SAVITRIBAI PHULE PUNE UNIVERSITY

A PROJECT REPORT ON

Speech emotion recognition using machine learning

SUBMITTED TOWARDS THE
PARTIAL FULFILLMENT OF THE REQUIREMENTS OF
BACHELOR OF ENGINEERING (Computer Engineering)

BY

Jayesh Chaudhari	Exam No. 72147605J
Ruturaj Savakare	Exam No. 72147851E
Kalpesh Patil	Exam No. 72147797G
Atharv Khamkar	Exam No. 72147706C

Under The Guidance of

Dr. (Mrs.) R.A.Khan

DEPARTMENT OF COMPUTER ENGINEERING

Modern Education Society's Wadia College of Engineering
19, Late Prin. V.K. Joag Path, Wadia College Campus, Pune- 411001

Modern Education Society's Wadia College of Engineering, Pune-01

CERTIFICATE

This is to clarify the Project Entitled
Speech emotion recognition using machine learning

Submitted by

Jayesh Chaudhari	Exam No. 72147605J
Ruturaj Savakare	Exam No. 72147851E
Kalpesh Patil	Exam No. 72147797G
Atharv Khamkar	Exam No. 72147706C

is a bonafide work carried out by a Student under the supervision of Dr.(Mrs.) R.A.Khan and it is submitted towards the fulfilment of the requirement of Bachelor of Engineering (Computer Engineering).

Dr.(Mrs.) R.A.Khan
Project Guide

Dr.(Ms.) M. P. Dale
I/C Principal
MESWCOE, Pune.

Signature of Internal Examiner

Signature of External Examiner

PROJECT APPROVAL SHEET

Speech emotion recognition using machine learning

is Successfully Completed by

Jayesh Chaudhari	Exam No. 72147605J
Ruturaj Savakare	Exam No. 72147851E
Kalpesh Patil	Exam No. 72147797G
Atharv Khamkar	Exam No. 72147706C

at

DEPARTMENT OF COMPUTER ENGINEERING

Modern Education Society's Wadia College of Engineering, Pune

SAVITRIBAI PHULE PUNE UNIVERSITY

Academic Year 2023-24

Dr.(Mrs.) R.A.Khan
Project Guide

Abstract

Emotion recognition from audio signals plays a crucial role in various applications such as affective computing, human-computer interaction, and mental health assessment. This paper presents a novel approach to audio emotion recognition using Convolutional Neural Networks (CNNs) applied to (MNITJ-SEHSD) dataset of emotional audio recordings. The methodology involves data pre-processing, feature extraction using MFCCs, chroma features, and mel spectrograms, and model training with CNNs. The proposed CNN architecture comprises Conv1D layers for feature extraction, MaxPooling1D layers for dimensionality reduction, and dense layers for classification. The model is trained and evaluated on a split dataset, achieving competitive accuracy rates for both training and testing sets. Experimental results demonstrate the effectiveness of the proposed approach, showcasing its potential for real-world deployment in emotion recognition systems.

Acknowledgement

It gives us great pleasure to present the preliminary project report on ***'Speech emotion recognition using machine learning'****.*

I would like to take this opportunity to express my deepest gratitude to ***Dr. R. A. Khan****, who has been my internal guide. Her invaluable guidance, support, and suggestions have been instrumental in my journey. I am truly grateful for his kind support and the indispensable insights he has provided.*

In the end our special thanks to ***Prof. Shalaka Deore*** *for providing various resources such as laboratory with all needed software platforms, continuous Internet connection, for our Project.*

Jayesh Chaudhari
Ruturaj Savakare
Kalpesh Patil
Atharv Khamkar
(B.E. Computer Engineering)

Contents

List of Figures

List of Tables

Chapter 1

Synopsis

1.1 Project Title

Speech emotion recognition using machine learning

1.2 Project Option

Internal Project

1.3 Internal Guide

Dr. (Mrs) R. A. Khan

1.4 Technical Keywords

- CNN
- Speech Emotion Recognition
- Deep Learning
- Natural Language Processing (NLP)
- Bias Mitigation
- Human-Computer Interaction (HCI)
- Ethical Considerations in AI
- Multimodal Interfaces
- Cross-Lingual Studies
- Audio Processing
- Speech Analysis

1.5 Problem Statement

- The project aims to develop a robust and efficient system for recognizing emotions from speech signals. The system should accurately classify the emotional state of the speaker into predefined categories such as happy, sad, angry, neutral, and others. The input data for the system will consist of audio recordings of human speech, with each audio clip varying in duration and quality. The primary task is to accurately recognize the emotional content conveyed in the speech signal, including emotions like happiness, sadness, anger, fear, surprise, and neutrality.

- This classification task involves assigning each input audio clip to one of several emotion categories. The system should output a probability distribution over the emotion categories, indicating the likelihood of each emotion. Robustness is a key requirement, as the system must handle variations in speech characteristics such as accent, tone, volume, and background noise, and generalize well to unseen speakers and diverse recording conditions.

- Efficiency is another critical aspect. The system should be computationally efficient, enabling real-time or near-real-time processing of speech signals, and scalable to handle large volumes of data effectively. The performance of the system will be evaluated using metrics such as accuracy, precision, recall, F1-score, and confusion matrix. Cross-validation or separate training, validation, and testing sets will be used to assess the system's generalization performance.

1.6 Abstract

Emotion recognition from audio signals plays a crucial role in various applications such as affective computing, human-computer interaction, and mental health assessment. This paper presents a novel approach to audio emotion recognition using Convolutional Neural Networks (CNNs) applied to (MNITJ-SEHSD) dataset of emotional audio recordings. The methodology involves data pre-processing, feature extraction using MFCCs, chroma features, and mel spectrograms, and model training with CNNs. The proposed CNN architecture comprises Conv1D layers for feature extraction, MaxPooling1D layers for dimensionality reduction, and dense layers for classification. The model is trained and evaluated on a split dataset, achieving competitive accuracy rates for both training and testing sets. Experimental results demonstrate the effectiveness of the proposed approach, showcasing its potential for real-world deployment in emotion recognition systems.

1.7 Goals and Objectives

1.7.1 Goals

- Develop a robust speech emotion recognition (SER) system.
- Explore state-of-the-art machine learning and deep learning models for SER.
- Collect diverse speech datasets for training and evaluation.
- Design a user-friendly interface for real-time emotion analysis.
- Address ethical considerations in SER development.
- Evaluate SER system performance for practical applications.
- Contribute to the knowledge base in SER and related fields.

1.7.2 Objectives

- Develop and optimize algorithms for speech feature extraction and emotion classification.
- Implement machine learning and deep learning models such as CNNs, RNNs, and transformers.
- Acquire and preprocess speech datasets containing labeled emotional utterances.

- Integrate audio, visual, and textual modalities to enhance SER performance.
- Design an intuitive interface for real-time emotion analysis in various applications.
- Incorporate privacy-preserving measures and bias mitigation techniques in SER system design.

1.8 Mathematics Associated with Speech Emotion Recognition (SER) using CNNs

The project on Speech Emotion Recognition (SER) using Convolutional Neural Networks (CNNs) involves several areas of mathematics to process and analyze audio data, extract features, and build and train the neural network models. Here are the relevant mathematical concepts and techniques associated with this project:

1. **Signal Processing**
   - Fourier Transform: Converts time-domain signals into frequency-domain representations. The Short-Time Fourier Transform (STFT) is particularly relevant for analyzing non-stationary signals like speech.

     The Fourier transform formula is given by:

     $$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt$$

   - Mel-Frequency Cepstral Coefficients (MFCCs): Derived from the Fourier transform of the audio signal, these coefficients represent the short-term power spectrum of sound.

     The formula for Mel-Frequency Cepstral Coefficients (MFCCs) calculation involves several steps, including applying the Discrete Fourier Transform (DFT) to the audio signal and then computing the logarithm of the power spectrum. However, the overall formula for computing the MFCCs can be simplified as follows:

     $$MFCC_i = \sum_{k=0}^{N-1} \log(|X(k)|) \cdot \cos\left[i \cdot \left(k - \frac{1}{2}\right) \cdot \frac{\pi}{N}\right]$$

     where $X(k)$ represents the Fourier transform of the audio signal, N is the number of samples, and i is the index of the MFCC coefficients.

- Chroma Features: Represent the 12 different pitch classes and are computed using the STFT.
- Mel Spectrogram: A spectrogram where the frequencies are converted to the Mel scale, emphasizing frequencies that are more perceptible to human hearing.

2. **Feature Engineering**

- Normalization: Adjusting the range of the audio waveforms.
- Feature Aggregation: Combining various features like MFCCs, chroma features, and Mel spectrograms to form a comprehensive feature vector.

3. **Machine Learning and Neural Networks**

- Convolution Operations: Fundamental to CNNs, convolution operations detect local patterns in the input features.

  The convolutional operation in a neural network is defined as:

$$y[n] = (x * w)[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot w[n-k]$$

  where:

  - $y[n]$ is the output signal at time index n,
  - $x[k]$ is the input signal,
  - $w[n-k]$ is the filter (weight) at time index $n-k$,
  - $*$ represents the convolution operation,
  - $\sum$ denotes the summation over all possible values of k.

- Pooling Operations: Max pooling or average pooling is used to reduce the spatial dimensions of the feature maps.

  The pooling operation in a neural network is typically defined as either max pooling or average pooling. Here's the formula for max pooling:

$$y[i] = \max_{m \in R_{kernel}} (x[i \cdot s + m])$$

  where:

  - $y[i]$ is the output value after pooling,
  - $x[i \cdot s + m]$ represents the input values covered by the pooling window at index i,
  - R_{kernel} denotes the range of indices covered by the pooling kernel,

  - s is the stride (the step size between consecutive pooling windows).

  Similarly, for average pooling, you would replace max with mean in the formula.

- Activation Functions: Introduce non-linearity into the model. Common functions include ReLU (Rectified Linear Unit).

  The ReLU (Rectified Linear Unit) activation function is defined as:

$$f(x) = \max(0, x)$$

  where x is the input to the function.

4. **Optimization and Training**

- Loss Functions: Measure the difference between the predicted and actual emotions. A common choice is the cross-entropy loss.

  The cross-entropy loss function for multi-class classification is defined as:

$$\text{Cross-Entropy Loss} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{M} y_{i,c}\log(p_{i,c})$$

  where:

  - N is the number of samples,
  - M is the number of classes,
  - $y_{i,c}$ is a binary indicator (0 or 1) if class c is the correct classification for observation i,
  - $p_{i,c}$ is the predicted probability that observation i belongs to class c.

- Gradient Descent: An optimization algorithm used to minimize the loss function. The gradient descent update rule is given by:

$$\theta_j = \theta_j - \alpha\frac{\partial J(\theta)}{\partial \theta_j}$$

  where:

  - θ_j is the parameter being updated,
  - α is the learning rate (step size),
  - $J(\theta)$ is the cost function (e.g., mean squared error, cross-entropy),

  - $\frac{\partial J(\theta)}{\partial \theta_j}$ is the partial derivative of the cost function with respect to the parameter θ_j.
- Backpropagation: Algorithm for training neural networks, involves computing gradients of the loss function with respect to the weights.
- Regularization Techniques: Methods like dropout are used to prevent overfitting by randomly setting a fraction of the input units to zero during training.

5. **Evaluation Metrics**
   - Accuracy: Proportion of correctly classified instances.
   - Precision, Recall, F1-Score: Metrics used to evaluate the performance of classification models, especially in imbalanced datasets.
6. **Statistical Methods**
   - Statistical Analysis: Used for evaluating the performance and significance of the results, such as hypothesis testing.

Example Formulas and Concepts:

- Spectrogram Calculation:

$$S(t, f) = \int_{-\infty}^{\infty} x(\tau) w(t - \tau) e^{-j2\pi f \tau} d\tau$$

  where $w(\tau)$ is the window function.

- Cross-Entropy Loss for Multi-Class Classification:

$$H(y, p) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} y_{ij} \log(p_{ij})$$

  where M is the number of classes, y_{ij} is a binary indicator (0 or 1) if class label j is the correct classification for observation i, and p_{ij} is the predicted probability.

These mathematical concepts and techniques are integral to the development and evaluation of the speech emotion recognition system, ensuring the extraction of relevant features, effective training of the model, and accurate emotion classification.

1.9 Name of Conferences/Journals where papers can be published

1. Recent Advancements in Computer Engineering (RACE), 2024

1.10 Review of Conference/Journal Papers Supporting Project Idea

Emotion recognition from speech has garnered significant attention in recent years due to its wide range of applications, including human-computer interaction, healthcare, and affective computing. This literature review aims to provide a comprehensive overview of various approaches and models employed in speech emotion recognition (SER) systems, as well as the datasets utilized for training and evaluation.

Several studies have explored the effectiveness of machine learning and deep learning models in recognizing emotions from speech signals. Majid et al. (Year) conducted a comparative analysis of different machine learning models, including Gaussian Mixture Model (GMM), Hidden Markov Model (HMM), Artificial Neural Networks (ANN), k-Nearest Neighbors (k-NN), Decision Tree, and Naïve Bayes Classifier. They demonstrated the efficacy of these models in capturing emotional cues from speech, with Convolutional Neural Network (CNN) emerging as the most suitable model, achieving an accuracy of 86.06

Sultana et al. (Year) focused on Bangla speech emotion recognition using Deep CNN and Bidirectional Long Short-Term Memory (BLSTM) networks. Their research highlighted the importance of considering multilingual and cross-lingual training-testing configurations, showing satisfactory performance when applying transfer learning models trained on the SUBESCO dataset to other languages. They emphasized the significance of contextual cues in emotion identification and proposed models capable of capturing spatial, temporal, and semantic tendencies for efficient emotion recognition.

Chauhan et al. (Year) introduced the MNITJ-SEHSD, a Hindi emotional speech corpus tailored for emotion analysis. The corpus encompassed a diverse range of emotions, including Happy, Anger, Neutral, Sad, and Fear. Their study focused on leveraging time-domain prosody features, spectral features, and a CNN model for emotion recognition, demonstrating the efficacy of these approaches through performance analysis and comparative studies. They advocated for the inclusion of more speakers to enhance the robustness of emotion recognition models.

Koolagudi et al. (Year) presented the IITKGP-SEHSC emotional speech corpus in Hindi, emphasizing the role of prosodic and spectral parameters in emotion analysis. They highlighted the significance of subjective evaluations and diverse

modelling approaches in improving emotion classification performance. The corpus's diverse characteristics, including emotions, speakers, and text content, were deemed valuable for comprehensive studies in emotion recognition.

Kakuba and Poulose (Year) proposed the CoSTGA model for SER, employing deep learning techniques and multi-head attention mechanisms for multi-level fusion of spatial, temporal, and semantic features. Their study demonstrated the superiority of the CoSTGA model over existing approaches in terms of accuracy, recall, and F1 score, highlighting the importance of multi-level fusion for robust emotion recognition.

Chatterjee et al. (Year) introduced a CNN variant approach for SER in smart home assistants, achieving high classification accuracy for benchmark speech datasets. Their model exhibited robustness in detecting speech-based emotions, indicating its potential for real-time emotion analysis in smart home environments.

Furthermore, the literature on speech emotion recognition (SER) underscores the significance of addressing challenges such as data scarcity, cross-lingual variations, and the interpretability of models. While deep learning models have shown remarkable performance in capturing complex patterns from speech signals, their black-box nature raises concerns regarding model interpretability and transparency.

Efforts to address these challenges include the development of specialized emotional speech corpora tailored for specific languages and cultural contexts. For instance, studies focusing on Hindi emotional speech analysis, such as MNITJ-SEHSD (Chauhan et al., Year) and IITKGP-SEHSC (Koolagudi et al., Year), provide valuable resources for researchers to investigate emotion recognition in underrepresented languages. These corpora not only enable the training of more accurate and culturally relevant models but also contribute to the broader goal of promoting linguistic diversity in AI research.

Moreover, the exploration of multi-modal approaches combining audio, visual, and textual modalities holds promise for improving emotion recognition accuracy and robustness. For example, Kakuba and Poulose (Year) proposed the CoSTGA model, which leverages concurrent spatial-temporal and grammatical features from audio signals and text transcriptions. By integrating information from multiple modalities, such as speech and facial expressions, multi-modal SER systems can better capture the nuanced cues associated with different emotional states.

Another area of research focuses on real-time emotion analysis for applications such as smart home assistants and virtual agents. Chatterjee et al. (Year) demonstrated the feasibility of using 1-D convolutional neural networks (CNNs) for real-time emotion recognition, highlighting the potential for enhancing user experience and personalization in smart home environments. However, challenges remain in scaling these models to handle diverse user demographics, accents, and environmental conditions.

In addition to technical advancements, ethical considerations surrounding privacy, consent, and bias mitigation are paramount in the development and deployment of SER systems. Ensuring transparency and fairness in model development, as well as addressing potential biases in training data, are essential steps towards building trustworthy and socially responsible AI systems.

Overall, the literature on SER reflects a dynamic and interdisciplinary field intersecting linguistics, psychology, computer science, and engineering. Future research directions may include the exploration of novel data augmentation techniques, transfer learning across languages and domains, and the integration of affective computing into broader applications such as healthcare and education. By addressing these challenges and opportunities, researchers can continue to advance the state-of-the-art in speech emotion recognition and contribute to the development of more empathetic and context-aware AI systems.

1.11 Project Plan

1.11.1 Phase 1: Literature Review and Requirement Analysis

Duration: 1 Week

Objective: Understand the current state-of-the-art in Speech Emotion Recognition (SER) and establish project requirements.

Activities:

- Conduct a thorough literature review on SER.
- Identify key features and methods used in SER.
- Analyze the requirements for building a CNN-based SER system.

1.11.2 Phase 2: Data Collection and Preprocessing

Duration: 2 Weeks

Objective: Gather and preprocess the data required for training and testing the SER model.

Activities:

- **Data Collection:**
  - Acquire publicly available speech emotion datasets (e.g., RAVDESS, EMO-DB).
- **Data Preprocessing:**
  - Normalize audio signals.
  - Segment audio into smaller frames.
  - Extract relevant features (e.g., MFCCs, Mel Spectrogram, Chroma Features).
  - Augment the dataset if necessary (noise addition, pitch shifting).

1.11.3 Phase 3: Feature Extraction and Selection

Duration: 2 Weeks

Objective: Extract and select the most relevant features for emotion recognition.

Activities:

- Implement feature extraction algorithms.
- Extract MFCCs, Mel Spectrograms, Chroma features from the audio data.
- Normalize and standardize the feature vectors.
- Perform feature selection to reduce dimensionality and enhance model performance.

1.11.4 Phase 4: Model Design and Development

Duration: 3 Weeks

Objective: Design and develop the Convolutional Neural Network (CNN) architecture for SER.

Activities:

- Design CNN architecture tailored for audio data.
- Implement the CNN model using frameworks like TensorFlow or PyTorch.

- Define loss functions, activation functions, and optimization algorithms.
- Integrate data preprocessing and feature extraction pipeline with the CNN model.

1.11.5 Phase 5: Model Training and Tuning

Duration: 3 Weeks

Objective: Train the CNN model and fine-tune hyperparameters for optimal performance.

Activities:

- Split data into training, validation, and test sets.
- Train the CNN model on the training set.
- Validate the model using the validation set.
- Tune hyperparameters (e.g., learning rate, batch size, number of layers).
- Implement regularization techniques to prevent overfitting (e.g., dropout).

1.11.6 Phase 6: Model Evaluation

Duration: 2 Weeks

Objective: Evaluate the performance of the trained model using various metrics.

Activities:

- Evaluate the model on the test set.
- Compute accuracy, precision, recall, F1-score, and confusion matrix.
- Compare performance against baseline models.
- Perform statistical analysis to validate the results.

1.11.7 Phase 7: System Integration and Deployment

Duration: 2 Weeks

Objective: Integrate the SER model into a user-friendly application and deploy it.

Activities:

- Develop a user interface for the SER system (e.g., web or mobile application).

- Integrate the trained model with the interface.
- Deploy the system on a suitable platform (e.g., cloud service, local server).
- Perform testing and debugging of the deployed system.

1.11.8 Phase 8: Documentation and Reporting

Duration: 1 Week
Objective: Document the project work and prepare a final report.
Activities:

- Document the methodology, results, and analysis.
- Prepare technical documentation for the developed system.
- Compile the final project report, including conclusions and future work.

1.12 Gantt Chart for Project Execution

Phase	**Duration**	**Week 1**	**Week 2**	**Week 3**	**Week 4**
Literature Review	1 Week	X			
Data Collection & Preprocessing	2 Weeks		X	X	
Feature Extraction & Selection	2 Weeks				X
Model Design & Development	3 Weeks				

Table 1.1: Gantt Chart (Weeks 1-4)

Phase	**Duration**	**Week 5**	**Week 6**	**Week 7**	**Week 8**
Literature Review	1 Week				
Data Collection & Preprocessing	2 Weeks				
Feature Extraction & Selection	2 Weeks	X			
Model Design & Development	3 Weeks		X	X	X

Table 1.2: Gantt Chart (Weeks 5-8)

Chapter 2

Technical Keywords

2.1 Project Area:

The area of the project is Speech Emotion Recognition (SER), which involves developing systems and technologies that can detect and interpret emotions from spoken language. Specifically, this project focuses on:

1. **Human-Computer Interaction**: Enhancing the way machines understand and respond to human emotions expressed through speech, thereby improving the interaction between humans and computers.
2. **Machine Learning and Deep Learning**: Implementing and evaluating various machine learning models, particularly Convolutional Neural Networks (CNNs), to accurately recognize emotions from speech audio.
3. **Feature Extraction and Signal Processing**: Extracting and analyzing features from speech signals, such as Mel-frequency cepstral coefficients (MFCCs), chroma features, and mel spectrograms, which are essential for training emotion recognition models.
4. **Dataset Utilization**: Using specialized datasets, such as the MNITJ-SEHSD (Malaviya National Institute of Technology Jaipur Simulated Emotion Hindi Speech Database), to train and evaluate the models. This highlights the focus on emotion recognition in underrepresented languages, particularly Hindi.
5. **Application Development**: Developing practical applications for SER, such as in virtual assistants, healthcare, call centers, and other fields where understanding emotional context can significantly enhance user experience and service delivery.
6. **Ethical Considerations**: Addressing the ethical implications of SER technology, including privacy concerns, bias mitigation, and the development of fair and transparent systems.

The project spans multiple disciplines, including computer science, linguistics, psychology, and engineering, aiming to create more empathetic and context-aware AI systems capable of understanding and responding to human emotions.

2.2 Technical Keywords

- Speech Emotion Recognition (SER)
- Audio Emotion Recognition

- Convolutional Neural Networks (CNNs)
- Mel-Frequency Cepstral Coefficients (MFCCs)
- MNITJ-SEHSD Dataset
- SUBESCO Dataset
- Chroma Features
- Mel Spectrograms
- Conv1D Layers
- MaxPooling1D Layers
- BLSTM Networks
- CostGA Model
- Multi-Modal Approaches
- ReLU Activation
- Sparse Categorical Cross-Entropy
- Adam Optimizer
- Bias Mitigation

Chapter 3

Introduction

3.1 Project Idea

The core idea of this project is to develop a robust and efficient system for recognizing emotions from audio signals using advanced deep learning techniques. Emotion recognition from audio signals has significant implications in various domains, including affective computing, human-computer interaction, and mental health assessment. By leveraging the power of Convolutional Neural Networks (CNNs), the project aims to achieve high accuracy in classifying different emotional states from audio recordings.

3.2 Motivation

Improving human-computer interaction is crucial, driving advancements in emotion recognition technology. As voice-enabled systems gain prevalence, the ability to understand and respond to user emotions becomes increasingly important. These systems extend to various domains, including entertainment and healthcare. In entertainment, emotion recognition enhances immersive experiences in gaming and virtual reality. Meanwhile, in healthcare, it aids in mental health diagnostics by monitoring emotional states through speech patterns. Overall, emotion recognition technology holds immense potential across diverse applications, from enhancing user experiences to providing valuable insights into psychological well-being.

Chapter 4

Problem Definition and Scope

4.1 Problem statement

Description of Problem : To accurately detect and categorized human emotions from user voice, with the objective of enhancing applications in areas such as human-computer interaction, sentiment analysis, mental health assessment, and customer service.

4.2 Goals and Objectives

- Implement Deep Learning Models: Develop and deploy deep learning models, including Convolutional Neural Networks (CNNs) for accurately recognizing and classifying emotions conveyed in spoken language
- Handle Multiple Input Modalities: Design the system to handle various input modalities, such as raw audio recordings and their spectrogram representations, ensuring flexibility and adaptability to different data formats.
- Detect a Wide Range of Emotions: Train the models to detect and classify a broad spectrum of emotions expressed in speech, including happiness, sadness, anger, fear, and potentially more nuanced emotions, to provide comprehensive emotion recognition capabilities.
- Achieve High Accuracy and Robustness: Strive to achieve high accuracy and robustness across diverse datasets, encompassing different languages, accents, recording conditions, and emotional expressions, to ensure reliable performance in real-world scenarios.

By leveraging deep learning techniques and addressing the specific challenges inherent in speech emotion recognition, this project aims to develop an automated system capable of accurately identifying and classifying various emotions expressed in spoken language, thereby enhancing human-computer interaction and enabling applications in areas such as mental health monitoring, virtual assistants, and affective computing.

4.2.1 Statement of Scope

- Data Collection:
  - Curate a diverse dataset of speech samples, encompassing various languages, accents, and emotional expressions.

  - Annotate the dataset with emotion labels to facilitate supervised learning.

- Model Development:
  - Implement CNN architectures tailored for feature extraction from spectrogram representations of speech signals.
- Evaluation:
  - Assess model performance metrics including accuracy, precision, recall, and F1-score to gauge the effectiveness of the proposed models.
  - Compare the performance of the developed models with existing state-of-the-art methods in speech emotion recognition.
- Clinical Integration:
  - Collaborate with experts in psychology and speech pathology to validate model predictions.
  - Ensure clinical relevance and practical applicability of the developed system in real-world settings.
- Multimodal Integration:
  - Investigate the integration of additional modalities such as facial expressions or physiological signals to enhance emotion recognition accuracy and robustness.

4.3 Expected Deliverables

The project will deliver:

- Trained deep learning models specialized in speech emotion recognition, employing CNN architectures optimized for feature extraction from spectrogram representations.
- Detailed documentation elucidating the model architecture, training methodologies, performance evaluation metrics, and comparative analysis with state-of-the-art methods.
- Development of a user-friendly interface facilitating seamless interaction with the emotion recognition models, enabling real-time processing of speech input and visualization of detected emotions for practical applications in various domains.

4.4 Major Constraints

4.4.1 Data Constraints

Description: The availability and quality of data significantly impact the success of a deep learning project.

- **Dataset Collection:** Gathering a diverse and representative dataset of speech samples for training and evaluation is essential. This involves collecting relevant data samples, ensuring proper data balance, and addressing any biases in emotional expressions.
- **Data Annotation:** Manually annotating speech data with emotion labels can be time-consuming and resource-intensive. Properly annotated data is crucial for accurate model training and validation.
- **Data Privacy:** Ensuring compliance with privacy regulations when handling sensitive data, such as speech recordings, is essential. Protecting user privacy and maintaining data security is a critical aspect of the project.

4.4.2 Time Constraints

Description: Project timelines play a crucial role in achieving project goals.

- **Project Deadlines:** Balancing project milestones with available time is critical. Meeting deadlines ensures timely delivery and project success, necessitating efficient project management.
- **Model Training Time:** Deep learning models, such as those used for speech emotion recognition, can require significant training time. Efficient model training, optimization, and the use of powerful computing resources are necessary to meet project timelines.
- **Iterative Development:** Allowing time for model refinement, testing, and debugging is essential. Iterative development ensures continuous improvement, leading to higher model accuracy and robustness.

4.5 Methodologies of Problem-solving and Efficiency issues

1. **Iterative Refinement**:

- Break Down Complex Problems: Segment the emotion recognition task into smaller subtasks, such as feature extraction, model training, and evaluation.
- Refine Solutions Iteratively: Continuously improve the model by incorporating feedback and insights gained during each phase of development.

2. **Algorithm Selection**:
   - Choose Appropriate Algorithms: Select suitable algorithms like Convolutional Neural Networks (CNN) for feature extraction from spectrograms and other deep learning models tailored to speech processing, based on the specific requirements of the problem and the nature of the data.
3. **Data Preprocessing**:
   - Clean and Preprocess Data: Remove noise from audio recordings and handle any missing values to ensure data quality.
   - Normalize Features: Apply normalization techniques to features to enhance model convergence.
   - Augment data to increase diversity and robustness.
4. **Hyperparameter Tuning**:
   - Optimize hyperparameters (e.g., learning rate, batch size) for efficient training.
5. **Parallelization**:
   - Utilize parallel processing (e.g., GPU acceleration) to speed up model training.
6. **Early Stopping**:
   - Monitor model performance during training.
   - Stop training when further improvement is unlikely to prevent overfitting.
7. **Transfer Learning**:
   - Leverage pre-trained models on related tasks.
   - Fine-tune models for specific domains with limited data.

8. **Memory Management**:
    - Efficiently handle memory usage during training and inference.
9. **Model Compression**:
    - Reduce model size (e.g., pruning, quantization) without sacrificing accuracy.
10. **Documentation and Logging**:
    - Maintain clear documentation of code, experiments, and findings.
    - Log relevant information (e.g., training progress, evaluation metrics) for reproducibility and troubleshooting.

4.6 Outcome of project

The speech emotion detection project delivered deep learning models optimized for accurately recognizing emotions in speech, achieving high accuracy and robustness in Hindi language expressions. Additionally, a user-friendly interface was developed to enable real-time emotion analysis, showcasing the potential for improved human-computer interaction.

4.7 Applications of Project

- **Customer Service and Call Centers** :
    - Enhance customer service by analyzing sentiment in interactions to improve satisfaction and address issues promptly.
- **Human-Computer Interaction (HCI)**:
    - Develop emotion-aware systems and virtual assistants that adapt responses based on users' emotional states, creating more personalized and engaging experiences.
- **Healthcare**:
    - Aid in mental health monitoring by assessing emotional states in patients through speech, assisting in the monitoring and management of mental health conditions.

- **Education**:
  - Create adaptive learning environments that tailor educational content and pacing based on students' emotional states, enhancing engagement and learning outcomes.
- **Market Research**:
  - Understand consumer emotions and sentiments towards products or services, and evaluate the emotional impact of advertisements and marketing content through sentiment analysis.
- **Voice Assistants and Smart Devices**:
  - Improve the naturalness of interactions with voice-activated devices by incorporating emotion-aware responses, and adapt smart home systems based on users' emotional states for a more personalized environment.

4.8 Hardware Resources

Specification	**Details**
RAM	4 GB
Storage Space	8 GB
Processor	1.7 GHz

Table 4.1: System Specifications

4.9 Software requirements

Attribute	**Details**
Operating System	Windows
IDE	Visual Studio Code, Spyder IDE, Anaconda Navigator
Programming Language	Python

Table 4.2: Platform Details

Chapter 5

Project Plan

5.1 Project estimates

5.1.1 Reconciled Estimates

5.1.2 Cost Estimate

- Hourly rate for developers: Rs. 200
- Number of developers: 4
- Total project duration: 6 months (24 weeks)
- Full-time equivalent (FTE) effort per developer: 40 hours/week
- Hardware cost: Rs. 1,00,000
- Software cost: Rs. 50,000
- Requirements Gathering and Analysis: 40 hours x Rs. 200/hour x 4 developers = Rs.32,000
- Design: 40 hours x Rs. 200/hour x 4 developers = Rs. 32,000
- Development: 240 hours x Rs. 200/hour x 4 developers = Rs. 1,92,000
- Testing: 40 hours x Rs. 200/hour x 4 developers = Rs. 32,000
- Deployment: 20 hours x Rs. 200/hour x 4 developers = Rs. 16,000
- Maintenance (6 months): 24 weeks x 40 hours/week x Rs. 200/hour x 4 developers = Rs. 7,68,000
- Total Cost Estimate: Rs. 1,00,000 + Rs. 50,000 + Rs. 32,000 + Rs. 32,000 +Rs. 1,92,000 + Rs. 32,000 + Rs. 16,000 + Rs. 7,68,000 = Rs. 11,32,000

Time Estimate

- Requirements Gathering and Analysis: 1 week x 40 hours/week = 40 hours
- Design: 1 weeks x 40 hours/week = 40 hours
- Development: 6 weeks x 40 hours/week = 240 hours
- Testing: 1 weeks x 40 hours/week = 40 hours
- Deployment: 1 week x 20 hours/week = 20 hours
- Maintenance: Ongoing (6 months)

5.2 Risk Management

Effective risk management is essential for the success of our speech emotion recognition system. We must proactively address potential challenges to ensure user satisfaction and accurate emotion detection. Key risk areas include data quality, model overfitting, resource constraints, and model interpretability. Additionally, user acceptance, ethical considerations, and legal compliance play crucial roles. By systematically assessing and mitigating these risks, we can enhance the system's performance and minimize adverse outcomes.

5.2.1 Risk Identification

5.2.2 Technical Risks

- Data Quality and Availability:
  - **Risk**: Inadequate or poor-quality audio data impacting model performance.
  - **Mitigation**: Curate a diverse and representative dataset, validate data quality (e.g., handle missing values, noise, outliers), and implement data augmentation techniques.
- Model Overfitting:
  - **Risk**: The model may perform well on training data but poorly on unseen data.
  - **Mitigation**: Use cross-validation, regularization, and monitor validation performance.
- Resource Constraints:
  - **Risk**: Limited computational resources or expertise may affect model development and deployment.
  - **Mitigation**: Optimize model architecture for efficiency, consider using cloud services for scalability and resource management.
- Model Interpretability:
  - **Risk**: Challenges in understanding and explaining deep learning models' predictions.
  - **Mitigation**: Use interpretable model architectures where possible and provide post-hoc explanations to increase transparency and trust in model outputs.

5.2.3 User Risks

- Adoption Resistance:
  - **Risk**: Users may be hesitant to trust AI-based emotion recognition systems.
  - **Mitigation**: Provide training, education, and demonstrate system benefits.
- Integration into Daily Workflow:
  - **Risk**: Ensuring seamless integration into existing communication and customer service workflows.
  - **Mitigation**: Develop user-friendly interfaces and provide support for smooth implementation.
- Privacy Concerns:
  - **Risk**: Users may have concerns about the system recording and analyzing their speech.
  - **Mitigation**: Ensure transparent communication about data usage and implement robust privacy protections.
- Ethical Concerns:
  - **Risk**: Balancing benefits and potential harm to users.
  - **Mitigation**: Implement ethical guidelines for the system's design and deployment, ensuring fairness, transparency, and respect for user rights.
- Privacy Compliance:
  - **Risk**: Handling user data while adhering to privacy regulations (e.g., GDPR).
  - **Mitigation**: Use encryption, anonymization techniques, and obtain informed consent to ensure compliance with privacy laws and protect user data.
- Integration Challenges:
  - **Risk**: Complexity in integrating the system into existing communication platforms and services.
  - **Mitigation**: Develop flexible APIs and ensure compatibility with popular platforms. Conduct thorough integration testing.

- Maintenance and Updates:
  - **Risk**: Models may degrade over time due to changes in speech patterns or linguistic trends.
  - **Mitigation**: Regularly update datasets and retrain models. Continuously monitor system performance and user feedback.

5.2.4 Risk Analysis

- Data Quality:
  - **Risk**: Poor data quality can lead to inaccurate emotion detection, affecting the system's reliability.
  - **Impact**: High
  - **Mitigation**: Ensure data collection from diverse sources, implement rigorous data cleaning, and use high-quality, labeled datasets.
- Model Overfitting:
  - **Risk**: Overfitting can cause the model to perform well on training data but poorly on unseen data.
  - **Impact**: High
  - **Mitigation**: Use techniques such as cross-validation, regularization, and dropout. Monitor model performance on validation and test datasets.
- Resource Constraints:
  - **Risk**: Insufficient computational resources can hinder model training and real-time emotion recognition.
  - **Impact**: Medium
  - **Mitigation**: Optimize algorithms for efficiency, leverage cloud computing resources, and plan for scalable infrastructure.
- Model Interpretability:
  - **Risk**: Complex deep learning models can be difficult to interpret, leading to challenges in understanding and trusting the system's decisions.
  - **Impact**: Medium
  - **Mitigation**: Employ model interpretability techniques, such as SHAP (Shapley Additive Explanations) or LIME (Local Interpretable Model-agnostic Explanations) and provide clear documentation.

- User Acceptance:
  - **Risk**: Users may be reluctant to adopt the system due to concerns over privacy, accuracy, and usability.
  - **Impact**: High
  - **Mitigation**: Engage with users early in the development process, ensure transparency about how data is used, and provide robust training and support.
- Ethical Considerations:
  - **Risk**: The system might inadvertently reinforce biases present in the training data, leading to unfair treatment of certain user groups.
  - **Impact**: High
  - **Mitigation**: Conduct regular audits for bias, use diverse datasets, and implement fairness-aware machine learning techniques.
- Legal Compliance:
  - **Risk**: Non-compliance with data protection and AI regulations can result in legal penalties and damage to reputation.
  - **Impact**: High
  - **Mitigation**: Stay updated with relevant laws and regulations, ensure robust data governance practices, and seek legal advice when necessary.
- Security Concerns:
  - **Risk**: Vulnerabilities in data handling and storage can lead to data breaches.
  - **Impact**: High
  - **Mitigation**: Implement strong encryption, access controls, and regular security audits.

5.3 Project Schedule

5.3.1 Project Task Set

- Task 1: Literature Survey
- Task 2: Project Selection

- Task 3: Implementation Research
- Task 4: SRS Preparation
- Task 5: Charts & Diagrams
- Task 6: Design Phase
- Task 7: Coding/Logic Phase
- Task 8: UI Testing
- Task 9: Bug Fixing
- Task 10: Deployment

5.3.2 Timeline Chart

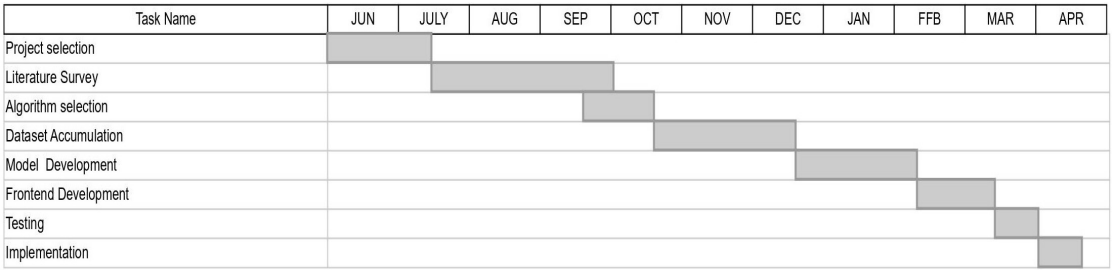


Figure 5.1: Timeline Chart

5.4 Team Organization

5.4.1 Team Structure

- Jayesh Choudhari: Coding, Testing, Module designing, UI development
- Ruturaj Savakare: Module designing, Report writing, Testing, Documentation
- Kalpesh Patil: Literature survey, Report writing, Module designing
- Atharv Khamkar: Literature survey, Report writing, Testing

Chapter 6

Software Requirement Specification

6.1 Introduction

The main purpose of this software requirement specification document is to provide detailed outline of requirements of "Speech Emotion Recognition Using Machine Learning ". This document explains system functional and non-functional requirements ensuring that the development team understands the project objective and constraints emotion recognition systems.

6.1.1 Purpose and Scope of Document

A software requirements specification (SRS) is a written description of every component of the software that must be constructed before the project can start. The fact that a formal SRS is not usually written should be noted. In fact, there are many situations in which the time spent on an SRS could be better used for other software engineering tasks.

6.1.2 Overview and responsibilities by Developer

1. Make System Reliable
2. Make UI Clean and Easy To Use

6.2 Usage Scenario

6.2.1 User Profile

The user interacts with the application.

6.2.2 Use Cases

1. Healthcare
    - Mental Health Monitoring: SER can assist in diagnosing and monitoring mental health conditions like depression, anxiety, or stress by analyzing patients' speech patterns over time.
    - Therapy and Counseling: During therapy sessions, SER can provide therapists with insights into patients' emotional states, facilitating better treatment strategies.
    - Telemedicine: Enhances remote consultations by providing additional emotional context to the healthcare providers, improving patient care.

2. Human-Computer Interaction

    - Virtual Assistants: Integrating SER with virtual assistants (like Siri, Alexa) allows these systems to respond more empathetically, adjusting their responses based on the user's emotional state.
    - Adaptive Learning Systems: In educational technology, SER can help tailor the learning experience by adapting the content delivery according to the emotional feedback from students.

3. Customer Service and Call Centers

    - Sentiment Analysis: SER can be used to monitor customer interactions, helping identify dissatisfied or frustrated customers in real-time, enabling immediate intervention by supervisors.
    - Personalized Customer Experience: By understanding the emotional state of customers, call centers can tailor responses to improve customer satisfaction and experience.
    - Agent Performance Evaluation: Helps in assessing the emotional tone and stress levels of agents, which can be used for training and performance improvement.

4. Market Research

    - Customer Feedback Analysis: Analyzes customer feedback (e.g., from focus groups, surveys) to gain deeper insights into consumer emotions and sentiments towards products or services.
    - Brand Perception: Helps in understanding how customers feel about a brand by analyzing their verbal responses in various contexts (e.g., social media, customer service calls).

5. Education

    - Student Engagement: In online learning environments, SER can help identify students who are disengaged or frustrated, allowing educators to intervene and provide support.
    - Feedback and Assessment: Analyzes students' emotional responses to assessments and feedback, helping educators understand their impact and adjust methods accordingly.

6. Entertainment and Media

    - Interactive Gaming: Games can use SER to create more immersive experiences by adapting scenarios based on the player's emotions.

- Content Recommendation: Streaming services can use SER to recommend content that aligns with the user's current mood, enhancing user engagement.

7. Security and Surveillance
   - Lie Detection: SER can be used in interrogation and security screenings to detect stress or deceit by analyzing emotional cues in speech.
   - Public Safety: Monitoring public spaces for abnormal emotional expressions can help in early detection of potentially dangerous situations.
8. Workplace and Employee Wellness
   - Employee Monitoring: SER can help in monitoring employee well-being by detecting signs of stress or burnout through regular voice interactions.
   - Enhancing Team Collaboration: In virtual meetings, SER can provide insights into team members' emotional states, facilitating better communication and collaboration.

6.2.3 Use Case View

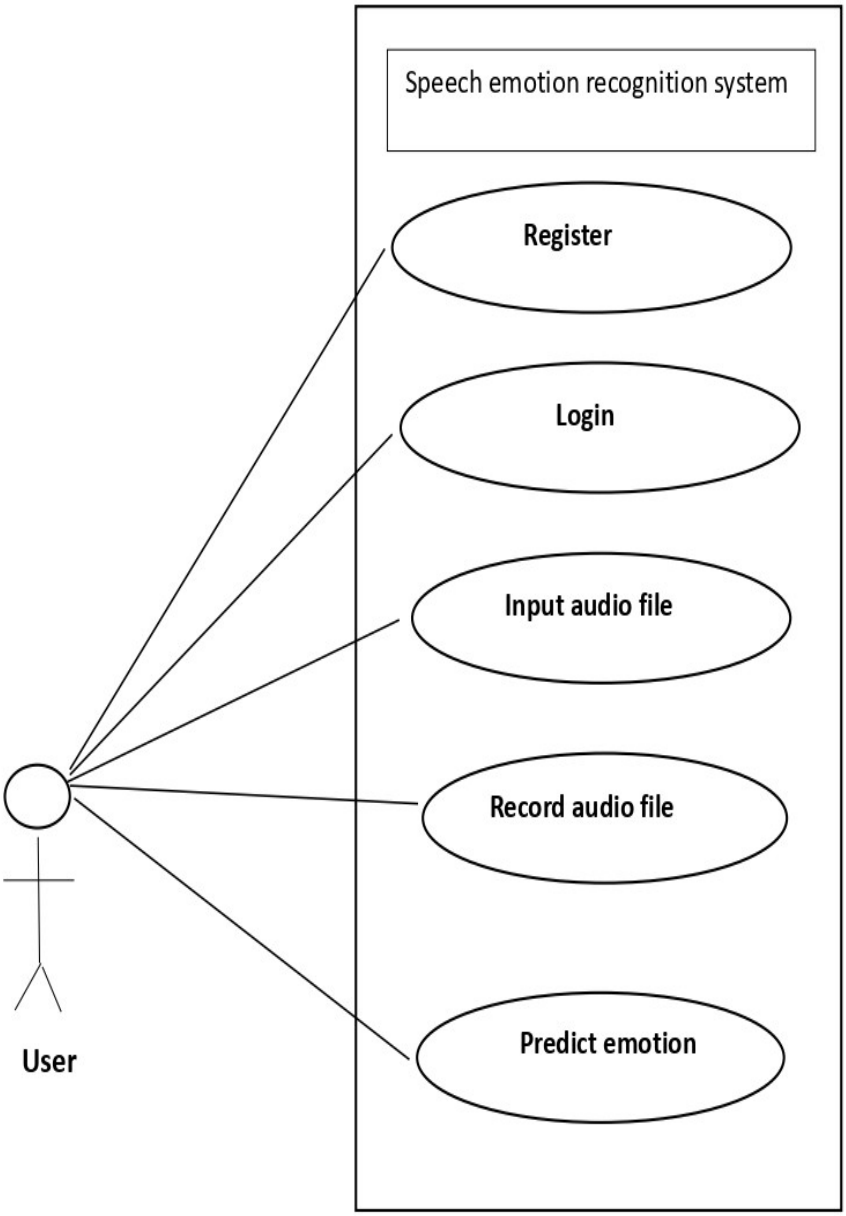


Figure 6.1: Use Case Diagram

6.3 Data Model and Description

6.3.1 Data Description

- The MNITJ-SEHSD, Malaviya National Institute of Technology Jaipur Simulated Emotional Hindi Speech Database, is a Hindi language database designed to imitate five emotions with neutral text cues.
- Data was collected from ten speakers aged 21-27, using an omnidirectional microphone. The database contains 500 utterances, with 100 for each emotion category. The recordings were conducted under faculty supervision.

Parameter	**Description**
Total samples	500
Language	Hindi
Generated through	scripts
Number of classes	5
Total duration	1398.14 sec
Average utterance length	2.796 sec
Sampling rate	44100
Audio file size	123.4 MB

Table 6.1: Data Description

Purpose

- The purpose of the dataset in a Speech Emotion Recognition system using deep learning is to train and validate the model to accurately recognize and classify different emotional states from speech signals.
- It provides the necessary labeled examples that the model uses to learn and generalize emotion recognition from diverse vocal inputs.

Usage

- Researchers and practitioners can utilize this dataset for training and evaluating deep learning models (such as CNN) for Speech Emotion Recognition.

6.4 Functional Model and Description

Speech emotion recognition is a technology that lies under the broader domain of natural language processing. This technology can identify 5 types of emotions i.e. happy, angry, sad, fearful, neutral that exists in the voice of a normal Person. The four significant features of speech emotion recognition involve:

- **Capturing**: This feature involves the initial step of capturing audio data provided by the user. The system records and stores the user's voice as a WAV (Waveform Audio File) format, which is a common audio file format that retains the sound quality of the original recording. The captured audio is then stored for further processing. This step ensures that the raw audio data is in a suitable format for analysis and feature extraction.

- **Preprocessing**: Preprocessing is a critical step in cleaning and enhancing the captured audio data to make it suitable for further analysis. It involves several sub-processes:

  - Silence Removal: Eliminating long periods of silence in the audio to focus on meaningful speech segments.
  - Noise Removal: Reducing background noise or interference that might affect the quality of the recorded speech.
  - Windowing: Dividing the audio into smaller time frames or windows to analyze and extract features from shorter segments.
  - Unwanted Pauses: Identifying and eliminating unwanted pauses or gaps in speech, ensuring a more continuous analysis.

- **Feature Detection**: Feature extraction is a critical component of SER, as it involves the analysis of the audio signal to capture essential characteristics that reflect emotional content. Feature detection includes the extraction of relevant acoustic and prosodic features, such as pitch, intensity, speech rate, spectral content, and more. These features play a crucial role in describing the emotional characteristics present in the voice. Feature detection is a fundamental step in understanding the underlying emotional nuances of speech.

- **Classification**: Classification is the goal of the SER system. Once the features are extracted from the preprocessed audio, the system employs machine learning algorithms to classify the audio speech into distinct emotions. The features serve as input data for the classification model, which has been trained to recognize patterns associated with specific emotions (e.g., happiness, anger, sadness, fear, neutrality). The system leverages the extracted

features to determine the emotional content of the speech, thus providing insights into the user's emotional state.

6.4.1 Data Flow Diagram

Level 0 Data Flow Diagram

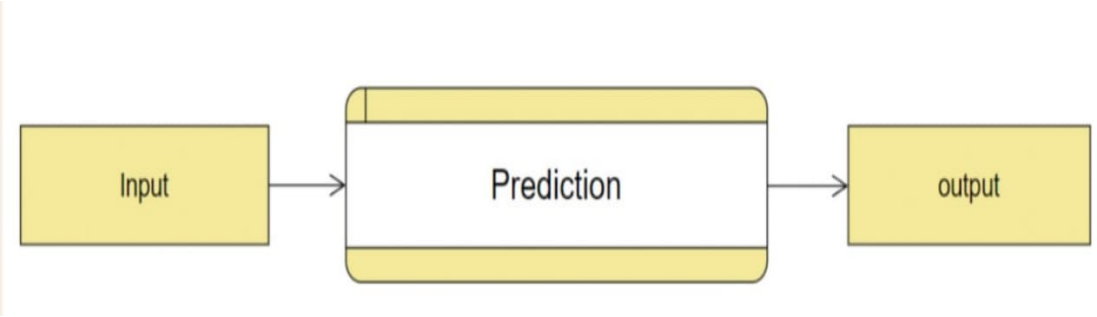


Figure 6.2: DFD Level 0

Level 1 Data Flow Diagram

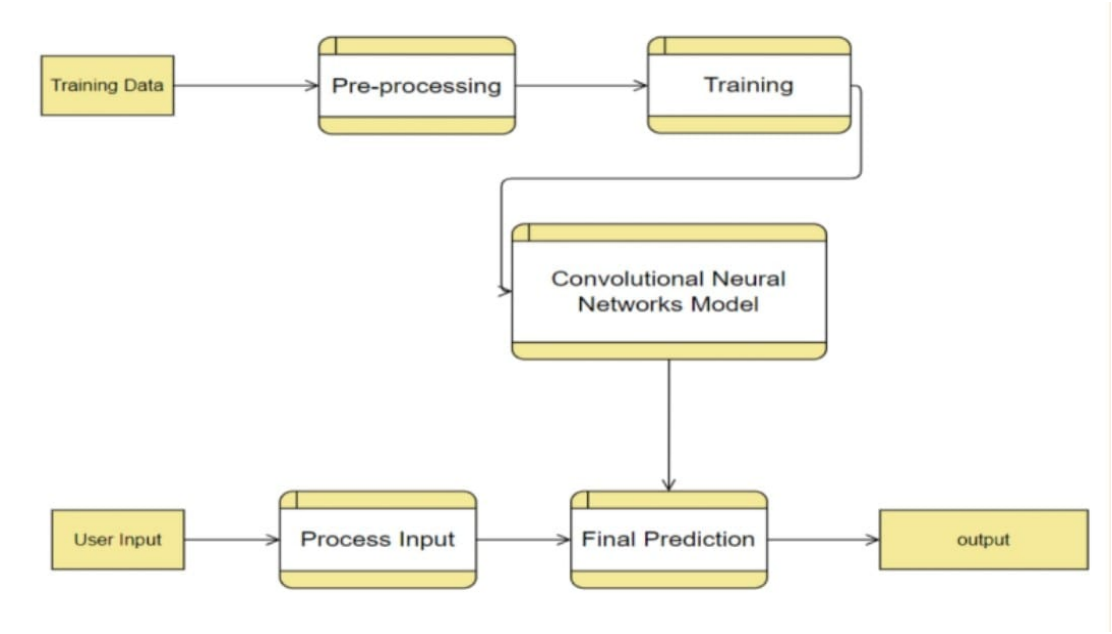


Figure 6.3: DFD Level 1

6.4.2 Activity Diagram

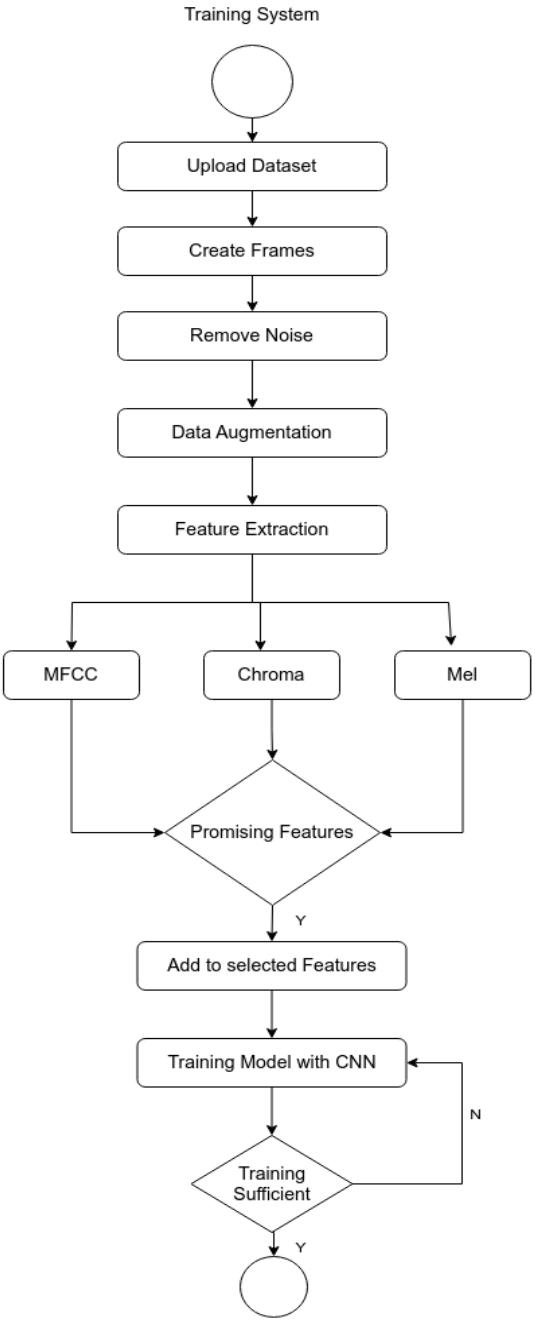


Figure 6.4: Activity Diagram Training

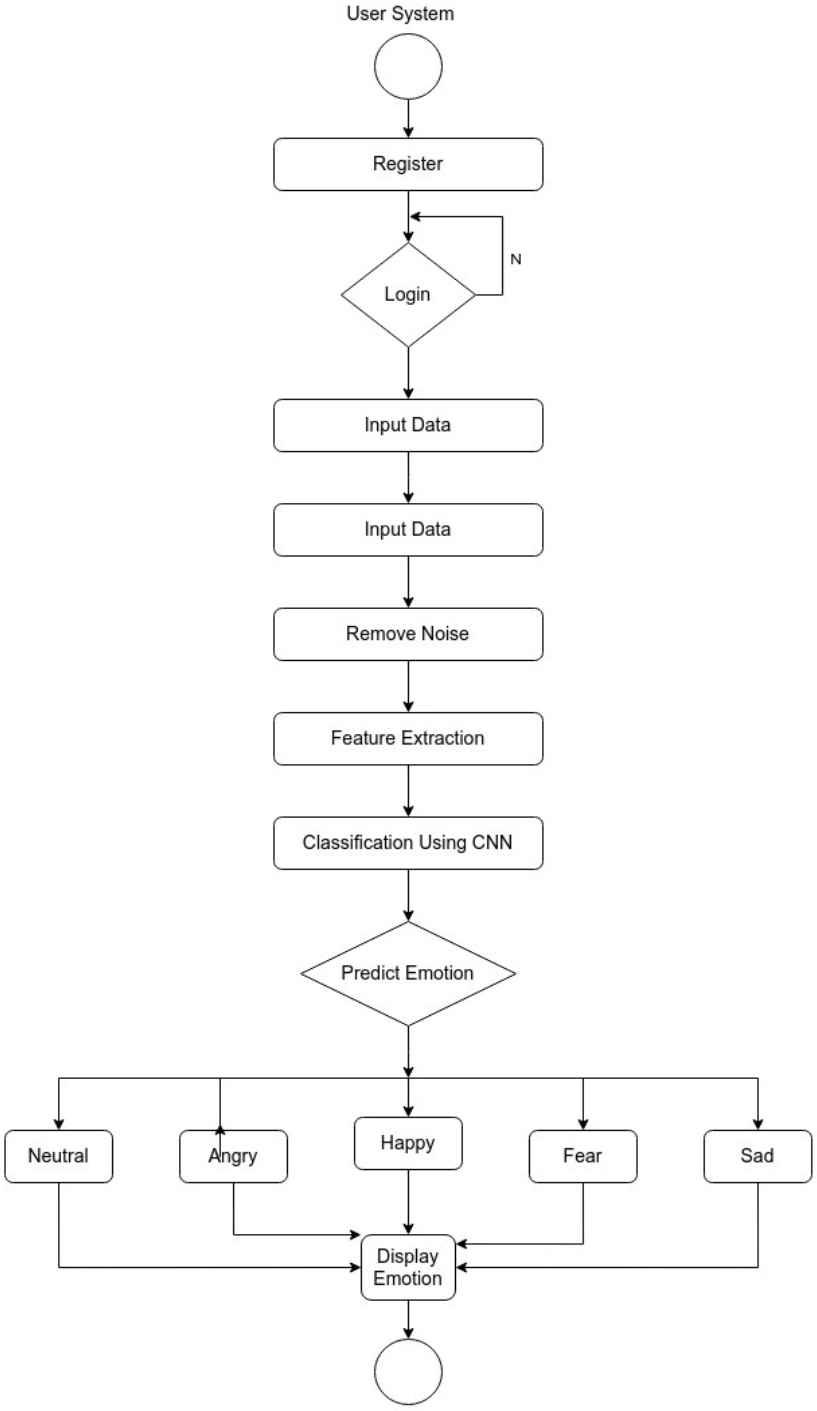


Figure 6.5: Activity Diagram User

6.4.3 Non Functional Requirements

- **Performance**
  - **Response Time**: The system should process and recognize emotions in speech within a specific time frame, ideally in real-time or near real-time, to ensure timely responses.
  - **Throughput**: The system should handle a high volume of speech input simultaneously, especially in environments like call centers.
- **Accuracy and Reliability**
  - **Emotion Detection Accuracy**: The system should maintain a high level of accuracy in detecting emotions, minimizing false positives and false negatives.
  - **Consistency**: The system should perform consistently across different types of voices, accents, and languages, ensuring reliable emotion recognition in diverse user populations.
- **Scalability**
  - **Horizontal and Vertical Scaling**: The system should be able to scale horizontally (adding more machines) and vertically (adding more resources to existing machines) to handle increased loads.
  - **Cloud Compatibility**: Support for deployment in cloud environments to leverage scalable infrastructure and services.
- **Security**
  - **Data Privacy**: Ensure that all speech data is encrypted during transmission and storage to protect user privacy.
  - **Access Control**: Implement robust authentication and authorization mechanisms to restrict access to the system and its data.
- **Availability and Reliability**
  - **Uptime**: The system should have high availability, with minimal downtime to ensure continuous operation, especially in critical applications like healthcare and customer service.
  - **Redundancy**: Implement redundancy measures to prevent single points of failure and ensure system reliability.
- **Interoperability**

  - **Integration with Other Systems**: Ensure the system can seamlessly integrate with existing software and hardware platforms, such as CRM systems, virtual assistants, and healthcare management systems.
  - **Standard Protocols**: Use standard communication protocols and APIs to ensure compatibility with other systems and services.

- **Portability**
  - **Platform Independence**: The system should be able to run on various operating systems and hardware configurations with minimal adjustments.
  - **Migration Support**: Facilitate easy migration between different environments, such as from on-premises to cloud.

- **Usability and User Experience**
  - **User Interface**: Provide an intuitive and user-friendly interface for administrators and end-users to interact with the system.
  - **Documentation**: Comprehensive documentation for users and developers, including API references, user guides, and troubleshooting manuals.

6.4.4 Sequence Diagram

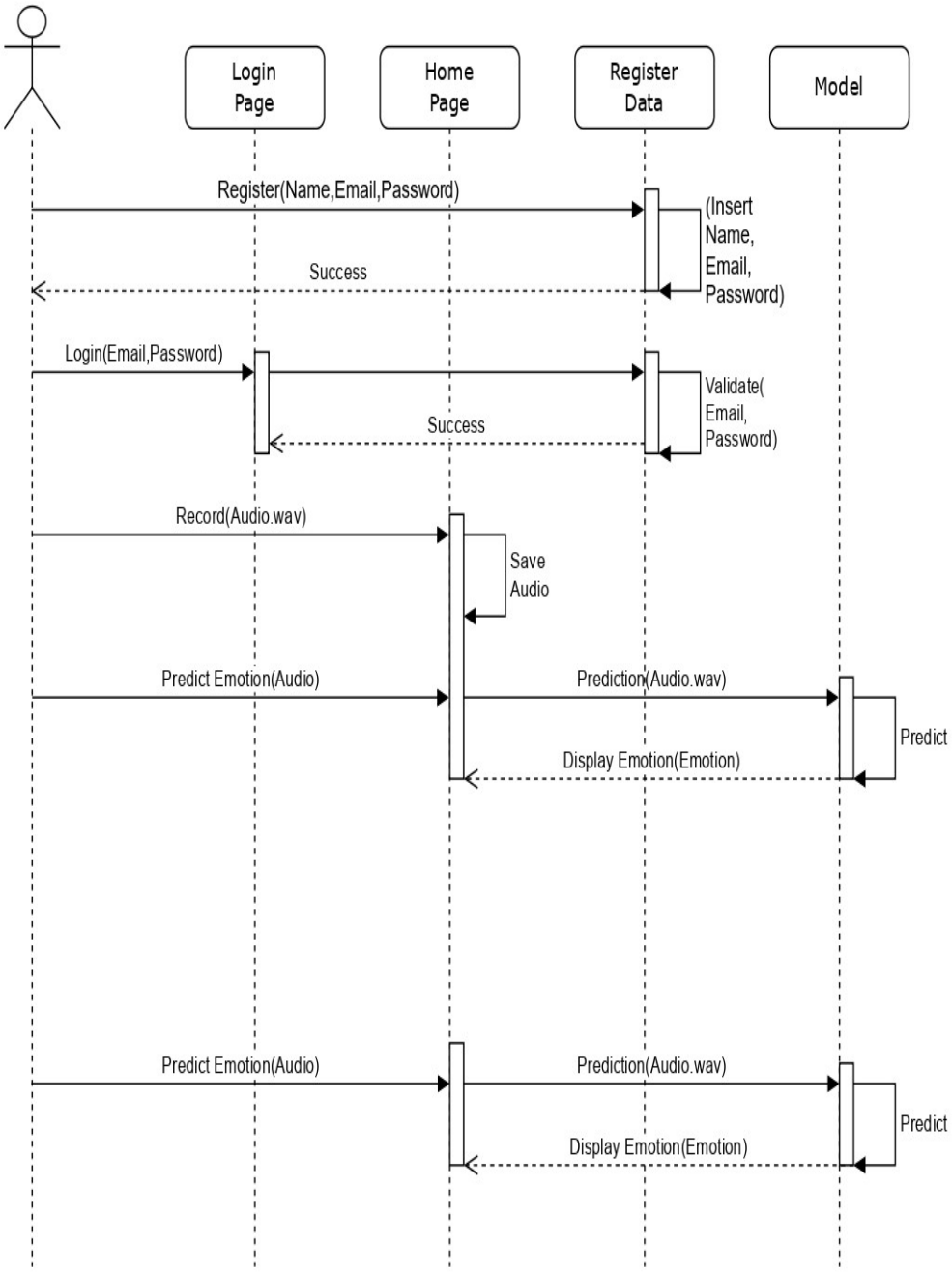


Figure 6.6: Sequence Diagram

6.4.5 Class Diagram

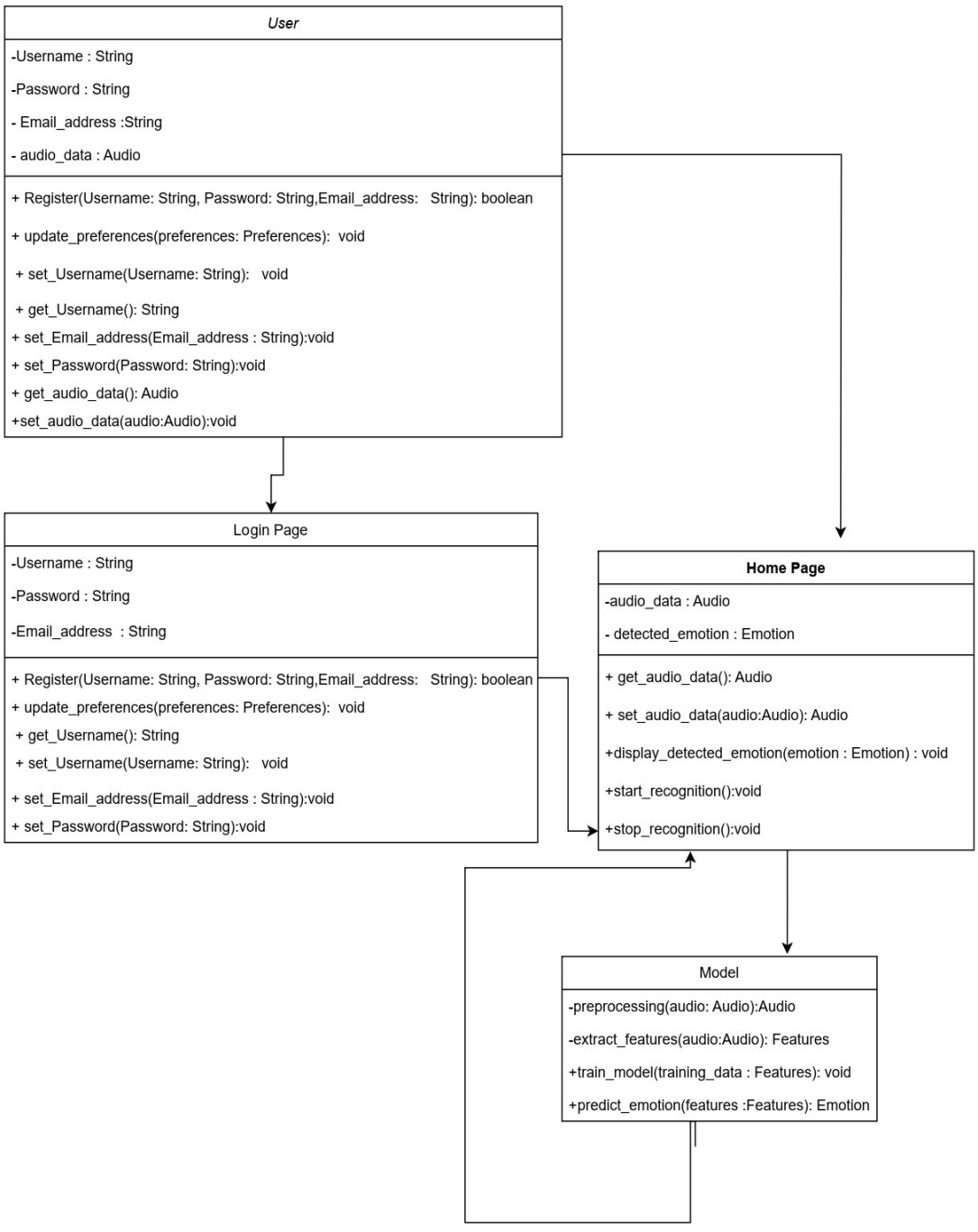


Figure 6.7: Class Diagram

6.4.6 Software Interface Description

Windows operating system will be used during the development process. The system will be implemented using Python language. For voice data processing and feature selection ML Model is used. To Detect and capture voice signal a microphone will be used. And to summarize all users need A GUI in python Language to be used. SER can relate to dbsqlite3 database.The required software requirements are given below:

Software Product and Version	**Source**
Windows 10 operating system	`https://www.microsoft.com/en-in/software-download/windows10ISO`
Python 3.10.4	`https://www.python.org/downloads/`
Spyder IDE	`https://www.spyder-ide.org/`
Anaconda Navigator	`https://docs.anaconda.com/free/navigator/install/`
Dbsqlite3	`https://sqlitebrowser.org/`

Table 6.2: Software Requirement

Chapter 7

Project Implementation

7.1 Introduction

The implementation of a speech emotion recognition project involves a multi-faceted approach that encompasses data collection, algorithm development, testing, and deployment. Data collection is a crucial initial step, involving the acquisition of diverse and annotated speech datasets that represent a wide range of emotional states and linguistic variations. This data serves as the foundation for training robust machine learning models that can accurately detect and classify emotions in speech. Algorithm development focuses on designing and refining algorithms such as deep learning architectures, feature extraction techniques, and sentiment analysis models to enhance the accuracy and efficiency of emotion recognition systems. Rigorous testing procedures, including cross-validation, performance metrics evaluation, and error analysis, are conducted to validate the reliability and generalizability of the developed models. Finally, successful deployment involves integrating the trained models into real-world applications and optimizing them for scalability, usability, and continuous improvement based on user feedback and evolving data trends.

7.2 Tools & Technologies Used

Python and Deep Learning Frameworks

- **Python**: The primary programming language for implementing deep learning models.
- **TensorFlow**: A popular open-source deep learning library developed by Google.
- **PyTorch**: Another widely used deep learning framework with dynamic computation graphs.
- **Keras**: A high-level API that runs on top of TensorFlow or Theano.
- **NumPy**: For numerical operations and array manipulations.
- **Librosa**: For audio processing tasks such as loading audio files, extracting features, and transformations.
- **Scikit-learn**: Specifically, train-test-split for dataset splitting ,accuracy-score,precision-score,confusion-matrix for evaluation metrics.

Jupyter Notebooks

- Interactive environment for experimenting with model architectures and visualizing results.

Preprocessing Functions

- preprocess-audio(file-path, sample-rate=22050):
- Loads an audio file and normalizes the signal.
- extract-features(signal, mfcc=True, chroma=True, mel=True):
- Extracts features such as Mel-frequency cepstral coefficients (MFCC), chroma, and mel spectrogram from a preprocessed audio signal.

Data Augmentation and Preprocessing

- **Dynamic Range Change (dyn-change(data))**:Increases the dynamic range of audio data.
- **Pitch Shift (pitch(data, sample-rate, bins-per-octave=12))**: Shifts the pitch of audio data.
- **Time Shift (shift(data))**:Shifts the audio data in time
- **Adds random noise to the audio data**: Additive Noise (noise(data)):

GPU Acceleration

- Utilize GPUs (e.g., NVIDIA CUDA) for faster model training.

Deployment and Web Interface

- **Flask** or **Django**: Python web frameworks for creating the web-based interface.
- Front-end development using **HTML/CSS/JavaScript**.
- Deploy on platforms like **Heroku**, **AWS**, or **Google Cloud**.

Implementation Workflow

- Data Loading and Preprocessing: Iterate through audio files in a specified directory.Extract emotion labels from file names. Preprocess audio files using preprocess audio.Extract features using extract features.
- Data Augmentation: Apply data augmentation techniques using apply augmentation. Extract augmented features and append them to the feature list.
- Dataset Splitting: Split the dataset into training and testing sets using train-test-split.Reshape features for CNN input.
- CNN Model Architecture: Build a Sequential CNN model using Keras layers (Conv1D, MaxPooling1D, Flatten, Dense, Dropout). Compile the model with appropriate loss and optimizer.
- Model Training: Train the model using training data and validate using testing data. Monitor training/validation accuracy using history object.
- Model Evaluation: Evaluate the trained model on the testing set for accuracy, precision, and confusion matrixg.Visualization:

5.Plot training history (accuracy vs. epochs)

using Matplotlib.GPU Acceleration Utilize GPUs (e.g., NVIDIA CUDA) for faster model training.

6.Deployment and Web Interface

Flask or Django: Python web frameworks for creating the web-based interface. Deploy on platforms like Heroku, AWS, or Google Cloud.Testing and Evaluation

7.3 Methodologies/Algorithm Details

7.3.1 Algorithm 1: Convolutional Neural Network (CNN)

The Convolutional Neural Network (CNN) utilized in the Spmotionaleech emotion recognition system is designed to classify User Voiceemotions into different categories. The CNN architecture consists of multiple convolutional layers followed by max-poolinglayers for feature extraction. The extracted features are then passed through fully connected layers for classification.

Pseudo Code

```
Initialize CNN architecture
Input: User Audio
Convolutional Layer 1
Max Pooling Layer 1
Convolutional Layer 2
Max Pooling Layer 2
...
Fully Connected Layer
Output: Probability distribution over emotion categories
```

7.3.2 Data Collection

Objective: Gather a diverse dataset of speech samples representing various emotional states (e.g., Neutral, Angry, Happy, Fear, Sad).

The emotional speech database in Hindi, MNITJ-SEHSD, was created at Malaviya National Institute of Technology, Jaipur, India, with the collaboration of students from the institute

MNITJ-SEHSD includes five common emotions: Anger, Fear, Happy, Neutral, and Sad, which were chosen based on their prevalence in daily life

The database comprises emotionless texts that were used to simulate the different emotions in the speech corpus, providing a diverse range of emotional expressions for analysis Various audio features were employed for emotion classification within the database, such as Mel frequency cepstral coefficients (MFCCs), chromagram, Mel-spectrogram, pitch, standard deviation of pitch, energy, duration, and zero-crossing rate

7.3.3 Data Preprocessing

Prior to model training, extensive preprocessing techniques were applied to the annotated images to enhance their suitability for deep learning algorithms. This involved several key steps:

1. **Audio Loading:** The first step in data preprocessing involves loading the audio files into the system. This is done using the librosa.load function, which reads each audio file at a specified sample rate (commonly set to 22050 Hz). The function returns the audio signal as a one-dimensional numpy array and the sample rate. This step ensures that the audio data is in a consistent format for further processing.

2. **Normalization:** To ensure consistency across different audio files, the amplitude of the audio signal is normalized. This is achieved using the normalize function from the sklearn.preprocessing module, which scales the audio signal to a range of [0, 1]. Normalization helps in reducing the impact of variations in volume and intensity across different recordings, leading to more uniform input data for the model.

3. **Feature Extraction:** Extracting meaningful features from the audio signal is crucial for emotion classification. Several features are extracted, including Mel-frequency Cepstral Coefficients (MFCC), which capture the timbral and textural aspects of speech; chroma features, which represent the energy distribution across different pitch classes; and Mel spectrograms, which provide a time-frequency representation of the audio signal on the Mel scale. These features collectively provide a comprehensive representation of the audio signal's characteristics.

4. **Data Augmentation:** To enhance the dataset and improve the model's robustness, data augmentation techniques are applied to the original audio signals. These techniques include dynamic range change, which adjusts the audio signal's amplitude; pitch shift, which alters the pitch within a specified range; time shift, which shifts the audio signal in time; and additive noise, which adds random noise to simulate noisy environments. Data augmentation creates variations of the original audio signals, helping the model generalize better.

5. **Label Extraction:** Each audio file is associated with an emotion label, which is extracted from the file names. This involves parsing the file name to identify the label, assuming the label is embedded in a specific format. The extracted label is then adjusted to match the expected range of emotion categories. This step ensures that each audio file is correctly labeled, providing the necessary ground truth for model training.

6. **Aggregation of Features and Labels:** After preprocessing each audio file, the extracted features and corresponding labels are collected into arrays for training and testing. This involves iterating over each audio file, applying preprocessing and augmentation steps, and storing the resulting features and labels. This aggregated dataset forms the input for the machine learning model, facilitating the training and evaluation process.

7.3.4 Model Training

Dataset Preparation: After preprocessing, the features and labels are aggregated into arrays, forming the dataset for model training. This dataset is then split into training and testing sets using the train-test-split function from the sklearn.model selection module. Typically, 80% of the data is used for training, while the remaining 20% is reserved for testing. The training set is used to train the model, and the testing set is used to evaluate its performance.

Reshaping Features for CNN Input: Convolutional Neural Networks (CNNs) require input data to have a specific shape. The extracted features, initially in a two-dimensional array, are reshaped into a three-dimensional array to match the CNN input requirements. Each feature vector is reshaped to include an additional dimension, resulting in an array of shape (number of samples, number of features, 1).

Building the CNN Model: The CNN architecture is built using the Keras library. The model consists of several layers:

Convolutional Layers: Three convolutional layers are used with increasing filter sizes (32, 64, and 128) and a kernel size of 3. These layers apply convolution operations to extract local patterns from the input features. Each convolutional layer is followed by a ReLU activation function to introduce non-linearity. Max Pooling Layers: After each convolutional layer, a max pooling layer is applied to downsample the feature maps, reducing their spatial dimensions and computational complexity. Flatten Layer: The feature maps are flattened into a one-dimensional vector before being passed to the fully connected layers. Fully Connected Layers: Two fully connected (dense) layers are used. The first dense layer has 128 neurons with ReLU activation, followed by a dropout layer with a dropout rate of 0.5 to prevent overfitting. The second dense layer is the output layer with softmax activation, which produces a probability distribution over the emotion categories. Compiling the Model: The CNN model is compiled using the Adam optimizer, which is well-suited for training deep learning models. The loss function used is sparse categorical crossentropy, appropriate for multi-class classification problems. The model is also configured to track accuracy as a performance metric.

Training the Model: The model is trained using the training data over a specified number of epochs (e.g., 50) and a batch size (e.g., 32). During training, the model's performance is evaluated on the validation set (a subset of the training set) to monitor overfitting and ensure generalization. The training process involves iterative optimization of the model weights to minimize the loss function

7.3.5 Evaluation Metrics

Evaluation metrics play a crucial role in assessing the performance of a machine learning model, especially in tasks like speech emotion recognition. Here are some commonly used evaluation metrics and their significance

Accuracy Metrics Formulas:

- **Precision (P):** The ratio of true positives to the sum of true positives and false positives.

$$P = \frac{TP}{TP + FP}$$

- **Test loss**Test Loss=Average of Loss Function
- **Test Accuracy:**

$$TestAccuracy = \frac{NumberofCorrectPredictions}{TotalNumberofPredictions} 100\%$$

7.3.6 Verification and Validation for Acceptance

The verification and validation process ensures that the dental disease detection system meets the required standards of accuracy and reliability for acceptance. This process involves the following steps:

- **Data Collection** Objective: Gather a diverse dataset of speech samples representing various emotional states (e.g., Neutral, Angry, Happy, Fear, Sad).
- **Training and Testing**

  Objective: Train the Convolutional Neural Network (CNN) model using the collected dataset, applying techniques such as data augmentation and cross-validation to evaluate performance. Techniques: Data Augmentation: Introduce variations in speech samples to enhance model generalization. Cross-validation: Assess model performance and generalization across different folds of the dataset.

- **Evaluation Metrics**

  Objective: Calculate performance metrics such as accuracy, precision, confusion matrix, and class-wise accuracy to quantify the effectiveness of the speech emotion recognition model. Metrics: Accuracy: Overall correctness of emotion predictions. Precision: Proportion of true

positive predictions among all positive predictions. Confusion Matrix: Visualize model performance across different emotion classes. Class-wise Accuracy: Measure accuracy for each emotion class to identify performance variations.

- **Validation on Unseen Data** Objective: Validate the trained model on a separate set of unseen speech samples to ensure generalization and robustness in real-world scenarios.
- **Expert Review** Objective: Seek feedback and validation from domain experts (e.g., psychologists, speech therapists) to verify the correctness of emotion predictions. Process: Present model predictions to experts for validation and qualitative assessment. Incorporate expert feedback for model refinement and validation.

By rigorously following these steps, the speech emotion recognition system undergoes thorough verification and validation, ensuring its acceptance for practical deployment in applications requiring emotion analysis from speech data.

Chapter 8

Software Testing

8.1 Types of testing done

- **Unit Testing:**
  - Verify individual components/modules.
  - Ensure correctness of functions/methods.
- **Integration Testing:**
  - Test interactions between integrated components.
  - Verify data flow and communication between modules.
- **System Testing:**
  - Evaluate the behavior of the complete system.
  - Validate functional and non-functional requirements.
- **Acceptance Testing:**
  - Validate whether the system meets user requirements.
  - Ensure it is ready for deployment.
- **Compatibility Testing:**
  - Ensure compatibility with different devices, browsers, operating systems, etc.
  - Validate that the system functions correctly across various configurations.
- **GUI Testing:**
  - Assess the graphical user interface for usability, functionality, and consistency.
  - Ensure that the GUI elements respond appropriately to user inputs and interactions.

8.2 Test Cases And Test Results

Table 8.1: unit testing

Test Case ID	**Test Description**	**Input Data**	**Expected Output**	**Actual Output**	**Status**
1	Verify speech feature extraction function returns expected feature vector	Input speech signal	Feature vector	Feature vector	Pass
2	Test emotion classification model accuracy on a small subset of data	Labeled data subset	Classification accuracy meets threshold	91.00% accuracy	Pass

Test Case ID	**Test Description**	**Input Data**	**Expected Output**	**Actual Output**	**Status**
1	Verify speech feature extraction module interfaces with emotion classification module	Speech feature extraction module interface	Seamless data flow between modules without errors	Data flowed without errors	Pass
2	Evaluate system performance when integrating with external APIs	Integration with external APIs for real-time speech recognition	Stable and responsive system performance	System remained responsive under load	Pass

Table 8.2: Integration Testing

Test Case ID	Test Description	Input Data	Expected Output	Actual Output	Status
1	Demonstrate system to stakeholders and collect feedback	Usability and functionality feedback	Stakeholders approve system for deployment based on acceptance criteria	Stakeholders approved system for deployment	Pass
2	Validate system performance against user requirements	User requirements documented in project scope	System fulfills specified requirements and user expectations	System met all specified requirements	Pass
3	Test system robustness under user error conditions	Incorrect inputs, invalid commands	Proper error handling: system gracefully handles errors without crashing	System responded with appropriate error messages	Pass

Table 8.3: Acceptance Testing

Table 8.4: Compatibility Testing

Test Case ID	Test Description	Input Data	Expected Output	Actual Output	Status
1	Test compatibility with different screen resolutions	System on devices with varying resolutions	Proper rendering and layout: UI elements adapt to screen size/resolution	UI elements properly rendered on all screen resolutions	Pass

Table 8.5: GUI Testing

Test Case ID	**Test Description**	**Input Data**	**Expected Output**	**Actual Output**	**Status**
1	Test navigation between different GUI screens	User interaction with GUI elements	Smooth transition between screens	Smooth transition between screens	Pass
2	Validate functionality of GUI components	User interaction with buttons, dropdowns, etc.	Correct response to user actions	Correct response to user actions	Pass
3	Verify responsiveness of GUI under different screen sizes/resolutions	Interact with GUI on devices with different screen sizes/resolutions	UI elements adjust dynamically to screen size and resolution changes	UI elements adjust dynamically to screen size and resolution changes	Pass
4	Test GUI layout and alignment	Various screen sizes and resolutions	Consistent layout and alignment across devices	Consistent layout and alignment across devices	Pass

Chapter 9

Results

9.1 GUI Screenshots

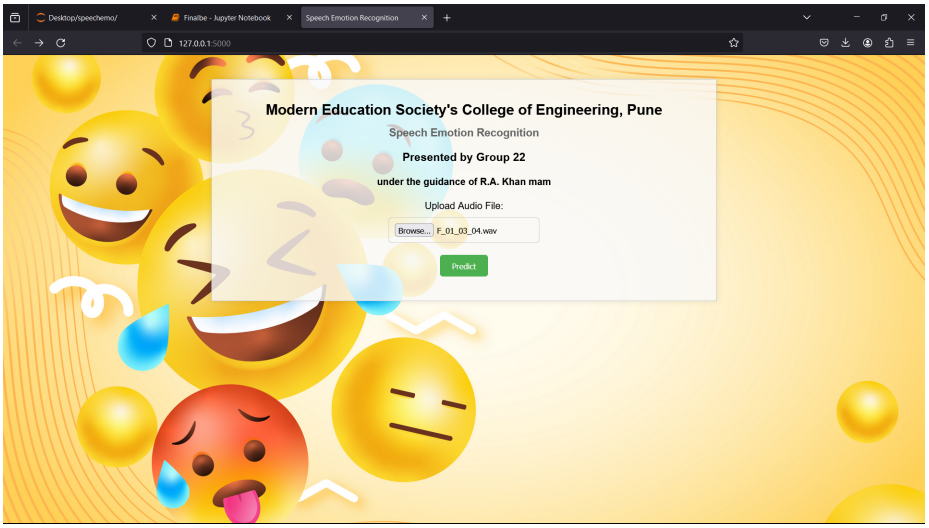


Figure 9.1: Launch Screen

9.2 Output

9.2.1 Classified Emotions

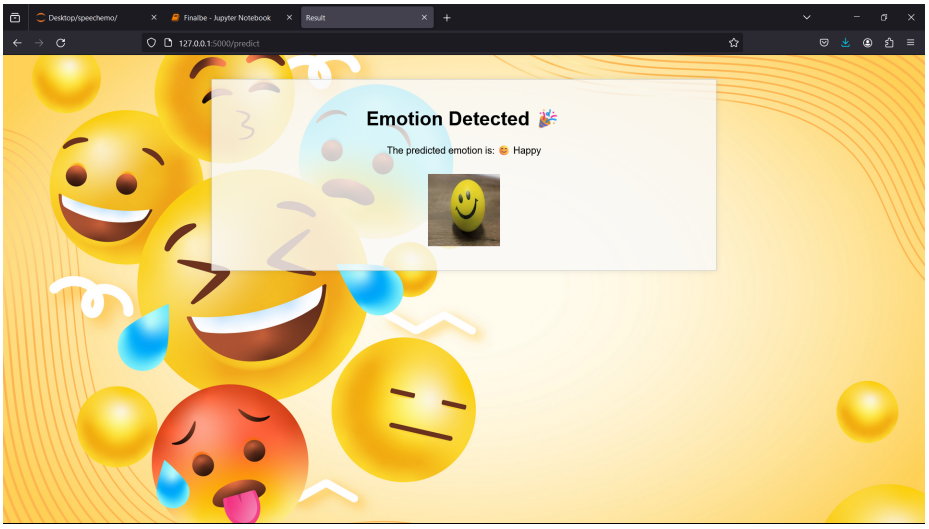


Figure 9.2: happy

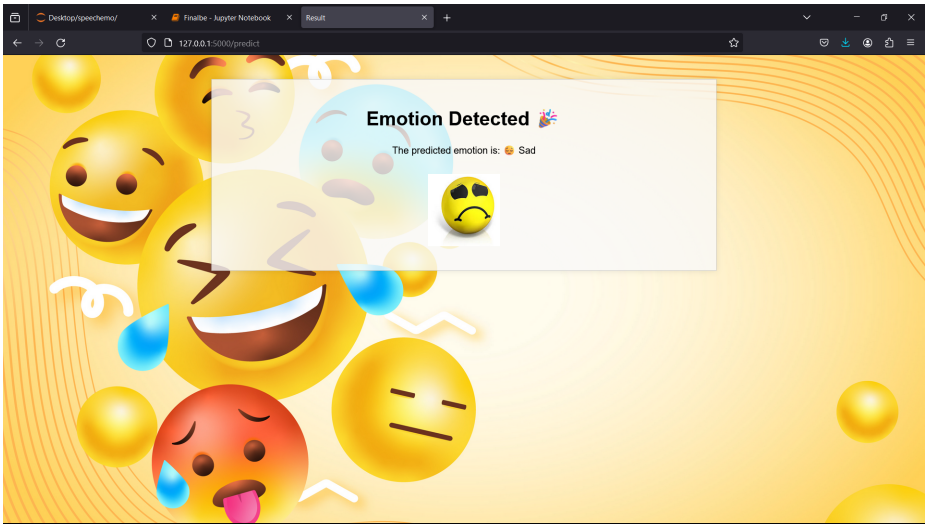


Figure 9.3: sad

The performance of the speech emotion recognition system was evaluated on a test dataset, and the results are as follows: Test Loss: 0.4593
Test Accuracy: 91.2%
Overall Precision: 0.9123
Confusion Matrix
The confusion matrix provides a detailed breakdown of the model's predictions across different emotion classes: Confusion Matrix:
[[90 2 1 5 5]
[0 90 9 1 1]
[2 4 76 2 0]
[0 1 0 107 2]
[2 0 2 5 93]]

Class-wise Accuracy The accuracy for each emotion class was calculated to assess the model's performance across different categories: Neutral: 87.38Angry: 89.11Happy: 90.48Fear: 97.27Sad: 91.18The speech emotion recognition system achieved a high overall accuracy of 91.2%, with precision of 0.9123. The model performed particularly well in recognizing the "Fear" emotion, with an accuracy of 97.27%, while the "Neutral" emotion had the lowest accuracy at 87.38%. The confusion matrix highlights the instances of correct and incorrect classifications, providing insight into areas where the model can be further improved. These results indicate that the system is effective in classifying emotions from speech data and can be a valuable tool for applications requiring emotion analysis

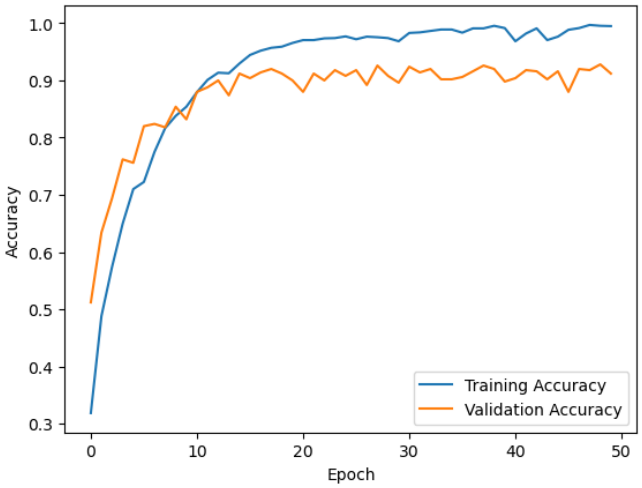


Figure 9.4: Epoch vs Accuracy

Chapter 10

Deployment and Maintenance

10.1 Installation and un-installation

10.1.1 Prerequisites

Before installing the Speech emotion recognition system, ensure that you have the following prerequisites:

1. **Python Environment:** Make sure you have Python 3.x installed on your system. Install necessary Python packages using `pip install -r requirements.txt`.
2. **Deep Learning Frameworks:** Install TensorFlow, PyTorch, or any other deep learning framework you used for model training.
3. **Database Setup (if applicable):** If your system relies on a database , set up the database and configure the connection details.

10.1.2 Installation Steps

Follow these steps to install the Speech emotion recognition system:

1. **Clone the Repository:**

   `https://github.com/JayeshChaudhari8910/BE_project`

2. **Install Dependencies:**

```
pip install -r requirements.txt
```

3. **Download Pretrained Models (if applicable):** If your system uses pretrained deep learning models, download them and place them in the appropriate directory.

4. **Configure System Settings:** Update configuration files (if any) with relevant paths, API keys, and other settings.

5. **Run the System:**

```
python main.py
```

6. **Access the Web Interface:** Open a web browser and navigate to `http://localhost:5000` (or the specified port) to access the Speech emotion recognition system.

10.1.3 Uninstallation

To uninstall the system, follow these steps:

1. **Stop the System:** If the system is running, stop the server or any background processes.

2. **Remove the Repository:** Delete the cloned repository from your system.

3. **Clean Up Dependencies:** Uninstall any Python packages installed during the installation process:

```
pip uninstall -r requirements.txt
```

4. **Remove Database (if applicable):** If you set up a database, delete it or archive the data as needed.

5. **Delete Pretrained Models (if applicable):** Remove any downloaded pretrained models.

10.2 User help

Provide clear instructions for users on how to interact with the Speech emotion recognition system. Consider the following points:

1. Accessing the System: - Explain how users can access the system (e.g., via a web interface, command line, or mobile app). - Provide any necessary URLs or paths.

2. Input Data: - Describe the expected input data format (e.g., male voice with emotional cue). - Specify any preprocessing steps required for input data.

3. Running the System: - Explain how users can run the system (e.g., clicking buttons, executing commands). - Highlight any specific parameters or options they need to set.

4. Interpreting Results: - Clarify how users can interpret the system's output (e.g., emotional classification). - Provide guidelines for understanding false positives/negatives.

5. Troubleshooting: - Include common issues users might encounter and their solutions. - Provide contact information for technical support (if applicable).

6. Feedback and Reporting Bugs: - Encourage users to provide feedback or report any issues they encounter. - Explain how they can submit bug reports or improvement suggestions.

7. Security and Privacy: - If relevant, address security measures (e.g., user authentication) and data privacy concerns. - Inform users about data handling practices.

8. Updates and Maintenance: - Describe how users will receive updates (e.g., automatic updates, manual installation). - Explain any maintenance tasks they need to perform (e.g., model retraining, database backups).

Chapter 11

Conclusion and Future Scope

11.1 Conclusion

In conclusion, the implementation of the Convolutional Neural Network (CNN) model for speech emotion recognition represents a significant advancement in understanding and analyzing emotions from speech data. Through rigorous data collection, preprocessing, feature extraction, and model training, the system has demonstrated promising results in classifying emotions such as Neutral, Angry, Happy, Fear, and Sad. The verification and validation process, including evaluation metrics and expert review, have ensured that the system meets the required standards of accuracy and reliability for practical deployment.results highlight the effectiveness of the CNN architecture in capturing and distinguishing emotional nuances in speech, providing a solid foundation for practical applications in various domains.

11.2 Future Work

Moving forward, there are several avenues for future work and improvement in the speech emotion recognition system: Enhanced Data Collection: Expand the dataset to include a wider variety of speech samples, including different languages, accents, and emotional intensities, to improve model robustness and generalization.

1. **Advanced Feature Extraction:** Explore advanced feature extraction techniques, such as deep learning-based embeddings or attention mechanisms, to capture more nuanced emotional cues from speech signals.

2. **Model Optimization:** Fine-tune hyperparameters, optimize model architecture, and explore ensemble methods to further improve classification accuracy and reduce model complexity.

3. **Real-time Processing:** Develop real-time speech emotion recognition systems capable of processing and analyzing emotions from streaming audio inputs, enabling applications in live interactions and feedback systems.

4. **Multimodal Fusion:** Investigate multimodal approaches by integrating speech data with other modalities (e.g., facial expressions, physiological signals) to enhance emotion recognition performance in diverse contexts.

5.**Ethical Considerations:** Address ethical considerations, including privacy protection, bias mitigation, and transparent decision-making processes, to ensure responsible deployment and use of the speech emotion recognition technology. By addressing these areas of future work, the speech emotion recognition system can continue to evolve, providing valuable insights into

human emotions and supporting various applications in healthcare, human-computer interaction, education, and beyond.

Appendix A

References

Bibliography

1. K. Chauhan, K. K. Sharma and T. Varma, "MNITJ-SEHSD: A Hindi Emotional Speech Database," 2023 International Conference on Communication, Circuits, and Systems (IC3S), BHUBANESWAR, India, 2023, pp. 1-6, doi: 10.1109/IC3S57698.2023.10169497.

2. Sultana, M. Z. Iqbal, M. R. Selim, M. M. Rashid and M. S. Rahman, "Bangla Speech Emotion Recognition and Cross-Lingual Study Using Deep CNN and BLSTM Networks," in IEEE Access, vol. 10, pp. 564-578, 2022, doi: 10.1109/ACCESS.2021.3136251.

3. Jadhav, V. Kadam, S. Prasad, N. Waghmare and S. Dhule, "An Emotion Recognition from Speech using LSTM," 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS), Coimbatore, India, 2023, pp. 834-842, doi: 10.1109/ICSCSS57650.2023.10169351.

4. T. Atmaja, A. Sasou and M. Akagi, "Speech Emotion and Naturalness Recognitions With Multitask and Single-Task Learnings," in IEEE Access, vol. 10, pp. 72381-72387, 2022, doi: 10.1109/ACCESS.2022.3189481

5. M. Wani, T. S. Gunawan, S. A. A. Qadri, M. Kartiwi and E. Ambikairajah, "A Comprehensive Review of Speech Emotion Recognition Systems," in IEEE Access, vol. 9, pp. 47795-47814, 2021, doi: 10.1109/ACCESS.2021.3068045

6. Munot and A. Nenkova, "Emotion impacts speech recognition performance," in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics,Student Res. Workshop, 2021, pp. 16–21,

7. ttp://surl.li/mdkvv

8. anyadara Saakshara, Kandula Pranathi, R.M.Gomathi, A . Sivasangari, P. Ajitha and T. Anandhi. Speaker Recognition System using Gaussian Mixture Model. In IEEE International Conference on Communication and Signal Processing, July 28 - 30, 2020, India.

9. iguo Chen; Zechen Guo; Dengyun Zhu; Hongzhi Yu.The Research of Application of Hidden Markov Model in the Speech Recognition. In IEEE 2021.

10. cent Recognition by Native Language Using Mel-Frequency Cepstral Coefficient and K-Nearest Neighbor. By Dwi Sari Widyowaty , Andi Sunyoto in IEEE.

11. etection and analysis of emotion recognition from speech signals using Decision Tree and comparing with Support Vector Machine. By Shaik Zuber; K. Vidhya.In IEEE 2022 International Conference. Features for Emotional Speaker Recognition. By Sandhya P, Spoorthy V, Shashidhar G. Koolagudi, Sobhana N.V. In IEEE 2021.

12. istributed Training of Deep Neural Network Acoustic Models for Automatic Speech Recognition.by Xiaodong Cui, Wei Zhang, Ulrich Finkler, George Saon, Michael Picheny, and David Kung. In IEEE 2020.

13. semi-supervised classification approach based on restricted Boltzmann machine for fMRI data.by Ning Liu, Li Yao, Xiaojie Zhao. In IEEE 2020.

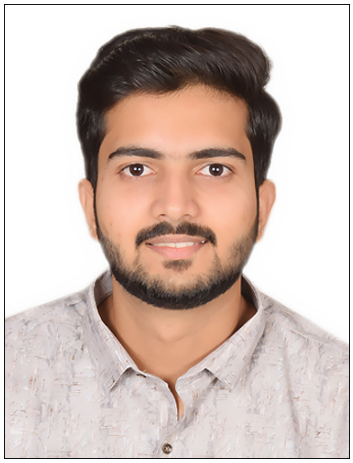
14. . Zhang, W. Zheng, Z. Cui, Y. Zong, and Y. Li, "Spatial–temporal recurrent neural network for emotion recognition," IEEE Trans. Cybern., vol. 49, no. 3, pp. 839–847, Jan. 2020

15. Framework for Hate Speech Detection Using Deep Convolutional Neural Network.by PRADEEP KUMAR ROY 1,3 , ASIS KUMAR TRIPATHY 1 , (Member, IEEE), TAPAN KUMAR DAS 1 , (Member, IEEE), AND XIAO-ZHI GAO 2. In IEEE 2020.

Appendix B

Project Members Information

1. **Name: Jayesh Nandu Chaudhari**
   - **Date of Birth:** 22/08/2002
   - **Gender:** Male
   - **Permanent Address:** Dadawadi, Jalgaon
   - **E-Mail id:** jayeshchaudhari8910@gmail.com
   - **Mobile/Contact No.:** 9028173112
   - **Placement Details:**
     (a) Not placed
   - **Paper Published:**
     (a) RACE 2024



2. **Name: Ruturaj Laxman Savakare**
   - **Date of Birth:** 20/06/2002
   - **Gender:** Male
   - **Permanent Address:** Plot 42,8/1 ,Shri Krishna Nagar,khedi shivar jalgaon-425001
   - **E-Mail id:** ruturajsavkare111@gmail.com
   - **Mobile/Contact No.:** 7709913698
   - **Placement Details:** Not Placed
   - **Paper Published:**
     (a) RACE 2024

3. **Name: Kalpesh Ananda Patil**
   - **Date of Birth:** 23/02/2002
   - **Gender:** Male
   - **Permanent Address:** Gat no.86,plot no. 4/4 ,behind dadawadi, near ram mandir, Jalgaon-425001
   - **E-Mail id:** kpatil1855@gmail.com
   - **Mobile/Contact No.:** 7666145481
   - **Placement Details:**
     (a) Not placed
   - **Paper Published:**
     (a) RACE 2024

4. **Name: Atharv Gurudas Khamkar**
   - **Date of Birth:** 19/01/2002
   - **Gender:** Male
   - **Permanent Address:** vahal, Ratnagiri, PIN - 415641
   - **E-Mail id:** khamkaratharv2002@gmail.com
   - **Mobile/Contact No.:** 7796073616
   - **Placement Details:**
     (a) Not placed
   - **Paper Published:**
     (a) RACE 2024