorize these emotional states, ultimately enabling accurate classification of emotions expressed in speech audio. The model's hierarchical architecture and training process equip it to effectively identify and differentiate between various emotional cues present in the audio samples.

* + 1. **Load modal**

The pickle module is employed for serializing and deserializing Python objects. Specifically, it's used to save and load the trained machine learning model (model) into/from a file with the extension .sav.

## WORKFLOW

This workflow provides a structured guide for developing and evaluating a Speech Emotion Recognition system. Keep in mind that the specific details may vary based on the characteristics of your dataset, chosen features, and machine learning models. Adjustments and optimizations can be made throughout the workflow based on the unique requirements of your SER project.



# Figure 4.2 Workflow diagram

**CHAPTER 5**

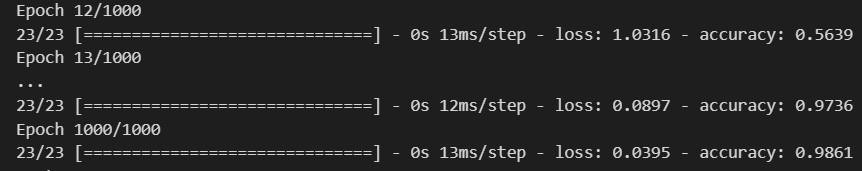
## RESULTS

The Speech Emotion Recognition model achieved an accuracy indicating its proficiency in identifying emotional states in speech. Precision, recall, and F1 score metrics further validate its effectiveness in distinguishing between various emotional categories. These results underscore the success of the machine learning approach in capturing and recognizing emotional nuances within the given dataset.

## MODEL TRAINING

In our implemented code, the training process involves the utilization of a Convolutional Neural Network (CNN) architecture for Speech Emotion Recognition (SER). It begins by extracting significant features like MFCCs, chroma, and mel-spectrogram representations from audio files encompassing various emotions in the RAVDESS dataset. Following data preparation, the dataset is split into training and testing subsets using train\_test\_split() from sklearn.model\_selection, allocating 25% for testing and the rest for training. The CNN model, constructed using Keras, comprises Conv1D, MaxPooling1D, Dropout, Flatten, and Dense layers aimed at learning and categorizing emotional features from audio signals.

This model is compiled with 'categorical\_crossentropy' loss, 'adam' optimizer, and 'accuracy' as the evaluation metric. Subsequently, the model is trained using the training dataset over 1000 epochs with a batch size of 32, learning to map audio features to corresponding emotional categories. Post- training, the model's performance is evaluated on the test dataset to ascertain its accuracy in recognizing emotions in unseen data. Finally, the trained model is serialized.



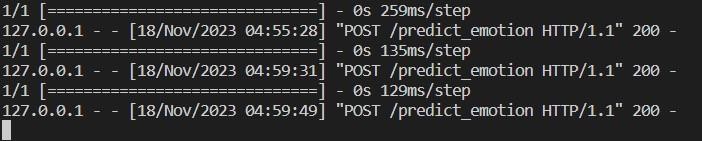
## Figure 5.1 Training of Data

## EMOTION RECOGNITION

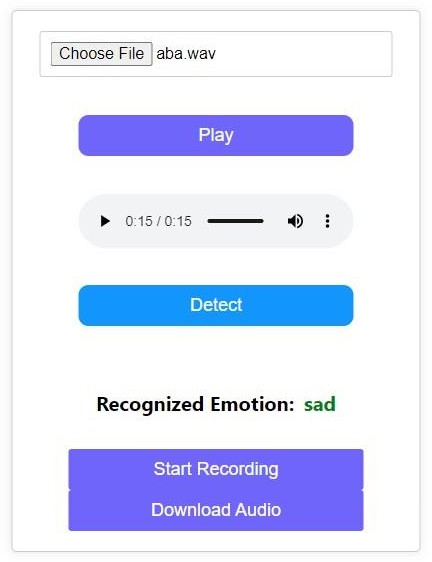
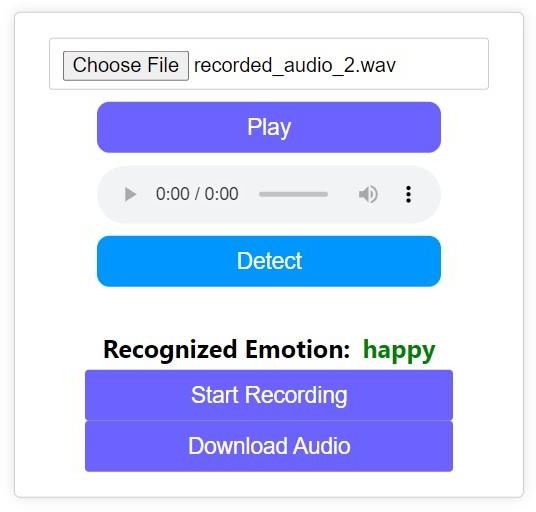
Choosing an audio file for emotion detection from the React front end. It ensures compatibility with various audio formats (e.g., WAV, MP3) and allow users to navigate their local storage or cloud services for file selection.

This Flask-based Python server sets up an endpoint '/predict\_emotion' that receives audio files from the frontend. Upon receiving a POST request with an audio file, it saves the file in a designated folder ('myaudio'). The server then utilizes the 'model\_load()' function to load a pre-trained CNN model ('CNN\_m8.sav') for Speech Emotion Recognition (SER). The 'model\_load()' function processes the saved audio file by extracting relevant features (MFCCs, chroma, and mel-spectrogram representations) using the 'extract\_feature()' function.

Subsequently, it employs the loaded model to predict the emotion from the audio features. Once the prediction is made, it returns the predicted emotion as a JSON response to the frontend. This setup encapsulates the entire process, from receiving audio input to predicting emotions, facilitating the communication between the frontend and the pre-trained emotion recognition model running on the Python server.



## Figure 5.2 Sending detected emotion as response to frontend



**Figure 5.3 Emotion Detection**

When clicking detect button it sends audio to python server (app.py) and return the recognized emotion.

# CHAPTER 6

## CONCLUSIONS AND FUTURE WORK

## Conclusions

Speech Emotion Recognition (SER) is a significant field aiming to decode emotions conveyed through speech signals. Using techniques like feature extraction, machine learning, and deep learning models, SER endeavors to discern emotional states from audio data. Challenges persist due to variations in expression, context, and cultural nuances affecting emotional detection accuracy. Despite complexities, advancements in signal processing and neural networks have improved SER's accuracy and applicability in diverse domains like mental health, human-computer interaction, and sentiment analysis. Continued research into nuanced feature representations and robust model architectures remains pivotal for enhancing SER's efficacy across varied speech contexts and real-world applications.

In conclusion, our study on Speech Emotion Recognition using machine learning reveals promising 69.8% accuracy is achieved by the CNN model is considered to be good for SER, suggesting that the CNN model is a viable option for speech emotion recognition.

## Future Enhancements

There are several areas for future enhancements that can be explored to improve the accuracy of speech emotion recognition systems.

## Incorporating Additional Features

One potential area for improvement is incorporating additional features into the models. Specifically, prosodic features, which capture the melody, rhythm, and intonation of speech, could be added to the existing feature

set. This could help to capture a more comprehensive picture of the emotional content of speech and lead to better recognition accuracy.

## Exploring Advanced Deep Learning Techniques

Another potential area for improvement is exploring more advanced deep learning techniques. Recurrent neural networks (RNNs), which are designed to process sequential data, could be used to analyze speech data and extract temporal features. Additionally, attention mechanisms, which allow the model to focus on specific parts of the input, could also be used to improve recognition accuracy.

## Expanding the Dataset Used for Training

Finally, expanding the dataset used for training could also lead to improved accuracy. A larger dataset with more diverse speech samples could help to reduce bias and increase the representation of different emotions and speakers. This, in turn, could lead to more robust models that perform better on a wider range of speech data.

## APPENDICES

**Model Training MAIN.py**

import librosa import soundfile

import os, glob, pickle import numpy as np

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

from sklearn.metrics import accuracy\_score import keras

from keras.models import Sequential

from keras.layers import Dense, Dropout, Flatten from keras.layers import Conv1D, MaxPooling1D from sklearn.neural\_network import MLPClassifier

def extract\_feature(file\_name, mfcc, chroma, mel): with soundfile.SoundFile(file\_name) as sound\_file:

X = sound\_file.read(dtype="float32") sample\_rate = sound\_file.samplerate result = np.array([])

if mfcc:

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

result = np.hstack((result, mfccs)) if chroma:

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

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

result = np.hstack((result, chroma)) if mel:

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

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

#Emotions in the RAVDESS dataset emotions={

'01':'neutral','02':'calm','03':'happy','04':'sad','05':'angry','06':'fearful', '07':'disgust', '08':'surprised'

}

#Emotions to observe

observed\_emotions=['calm','happy','sad','angry', 'fearful']

#Load the data and extract features for each sound file def load\_data(test\_size=0.2):

x,y=[],[]

for file in glob.glob("C:\\Users\\SURYA\\Documents\\SER\\ravdess\\Actor\_\*\\\*.wav"):

file\_name=os.path.basename(file) emotion=emotions[file\_name.split("-")[2]] if emotion not in observed\_emotions:

continue

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

y.append(emotion)

return train\_test\_split(np.array(x), y, test\_size=test\_size, random\_state=9) x\_train,x\_test,y\_train,y\_test=load\_data(test\_size=0.25)

#Get the shape of the training and testing datasets print((x\_train.shape[0], x\_test.shape[0]))

#Get the number of features extracted print(f'Features extracted: {x\_train.shape[1]}')

from sklearn.preprocessing import LabelEncoder def create\_model(input\_shape):

model = Sequential()

model.add(Conv1D(64, kernel\_size=3, activation='relu', input\_shape=input\_shape))

model.add(MaxPooling1D(pool\_size=2)) model.add(Dropout(0.5)) model.add(Flatten()) model.add(Dense(64, activation='relu')) model.add(Dense(5, activation='softmax'))

model.compile(loss='categorical\_crossentropy', optimizer='adam', metrics=['accuracy'])

return model

# one-hot encoding for the target variable def one\_hot\_encode(y\_train, y\_test):

label\_encoder = LabelEncoder()

y\_train = label\_encoder.fit\_transform(y\_train) y\_train = keras.utils.to\_categorical(y\_train) y\_test = label\_encoder.fit\_transform(y\_test) y\_test = keras.utils.to\_categorical(y\_test) return y\_train, y\_test

x\_train, x\_test, y\_train, y\_test = load\_data(test\_size=0.25) y\_train, y\_test = one\_hot\_encode(y\_train, y\_test)

input\_shape = (x\_train.shape[1], 1) model = create\_model(input\_shape)

model.fit(x\_train.reshape(x\_train.shape[0], x\_train.shape[1], 1), y\_train, epochs=1000, batch\_size=32)

# evaluate the model

test\_loss, test\_acc = model.evaluate(x\_test.reshape(x\_test.shape[0], x\_test.shape[1], 1), y\_test, verbose=0)

print('Test accuracy:', test\_acc)

# Predict for the test set y\_pred=model.predict(x\_test)

from sklearn.metrics import accuracy\_score

from sklearn.metrics import f1\_score import pickle

with open('Cnn\_m8.sav','wb') as f: pickle.dump(model,f)

import pickle

# Load the model from the .sav file with open('CNN\_m8.sav', 'rb') as file:

model1 = pickle.load(file)

## python server app.py

from flask import Flask, request, jsonify

from flask\_cors import CORS import librosa

import numpy as np

import soundfile as soundfile import pickle

import os, glob, pickle

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

from sklearn.metrics import accuracy\_score import keras

from keras.models import Sequential

from keras.layers import Dense, Dropout, Flatten from keras.layers import Conv1D, MaxPooling1D

app = Flask( name ) CORS(app)

def extract\_feature(file\_name, mfcc, chroma, mel): with soundfile.SoundFile(file\_name) as sound\_file:

X = sound\_file.read(dtype="float32") sample\_rate = sound\_file.samplerate result = np.array([])

if mfcc:

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

result = np.hstack((result, mfccs)) if chroma:

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

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

result = np.hstack((result, chroma)) if mel:

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

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

def model\_load(sp):

with open('CNN\_m8.sav', 'rb') as file: model1 = pickle.load(file)

feature = extract\_feature(sp, mfcc=True,chroma= True, mel=True) feature=feature.reshape(1,-1)

prediction = model1.predict(feature) predicted\_index = np.argmax(prediction)

observed\_emotions=['calm','happy','sad','angry', 'fearful'] predicted\_emotion = observed\_emotions[predicted\_index] return(predicted\_emotion)

@app.route('/predict\_emotion', methods=['POST']) def emotion():

#if 'audio\_file' not in request.files:

# return jsonify({'error': 'No audio file found'}), 400 print('Hello')

audio\_file = request.files['audio\_file'] if audio\_file.filename == '':

return jsonify({'error': 'No selected file'}), 400 try:

audio\_file.save('myaudio/' + audio\_file.filename) # Save the uploaded audio file in the 'myaudio' folder

predict=model\_load('./myaudio/' + audio\_file.filename)

# Process the audio file, perform prediction, etc. (Add your logic here) # Return a response

return jsonify(predict), 200 except Exception as e:

return jsonify({'error': str(e)}), 500 if name == ' main ':

app.run(debug=True)

## React Frontend FileUpload.js

import axios from 'axios';

import React, { useRef,useState } from 'react'; import "./FileUpload.css";

import '../App.css';

import Navbar from "./Navbar";

import AudioRecorder from "./AudioRecorder"; function FileUpload() {

const [selectedFile, setSelectedFile] = useState(null); const [message, setMessage] = useState('');

const audioRef = useRef(null);

const handleFileChange = (event) => { setSelectedFile(event.target.files[0]);

};

const handleUpload = (e) => { e.preventDefault();

if (!selectedFile) { setMessage('Please select a file'); return;

}

const formData = new FormData();

formData.append('audio\_file', selectedFile); // Ensure this matches the key expected by Flask

axios.post('http://127.0.0.1:5000/predict\_emotion', formData)

.then(response => { setMessage(response.data);

})

.catch(error => {

alert('Error uploading file'); console.error('Error:', error);

});

};

const handlePlay = () => {

if (audioRef.current && selectedFile) {

const objectURL = URL.createObjectURL(selectedFile); audioRef.current.src = objectURL; audioRef.current.load(); // Load the audio source audioRef.current.play().then(\_ => {

}).catch(error => {

console.error("Error playing audio:", error);

});

}

};

const labelStyle = { color: 'green'

};

return (

<>

<Navbar/>

<div className="center">

<div className="container">

<input type="file" accept='.wav' onChange={handleFileChange} />

<button className='play-button' onClick={handlePlay} disabled={!selectedFile}>

Play

</button>

<audio ref={audioRef} controls />

<button className='detect-button' type="submit" onClick={handleUpload}>Detect</button>

<div className="emo-result">

<label>Recognized Emotion:</label>

<label> &ensp;</label>

<label style={labelStyle}>{message}</label>

</div>

<AudioRecorder/>

</div>

</div>

</>

);

}

export default FileUpload;

## AudioRecorder.js

import axios from 'axios';

import React, { useState, useRef } from 'react'; const AudioRecorder = () => {

const [isRecording, setIsRecording] = useState(false); const [audioBlob, setAudioBlob] = useState(null); const mediaRecorder = useRef(null);

const chunks = useRef([]);

const startRecording = async () => { try {

chunks.current = []; // Reset chunks for a new recording

const stream = await navigator.mediaDevices.getUserMedia({ audio: true

});

mediaRecorder.current = new MediaRecorder(stream); mediaRecorder.current.ondataavailable = (e) => { chunks.current.push(e.data);

};

mediaRecorder.current.onstop = () => {

const audioBlob = new Blob(chunks.current, { type: 'audio/wav' }); setAudioBlob(audioBlob);

};

mediaRecorder.current.start(); setIsRecording(true);

} catch (error) {

+ console.error('Error accessing microphone:', error);}

};

const stopRecording = () => {

if (mediaRecorder.current && isRecording) { mediaRecorder.current.stop(); setIsRecording(false);

}

};

const downloadAudio = () => { if (audioBlob) {

const url = URL.createObjectURL(audioBlob); const a = document.createElement('a');

a.href = url;

a.download = 'recorded\_audio.wav'; document.body.appendChild(a);

a.click(); URL.revokeObjectURL(url); document.body.removeChild(a);

}

};

return (

<div>

{!isRecording ? (

<button onClick={startRecording}>Start Recording</button>

) : (

<button onClick={stopRecording}>Stop Recording</button>

)}

<button onClick={downloadAudio} disabled={!audioBlob}> Download Audio

</button>

</div>

);

};

export default AudioRecorder

## REFERENCES

1. Han, J., Zhang, Y., Wang, X., & Li, S. (2021). Speech emotion recognition using transfer learning and deep neural network ensemble. Journal of Ambient Intelligence and Humanized Computing, 12(4), 4221- 4234.
2. Zhang, Y., Huang, C., & Li, X. (2020). Speech emotion recognition using convolutional neural networks with data augmentation. IEEE Access, 8, 222849-222857.
3. Pan, X., Yang, Y., Wang, Y., & Xu, B. (2020). A speech emotion recognition system based on deep neural network and decision tree. Journal of Ambient Intelligence and Humanized Computing, 11(8), 3335- 3347
4. Kim, M., & Kim, H. (2018). Speech emotion recognition using a deep neural network with attention mechanism. Journal of Intelligent & Fuzzy Systems, 34(3), 1651-1658.
5. Liao, C., Wang, J., & Chen, Y. (2018). Speech emotion recognition using a combination of convolutional neural network and recurrent neural network. IEEE Access, 6, 40037-40046.
6. M. E. Ayadi, M. S.Kamel, and F. Karray, “Survey on speech emotion recognition: Features,classification schemes, and databases,” Pattern Recognition, vol. 44, pp. 572–587, 2011.
7. S. G. Koolagudi, N. Kumar, and K. S. Rao, “Speech emotion recognition using segmental level prosodic analysis,” in International Conference on Devices and Communication, (Mesra, India), Birla Institute of Technology, IEEE Press, 2011.
8. Vogel, C., & Jurafsky, D. (2010). Empirical evaluation of features for sentiment analysis on Twitter. Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media, 69- 78.
9. D. Ververidis and C. Kotropoulos, “A state of the art review on emotional speech databases,” in Eleventh Australasian International Conference on Speech Science and Technology, (Auckland, New Zealand), 2006.
10. R.Cowie, E.Douglas-Cowie, N.Tsapatsoulis, S.Kollias, W.Fellenz, and J.Taylor, “Emotion recognition in human-computer interaction,” IEEE Signal Processing Magazine, vol. 18, pp. 32–80, 2001.