



كلية الحاسبات والذكاء الاصطناعي



Helwan University

Faculty of Computers and Artificial Intelligence

Information Systems Department

Speech Emotion Recognition (SER)

Bachelor's Thesis Presented by:

[Amal Taha Shaaban Mohamed (202000148)]

[Demiana nan zakria zaki (202000295)]

[Hala Ezzat Mahmoud Abass (202001021)]

[Salma Yasser Abdl-Sattar Radwan (202000396)]

[Sohaila Abdelkarim Fathy Awad (202000409)]

[Viola Amin Shawkey gerges (202000653)]

Submitted in partial fulfillment of the requirements for the degree of Bachelor
of Science in Computers & Artificial Intelligence at the Department of
Information Systems,

Faculty of Computers & Artificial Intelligence, Helwan University.

Supervised by:

Dr. Ahmed Al-sayed

June 2024

Abstract

This project presents the development of a comprehensive application that integrates a machine learning-based emotion detection model for speech analysis with a backend system and user interfaces for both clients and employees. The key components of this project include the implementation of a deep learning-based emotion recognition model, the creation of a Flask-powered backend API with a MySQL database, and the development of intuitive user interfaces for clients and employees. The emotion detection model is built using a Convolutional Neural Network (CNN) architecture and is trained on datasets such as **RAVDESS**, **TASS**, **CREMA** and **SAVEE**, achieving an impressive accuracy of 96.34%. This model serves as the core of the speech analysis functionality, allowing clients to submit voice recordings that are then processed to identify the underlying emotions. The backend system, developed using Flask and MySQL, provides the necessary infrastructure to handle user interactions, news management, complaint handling, and integration with the emotion detection model. Clients can create accounts, view news updates, submit complaints, and send voice recordings, while employees can retrieve user complaints, access voice recordings, and receive the emotion analysis results. To provide a seamless user experience, the project also includes the development of Android-based user interfaces for both clients and employees. These interfaces enable users to perform various tasks, such as logging in, viewing news, submitting complaints, and sending voice recordings, while employees can manage user complaints and access the emotion analysis results. The integration of the deep learning-based emotion detection model, the robust backend system, and the user-friendly interfaces creates a comprehensive application that addresses the needs of both clients and employees. This project showcases the powerful combination of advanced natural language processing techniques and well-designed software architecture to deliver a user-centric solution.

Acknowledgment

We extend our deepest appreciation to **Dr. Ahmed Elsayed** for his exceptional guidance and steadfast support throughout our graduation project at Helwan University. Under his mentorship, we were privileged not only to embark on this endeavor but also to benefit from his insightful guidance and direction. Dr. Elsayed's unwavering encouragement, profound understanding of the project's scope, and invaluable feedback were instrumental in steering our research towards success.

His expertise and assistance in the realm of Speech Emotion Recognition were invaluable in shaping our research and refining our methodologies. Dr. Elsayed's dedication to our project, insightful advice, and willingness to share his knowledge significantly enriched our work.

We are deeply grateful to our families, especially **our parents**, for their unwavering support, both emotionally and financially, throughout our academic journey at Helwan University. Their boundless encouragement, patience, and sacrifices have been the cornerstone of our success. We owe them a debt of gratitude for their endless prayers and relentless encouragement, which propelled us towards achievement.

Finally, we express our appreciation to the **Faculty Of Computers & Artificial Intelligence** for providing an enriching academic environment conducive to our growth and development. It is through their dedication and commitment to excellence that we were able to represent the best of what computer science graduates from Helwan University aspire to achieve.

List of Figures

Figure 1 : *Basic-CNN-architecture*

Figure 2 : *Schematic-of-the-RNN-architecture*

Figure 3 : The-structure-of-the-Long-Short-Term

Figure 4 : Papers

Figure 5 : Use Case Diagram

Figure 6 : Class Diagram

Figure 7 : Sequence Diagram

Figure 8 : Activity Diagram

Table of Contents

Chapter 1: Introduction

1.1 Overview.....	8
1.2 Objectives.....	8
1.3 Purpose.....	9
1.4 Scope.....	9
1.5 General Constraints.....	10

Chapter 2: Literature Review

2.1 Introduction.....	12
2.2 Background.....	13
2.3 SLR Methodology.....	16
2.3.1 Research Questions.....	16
2.3.2 Selection of Keywords.....	17
2.3.3 Formation of Query String.....	17
2.3.4 Selection of Search Space.....	17
2.3.5 Inclusion Criteria (IC).....	18
2.3.6 Exclusion Criteria (EC).....	18
2.4 Research Papers.....	18

Chapter 3: System Analysis & Design

3.1 System Analysis.....	39
3.1.1 Functional Requirements.....	39
- For the Employee.....	39
- For the User.....	39

3.1.2 Non-Functional Requirements.....	40
3.2 Software Design.....	41
3.2.1 Use Case Diagram.....	41
3.2.2 Class Diagram.....	42
3.2.3 Sequence Diagram.....	43
3.2.4 Activity Diagram.....	45

Chapter 4: Implementation & Results

4.1 System Requirements.....	48
4.1.1 Deep Learning Model.....	48
4.1.2 Backend.....	49
4.1.3 Android App Development.....	49
4.2 Model.....	50
4.2.1 Datasets Description.....	50
4.2.2 Data Visualisation and Exploration.....	51
4.2.3 Data augmentation.....	52
4.2.4 Feature extraction.....	53
4.2.5 Data Preparation	54
4.2.6 CNN Model	55
4.2.7 Experimental Model	57
4.3 Backend.....	61
4.3.1 Employee Functions.....	61
4.3.2 User Functions.....	62
4.4 Interface app.....	70

Chapter 5: Testing

5.1 Testing 80

5.1.1 Backend..... 80

Chapter 6: Conclusion & Future work

6.1 Conclusion..... 88

6.2 Future work..... 89

References..... 90

Chapter 1

Introduction

Introduction:

1.1 Overview

In today's digital age, effective communication and user engagement are paramount. Our graduation project focuses on developing a mobile application with two distinct interfaces tailored for users and employees. This application not only facilitates routine interactions but also incorporates advanced features such as speech-to-emotion conversion. The primary innovation of this project lies in its ability to analyze voice recordings from users and interpret their emotions using a deep learning model, which translates these emotions into corresponding emojis. Additionally, the application allows employees to manage news updates, handle user complaints, and access user-related data, thereby enhancing overall user experience and operational efficiency.

1.2 Objectives

The objectives of our project are clearly defined to ensure the application meets its intended goals. These objectives are:

- **Develop User and Employee Interfaces:**
 - **User Interface:** Enable users to create accounts, log in, view news, send voice recordings, and submit complaints.
 - **Employee Interface:** Allow employees to log in, upload news, receive and analyze user recordings, respond to complaints, and access user data.
- **Implement Secure Authentication:** Ensure that both users and employees can securely log into the system, protecting sensitive information.
- **News Management:**
 - **News Submission:** Enable employees to add news about specific things.
 - **Show News:** The users can see the news added by employees.
- **Voice Recording Analysis:**
 - **Recording Submission:** Enable users to send voice recordings to employees.
 - **Emotion Detection:** Utilize a deep learning model to analyze these recordings and predict the underlying emotions, represented as text.
- **Complaint Handling:**

- **Complaint Submission:** Allow users to provide a detailed description of their problem, along with their contact phone number for communication purposes.
- **Complaints Retrieve:** Employees handle user complaints, work to resolve them, and communicate with the users via phone.

1.3 Purpose

The purpose of this project is multi-faceted, aiming to integrate emotional intelligence into digital communication and improve overall user satisfaction. Specifically, the project aspires to:

- **Enhance Communication:** By analyzing and displaying emotions through emojis, the application aims to make digital communication more intuitive and emotionally aware.
- **Improve User Experience:** Provide users with a platform where they can easily interact, receive updates, and have their issues addressed promptly.
- **Support Employees:** Equip employees with tools to efficiently manage user interactions, analyze emotional data, and respond to complaints, leading to improved service quality.

1.4 Scope

The project encompasses various activities that are crucial for its successful completion. These activities include:

- **Planning:**
 - Define project goals, requirements, timelines, and resources.
 - Identify key milestones and deliverables.
- **Designing:**
 - Create wireframes and mockups for user and employee interfaces.
 - Develop the system architecture and design the database schema.
- **Coding:**
 - Implement user authentication and session management.
 - Develop features for news management, voice recording submission, and complaint handling.
 - Integrate the deep learning model for emotion detection.
- **Testing:**
 - Conduct unit testing, integration testing, and system testing to ensure functionality.

- Perform user acceptance testing to gather feedback and make necessary improvements.
- Optimize the application for performance and compatibility across different devices.
- **Documentation:**
 - Prepare detailed documentation including system architecture, API references, user manuals, and test reports.
 - Ensure documentation is clear and comprehensive for future maintenance and enhancements.

1.5 General Constraints

Throughout the development of this project, several constraints were encountered that affected the progress and outcome. These constraints include:

- **Time Constraints:** The limited timeframe for the project-imposed challenges on thorough testing and refinement of features, requiring prioritization of core functionalities over additional enhancements.
- **Data Availability:** Obtaining a diverse and high-quality dataset for training the deep learning model was challenging. This limitation impacted the accuracy and reliability of the emotion detection feature.
- **Technical Limitations:** Achieving real-time processing for voice recordings and ensuring the application works seamlessly across various devices required significant optimization efforts and innovative solutions.
- **Scope Definition:** Initial ambiguity in the project scope led to scope creep and required adjustments during the development phase to align with the project goals.
- **Resource Constraints:** Limited access to advanced development tools, computing resources, and expert guidance posed challenges in implementing and testing complex features.

Chapter 2

Literature Review

Literature Review:

2.1 Introduction

Speech Emotion Recognition (SER) involves using computational analysis of vocal cues to identify and understand the emotional content expressed in human speech. It has applications in areas such as virtual assistants, mental health, and human-computer interaction. The future of SER looks promising, with plans to address individual variations, delicate expression detection, and dataset generalization. These challenges present opportunities for further discoveries and developments, leading to a better understanding of emotional complexity in speech.

SER has applications beyond research and is used in fields like mental health diagnoses and human-computer interaction. This emerging field not only changes our understanding of spoken language but also creates new opportunities for emotion and technology to coexist harmoniously in expressive communication.

A wide range of datasets form the basis for deciphering the complex fabric of human emotion in Speech Emotion Recognition (SER). Various frameworks, such as attentive convolutional neural networks and deep multi-layered neural networks, are used to investigate emotional subtleties in spoken language using signal processing and machine learning techniques.

The effectiveness of these approaches in identifying emotions is measured against performance criteria such as F1 score, accuracy, recall, and precision. Results, often presented as confusion matrices and recognition rates, provide information on the effectiveness of SER models and illustrate their performance on datasets like IEMOCAP and EMO-DB

2.2 Background

CNN

One class of deep learning models called Convolutional Neural Networks (CNNs) is made especially for handling grid-like input, such as pictures and movies.

CNNs are widely used in computer vision because of their exceptional capacity to automatically identify and extract hierarchical features from input data.

Convolutional layers, which employ filters that glide over the input to capture spatial hierarchies and patterns, are the main source of innovation.

CNNs are crucial for tasks like object detection, segmentation, and picture classification because they can perceive and comprehend complicated visual information by utilizing processes like pooling and fully linked layers.

Because of their adaptability and effectiveness, CNNs are now a key component of artificial intelligence, especially in applications where the ability to comprehend and interpret visual data is essential.

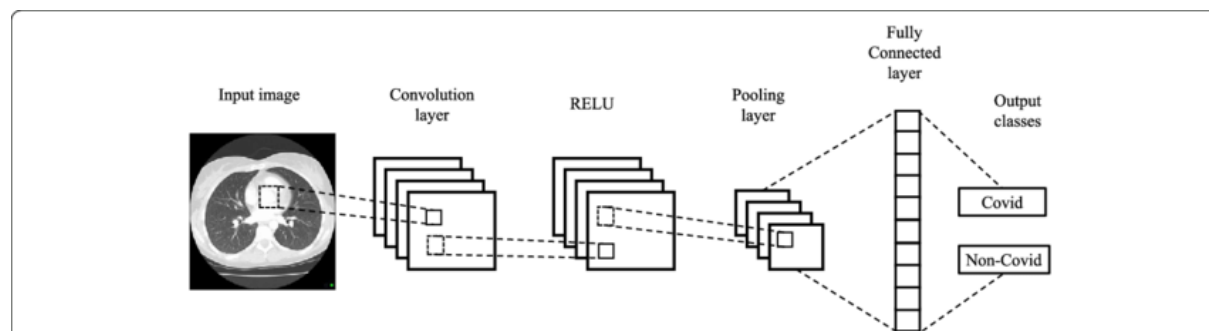


Figure1 Basic-CNN-architecture

RNN

A class of artificial neural networks called recurrent neural networks (RNNs) is made to process data sequentially. RNNs, in contrast to conventional feedforward neural networks, feature connections that form internal loops that enable the network to remember earlier inputs for calculations. RNNs are ideally suited for applications like speech recognition, time series prediction, and natural language processing because of their innate sequential memory.

Recurrent neural networks (RNNs) handle sequential data by repeatedly processing each element while preserving a hidden state that retains information from earlier inputs. RNNs are useful for capturing patterns and correlations throughout time because of their capacity to consider the context and temporal dependencies in data. Nevertheless, long-term dependencies and the "vanishing gradient" issue can be problems for conventional RNNs, which reduces their ability to extract information from far-off historical inputs.

Many RNN variations, including Gated Recurrent Units (GRUs) and Long Short-Term Memory (LSTM) networks, have been created to overcome these problems. These architectures improve RNNs' capacity to represent intricate temporal patterns by incorporating specialized methods that better capture and transfer information across longer sequences.

Speech recognition, time series analysis, and natural language processing are just a few of the many domains in which RNNs are used. RNNs continue to be a fundamental technology for applications requiring a sophisticated grasp of sequential information, even with the introduction of more sophisticated models.

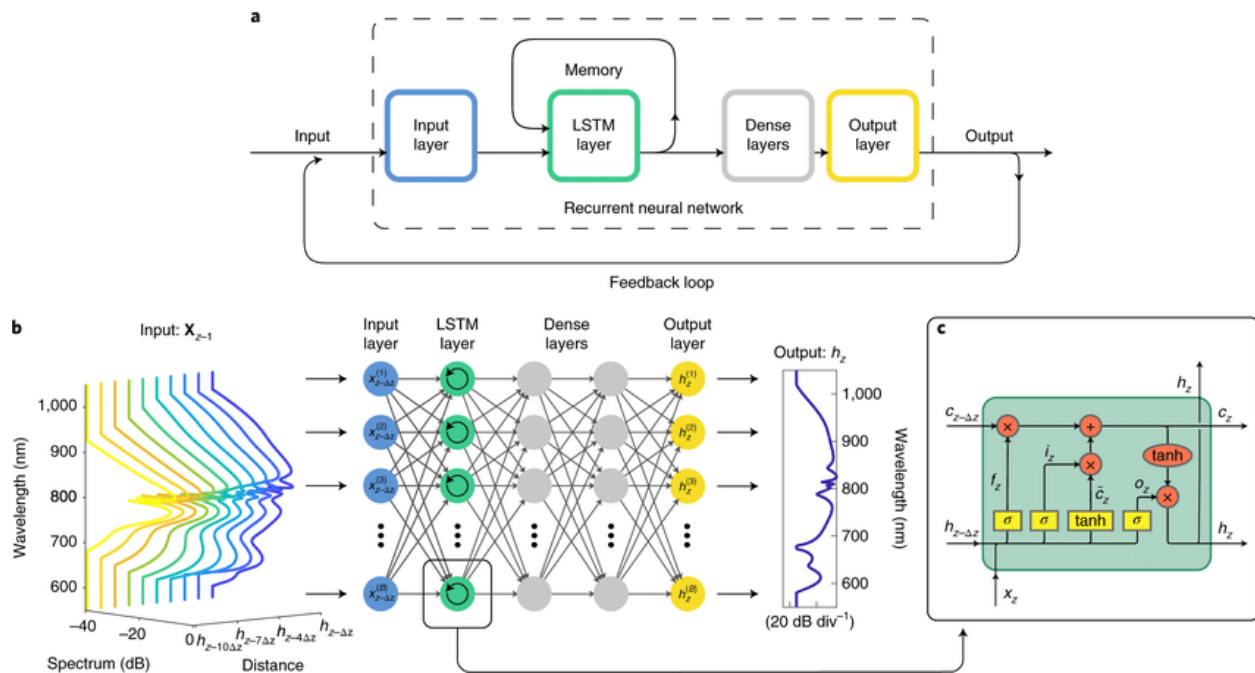


Figure2

Schematic-of-the-RNN-architecture

LSTM

LSTMs can selectively store, retrieve, and forget information over long sequences because they have memory cells and a gating mechanism. With the help of this architecture, the vanishing gradient issue that plagues conventional RNNs is lessened, enabling LSTMs to retain and make use of data from far-off historical inputs. The input, forget, and output gates in the architecture control the information flow, improving the model's capacity to identify pertinent patterns over a range of time scales.

Recurrent neural network (RNN) architecture with Long Short-Term Memory (LSTM) was created to address issues with identifying and preserving long-term dependencies in sequential input. LSTMs were developed to overcome the drawbacks of conventional RNNs and have shown promise in time series prediction, natural language processing, and other applications where a knowledge of temporal correlations

As internal memory storage, memory cells in long short-term memory (LSTMs) facilitate the acquisition of long-term dependence and guard against the gradual loss of important knowledge. Because of this, LSTMs work especially well in situations where recollection of previous events and contextual comprehension are crucial.

LSTMs are now a mainstay of sequential data deep learning and are used extensively in speech recognition, machine translation, and sentiment analysis, among other applications. Their effectiveness is attributed to their capacity to tackle the difficulties involved in representing sequential data, rendering them a significant breakthrough in the realm of recurrent neural networks.

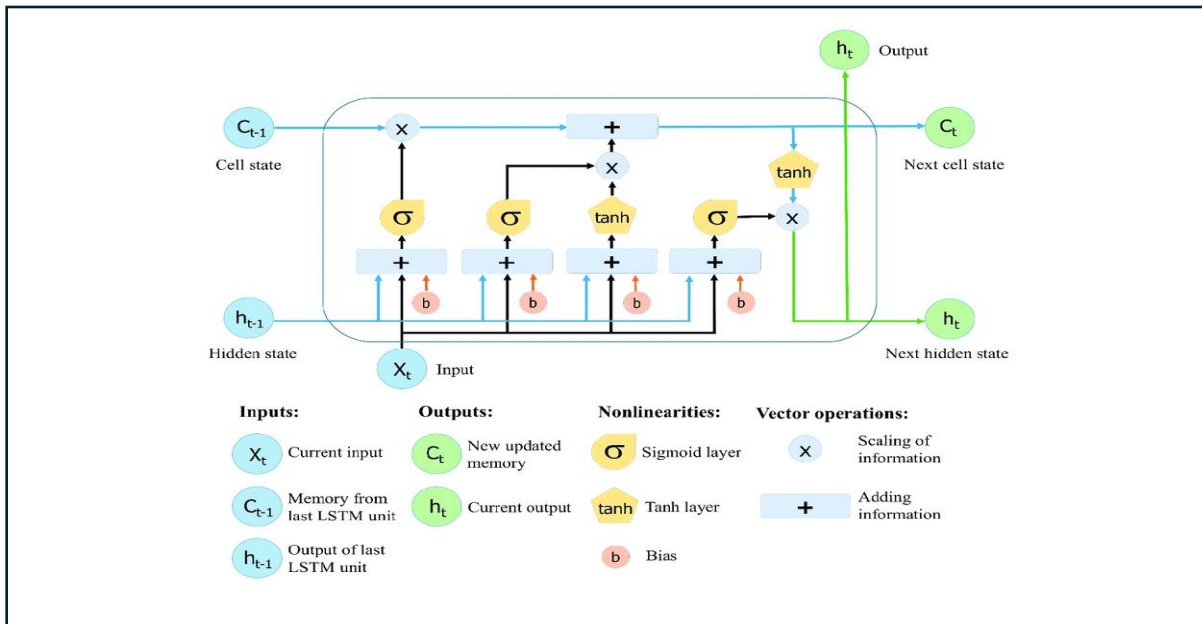


Figure3 The-structure-of-the-Long-Short-Term

Research Protocol

This study focuses on researching speech-to-emotion recognition. The literature review aims to investigate the selected research articles deeply. A research protocol must be adhered to formalize the review process. This process involves following various steps in a specific sequence.

2.3 SLR Methodology

2.3.1 Research Questions

The research questions should accurately represent the main objectives of a literature review. During the thorough review of the most relevant extracted articles, we will attempt to find the answers to these questions.

The research questions for this literature review are as follows:

- RQ1: What datasets are typically used in speech emotion recognition research papers?
- RQ2: What is the methodology/framework used in that paper?
- RQ3: What are the applications/problems in that paper?
- RQ4: What are the performance metrics?
- RQ5: What are the results?
- RQ6: What is the future study?
- RQ7: What are the challenges?

2.3.2 Selection of Keywords

The extraction of keywords from research questions is considered a fundamental step in the research process. The keywords will be systematically organized to form a query string for extracting the articles.

The keywords extracted from the research questions are: Machine Learning, Deep Learning, Speech, Emotion Recognition, and Artificial Intelligence.

2.3.3 Formation of Query String

The query string is formed by using various combinations of selected keywords and is used to extract the relevant research articles from the selected libraries.

The search query has been finalized with the extracted keywords. (Speech or "Emotion Recognition" or "Emotion Detection" or "Speech Analysis" or "Speech Processing") AND (DL or "Deep Learning" OR "Machine Learning" OR ML OR AI) AND (CNN or RNN or LSTM or Transformer or SVM)

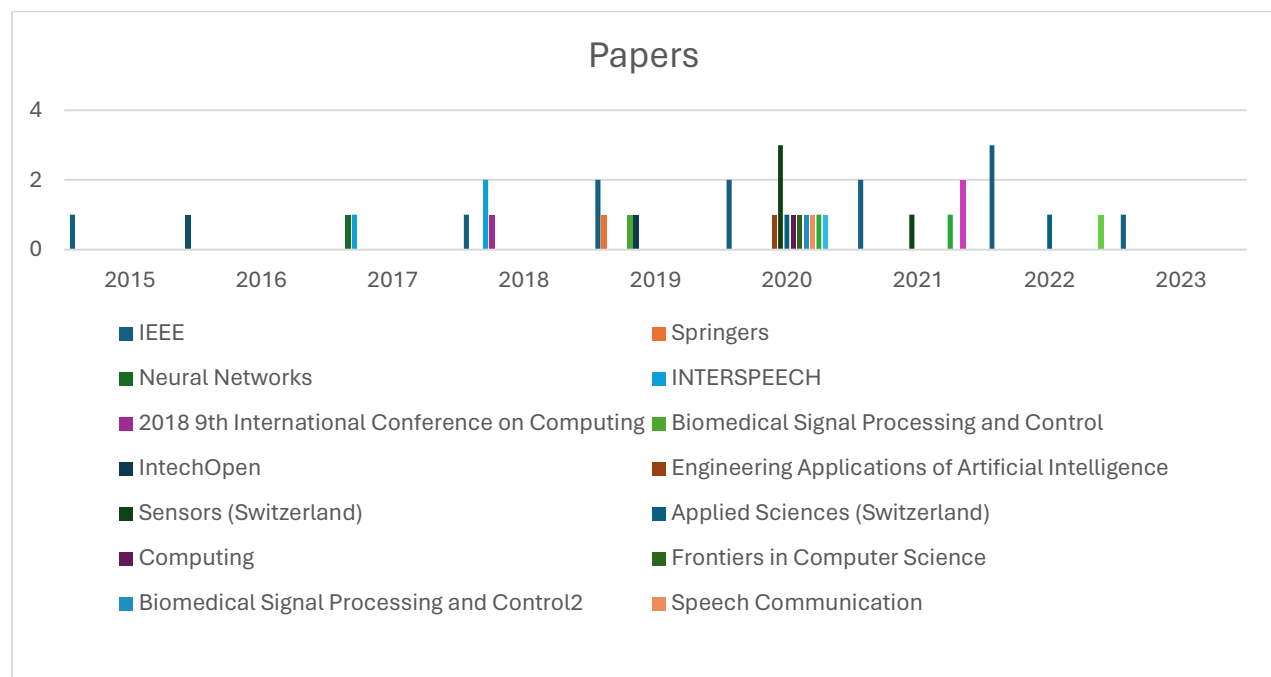


Figure 4 Papers

2.3.4 Selection of Search Space

Three popular online search libraries have been chosen to extract the relevant research literature: IEEE, Elsevier, and Springer. The three selected libraries have distinct characteristics and offer various options for searching relevant materials. Furthermore, the query had to be searched multiple times using different combinations of selected keywords.

2.3.5 Inclusion Criteria (IC)

Inclusion criteria are formed with the following rules:

- IC1: papers published from 2012 to the present date.
- IC2: papers that focus on speech emotion recognition.
- IC3: papers with hog citation.
- IC4: papers that employ deep learning or neural network-based methods.
- IC5: papers conducted on the English language.
- IC6: papers focus on the application domain.

2.3.6 Exclusion Criteria (EC)

Exclusion criteria are formed with the following rules:

- EC1: papers published before 2012.
- EC2: papers that focus on recognizing emotions from text or facial expressions.
- EC3: papers that are not in English language.

2.4 Research Papers

The study [1] Three databases are used in this work to evaluate the ADRNN model: the Berlin EmoDB corpus, IEMOCAP, and a cross-corpus mixture of the two. The study discusses a variety of uses for the developed networks, including human-machine interfaces, social education, and healthcare. The model overcomes individual weaknesses by combining different network strengths through the introduction of a revolutionary architecture known as ADRNN. With a noteworthy 48.8% Unweighted Average Recall (UAR) on the development dataset, the ADRNN network demonstrates improved generalization in its ability to identify speech emotions by employing 3-D Log-Mel spectrum characteristics from raw speech signals. The study uses UAR as the performance metric and addresses the difficulties in Speech Emotion Recognition (SER), such as the absence of a common dataset, the difficulty of subjectively classifying emotions, and the intricacy of capturing temporal dynamics in speech signals. It also highlights future work that will concentrate on investigating various feature extraction techniques, investigating transfer learning, and maybe integrating multimodal data.

The study [2] examines different deep learning architectures for Speech Emotion Recognition. The authors utilize the Interactive Emotional Dyadic Motion Capture (IEMOCAP) database, which contains audio-visual recordings divided into five sessions. They propose a deep multi-layered neural network capable of handling utterances of varying length as the basis of their methodology. The system achieves state-of-the-art performance on the IEMOCAP database, with 63.8% test accuracy and 60.5% test unweighted average recall (UAR), demonstrating improved results compared to previous research using other approaches on the same database. The study also

highlights challenges in Speech Emotion Recognition, such as speaker heterogeneity in emotional expression.

The article [3] examines the impact of input features, signal length, and acted speech on speech emotion recognition. The study utilizes the Interactive Emotional Motion Capture (IEMOCAP) database for experiments. To enhance recognition performance by extracting relevant information from input characteristics, the proposed approach employs an attentive convolutional neural network (ACNN) that combines convolutional neural networks and attention processes. The research demonstrates that regardless of the input feature selection, recognition performance is significantly influenced by the type of speech data. When applied to the improvised speech data in the IEMOCAP database, the ACNN model yields state-of-the-art results.

A lightweight fully convolutional neural network, LIGHT-SERNET, has been introduced for speech emotion recognition in a recent paper titled [4]. The model was evaluated using two datasets: the Berlin Emotion Database (EMO-DB) and the Interactive Emotional Dyadic Motion Captures (IEMOCAP). On the IEMOCAP (scripted+improvised) dataset, the proposed model demonstrated superior performance in terms of unweighted accuracy (UA), weighted accuracy (WA), and F1 score compared to previous models. It addresses the resource consumption issue by providing a lightweight and effective solution, making it suitable for deployment in systems with limited hardware resources.

A novel method for emotion recognition is proposed in the research paper [5]. The authors collected 339 audio recordings, each lasting around two seconds, representing one of four emotions: neutral, angry, joyful, or sad. They introduced a strategy for effective emotion recognition and training, utilizing Neural Structured Learning (NSL), Mel Frequency Cepstrum Coefficients (MFCC), and discriminant analysis. The results, including the mean recall rate in Table 1 and overall recognition rates in Fig. 18, demonstrated the effectiveness of this strategy compared to conventional approaches. The research also highlighted the susceptibility of machine learning models to adversarial cases, although it did not explicitly outline the challenges. Future research may explore advanced deep learning methods.

A study [6] introduces a unique framework for identifying emotions in voice signals. The authors utilized the IEMOCAP and EMO-DB benchmark datasets, which contain voice data from ten actors expressing a range of emotions such as happiness, sadness, rage, and neutrality. The proposed approach involves pre-processing, deep neural network (DNN) classification, and genetic algorithm feature selection. The system achieved high recognition rates, with the highest accuracy rates in the IEMOCAP studies being 85% for anger, 74% for sorrow, and 81% for neutrality. In the EMO-DB experiments, the best accuracy rates were 96% for anger, 99% for boredom, disgust, and fear, and 92% for sadness. Despite the challenges in feature selection and handling, the system demonstrated promising results.

A deep learning method for voice emotion recognition is outlined in the publication [7]. The authors utilized three datasets, namely IEMOCAP, EMODB, and RAVDESS, containing audiovisual data from different speakers. The proposed methodology incorporates deep Bidirectional Long Short-Term Memory (BiLSTM) and Convolutional Neural Network (CNN)

with Radial Basis Function Network (RBFN), along with normalization approaches. The model achieved an accuracy of 85.57% on the IEMOCAP dataset, 72.25% on the EMO-DB dataset, and 77.02% on the RAVDESS dataset. However, the authors encountered challenges such as computing expenses, discriminative feature selection, and determining an appropriate period for audio segmentation. Future research is recommended to focus on improving the recognition rate of specific emotions and expanding the scope of the study.

The DEAP dataset is utilized in the publication [8] for emotion identification. The authors propose a deep learning method based on a parallel Convolutional Recurrent Neural Network (CRNN) to identify human emotion states from multi-channel EEG recordings. The CRNN comprises a convolutional neural network (CNN) and a recurrent neural network (RNN) as two parallel branches. The CNN extracts spatial features from EEG signals, while the RNN captures temporal relationships between these features. The suggested method faces challenges such as the high dimensionality and complexity of EEG signals, individual variations in EEG signals, and the need for a substantial amount of labeled data for training. Despite these difficulties, the approach shows promising results in identifying human emotions.

The publication [9] is based on the DEAP dataset created by researchers at Queen Mary University of London. The approach and framework utilized in this work for binary emotion categorization are founded on convolutional neural networks (CNNs), incorporating variables from the time, frequency, and valence and arousal domains. The study's outcomes demonstrated that models with incorporated temporal and frequency domain information performed better in EEG emotion identification. The study also covers the benefits and drawbacks of utilizing CNN, including its ability to lower network complexity, minimize the number of training parameters, and increase fault tolerance and robustness. The authors recommend that future research concentrate on creating.

In Study [10], The publication introduces a novel emotion recognition framework for conversations called ICON. The framework utilizes two benchmark datasets, SEMAINE and IEMOCAP, to demonstrate its effectiveness in extracting emotional dynamics from language, audio, and visual modalities. The study shows that ICON outperforms other models on both datasets. While the authors did not address specific challenges, they highlighted ICON's potential for use in other dialogue-based applications and its expansion to multi-party discussions. Future research will involve testing ICON on more relevant applications and evaluating its performance across various modalities using metrics for assessment.

In Study [11], Regarding voice emotion recognition, this paper uses a semi-supervised autoencoder approach with the GeWEC dataset for whispered and normal paired utterances. The model overcomes the lack of labeled data by outperforming other approaches and achieving state-of-the-art performance on many databases. The suggested model performs exceptionally well when measured by the unweighted average recall (UAR) metric, attaining 66.3% UAR on the IEMOCAP database and 60.5% on MSP-IMPROV. The inherent problems of semi-supervised learning and the complexity of emotional expressions are acknowledged, even though the research doesn't specifically list issues. Future research recommends expanding the suggested model to recurrent algorithms and investigating more complex semi-supervised algorithms.

In Study [12], To tackle the issues of low labeled data and significant variability in emotional expressions, this paper uses the MSP-Podcast corpus, which has been annotated for emotional qualities, in the context of speech emotion recognition. In this paper, ladder networks are introduced as a framework and the use of speech emotion recognition in many sectors, such as security and healthcare, is discussed. The study uses this paradigm to evaluate implementations, both supervised and unsupervised, against baselines for single-task and multi-task learning. When it comes to emotional characteristics, such as dominance and arousal, the ladder network performs better than baselines on both test and development sets. The ladder network was assessed using the Concordance Correlation Coefficient.

In Study [13], This research proposes a Cooperative Learning framework for emotion recognition in speech tasks, using the Speech Under Simulated and Acted Stress database and the FAU Aibo Emotion Corpus. The system, which includes Co-EM, Co-SSVM, and Co-SSL algorithms, is intended for real-world applications in human-computer interaction and social robots. It routinely outperforms stand-alone Active and Semi-Supervised Learning methods. Notably, with a 75% reduction of labeled occurrences, Cooperative Learning offers cost-effectiveness and produces competitive Unweighted Average Recall (UAR) outcomes, nearly matching baseline performance. Future work directions include investigating diverse feature types, integrating deep neural networks, and expanding Cooperative Learning into various domains like image and video analysis, showcasing its potential for broader applications. These efforts are made possible despite certain limitations, such as a small dataset and a single-language focus.

In Study [14], To improve deep neural network training efficiency for voice emotion identification, curriculum learning is introduced in this paper. Creating curricula based on indicators for inter-evaluation agreement and evaluator disagreement aims to improve system performance. The study shows that curriculum learning works much better than models without a curriculum or with randomly chosen bins, especially when criteria 2 and 3 are met. Notably, criteria 3 exhibits statistical superiority over baselines in arousal, valence, and dominance when utilizing the minimax entropy framework. The main evaluation metric, the F1-score, emphasizes recall rates and precision for each class. Even with these developments, problems still exist, such as the need to recognize subtle expressions, individual variations, environmental factors, and dataset generalization. Future work will focus on continuous curriculum learning refinement for improved speech-emotion recognition systems.

In Study [15], The MAS, IEMOCAP, and FAU-AIBO evaluations performed by the MCIL approach are crucial for applications in call centers, social robots, and healthcare. MCIL uses several classifiers and interactive learning to address emotion ambiguity; it ranks third on FAU-AIBO but outperforms MAS and IEMOCAP. A Kappa value is used to quantify the consistency and accuracy of classification in evaluation metrics. Difficulties include the requirement for a cohesive strategy and emotional ambiguity, which are covered in Sections I and III. Further research endeavors to investigate feature extraction, and sophisticated interactive learning, expand MCIL for intricate assignments, and use it for the identification of emotions in images and videos, as delineated in Section VI.

In Study [16], Pitt and DAIC-WOZ datasets. Using Transformer models, the suggested SpeechFormer++ framework performs better on four different corpora. Unit and word encoders for structure-based speech unit learning and a merging block for aggregation are integrated into the efficient and hierarchical architecture. The weighted average F1 (WF1), weighted accuracy (WA), and unweighted accuracy (UA) show notable gains over the competition. Difficulties include extracting complex features and modeling dynamic speech changes. Subsequent research endeavors to investigate SpeechFormer++ in additional speech-related assignments, low-resource situations, comprehensibility, adaptability to diverse fields, and resilience against hostile assaults.

In Study [17], The research addresses emotion recognition, sentiment analysis, and speaker identification through paralinguistic speech processing the study addresses applications in mental health monitoring, human-robot interaction, customer service, education, and entertainment. It does this by using the IEMOCAP and MELD datasets and introducing Friends-ELE for conversational emotion recognition. The suggested CTNet framework emphasizes multimodal characteristics, including speaker embeddings, and uses transformer-based structures to achieve better results on the IEMOCAP and MELD datasets. Weighted and unweighted average accuracy (WAA and UAA) and weighted and unweighted average F1 (WAF1 and UAF1) are two examples of evaluation measures. Variability in emotional expressions, a lack of labeled data, multimodal fusion complexity, speaker and contextual dependencies, and interpretability problems are some of the challenges associated with conversational emotion recognition. It is suggested that future studies investigate CTNet on other datasets, add more modalities, evaluate generalization across languages and cultures, use transfer learning, and improve interpretability.

In Study [18], For voice emotion recognition, the paper makes use of a self-constructed emotion library and the CASIA Chinese Emotional Voice Database. Outperforming KNN, Softmax, SVM, sparse representation, and neural network algorithms, the proposed stacked kernel sparse deep model achieves high accuracy (87.5%) and F1-score (0.875) on CASIA and 86.7% accuracy and 0.865 F1-score on the self-built library. It also incorporates auto-encoder, denoising auto-encoder, and sparse auto-encoder. Evaluation measures include F1-score and accuracy, and problems with feature learning efficacy and ambiguous emotion model definitions have been noted. It is suggested that future studies look into improved feature extraction techniques, combine various extraction methods, use deep learning for abstract features, and build more representative and diverse speech emotion databases.

In Study [19], Using the LSSSED and IEMOCAP datasets, the research presents ISNet, a solution for individual differences in speech emotion recognition. ISNet provides metrics at the individual level and enhances overall performance through benchmark supervision and standardization. Applications for it include voice assistants, advertising, customer support, and psychological testing. Weighted accuracy (WA) and unweighted accuracy (UA) are two metrics used in evaluation. Individual differences provide challenges, and the report proposes further research on multimodal information, sophisticated models, generalization, and potential applications in voice synthesis and speaker recognition.

In the study [20], The authors of this study present two brand-new datasets for audio-visual speech recognition: LRS2-BBC and LRS3-TED. Their research focuses on applications such as

human-robot interaction, hearing-impaired people, and voice recognition in loud environments. The suggested model combines an audio feature extraction and fusion Transformer-based model with a spatiotemporal Resnet for visual feature extraction. Two lip-reading models—a Transformer-based model and a CTC-based model—are included in the comparative analysis to highlight how complementary audio speech recognition and lip reading are. On the LRS2-BBC and LRS3-TED datasets, the model achieves state-of-the-art performance with a WER of 14.3% and SER of 47.7% on LRS2-BBC and a WER of 16.7% and SER of 54.2% on LRS3-TED. WER, SER, and CER are included in the assessment metrics. The difficulties with audio-visual speech recognition are emphasized, including the lack of large-scale datasets and visual diversity. To improve model performance, future work should investigate sophisticated techniques for extracting visual features, attention mechanisms, and unsupervised pre-training.

In the study [21], an emotion recognition method is introduced and evaluated on a large amount of emotion-related Big Data. Bimodal inputs of speech and video from fifty university-level students who have been trained to mimic normal, sad, and happy emotional expressions are used. The suggested system uses a deep learning methodology and is suitable for social robotics, affective computing, and human-computer interaction. The system integrates a sophisticated key frame selection technique, integrating grayscale images in the three-dimensional CNN with images of a local binary pattern (LBP) and interlaced derivative pattern (IDP), using a two-dimensional CNN for audio signals and a three-dimensional CNN for video signals. An extreme learning machine is used to accomplish feature fusion (ELM). With evaluation metrics including accuracy, precision, recall, and F1-score, the results show an impressive 92.5% accuracy, surpassing the performance of state-of-the-art methods. Challenges encompass feature selection, deep learning model design, and dataset availability. Future work aims to enhance accuracy through advanced deep-learning models and larger datasets.

The study [22], evaluates emotion recognition using a large-scale emotion Big Data set that includes speech and video inputs from fifty university-level students who have been trained to mimic normal, sad, and happy emotions. Applications in speaker identification, emotion recognition, and speech recognition in HCI systems are highlighted in the paper. Mel and Gammatone Frequency Cepstral Coefficients (MFCC and GFCC) are combined in the suggested Deep C-RNN technique to enable emotion analysis across various audio signal versions. The model outperforms baseline techniques with an accuracy that exceeds 80% and minimal loss during training and testing. Metrics for evaluation consist of accuracy rate and loss. The need for representative data encompassing the whole spectrum of emotion expression, the difficulties of feature recognition and fusion, complexity, as well as the need for talented actors to portray emotions. It is recommended that future work examine various joint representation functionalities and benchmark databases to improve model performance.

The study [23], The goal of the suggested method is to improve speech emotion identification accuracy across a range of industries, including entertainment, education, and healthcare. The methodology involves preprocessing the data, training a logistic regression meta-classifier, merging estimates by stacking, training base classifiers with five-fold cross-validation, and combining BLSTM and CNN models using a Stacking technique. Findings show that the method

outperforms baseline models in speech emotion recognition, with good recall, accuracy, and F1-score. Evaluation metrics include F1-score, accuracy, precision, recall, and confusion matrix utilization. Emotion unpredictability, lack of consistent datasets, and difficulties extracting relevant features from speech signals are challenges in speech emotion recognition. Future research attempts to investigate substitute deep learning methods and Use feature extraction techniques to improve speech emotion recognition even more.

In the study [24], The adaptability of the ADRNN model is evaluated on three databases: the Berlin EmoDB corpus, IEMOCAP, and a cross-corpus between the two. The networks that have been built have the potential to be used in human-machine interfacing, social education, and healthcare. The research introduces the ADRNN architecture and uses 3-D Log-Mel spectrum features taken from unprocessed voice signals to make use of the model's capacity to fuse disparate networks. Beyond its effectiveness in speech emotion identification, ADRNN also demonstrates strong generalization skills, as evidenced by its impressive 48.8% Unweighted Average Recall (UAR) on the development dataset. Subjectivity in emotion labeling, the difficulty of capturing temporal dynamics in speech signals, and the lack of a standardized dataset are recognized as challenges in speech emotion recognition (SER). To further progress the subject, the research proposes future work that will examine multimodal data, investigate the impact of different feature extraction approaches, and investigate transfer learning.

In the study [25], The suggested method is based on the Interactive Emotional Dyadic Motion Capture (IEMOCAP) dataset, which has applications in speech-based mental health diagnosis, emotion-aware virtual assistants, and human-computer interaction. The approach successfully extracts lexical and auditory information from speech for emotion recognition by introducing a multi-scale Convolutional Neural Network (CNN) with an attention mechanism. On the IEMOCAP dataset, the model outperformed earlier state-of-the-art techniques, as evidenced by its weighted accuracy (WA) of 73.5% and unweighted accuracy (UA) of 68.9%. The main assessment measures are UA and WA. The lack of extensive labeled datasets for speech emotion identification and the difficulty of modeling the interaction between lexical and acoustic information are regarded as challenges. To improve the performance of the suggested model even more, future research will investigate the combination of transfer learning and data augmentation approaches.

In the study [26], Using 46 French-speaking individuals' voice recordings divided into train, validation, and test sets and stratified by age and gender, the RECOLA database is used. The expanded Geneva minimalistic acoustic feature set (eGeMAPS) is used to extract acoustic features. In addition to real-time emotion identification for personalized services and entertainment, the proposed end-to-end Convolutional Recurrent Neural Network (CRNN) shows potential for applications in human-robot interaction, virtual reality, and mental health diagnostics. Convolutional, LSTM, and fully connected layers make up the CRNN architecture. It is trained to maximize the concordance correlation coefficient (ρ_c) to predict arousal and valence levels directly from unprocessed speech data. The results show that the CRNN is better than conventional feature-based techniques in terms of ρ_c . Among the difficulties include speech signal variability brought on by contextual, cultural, and individual factors; also, there is no standardized dataset for emotion recognition. It is suggested that future research increase the generality of the model across

languages and cultures and investigate the integration of multimodal input, including speech and facial expressions, to improve the recognition of emotions.

In the study [27], The AFEW5.0 and BAUM-1s are two difficult spontaneous emotional speech datasets that the authors test. The study highlights the value of automatic emotion recognition in spontaneous speech for improving service quality and user experience. It covers applications in healthcare, customer service, and human-robot interaction. The suggested technique uses multi-convolutional Neural Networks (CNNs) to learn deep multimodal audio features, presenting a novel way for spontaneous Speech Emotion Recognition (SER). The approach learns deep multimodal segment-level characteristics from the original 1D audio input by employing multiple CNNs, and it demonstrates great accuracy in identifying emotions in spontaneous speech. On the experimental datasets, the strategy compares favorably with state-of-the-art techniques and performs better than previous works using hand-designed features. Accuracy, F1-score, and a confusion matrix are examples of evaluation metrics. The lack of a standardized emotion recognition dataset and speech signal variability brought on by personal, cultural, and environmental influences are issues that are addressed in the article. To further improve the performance of the suggested strategy, future research is suggested to investigate additional deep learning models, such as transformers and Recurrent Neural Networks (RNNs).

In Study [28], The datasets used in the article to classify customer happiness based on speech include IEMOCAP, RAVDESS, EMODB, and KONECTADB. It presents a new method for modeling prosody, articulation, and phonation that outperforms in uncontrolled acoustic environments and produces competitive results on common emotional speech databases. The confusion matrix, F1-score, and accuracy are examples of performance measures. Difficulties include inconsistent emotional expression, lack of a uniform annotation technique, and challenges in gathering real-world data. Applications include producing emotionally intelligent virtual assistants, boosting human-robot interaction, and improving customer service in contact centers. Considered emotions include neutral, happy, sadness, and rage. Future research attempts to investigate deep learning methods for feature extraction and classification, as well as to improve the system's resilience to various acoustic environments.

In Study [29], MLT-DNet is introduced for Speech Emotion Recognition using the IEMOCAP dataset. For better performance, the unique strategy uses a Multi-Learning Trick along with a 1D Dilated CNN. Anger, sadness, and neutral emotions provide better results when measured using accuracy, recall, F1_score, and confusion matrices. The model outperforms baselines in spite of obstacles like subjective labeling and emotional fluctuation. Subsequent paths seek to improve the detection of happiness, investigate a variety of datasets, and use transfer learning in a range of settings. The research highlights the model's possible uses in benchmark datasets, which represents a noteworthy advancement in speech emotion recognition techniques.

In Study [30], Using the IEMOCAP and RAVDESS datasets, presents a Deep Stride Convolutional Neural Network (DSCNN) for Speech Emotion Recognition (SER). With 71.5% and 75.6% accuracy, respectively, DSCNN outperforms the most advanced methods. F1-score, recall, accuracy, and precision are examples of performance measurements. Subsequent research endeavors are intended to investigate alternative deep learning architectures and integrate multi-

modalities such as physiological inputs and facial expressions. The impact of ambient elements including background noise, robust feature selection, and the heterogeneity of emotional expression across cultures are some of the challenges that are explored. Applications in human-robot interaction, emergency call centers, and healthcare highlight how versatile SER is. Considered emotions include neutral, sad, glad, and angry.

In Study [31], Using formant features and phoneme type convergence, using six databases for Speech Emotion Recognition (SER), including Berlin EmoDB and RAVDESS. The methodology demonstrates robustness both inside and between corpora, produces high accuracy, and saves time and computational expenses. Deep learning for feature extraction and categorization may be investigated in future research. Difficulties include getting natural emotional voice data and non-standard emotion annotation. Applications include speech treatment, mental health diagnostics, and human-computer interaction. The emotions that are considered neutral, sadness, happiness, fear, disgust, and anger.

In Study [32], A unique method for speech emotion recognition based on Bi-LSTM with Directional Self-Attention is presented in the paper "Speech emotion recognition using recurrent neural networks with directional self-attention." The suggested approach improves audio signal relevance and captures temporal connections in speech frames. Superior recognition accuracy for emotions like anger, happiness, neutrality, and sorrow is demonstrated by performance evaluation on the IEMOCAP and EMO-DB databases. The method outperforms other compared algorithms, achieving weighted accuracy of 70.5% and unweighted accuracy of 68.5% on IEMOCAP and weighted accuracy of 63.5% and unweighted accuracy of 62.5% on EMO-DB. Future research will examine deep learning architectures and other aspects. Applications for the suggested method can be found in speech therapy, human-computer interaction, and affective computing.

In Study [33], AlexNet a pre-trained image classification network is used to perform real-time Speech Emotion Recognition (SER) using the EMO-DB dataset. With a 70.5% accuracy rate, the authors recommend that future research look at alternative networks like VGG-16 or ResNet and use transfer learning to improve the SER system. There are other obstacles, such as linguistic and cultural differences in emotional expression and the possibility that the acted emotions in the EMO-DB dataset are not realistic. Applications for everything from customer service to mental health highlight the potential benefits of real-time SER systems. Anger, boredom, disgust, anxiety/fear, happiness, sadness, and neutral emotions are among those considered.

In Study [34], Uses feature extraction methods to enhance voice emotion recognition. The study makes use of the English Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS). Feature extraction, selection, and classification are the steps in the process, and algorithms including MFCC, DWT, pitch, energy, and ZCR are used. For feature classification, classifiers like SVM, Decision Tree, and LDA are used; the Decision Tree has the highest accuracy, at 85%. Variability in speaking styles, choosing culturally invariant algorithms, database quality, and the absence of a consistent evaluation procedure are some of the difficulties in speech emotion recognition. Deep learning approaches, classifiers, and sophisticated algorithms are potential areas for future research. Text-to-speech synthesis, multimedia management, education,

entertainment, and medical diagnosis are among the industries that use speech emotion recognition.

In Study [35], In order to improve features, uses three datasets (EMO-DB, RAVDESS, and SAVEE) for speech emotion recognition. A genetic approach based on clustering is introduced. The suggested method achieves 87.5% accuracy and 87.4% unweighted recall in speaker-dependent scenarios, and 85.5% accuracy and 85.3% unweighted recall in speaker-independent scenarios, outperforming the usual genetic algorithm. It is recommended that multimodal data integration, transfer learning, and deep learning approaches be investigated in future research. The lack of a common dataset, the variation in emotional displays between cultures, and the crucial function of feature engineering are among the difficulties. Applications include monitoring mental health, human-computer interaction, and affective computing. The study probably includes a range of emotions, including happiness, sadness, rage, fear, and surprise, even though no emotions are mentioned.

In Study [36], Uses a deep convolutional neural network framework for Speech Emotion Recognition (SER) using the RAVDESS, EMO-DB, and IEMOCAP datasets. Their VGG-16 model outperforms earlier EMO-DB work with an accuracy of 71%. Future research may investigate various feature extraction techniques and recurrent neural networks. Difficulties include the requirement for various datasets, the variety of emotional manifestations, and the absence of universal emotional categories. Applications include speech treatment, mental health diagnostics, and human-computer interaction. Considered emotions include fear, surprise, disgust, fury, happiness, sadness, neutrality, frustration, excitement, and others.

In Study [37] ,Using deep convolutional neural networks (DCNNs) and correlation-based feature selection (CFS), investigates speech emotion recognition (SER) across a variety of datasets, including Emo-DB, SAVEE, RAVDESS, and IEMOCAP. The weighted average recall (WAR) statistic is used in performance evaluation. The Emo-DB dataset with SVM classifier yielded the greatest accuracy of 91.11%. For improved SER performance, it is advised that future research investigate various deep learning architectures, transfer learning strategies, and multimodal data integration. The research uses feature selection techniques and data resampling to solve high-dimensional feature spaces and imbalanced datasets, without specifically mentioning any issues. systems include affective computing, mental health diagnostics, and human-computer interaction, highlighting SER's potential to optimize speech-based systems. The following emotions are taken into consideration: fear (Emo-DB), anger, frustration, happiness, disgust, neutral, surprise, sadness (SAVEE).

In Study [38], Introduces a Two-Layer Fuzzy Multiple Random Forest (TLFMRF) that uses fuzzy C-means clustering to combine personalized and non-personalized data for speech emotion identification. With an average accuracy of 79.08%, TLFMRF beats models like BPNN and RF when tested on datasets provided by the authors and a prior study. Difficulties include differing emotional displays, an absence of standardized databases, and trouble differentiating between comparable feelings. Future research attempts to expand TLFMRF to applications such as human-robot interaction, investigate alternative characteristics, and enhance recognition accuracy for difficult emotions. Applications of TLFMRF reveal how successful it is in emotional social robot

systems; it can track six fundamental emotions in real time and has potential applications in affective computing and human-robot interaction. Anger, fear, joyful, neutral, depressed, and surprised are among the recognized emotions, demonstrating the model's adaptability in capturing a diverse range of emotional states.

In Study [39], A multilingual Speech Emotion Recognition (SER) model based on transformers is presented and tested on datasets, such as BAVED. The approach uses feature fusion and data augmentation to achieve an astounding 95% accuracy on BAVED. Future research will compare models across various databases and extract context related to emotions. Domain-dependent feature extraction and the lack of a definitive machine learning algorithm for multilingual databases are challenges. Applications in healthcare, education, and entertainment demonstrate how versatile the approach is. Although no specific emotions are listed, the assessment datasets suggest a wide variety, which distinguishes this article as a noteworthy development in SER techniques.

Using the EYASE dataset—which was taken from a television show—the study [40] focuses on Egyptian Arabic speech emotion recognition, with emotions like surprise, delight, fury, and neutrality. The study uses classifiers such as k-NN, SVM, and DT, using a framework of prosodic, spectral, and wavelet characteristics. It reports better performance using the combined feature set and k-NN. F1-score, AUC-ROC, recall, accuracy, and precision are examples of performance measures. Difficulties include dialectal differences, limited Arabic databases, and the subjectivity of emotions. Future research proposes expanding the framework to additional Arabic dialects, investigating practical applications, and improving methods. Applications include helping people with communication difficulties and human-computer interaction.

In the study of [41], Used a CNN and a unique filter bank design to improve speech emotion recognition using the IEMOCAP and MELD datasets. With the help of data augmentation, their suggested filter bank outperformed conventional techniques and produced better results on the MELD dataset. The research evaluated kappa values, F-scores, accuracy, and precision. It also noted issues with dataset standardization and ethical considerations in practical applications. To increase accuracy, future directions include utilizing multimodal data, implementing transfer learning, and investigating various filter designs. With potential uses in human-robot interaction and mental health monitoring, the research advances the field of emotion recognition.

In this study of [42], Applies multinomial logistic regression, support vector machines, and recurrent neural network classifiers with Mel-Frequency Cepstral Coefficients and linear predictive coding for feature extraction to the emotion classification of the Spanish and Berlin emotional speech databases. On the Spanish database, performance measures show that RNN with MFCC and MS features gets the maximum accuracy at 70.5%. While tackling issues like noise and speaker variability, future research directions include investigating deep learning approaches and alternate feature extraction methods. The paper highlights the broad influence of voice emotion recognition in improving numerous domains by discussing a variety of applications, from human-robot interaction to education and the commercial sector.

In the study of [43], The paper covers several datasets for speech emotion identification, such as IEMOCAP and VAM. Although it lacks a defined approach, it examines a variety of deep learning techniques, including LSTMs, DNNs, and attention mechanisms. It does not provide precise measurements for the emphasized research, even though it discusses other performance indicators and presents a comparison table. The study emphasizes the difficulties in recognizing emotions in speech and calls for the use of standard datasets, cross-corpus testing, and multimodal integration. Multimodal improvements, model generalization, and dataset standardization are anticipated areas of future investigation. The study, which focuses on deep learning applications, analyzes models for voice emotion identification, including CNNs and LSTMs, highlighting difficulties and outlining potential directions for further investigation.

In the study of [44], the authors combined Radial Basis Function Network with Convolutional Neural Network and Bidirectional Long Short-Term Memory, using IEMOCAP, Emo-DB, and RAVDESS datasets. With a 72.25% accuracy rate, 85.57% accuracy rate, and 77.02% accuracy rate for IEMOCAP, Emo-DB, and RAVDESS, respectively, the suggested framework shows promise in real-time emotion recognition. To improve performance, future research should investigate various feature extraction and normalization strategies. The lack of a common dataset and the difficulty of precisely identifying emotions because of the variety of emotional displays and human speech are among the difficulties. apps for the model include healthcare, education, and entertainment. It may be used to monitor patient emotions, enhance educational materials, and improve user experiences in entertainment apps.

In the study of [45], The paper uses a CNN based on ResNet34 to detect vocal pathology with 98.5% accuracy. F1-score, recall, accuracy, and precision are examples of performance measurements. The impact of various frequency zones will be investigated in future studies, and one of the obstacles is the lack of a uniform dataset. The suggested approach has potential for early voice issue detection in clinical settings.

In the study of [46], Using the EMO-DB dataset, the research assesses a CNN-based speech emotion identification system that outperforms state-of-the-art techniques with an accuracy of 87.5%. Accuracy, precision, recall, F1-score, True-Positive, False-Negative, and True-Negative are some examples of performance measures. Future research might examine more complex deep learning methods like LSTMs and RNNs. Subjectivity in emotional labeling and the lack of a common dataset are challenges. Applications for the proposed system include speech treatment, mental health diagnostics, and human-computer interaction.

In the study of [47], To improve English speech recognition, the research makes use of a dataset of 863 English speech samples and a deep neural network with one input layer, five hidden layers, and one output layer. The algorithm's advantage over conventional approaches is demonstrated by performance measures such as word error rate (WER) and sentence error rate (SER), especially when five frames of input data are used. Convolutional and recurrent neural networks will be the subject of future research in an effort to increase accuracy even further. Dealing with different accents and the requirement for large amounts of training data are challenges. Voice assistants, language translation, and speech-to-text transcription are just a few of the applications.

In the study of [48], Focuses on using a dataset tagged for fear and neutral expressions to detect fear in speech. For feature extraction, the method combines k-nearest neighbor and Deep Neural Network methods with Mel Frequency Cepstral Coefficients (MFCC) and Linear Predictive Coding (LPC). With its remarkable F1 score of 0.95, great accuracy, precision, recall, and recall, the suggested approach has promise for emotion recognition. Subsequent research endeavors will delve into other methodologies and algorithms, proposing potential uses in palliative care to track patients' emotional states. The absence of consistent datasets and precise emotion categorization are issues. Technology finds use in smart homes and commercial settings, as well as in healthcare, specifically in palliative care.

Paper	Year	Dataset	Methodology	Evaluation Metric	Result
[1]	2019	Berlin EmoDB, IEMOCAP	1D CNN LSTM and a 2D CNN LSTM	accuracy, precision, recall, and F1 score.	achieved excellent performance on recognizing speech emotion, especially the 2D CNN LSTM network
[2]	2017	IEMOCAP	deep multi-layered neural network	test accuracy &(UAR)	test accuracy of 63.8% and a test unweighted average recall (UAR) of 60.5%.
[3]	2017	IEMOCAP	ACNN	accuracy, precision, recall, F1-score	The ACNN model, achieved state-of-the-art results
[4]	2022	(IEMOCAP) & Berlin (EMO-DB)	lightweight fully convolutional neural network called LIGHT-SERNET.	(UA), (WA), and F1 score	UA of 94.15%, a WA of 94.21%, and an F1 score of 94.16%
[5]	2020	IEMOCAP	(MFCC), discriminant analysis, and Neural Structured Learning (NSL)	accuracy, recall, precision, and F1-score	testing the features from an audio clip in a testing fold took only 0.006 seconds

[6]	2020	(IEMOCAP) and (EMO-DB)	pre-processing and feature selection and classification	accuracy, precision, recall, and the F1-score	In the IEMOCAP experiments, the highest recognition results obtained for anger were 85%, In the EMO-DB experiments, the highest accuracy results of 96% were obtained for anger.
[7]	2020	IEMOCAP, EMO-DB, and RAVDESS	Radial Basis Function Network (RBFN), Convolutional Neural Network (CNN), and deep Bidirectional Long Short-Term Memory (BiLSTM)	accuracy, precision, recall, F1 score, weighted accuracy, and unweighted accuracy	The model achieves the highest accuracy on Emo-DB dataset with 85.57%
[8]	2018	DEAP	(CRNN)	accuracy, precision, recall, F1 score, accuracy	The paper presents several tables and figures showing the results of the proposed method in recognizing human emotion states based on multi-channel EEG signals.
[9]	2019	DEAP	(CNN)	accuracy, precision, recall, F1 score, accuracy,	Models based on the combined features in time and frequency domain had higher performance on EEG emotion recognition.
[10]	2018	IEMOCAP and SEMAIN	(ICON)	accuracy, precision, recall, F1 score, and accuracy	ICON performs better than the compared models with significant performance increase in emotions (?2.1% acc.) on the IEMOCAP dataset.
[11]	2018	The Geneva Whispered Emotion Corpus (GeWEC)	Semi supervised Autoencoder	Unweighted Average Recall (UAR)	Unweighted Average Recall (UAR) of 66.3% on the IEMOCAP database and 60.5% on the MSP-IMPROV database.
[12]	2020	MSP-Podcast corpus	Cooperative Learning (STL), (MTL)	Concordance Correlation Coefficient (CCC)

[13]	2015	FAU Aibo Emotion Corpus Speech	Cooperative Learning Co-EM, Co-SSVM, and Co-SSL,	Unweighted Average Recall (UAR)	Best UARs obtained with CL algorithms in the four experimental scenarios: 67.2%, 67.2%, 67.6%, 64.9% Baseline performance of the models trained on the whole pool of labeled data: 67.7%, 67.7%, 67.2%, 64.6%
[14]	2019	-----	Curriculum Learning DNNs	F1-score	-----
[15]	2022	(MAS) (IEMOCAP), and FAU-AIBO	(MCIL)	accuracy	MAS: MCIL achieves an accuracy of 70% IEMOCAP: MCIL achieves an accuracy of 63.10% FAU-AIBO: MCIL achieves an accuracy of 47.50%.
[16]	2023	Pitt dataset and DAIC-WOZ dataset	Speech Former	Weighted accuracy (WA), Unweighted accuracy (UA), Weighted average F1 (WF1), Macro average F1 (MF1) Top of Form	IEMOCAP: Improvement on WA: 2.9% - 4.8% Improvement on UA: 4.6% - 5.6% Improvement on WF1: 4.5% - 6.4% MELD: Improvement on WA: 1.0% - 3.3% Improvement on UA: 1.9% - 3.0% Improvement on WF1: 1.2% - 2.4% Pitt: Improvement on WA: 1.9% - 7.3% Improvement on UA: 2.5% - 7.6% Improvement on WF1: 3.4% - 6.3% DAIC-WOZ:

					Improvement on WA: 5.7% - 8.5% Improvement on UA: 2.3% - 2.5%
[17]	2021	IEMOCAP, MELD, Emotion Lines dataset, Friends-ELE dataset	CTNet	(WAA) , (UAA), (WAF1), (UAF1)	-----
[18]	2019	CASIA Chinese Emotional Speech Database, self-built emotion library	Stacked Kernel Sparse Deep Model KNN, Softmax, SVM, sparse representation, and neural network	Accuracy F1-score Top of Form	On the CASIA Chinese Emotional Speech Database: Accuracy: 87.5% F1-score: 0.875 On the self-built emotion library: Accuracy: 86.7% F1-score: 0.865
[19]	2022	LSSED, IEMOCAP	ISNet	(WA), (UA)
[20]	2022	LRS2-BBC and LRS3-TED.	CTC-based model and a Transformer-based model	WER, SER, CER	their model achieves a WER of 14.3% and a SER of 47.7% on LRS2-BBC, and a WER of 16.7% and a SER of 54.2% on LRS3-TED
[21]	2019	Big data of emotion, The database	uses a two-dimensional CNN& three-dimensional CNN	precision, recall, and F1-score.	achieved an accuracy of 92.5%
[22]	2021	the Berlin (Emo-DB) and the (IEMOCAP) database.	C-RNN approach	accuracy rate and loss	An accuracy rate of more than 80%

[23]	2021	IEMOCAP	BLSTM and CNN	accuracy, precision, recall, F1-score, and confusion matrix	proposed approach outperforms the baseline models and achieves high accuracy, precision, recall, and F1-score
[24]	2019	ADRNN model	The model utilizes 3-D Log-Mel spectrum features extracted from raw speech signals and then feeds these features into the model to classify after postprocessing	Unweighted Average Recall (UAR)	achieving 48.8% UAR on the development dataset
[25]	2021	(IEMOCAP)	multi-scale (CNN)	weighted accuracy (WA) and unweighted accuracy (UA)	achieving a WA of 73.5% and UA of 68.9%
[26]	2016	RECOLA	The CRNN consists of convolutional layers, LSTM layers, and fully connected layers	concordance correlation coefficient	the proposed end-to-end CRNN outperforms traditional feature-based methods in terms
[28]	2022	IEMOCAP, RAVDESS, EMODB	Feature extraction, feature selection, classification	Accuracy, F1-score, confusion matrix	The results are compared with three different approaches: i-vectors, x-vectors, and the Interspeech 2010 Paralinguistics Challenge (I2010PC) feature set
[29]	2021	IEMOCAP	1D Dilated CNN architecture, Multi-Learning Trick (MLT) approach	precision, recall, F1_score, weighted and un-weighted accuracies, the confusion matrix	shows the best result for anger, sadness, and neutral classes but exhibits weaker performance for happiness labels
[49]	2020	IEMOCAP, Emo-DB, RAVDESS.	deep BiLSTM	accuracy, F1-score, and BER (Bit Error Rate)	The results show that the proposed approach outperforms baseline methods on all three datasets in terms of accuracy, F1-score, and BER
[30]	2020	IEMOCAP, RAVDESS	DSCNN architecture,	accuracy, precision, recall, and F1-score	IEMOCAP, RAVDESS achieving an accuracy of 71.5% and 75.6%, respectively

[31]	2021	Berlin EmoDB, RAVDESS, IEMOCAP, ShEMO, DEMoS, MSP-Improv	pre-processing, feature extraction, the extracted features are then used as input for classifiers	accuracy (weighted ,unweighted), time,cost, &robustness (within corpus and cross-corpus)	The results show that the proposed approach achieves high accuracy while reducing time and computational costs compared to other state-of-the-art methods
[32]	2021	IEMOCAP, EMO-DB)	Bi-directional Long-Short Term Memory (Bi-LSTM) with Directional Self-Attention mechanism	accuracy	achieved the highest weighted accuracy (WA) of 70.5% and unweighted accuracy (UA) of 68.5% on the IEMOCAP database, outperforming the other compared algorithms. On the EMO-DB database, the proposed approach achieved a WA of 63.5% and UA of 62.5%, which is also better than the other compared algorithms.
[33]	2020	EMO-DB	pre-trained image classification network, AlexNet	accuracy, precision, recall, and F1-score	achieved an accuracy of 70.5% using the pre-trained image classification network, AlexNet, for real-time speech emotion recognition
[34]	2020	RAVDESS	Different classifiers such as SVM, Decision Tree, and LDA	accuracy	The Decision Tree classifier achieved the highest accuracy of 85%
[35]	2021	EMO-DB, RAVDESS, and SAVEE	feature optimization approach, GA, DBSCAN, and SVM	unweighted recall (UAR) and accuracy	achieving an accuracy of 87.5% and UAR of 87.4% for the speaker-dependent scenario, and an accuracy of 85.5% and UAR of 85.3% for the speaker-independent scenario.
[36]	2020	RAVDESS, EMO-DB, and IEMOCAP	feature extraction followed by the baseline deep learning model	accuracy, precision, recall, and F1-score	accuracy of 71% using Convolutional Neural Network VGG-16 as a classifier

[37]	2020	Emo-DB, SAVEE, RAVDESS, IEMOCAP	CFS, DCNN	the weighted average recall (WAR) metric	The highest accuracy of 91.11% was obtained with the Emo-DB dataset using the SVM classifier. The SVM achieved accuracies of 79.08%, 80.97%, and the MLP achieved 80.00% with the SAVEE, RAVDESS, and IEMOCAP datasets, respectively
[38]	2020	CASIA corpus, Berlin EmoDB	TLFMRF, BPNN, RF	accuracy	The average accuracies for TLFMRF, BPNN, and RF are $79.08 \pm 0.35\%$, $70.84 \pm 0.55\%$, and $79.33 \pm 0.49\%$, respectively
[39]	2022	BAVED, EMO-DB, SAVEE, and EMOVO	preprocessing the audio data, extracting features, and training the transformer-based model using a multi-task learning approach	accuracy, precision, recall, and F1-score	achieving an average accuracy of 95% for the BAVED dataset
[40]	2020	EYASE	prosodic, spectral, and wavelet features for Egyptian Arabic speech emotion recognition	accuracy, precision, recall, F1-score, AUC-ROC	The reported accuracy rates for different emotion recognition tasks ranged from 70% to 90%

[41]	2020	IEMOCAP MELD	convolutional neural network (CNN)	accuracy, precision, F-score, kappa	Accuracy of 70.40%, precision of 72.88%, F-score of 70.11%, Kappa of 60.49%
[42]	2020	Berlin	(MLR), Support Vector Machine (SVM), Recurrent Neural Network (RNN)	accuracy, precision, recall, F1-score	accuracy for RNN with MFCC and MS 70.5%
[43]	2021	IEMOCAP VAM	DNNs, LSTMs	accuracy, precision, recall,	accuracy range from around 50% to 99%

[44]	2020	IEMOCAP, Emo-DB, RAVDESS	Radial Basis Function Network (RBFN), (CNN) & (BiLSTM)	accuracy	accuracy for IEMOCAP 72.25%, accuracy for EMO-DB 85.57%, accuracy for RAVDESS 77.02%
[45]	2020	SVD	Convolutional Neural Network (CNN)	accuracy, precision, recall & F1-score	accuracy of 98.5%
[46]	2020	EMO-DBEMO-DB	convolutional neural network (CNN)	accuracy, precision, recall, F1-score, confusion matrix	accuracy of 87.5%
[47]	2020	English speech dataset consisting of 863 speech samples	deep neural networks	word error rate (WER) sentence error rate (SER)	the deep neural network increases, the five frames are currently the optimal values.
[48]	2018	Berlin	Deep Neural Network k-nearest neighbor algorithms	accuracy, precision, recall, F1-score	F1 score of 0.95

Chapter 3

System Analysis & Design

System Analysis and Design:

3.1 System Analysis

3.1.1 Functional Requirements

Functional requirements outline specific actions and behaviors that the system must perform to meet user needs. Here are the functional requirements for both the user and employee sides of the system:

For the Employee:

1. **User Authentication:**
 - Employees must be able to securely log into the system using unique credentials.
2. **News Management:**
 - Employees should have the ability to upload, edit, and delete news articles or updates that users can view.
3. **Recording Analysis:**
 - Employees must be able to receive, play, and submit user recordings to the backend for analysis using the deep learning model. The system should display the predicted emojis returned by the model.
4. **Complaint Handling:**
 - Employees should receive complaints from users, view them, and respond with appropriate solutions or actions.
5. **User Data Retrieval:**
 - Employees should have the capability to retrieve relevant data related to specific users registered in the system, such as recording history, complaint logs, or account details.

For the User:

1. **Account Creation and Authentication:**
 - Users must be able to create new accounts securely and log into the system using unique credentials.
2. **Recording Submission:**
 - Users should have the ability to record their voice, submit the recording to the system, and optionally include a message or context.
3. **Complaint Submission:**
 - Users should be able to submit complaints detailing their needs, issues, or feedback.
4. **News Viewing:**
 - Users should be able to view news articles or updates uploaded by employees.

3.1.2 Non-Functional Requirements

Non-functional requirements define the qualities and characteristics of the system, including performance, usability, security, and reliability. Here are the non-functional requirements for your system:

1. **Performance:**
 - The system should respond to user interactions promptly, with minimal latency in uploading, analyzing recordings, or displaying news updates.
2. **Usability:**
 - The user interface should be intuitive and user-friendly, catering to users with varying levels of technical proficiency.
3. **Security:**
 - User authentication and data transmission should be encrypted to ensure confidentiality and prevent unauthorized access to sensitive information.
4. **Reliability:**
 - The system should be available and operational whenever users or employees need to access it, with minimal downtime or service interruptions.
5. **Scalability:**
 - The system should be able to handle an increasing number of users, recordings, and news articles without significant degradation in performance or functionality.
6. **Compatibility:**
 - The system should be compatible with a variety of devices and web browsers to ensure accessibility for users and employees across different platforms.
7. **Maintenance:**
 - The system should be easy to maintain and update, with clear documentation and [support](#) resources available for administrators and developers.

3.2 Software Design

3.2.1 Use Case Diagram:

Use case diagrams illustrate the interactions between users (actors) and the system to achieve specific goals. They visually represent the functionality offered by the system from the user's perspective.

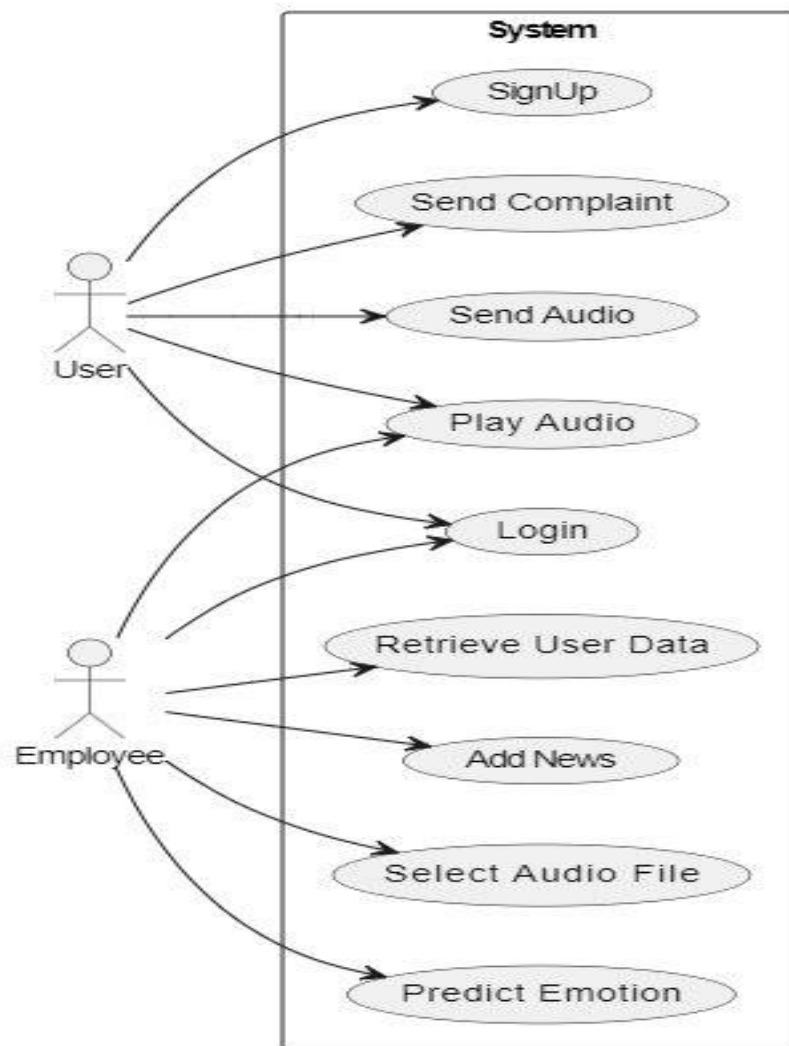


Figure 5

3.2.2 Class Diagram

A class diagram is a type of UML diagram that illustrates the structure of a system by displaying the classes, their attributes, methods, and the relationships among them. In the context of emotion recognition, a class diagram can be utilized to depict the classes involved in the system, including the main components, their properties, and their interactions.

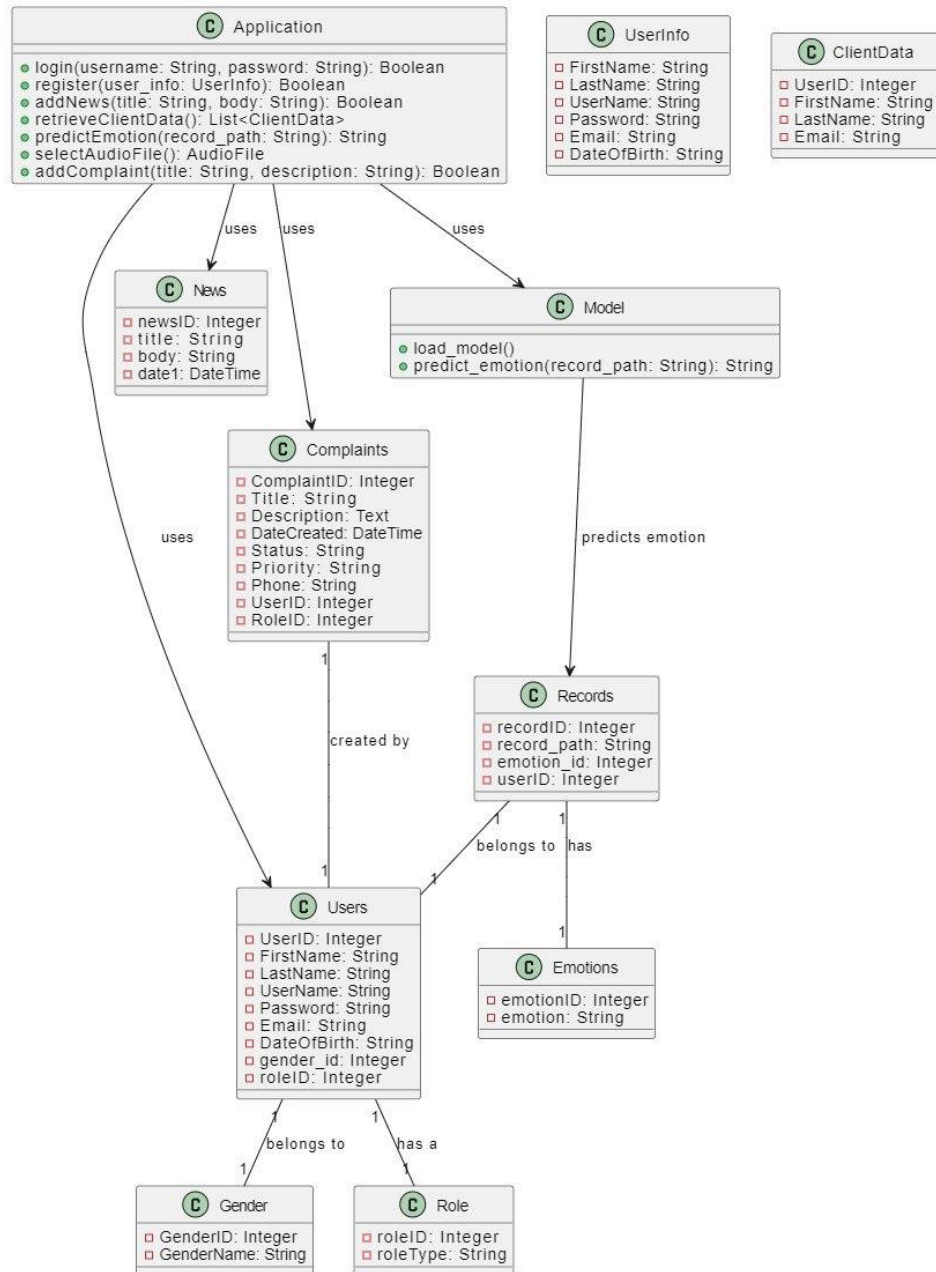


Figure 6

3.2.3 Sequence Diagram

It depicts the flow of messages or interactions between objects or components in a system over time. It displays the sequence of actions and messages exchanged among the various elements of a system to achieve a specific behavior or accomplish a particular task.

For the Employee:

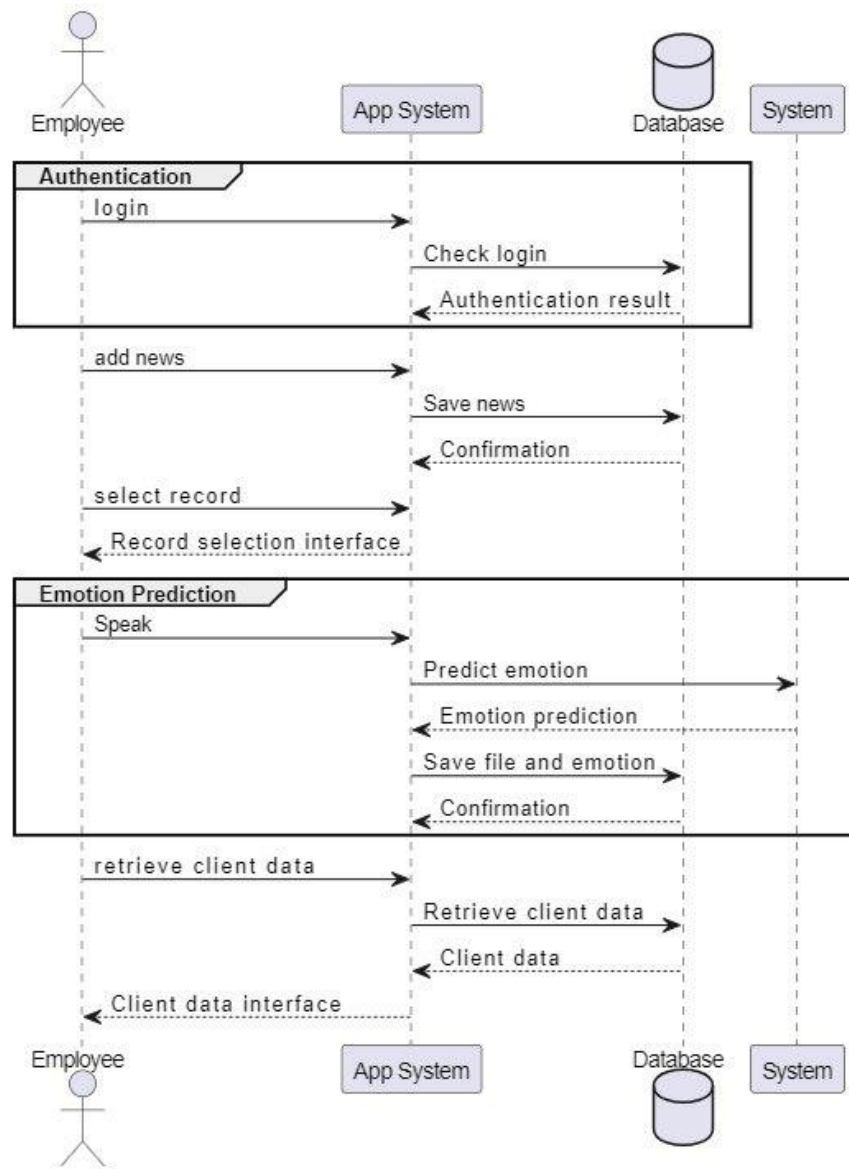
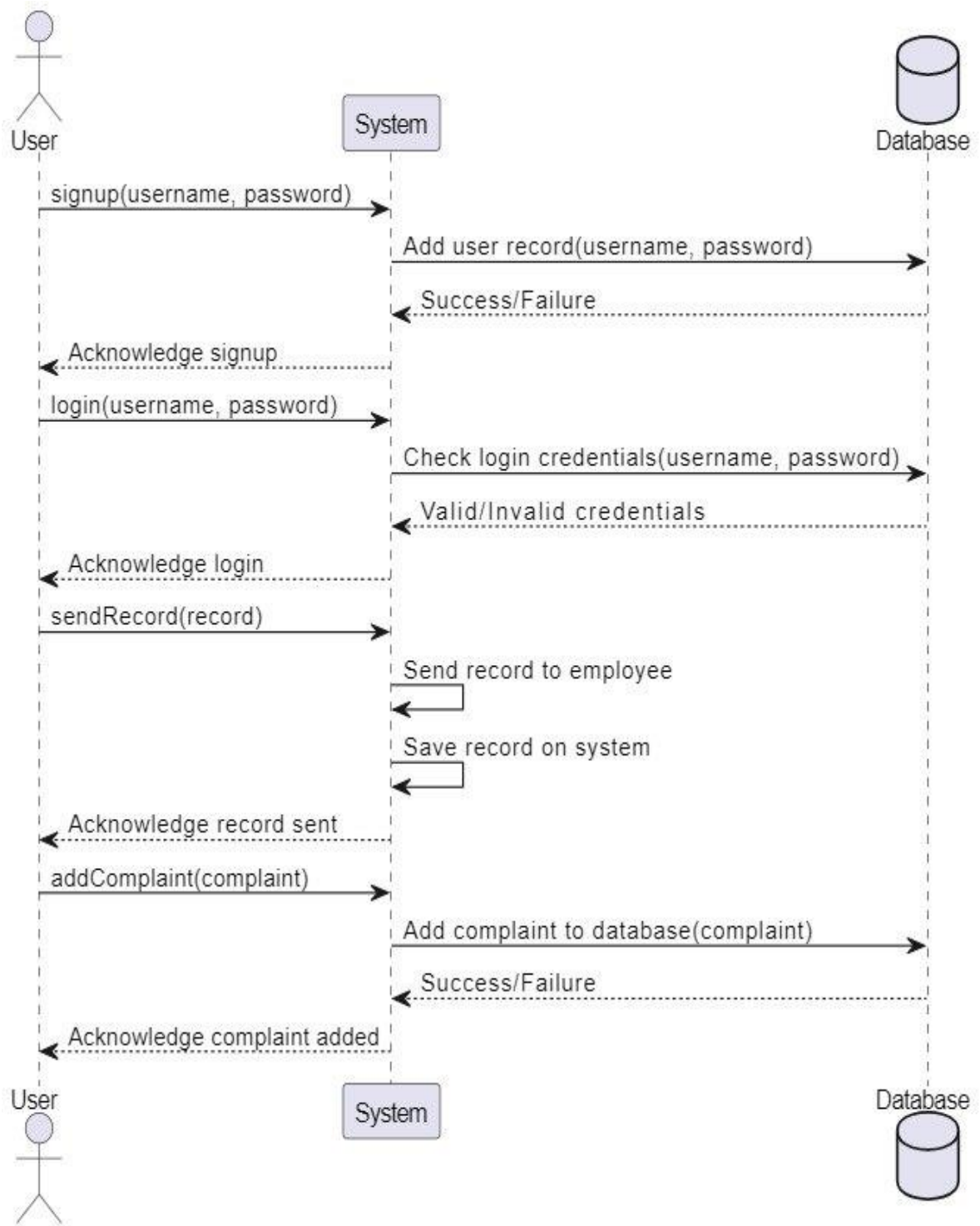


Figure 7

For the User:

3.2.4 Activity Diagram

It is a tool used to represent the sequence of activities and the flow of control within a system, focusing on the dynamic aspects of the system.

For the Employee:

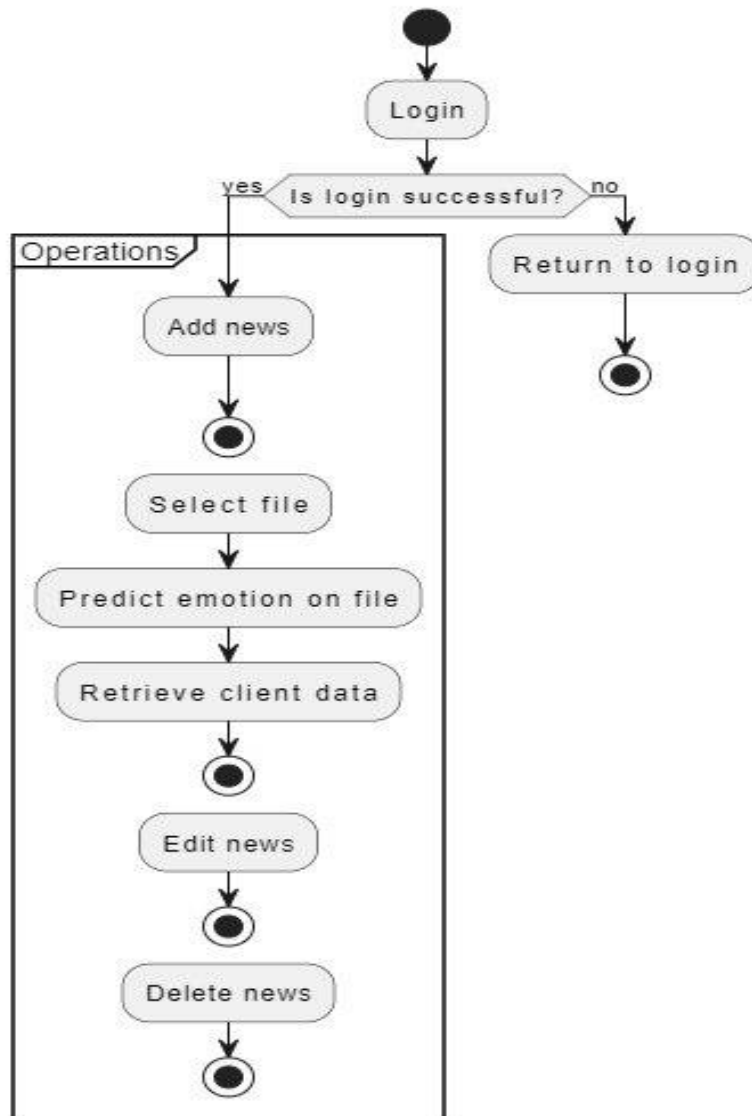
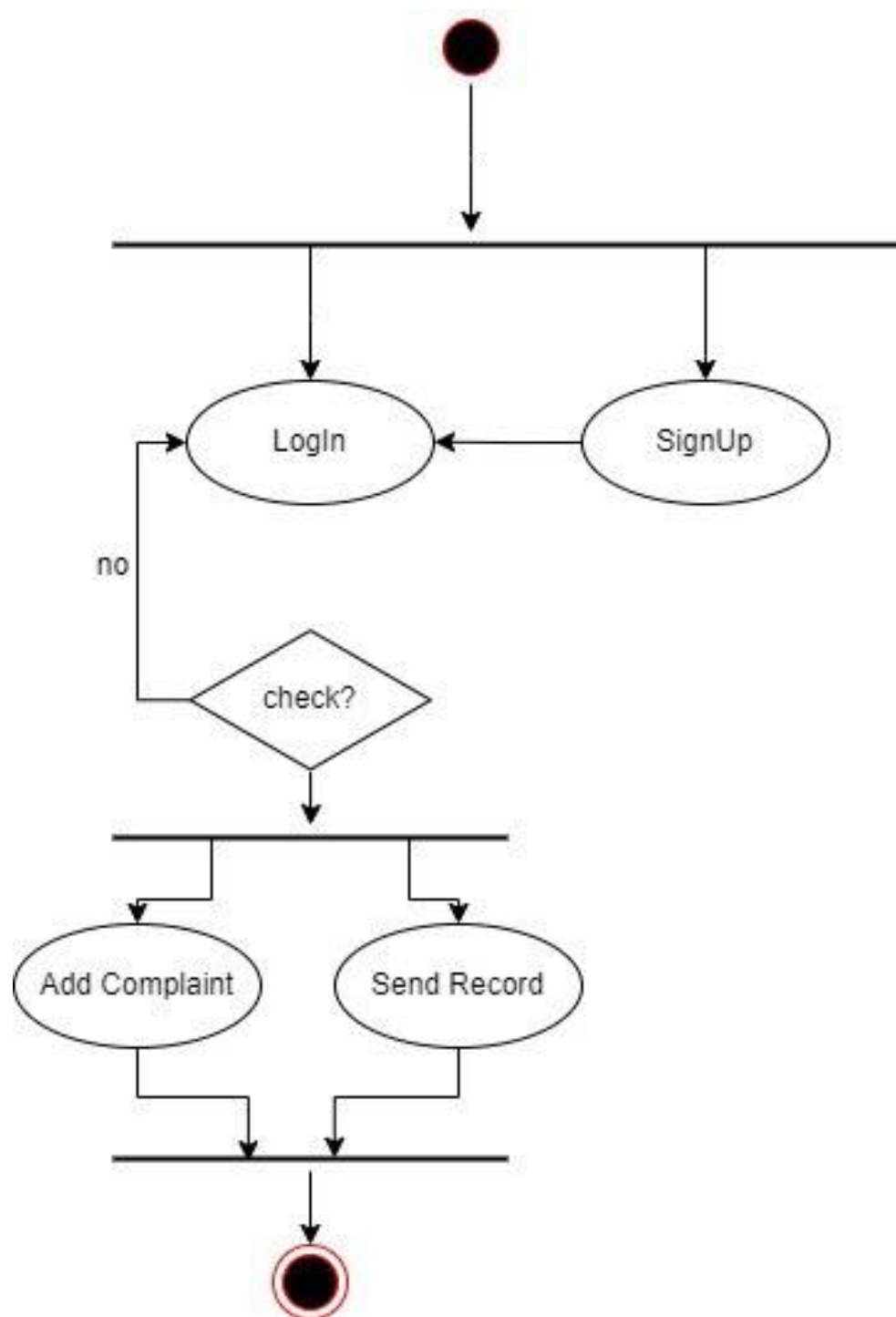


Figure 8

For the User:



Chapter 4

Implementation

Implementation & Results:

4.1 System Requirements

4.1.1 Model

- **Python:** Python is a widely used high-level programming language known for its simplicity and extensive libraries, making it ideal for speech emotion recognition projects. Key advantages include:
 - **Easy to Learn:** Python's syntax is clear and intuitive.
 - **Rich Library Ecosystem:** Extensive libraries for speech processing, machine learning, and data analysis.
 - **Large Community:** Active community support and abundant resources.
 - **Cross-Platform Compatibility:** Runs on various operating systems.
 - **Rapid Prototyping:** Quickly develop and test new ideas.
- **Librosa:** A Python library for audio and music analysis. It provides functionalities for loading, processing, and analyzing audio data, including tools for feature extraction, time-frequency analysis, and signal processing.
- **NumPy:** A Python library for scientific computing offering tools for working with arrays and matrices, essential for data analysis and processing tasks in speech emotion recognition projects.
- **Matplotlib:** A Python library used for creating visualizations such as line plots, bar graphs, scatter plots, and histograms. It is crucial for visualizing data and results in scientific research and data analysis.
- **Pandas:** An open-source Python library built on NumPy, providing tools for data manipulation and analysis. It is particularly useful for working with structured data in tabular form.
- **Scikit-learn (sklearn):** An open-source Python library for machine learning offering a broad array of tools and algorithms for tasks like classification, regression, clustering, and model selection. It integrates seamlessly with other scientific computing libraries.
- **IPython.display:** A module within IPython that provides functions for displaying rich media objects such as HTML, images, audio, and video directly within the IPython environment.
- **Seaborn:** A Python data visualization library built on top of Matplotlib. It provides a high-level interface for creating appealing and informative statistical graphics, useful for visualizing data distributions and relationships between variables.
- **TensorFlow:** An open-source library developed by Google Brain for numerical computation and machine learning. It offers a comprehensive ecosystem for building and deploying machine learning models, supporting deep neural networks and distributed computing.
- **Keras:** A high-level neural networks API written in Python, designed to be user-friendly, modular, and easy to extend. It serves as an interface for building, training, and deploying deep learning models and supports multiple backends, including TensorFlow.

4.1.2 Backend

- **Flask:** A lightweight and popular web framework for Python, ideal for quickly building web applications. It provides routing, templating, and easy handling of HTTP requests and responses, making it suitable for small to medium-sized projects.
- **MySQL:** A widely used open-source relational database management system (RDBMS) that offers efficient storage, management, and retrieval of structured data. It is commonly utilized for web applications to provide reliable data storage and access.
- **RESTAPI:** is a framework for building web services that use standard HTTP methods to interact with resources via unique URLs. It ensures each request is stateless and self-contained, commonly using JSON or XML for data. REST APIs are scalable and enable efficient client-server interactions.
- **SQLAlchemy:** is a Python library for working with databases using an ORM approach. It supports various database systems and provides a consistent API. With SQLAlchemy, developers can interact with databases using Python objects and methods, simplifying database operations.

4.1.3 Android App Development

- **Java:** The primary programming language used for Android app development. It offers a robust and flexible platform for creating visually appealing, interactive, and platform-specific mobile applications with a comprehensive set of APIs and tools for building feature-rich and scalable apps

4.2 Model

The core functionality of our mobile application is driven by a sophisticated deep learning model specifically designed for speech emotion recognition. This model employs advanced neural network architectures, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), to process and analyze voice recordings, detecting emotional cues and translating them into corresponding emojis. The primary goal of this model is to enhance user interaction by providing real-time emotion detection with high accuracy

4.2.1 Datasets Description

The model was trained and validated using several well-known datasets widely used in the field of speech emotion recognition. Here are the detailed descriptions of each dataset:

- **RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song):**
 - **Description:** Contains 7,356 audio and video files from 24 professional actors each expressing emotions with varying intensity.
 - **Filename Identifiers:** The filenames encode multiple pieces of metadata, such as modality, vocal channel, emotion, emotional intensity, statement, repetition, and actor.
 - **Emotions:** Neutral, calm, happy, sad, angry, fearful, disgust, and surprised.

- **Usage:** Used for training and validating the model to ensure a wide range of emotional expressions are covered.



- **CREMA-D (Crowd-sourced Emotional Multimodal)**

- **Description:** Consists of 7,442 original clips from various ethnicities, expressing one of six different emotions at four different emotion levels.
- **Emotions:** Anger, disgust, fear, happy, neutral, and sad.
- **Usage:** Provides diverse vocal expressions to improve the model's generalization capability.



- **TESS (Toronto Emotional Speech Set):**

- **Description:** Contains recordings of 200 target words spoken within the carrier phrase "Say the word _" by two actresses aged 26 and 64 years. Each target word was recorded portraying seven different emotions.
- **Emotions:** Anger, disgust, fear, happiness, pleasant surprise, sadness, and neutral.
- **Usage:** Augments the training dataset with additional examples of emotional speech to ensure the model can recognize subtle differences in vocal tone.



- **SAVEE (Surrey Audio-Visual Expressed Emotion):**

- **Description:** Recorded from four native English male speakers, aged 27 to 31 years, expressing seven different emotions.
- **Emotions:** Anger, disgust, fear, happiness, sadness, surprise, and neutral.
- **Usage:** Provides a comprehensive set of recordings to cover various emotions and ensure robust model training.



After integrating these four datasets, the final distribution of emotion samples is as follows:

- Disgust: 1,923 samples
- Fear: 1,923 samples
- Sad: 1,923 samples
- Happy: 1,923 samples
- Angry: 1,923 samples
- Neutral: 1,895 samples
- Surprise: 652 samples

This distribution highlights the model's exposure to a balanced variety of emotional expressions, although some emotions like "surprise" have fewer samples. This comprehensive integration ensures that the model can generalize well across different emotional states.

Integration

In [11]:

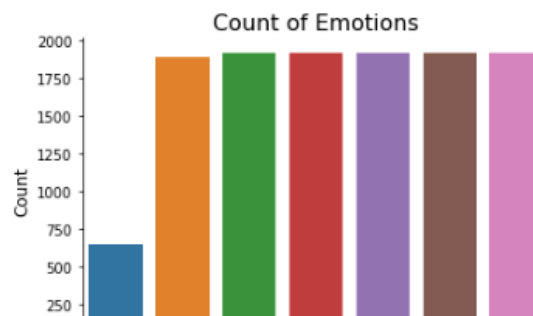
```
# creating Dataframe using all the 4 dataframes we created so far.
data_path = pd.concat([ravdess_df, Crema_df, Tess_df, Savee_df], axis = 0)
data_path.to_csv("data_path.csv", index=False)
data_path.head()
```

Out[11]:

	Emotions	Path
0	surprise	/kaggle/input/ravdess-emotional-speech-audio/a...
1	neutral	/kaggle/input/ravdess-emotional-speech-audio/a...
2	disgust	/kaggle/input/ravdess-emotional-speech-audio/a...
3	disgust	/kaggle/input/ravdess-emotional-speech-audio/a...
4	neutral	/kaggle/input/ravdess-emotional-speech-audio/a...

4.2.2 Data Visualisation and Exploration

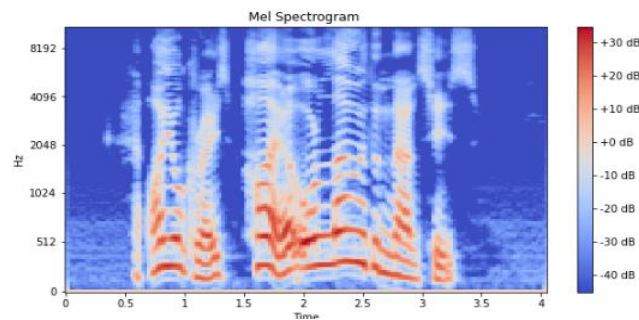
1- Count of Emotions



2- Log Mel Spectrogram

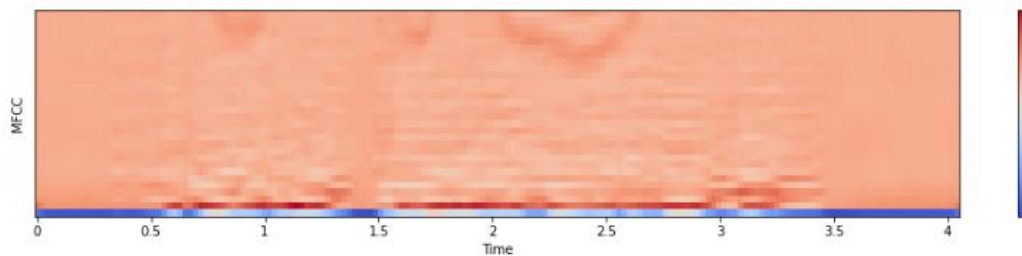
A log Mel spectrogram is a visual representation of sound using the Mel scale for frequency and a logarithmic scale for amplitude. It is widely used in audio and speech processing to highlight important frequency features.

Out[16]: <matplotlib.colorbar.Colorbar at 0x7d69a9104550>



3- MFCC

Mel-Frequency Cepstral Coefficients (MFCCs) are features that represent the power spectrum of sound on the Mel scale. They are commonly used in speech and audio recognition for their ability to capture essential sound characteristics

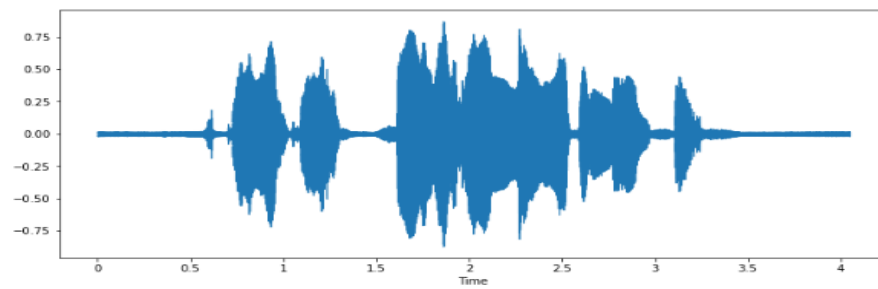
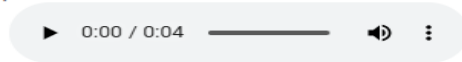


4.2.3 Data Augmentation

Data augmentation involves techniques used to increase the diversity and quantity of data without actually collecting new data. In audio processing, this can include adding noise, changing pitch, or time-stretching audio signals. These methods help improve the robustness and generalization of machine learning models by providing them with more varied training examples.

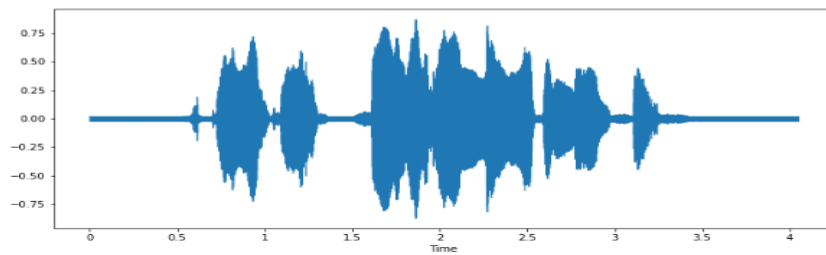
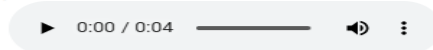
1- Normal Audio

Out[19]:



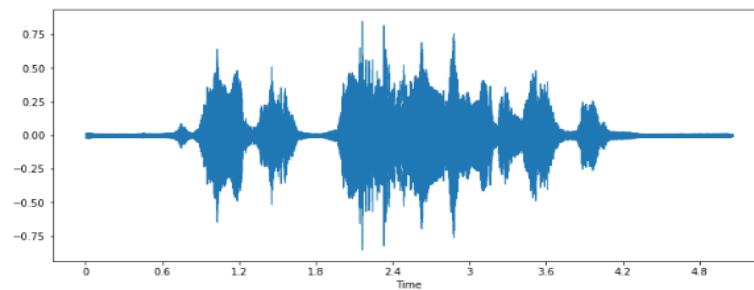
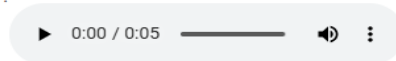
2- Audio with Noise

Out[20]:



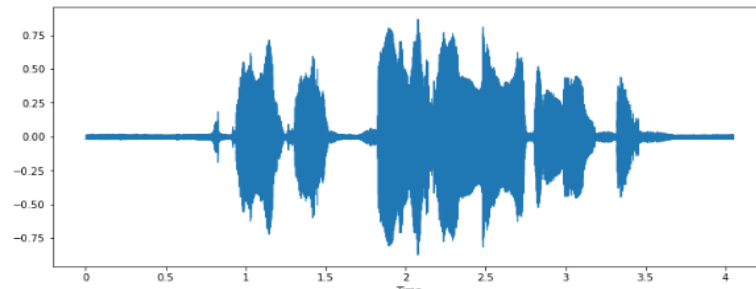
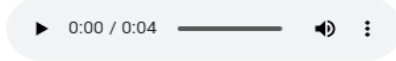
3- Stretched Audio

Out[21]:



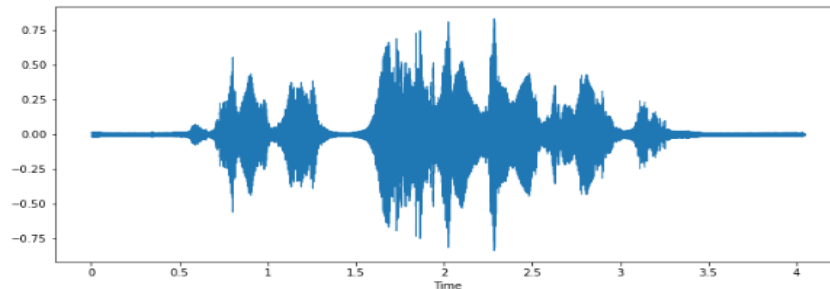
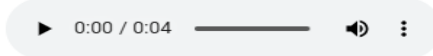
4- Shifted Audio

Out[22]:



5- Audio with Pitch

Out[23]:



4.2.4 Feature Extraction

Feature extraction is the process of transforming raw data into a set of measurable attributes that can be used for analysis. In audio processing, it involves converting sound waves into features like MFCCs or spectrograms, which highlight essential characteristics of the audio signal for tasks such as speech recognition or emotion detection.

Time spent:

12000 audio has been processed

12162it [1:33:14, 2.17it/s]

Done

Time: 5594.533975138

Saving features:

Out[29]:

	0	1	2	3	4	5	6
0	0.331543	0.471680	0.564941	0.452148	0.374512	0.296875	0.265137
1	0.238770	0.361816	0.478516	0.473633	0.485352	0.476074	0.472656
2	0.299805	0.419922	0.525879	0.459473	0.378418	0.326172	0.282227
3	0.252930	0.382812	0.497559	0.497070	0.487793	0.472656	0.482422
4	0.400879	0.591309	0.783203	0.777832	0.771973	0.777832	0.771973

4.2.5 Data Preparation

Before building a machine learning model, we must split our data into training and testing sets. After splitting the data, it is necessary to reshape it using the reshape function, enabling the dimensions of an array to be changed without altering its data content. After scaling the data using the StandardScaler function, the features are standardized to have a mean of 0 and a standard deviation of 1. This process helps to enhance the performance of the model.

```
In [37]: from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(X, Y, random_state=42, test_size=0.2, shuffle=True)
x_train.shape, y_train.shape, x_test.shape, y_test.shape

Out[37]: ((38918, 2376), (38918, 7), (9730, 2376), (9730, 7))

In [38]: #reshape for lstm
x_train = x_train.reshape(x_train.shape[0], x_train.shape[1], 1)
x_test = x_test.reshape(x_test.shape[0], x_test.shape[1], 1)

In [39]: # scaling our data with sklearn's Standard scaler
scaler = StandardScaler()
x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)
x_train.shape, y_train.shape, x_test.shape, y_test.shape

Out[39]: ((38918, 2376), (38918, 7), (9730, 2376), (9730, 7))
```


4.2.6 CNN Model

```
In [47]: import tensorflow.keras.layers as L

model = tf.keras.Sequential([
    L.Conv1D(512, kernel_size=5, strides=1, padding='same', activation='relu', input_shape=(X_train.
    shape[1],1)),
    L.BatchNormalization(),
    L.MaxPool1D(pool_size=5, strides=2, padding='same'),

    L.Conv1D(512, kernel_size=5, strides=1, padding='same', activation='relu'),
    L.BatchNormalization(),
    L.MaxPool1D(pool_size=5, strides=2, padding='same'),
    Dropout(0.2), # Add dropout layer after the second max pooling layer

    L.Conv1D(256, kernel_size=5, strides=1, padding='same', activation='relu'),
    L.BatchNormalization(),
    L.MaxPool1D(pool_size=5, strides=2, padding='same'),

    L.Conv1D(256, kernel_size=3, strides=1, padding='same', activation='relu'),
    L.BatchNormalization(),
    L.MaxPool1D(pool_size=5, strides=2, padding='same'),
    Dropout(0.2), # Add dropout layer after the fourth max pooling layer

    L.Conv1D(128, kernel_size=3, strides=1, padding='same', activation='relu'),
    L.BatchNormalization(),
    L.MaxPool1D(pool_size=3, strides=2, padding='same'),
    Dropout(0.2), # Add dropout layer after the fifth max pooling layer

    L.Flatten(),
    L.Dense(512, activation='relu'),
    L.BatchNormalization(),
    L.Dense(7, activation='softmax')
])
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics='accuracy')
model.summary()
```

Fit model:

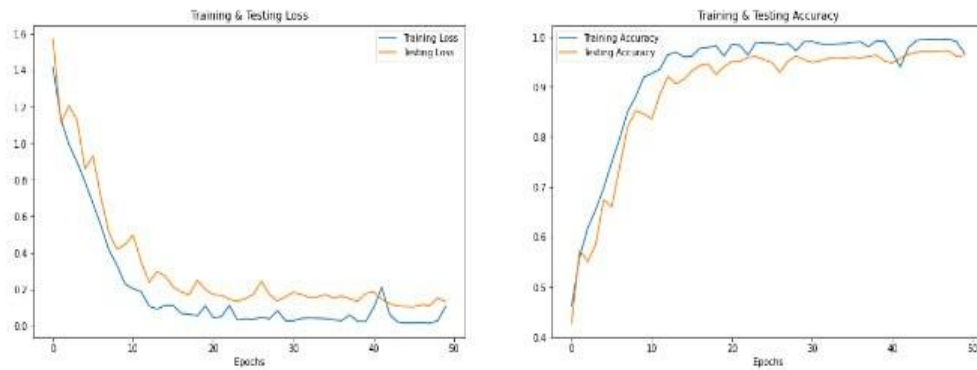
```
In [48]: history=model.fit(x_traincnn, y_train, epochs=50, validation_data=(x_testcnn, y_test), batch_size=
64, callbacks=[early_stop, lr_reduction, model_checkpoint])
```

4.2.7 Experimental model

Accuracy on test data

```
305/305 [=====] - 7s 24ms/step - loss: 0.1318 - accuracy:
0.9634
Accuracy of our model on test data : 96.34121060371399 %
```

Plotting accuracy & loss

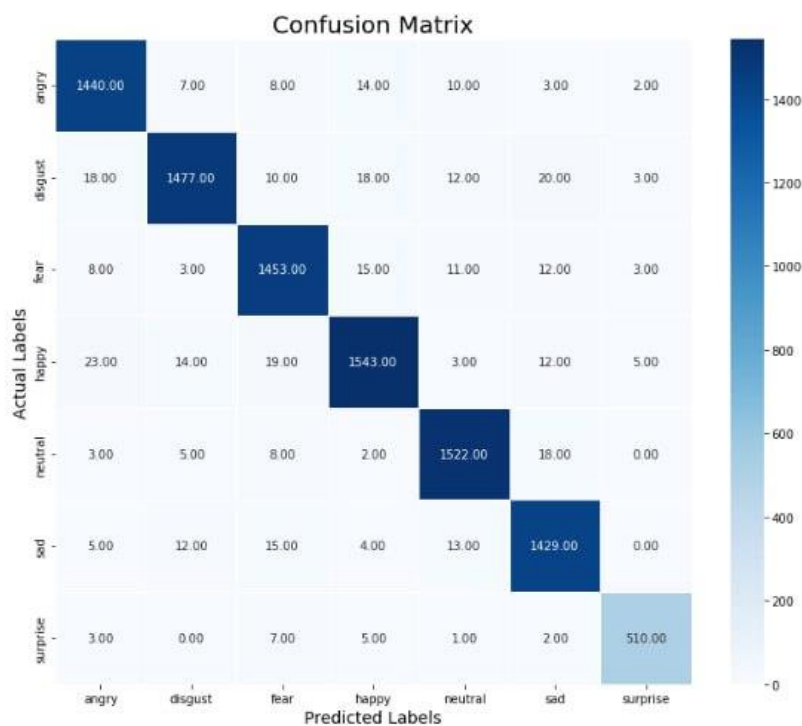


predicting on test data

Out[50]:

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

Confusion Matrix



Classification Report

	precision	recall	f1-score	support
angry	0.96	0.97	0.97	1484
disgust	0.97	0.95	0.96	1558
fear	0.96	0.97	0.96	1505
happy	0.96	0.95	0.96	1619
neutral	0.97	0.98	0.97	1558
sad	0.96	0.97	0.96	1478
surprise	0.98	0.97	0.97	528
accuracy			0.96	9730
macro avg	0.96	0.96	0.96	9730
weighted avg	0.96	0.96	0.96	9730

Saving Model

```
from tensorflow.keras.models import Sequential, model_from_json
model_json = model.to_json()
with open("CNN_model.json", "w") as json_file:
    json_file.write(model_json)
# serialize weights to HDF5
model.save_weights("CNN_model_weights.h5")
print("Saved model to disk")
```

Saved model to disk

Loaded model

In [60]:

```
from tensorflow.keras.models import Sequential, model_from_json
json_file = open('/kaggle/working/CNN_model.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
loaded_model = model_from_json(loaded_model_json)
# load weights into new model
loaded_model.load_weights("/kaggle/working/best_model1_weights.h5")
print("Loaded model from disk")
```

Loaded model from disk

Accuracy for loaded model

In [61]:

```
loaded_model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
score = loaded_model.evaluate(x_testcnn, y_test)
print("%s: %.2f%%" % (loaded_model.metrics_names[1], score[1]*100))
```

```
305/305 [=====] - 8s 24ms/step - loss: 0.1117 - accuracy:
0.9725
accuracy: 97.25%
```

Saving and Loading our Standard Scaler and encoder

```
In [62]: import pickle

# Saving scaler
with open('scaler2.pickle', 'wb') as f:
    pickle.dump(scaler, f)

# Loading scaler
with open('scaler2.pickle', 'rb') as f:
    scaler2 = pickle.load(f)

# Saving encoder
with open('encoder2.pickle', 'wb') as f:
    pickle.dump(encoder, f)

# Loading encoder
with open('encoder2.pickle', 'rb') as f:
    encoder2 = pickle.load(f)

print("Done")
```

Done

Test script

That can predict new record

Predict Neutral:

```
In [70]: prediction("/kaggle/input/ravdess-emotional-speech-audio/Actor_02/03-01-01-01-01-01-02.wav")

neutral
```

Predict Angry:

```
In [72]: prediction("/kaggle/input/ravdess-emotional-speech-audio/Actor_01/03-01-05-01-02-02-01.wav")

angry
```

Predict Sad:

```
In [73]: prediction("/kaggle/input/ravdess-emotional-speech-audio/Actor_21/03-01-04-02-02-02-21.wav")

sad
```

Predict Surprise:

```
In [75]: prediction("/kaggle/input/ravdess-emotional-speech-audio/Actor_01/03-01-08-01-01-01-01.wav")

surprise
```

Predict Disgust:

```
In [76]: prediction("/kaggle/input/ravdess-emotional-speech-audio/Actor_01/03-01-07-01-01-01-01.wav")

disgust
```

4.3 Backend

4.3.1 Employee Functions & APIs

- **Login**
 - **Endpoint:** /users/Login
 - **Method:** POST
 - **Description:** Employee can log into the system using their username and password. The system returns a message indicating the success or failure of the login attempt along with the user's role ID and user ID if successful.

```
@users_bp.route('/Login', methods=['POST'])
def login():
    data = request.get_json()
    username = data['UserName']
    password = data['Password']

    if not username or not password:
        return abort(400, "Username and password are required.")

    user = Users.query.filter_by(UserName=username).first()

    if user and check_password_hash(user.Password, password):
        return jsonify({'message': 'Login successful.', 'role_id': user.roleID, 'user_id': user.UserID})
    else:
        return jsonify({'message': 'Invalid username or password.'}), 401
```

- **Get All Complaints (get_complaints)**

- **Endpoint:** /complaints/getAll/user_id
- **Method:** GET
- **Description:** Retrieves all complaints associated with a specific user. Returns a list of complaints with details such as title, description, date created, status, priority, role ID, phone number, and user information.

```
@Complaints_bp.route('/getAll/<int:user_id>', methods=['GET'])
def get_complaints(user_id):
    user = Users.query.get_or_404(user_id)
    complaints = db.session.query(Complaints, Users).join(Users, Users.UserID == Complaints.UserID).filter(Complaints.UserID == user_id).all()

    complaints_list = [
        {
            'complaint_id': complaint.ComplaintID,
            'title': complaint.Title,
            'description': complaint.Description,
            'date_created': complaint.DateCreated,
            'status': complaint.Status,
            'priority': complaint.Priority,
            'phone': complaint.Phone,
            'first_name': user.FirstName,
            'last_name': user.LastName
        } for complaint, user in complaints
    ]

    return jsonify(complaints_list)
```

- **Upload News (insert_news)**

- **Endpoint:** /News/insert_news/<userID>
- **Method:** POST
- **Description:** Allows employees to upload news articles for users to view. The employee provides the title, content, and optionally an image.

```
from app.database import db
from app.models import News
from flask import Blueprint, request, jsonify, abort

News_bp = Blueprint('News', __name__, url_prefix='/News')

# Define a route for inserting news for Employee , UserID here is the employee
@News_bp.route('/insert_news/<int:UserID>', methods=['POST'])
def insert_news(UserID):
    data = request.get_json()

    if 'title' not in data or 'body' not in data:
        return jsonify({'error': 'Title and body are required'}), 400

    news_item = News(
        title=data['title'],
        body=data['body'],
        UserID=UserID,
    )
    db.session.add(news_item)
    db.session.commit()

    return jsonify({'message': 'News inserted successfully', 'newsID': news_item.newsID})
```


- **Process User Recording (process_recording)**

- **Endpoint:** /Deploy/predict-emotion/<userID>
- **Method:** POST
- **Description:** Employees can send user recordings to the backend for analysis. The system processes the audio file and returns the predicted emotions as emojis. The system also saves the predicted file and its emotion on the database for further use.

```
#Save Predicted file and emotion for employee
def PredictedRecordSave(file,prediction_result,user_id):
    emotion = Emotions.query.filter_by(emotion=prediction_result).first()
    # Extract the filename and ensure its safety
    filename = secure_filename(file.filename)
    if not emotion:
        # If it doesn't exist, create a new emotion record
        emotion = Emotions(emotion=prediction_result)
        db.session.add(emotion)
        db.session.commit()

    # Save the record data and link it to the emotion
    new_record = Records(record_path=filename, emotion_id=emotion.emotionID,userID=user_id)
    db.session.add(new_record)
    db.session.commit()

    return new_record
```

```
def prediction(path):
    res = get_predict_feat(path)
    predictions = loaded_model.predict(res)
    y_pred = encoder2.inverse_transform(predictions)
    return y_pred[0][0]
```

```
# Prediction endpoint for employee , user_id here is the employee
@deploy_bp.route('/predict-emotion/<int:user_id>',methods= ['POST'])
def predict(user_id):
    print("Predict endpoint accessed")
    file = request.files['file']

    if not file:
        return jsonify({'error': 'Empty file'})

    prediction_result = prediction(file)
    #print("000000000000000000000000",file)
    new_record =PredictedRecordSave(file, prediction_result,user_id)

    return jsonify(
        'message': 'File and prediction result saved successfully.',
        #'record_id': new_record.recordID,
        #'file_path': new_record.record_path,
        'emotion': new_record.emotion.emotion,
        #'userID':user_id
```


- **Retrieve all users (all-users)**
 - **Endpoint:** /all-users
 - **Method:** GET
 - **Description:** Employee retrieve all users usernames and IDs for further use.

```
@users_bp.route('/all-users', methods=['GET'])
def get_all_users():
    users = Users.query.with_entities(Users.UserID, Users.UserName).filter_by(roleID=2).all()
    users_list = [{'UserID': user.UserID, 'UserName': user.UserName} for user in users]
    return jsonify(users_list)
```

4.2.2 User Functions

- **Create New Account (register)**
 - **Endpoint:** /users/SignUp
 - **Method:** POST
 - **Description:** Allows users to create a new account in the system. Users provide their first name, last name, username, password, email, date of birth, gender, ID, and role ID (default is 2 for client and 1 for employee).

```
users_bp = Blueprint('users', __name__, url_prefix='/users')

@users_bp.route('/SignUp', methods=['POST'])
def register():
    data = request.get_json()

    hashed_password = generate_password_hash(data['Password'])

    new_user = Users(
        FirstName = data['FirstName'],
        LastName = data['LastName'],
        UserName = data['UserName'],
        Password=hashed_password,
        Email = data['Email'],
        DateOfBirth = data['DateOfBirth'],
        gender_id = data['gender_id'],
        roleID = data.get('role', 2) # 2 for client
    )

    db.session.add(new_user)
    db.session.commit()
    return jsonify(new_user.to_dict())
```

- **Login (login)**
 - **Endpoint:** /users/Login
 - **Method:** POST
 - **Description:** Users can log into the system using their username and password.

```
@users_bp.route('/Login', methods=['POST'])
def login():
    data = request.get_json()
    username = data['UserName']
    password = data['Password']

    if not username or not password:
        return abort(400, "Username and password are required.")

    user = Users.query.filter_by(UserName=username).first()

    if user and check_password_hash(user.Password, password):
        return jsonify({'message': 'Login successful.', 'role_id': user.roleID, 'user_id': user.UserID})
    else:
        return jsonify({'message': 'Invalid username or password.'}), 401
```

- **Send Complaint (add_complaint)**
 - **Endpoint:** /complaints/add_complaint/<userID>
 - **Method:** POST
 - **Description:** Users can send a complaint to the system.

```
@Complaints_bp.route('/add_complaint/<int:user_id>', methods=['POST'])
def add_complaint(user_id):
    data = request.get_json()
    new_complaint = Complaints(
        UserID=user_id,
        Title=data['title'],
        Description=data['description'],
        Status=data.get('status', 'Open'),
        Priority=data.get('priority', 'Medium'),
        Phone=data['phone']
    )

    db.session.add(new_complaint)
    db.session.commit()
    complaint_id = new_complaint.ComplaintID
    return jsonify({'message': 'Complaint added successfully.', 'complaint_id': complaint_id})
```

- **Send Recording (save_record_client)**

- **Endpoint:** /deploy/save_record_client/<userID>
- **Method:** POST
- **Description:** Users can send a recording to the backend for analysis. The system saves the record for employee.

```
# send record and saving it 'form Client'
@Record_bp.route('/save-record-client/<int:user_id>', methods=['POST'])
def save_record_client(user_id):
    file = request.files['file']

    if not file or user_id is None:
        return jsonify({'error': 'Empty'})

    savedRecord = storeRecordClient(file, user_id)

    return jsonify({'emotion': 'File saved successfully ', 'file_path': savedRecord})
```

- **Get News (get_news)**

- **Endpoint:** /News/get_news
- **Method:** GET
- **Description:** It retrieves all news records from the database.

```
# Define a route for retrieving all news for Client
@News_bp.route('/get_news', methods=['GET'])
def get_news():
    data = News.query.all()

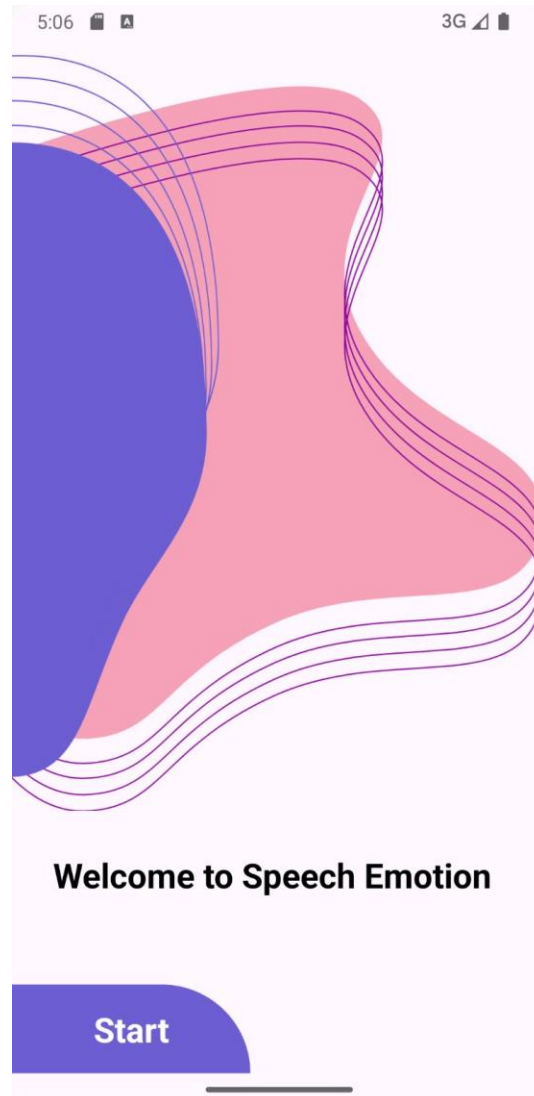
    news_list = []

    for n in data:
        news_item = {
            'title': n.title,
            'body': n.body,
            'date1': n.date1.strftime("%Y-%m-%d %H:%M:%S")
        }
        news_list.append(news_item)

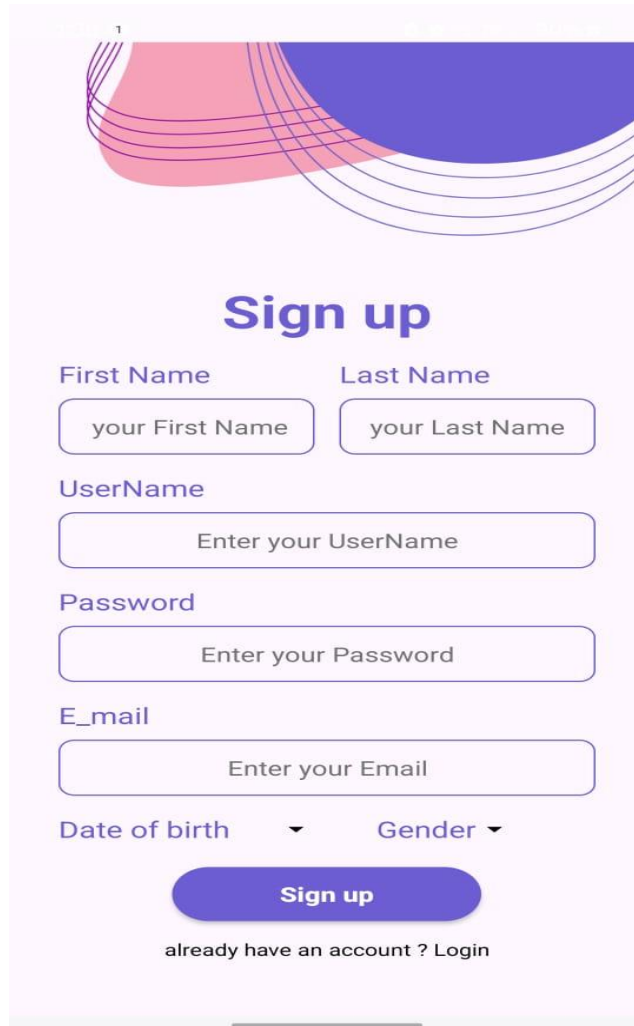
    return jsonify(news_list)
```

4.4 Interface App

- **Home Page**



- **Sign Up Page**



Sign up

First Name

Last Name

UserName

Password

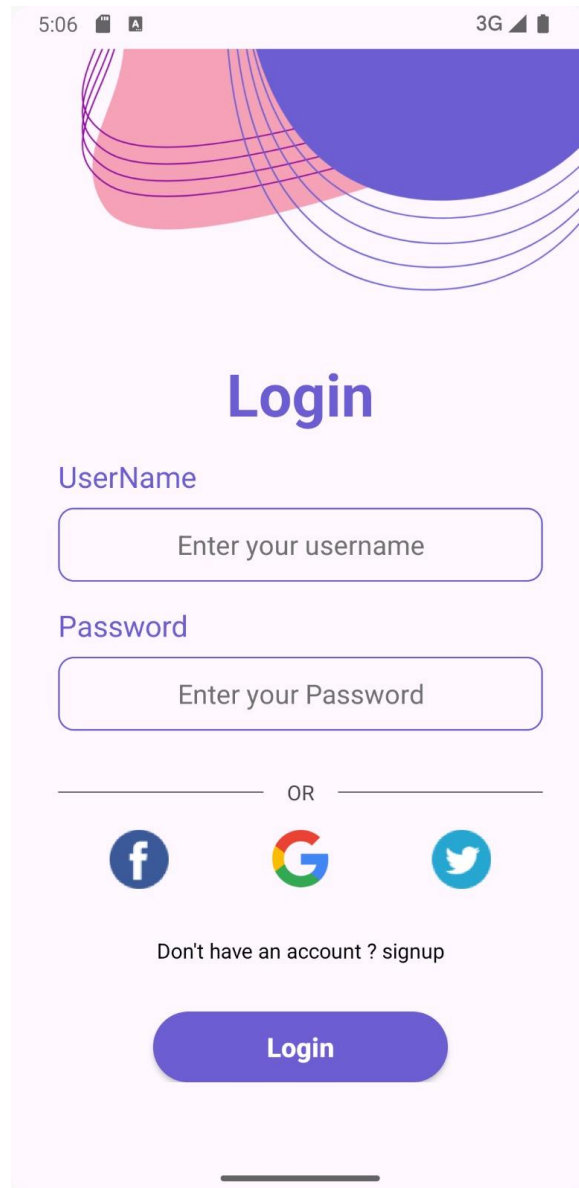
E_mail

Date of birth Gender

Sign up

already have an account ? [Login](#)

- **Login Page**



A mobile app login page with a light purple background. At the top, there's a decorative header with overlapping pink and blue curved shapes. The status bar at the very top shows the time 5:06, signal strength, and battery level. The word "Login" is centered in a large, bold, dark blue font. Below it, the label "UserName" is followed by a rounded rectangular input field containing the placeholder text "Enter your username". Below that, the label "Password" is followed by a similar rounded rectangular input field containing the placeholder text "Enter your Password". A horizontal line with the text "OR" in the center separates the password field from the social login options. There are three circular icons: Facebook (blue with white 'f'), Google (multi-colored 'G'), and Twitter (blue with white bird). Below these icons, the text "Don't have an account ? signup" is displayed. At the bottom, there is a large, rounded purple button with the word "Login" in white text. A thin horizontal line is at the very bottom of the page.

5:06 3G

Login

UserName

Enter your username

Password

Enter your Password

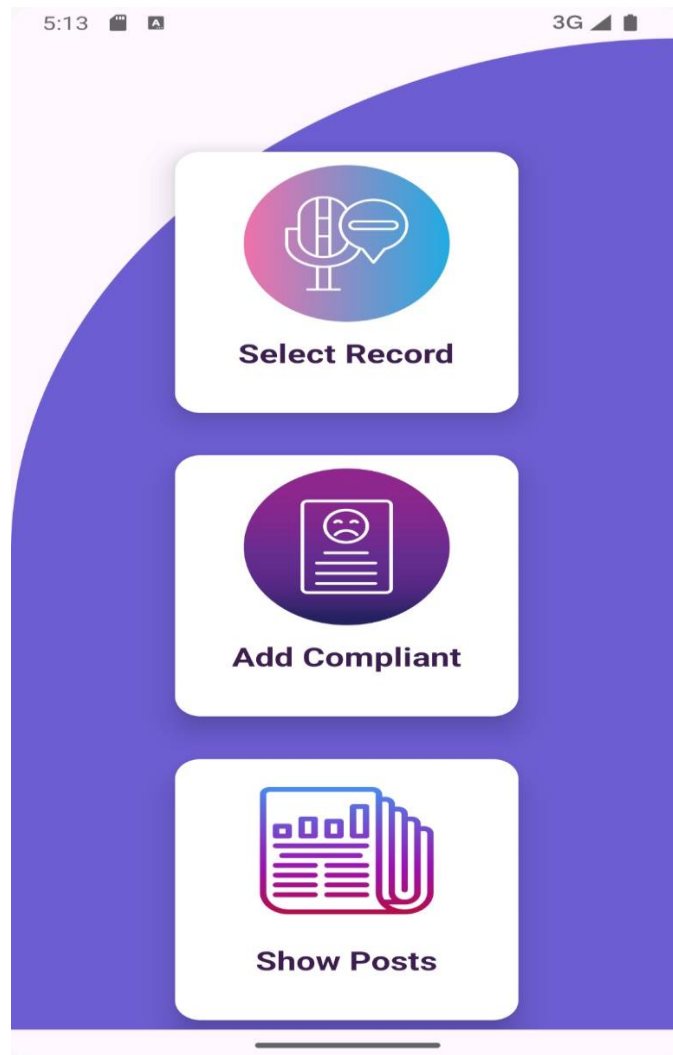
OR

f G t

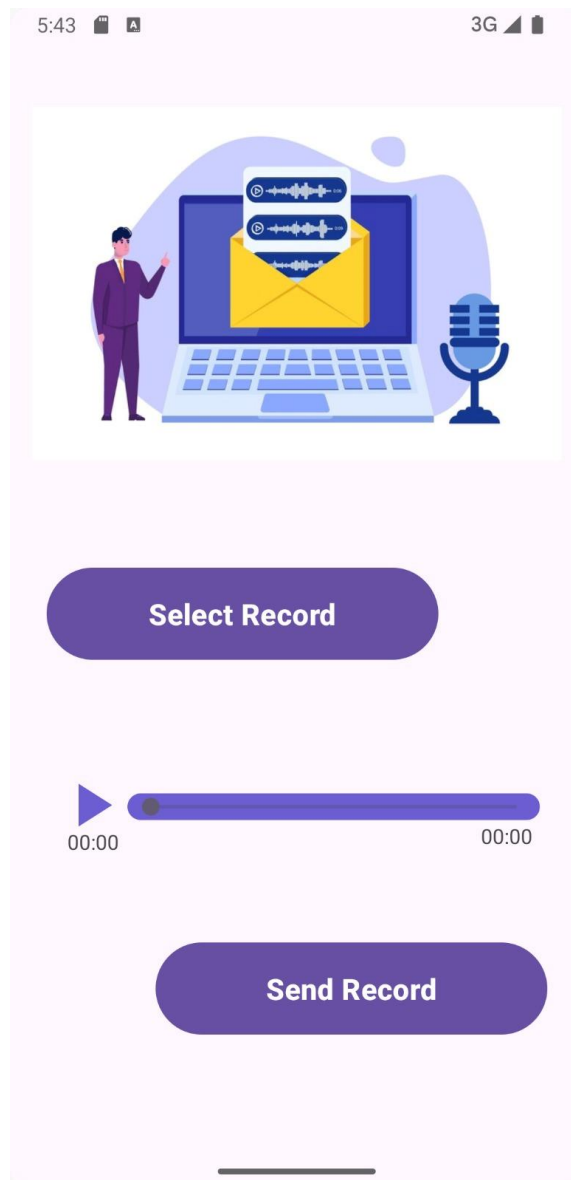
Don't have an account ? [signup](#)

Login

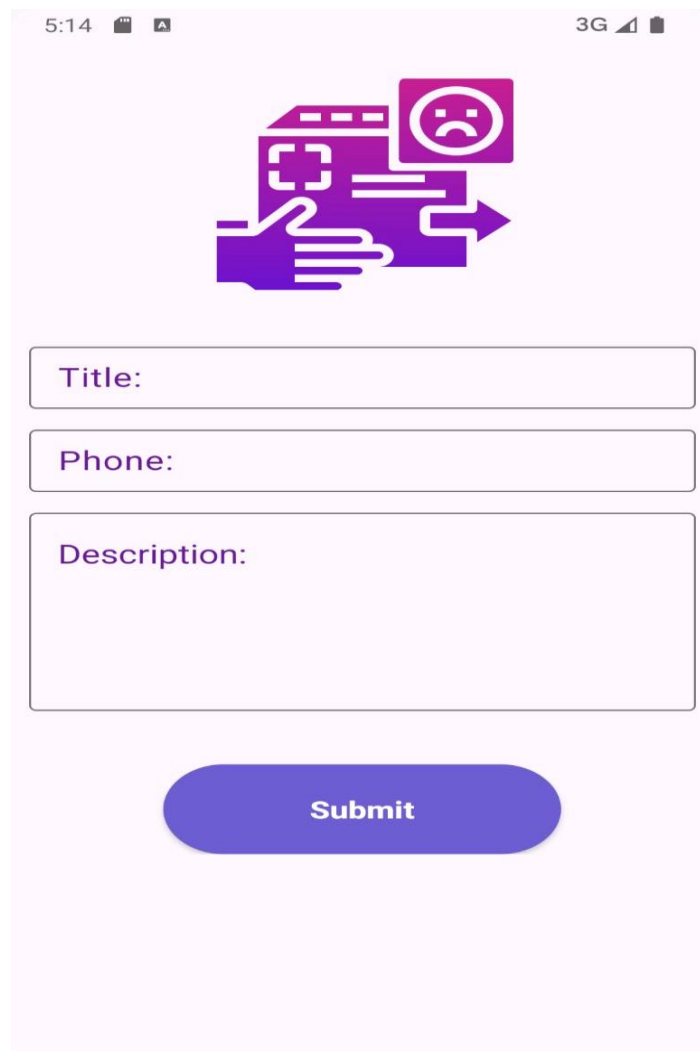
- **User Home page**




- **Send Record Page**



- **Add Compliant Page**



5:14 3G



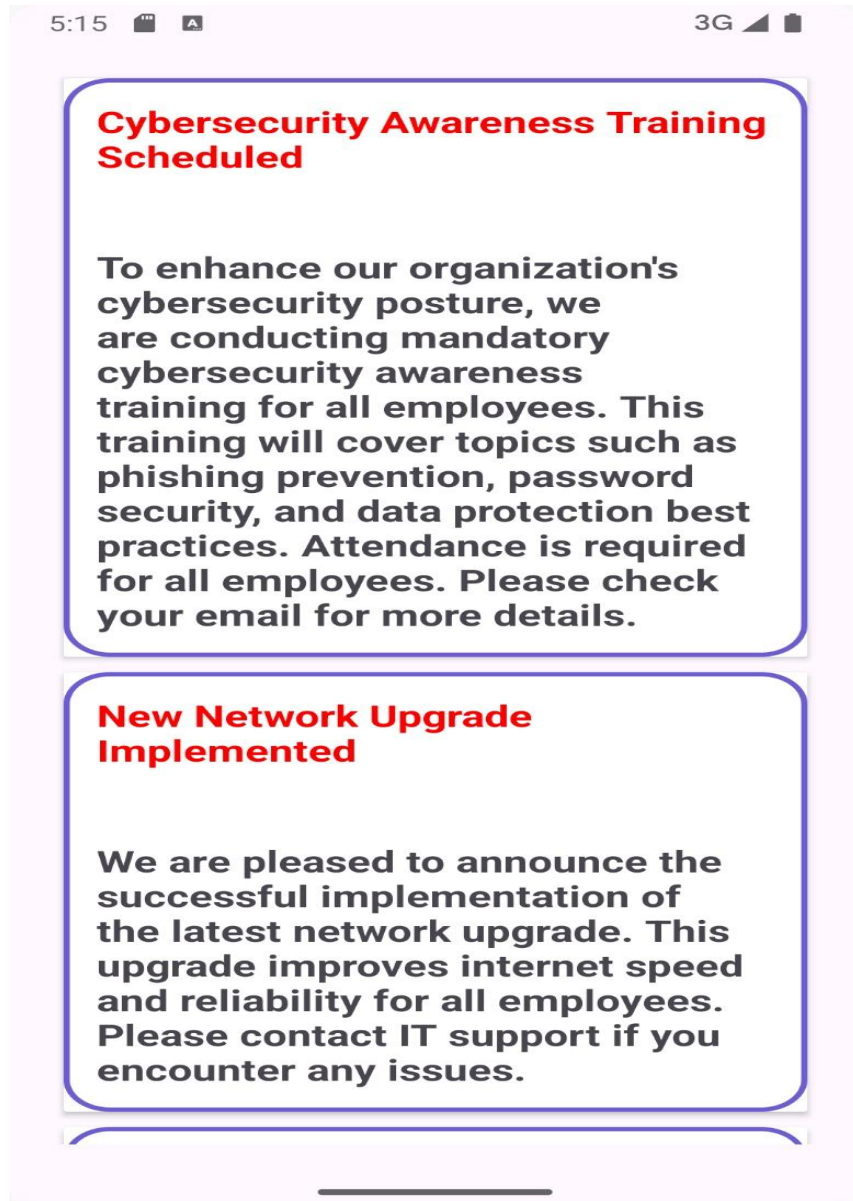
Title:

Phone:

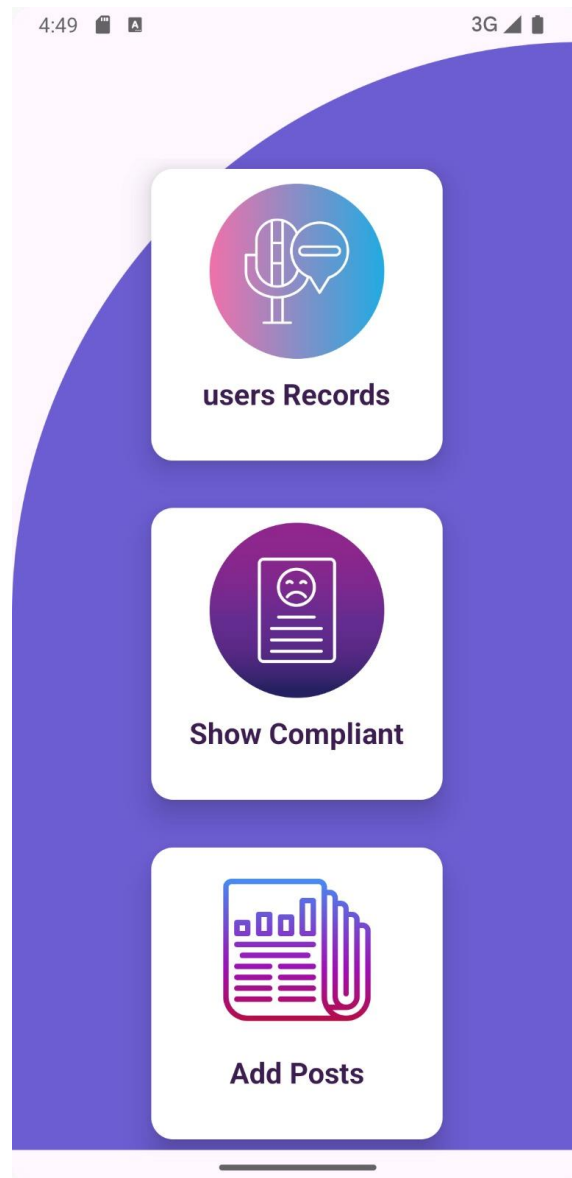
Description:

Submit

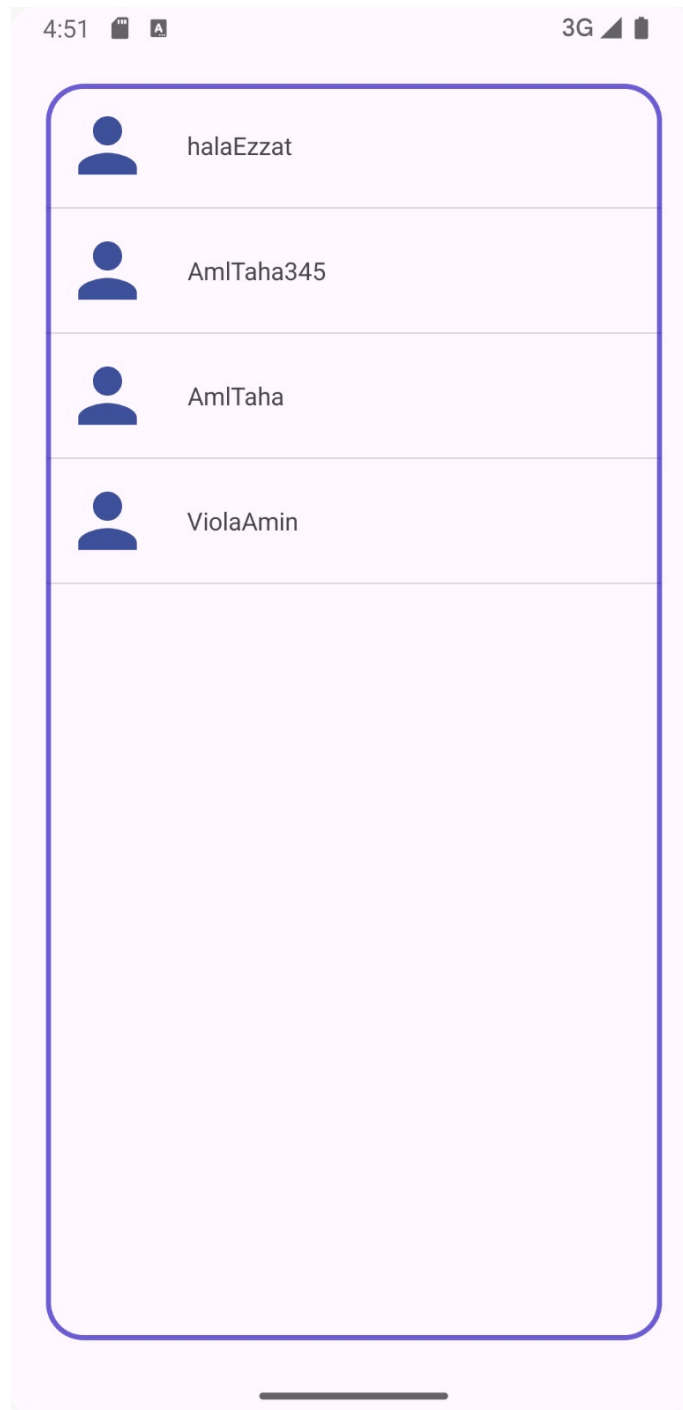
- **Show Posts Page**



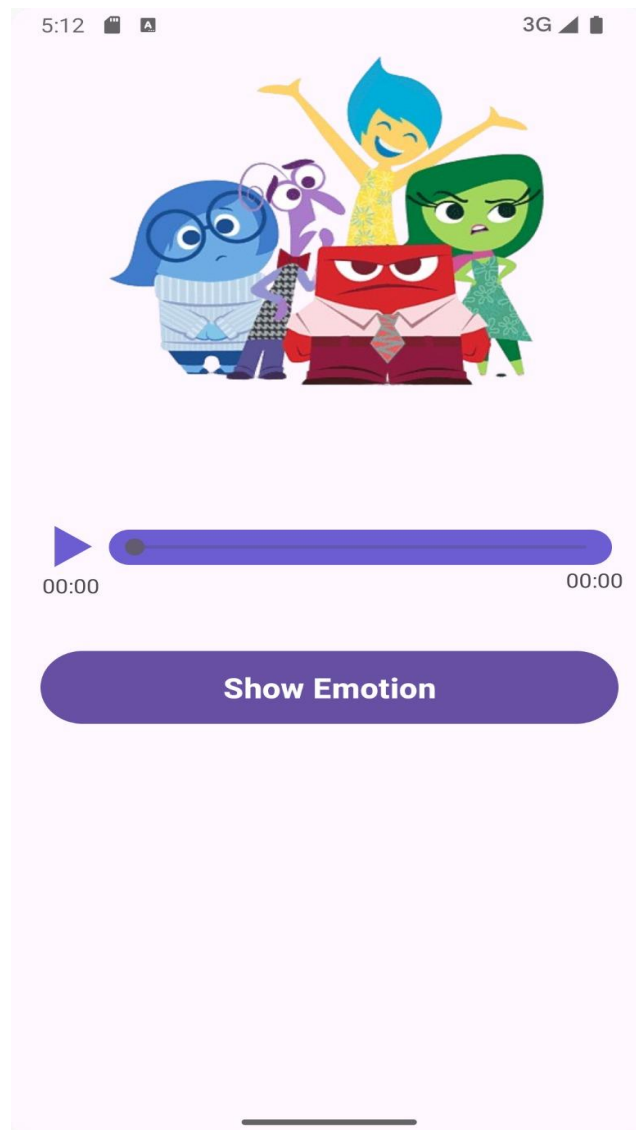
- **Employee Home Page**



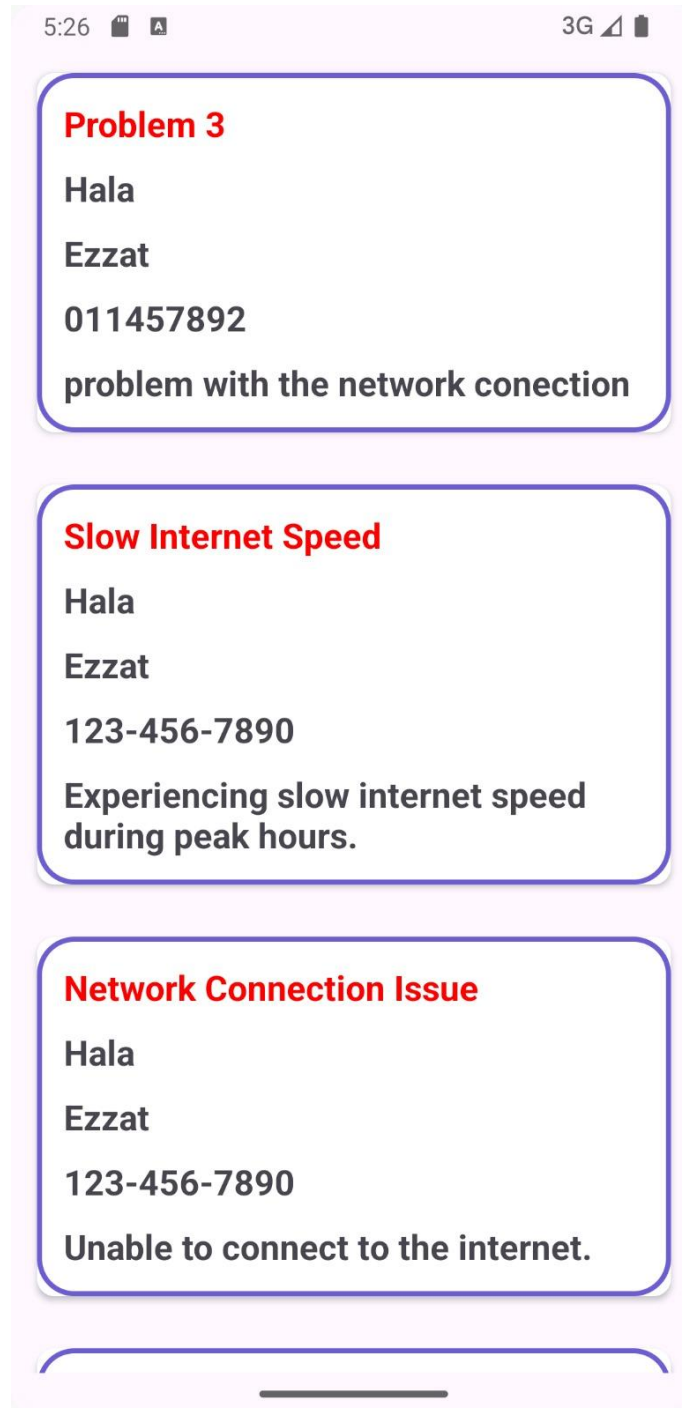
- **Retrieve Users Page**



- **Predict Record Page**



- **Show Complaints Page**



- **Add Posts Page**

The screenshot shows a mobile application interface. At the top, the status bar displays the time 4:53, signal strength, and battery level. The background consists of two news articles. The first article is titled "Cybersecurity Awareness Training Scheduled" in red text. The second article is partially visible below it. A white dialog box is centered on the screen, containing the following elements:

- A label "Post title :" followed by a text input field with the placeholder text "Enter Post Title".
- A text input field with the placeholder text "Enter Your Post".
- Two buttons at the bottom right: "Cancel" and "Insert".

At the bottom of the app interface, there is a dark blue bar containing the text "bb" in red, the text "bbbb" in white, and a dark blue circular button with a white plus sign.

Chapter 5

Testing

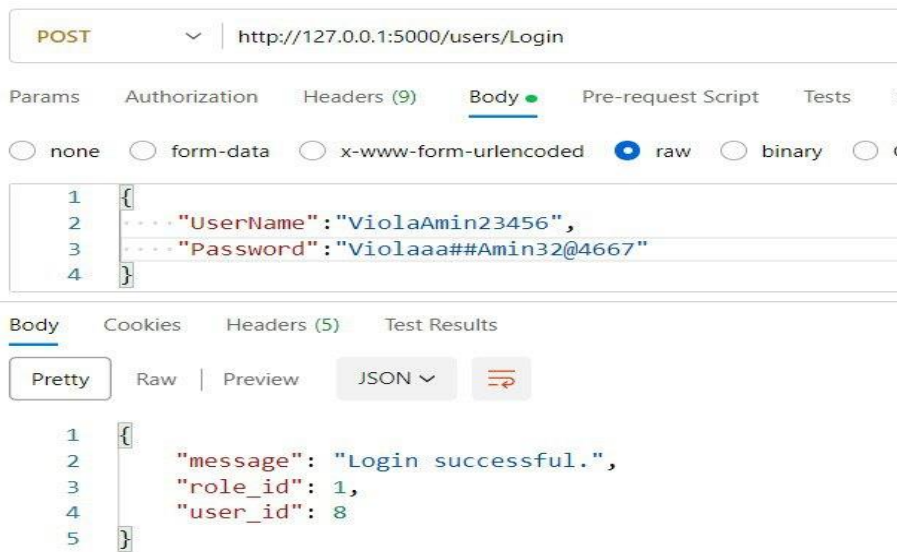
5.1 Testing

5.1.1 Backend

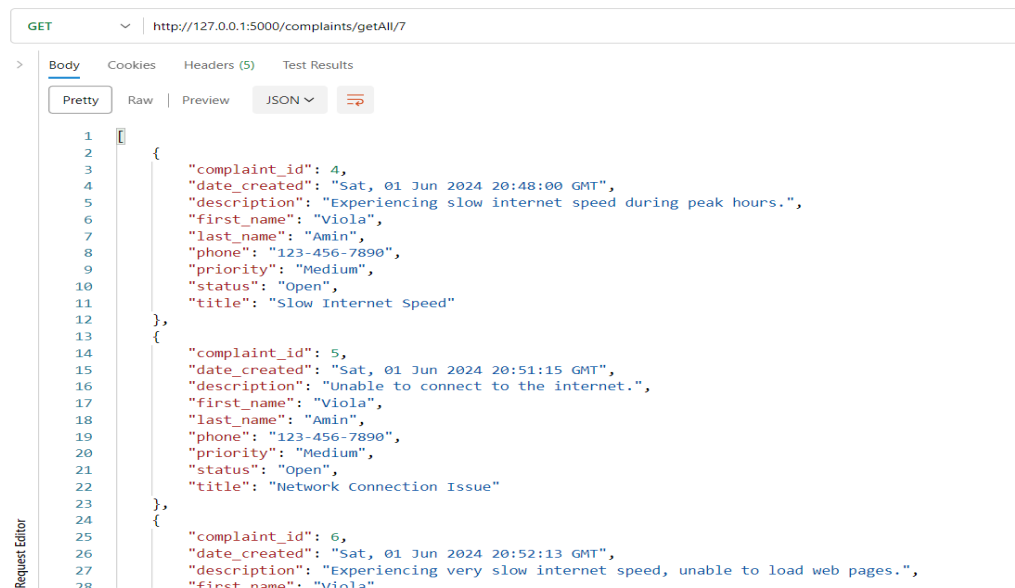
We used Postman to send requests and retrieve the results from my backend API

For employee

- **Login**



- **Get All Complaints**



• Upload News

POST ▼ | http://127.0.0.1:5000/News/insert_news/2

Body ▼ Code Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded

```

1 {
2   ...."title": "New Network
3   ...."body": "We are pleased
   ....to announce the
   ....successful
   ....implementation of the
   ....latest network
   ....upgrade. This upgrade
   ....improves internet
   ....speed and reliability
   ....for all employees.
   ....Please contact IT
   ....support if you
   ....encounter any issues."
4 }

```

Body Cookies Headers (5) Test Results

Pretty Raw Preview JSON ▼ ≡

```

1 {
2   "message": "News inserted successfully",
3   "newsID": 2
4 }

```

• Process User Recording

POST ▼ | http://127.0.0.1:5000/Deploy/predict-emotion/2

Body ▼ Code Cookies

☐ none ☒ form-data ☐ x-www-form-urlencoded

	Key	Value
<input checked="" type="checkbox"/>	file	D:\GPD... X
<input checked="" type="checkbox"/>	Key	Value

Body Cookies Headers (5) Test Results

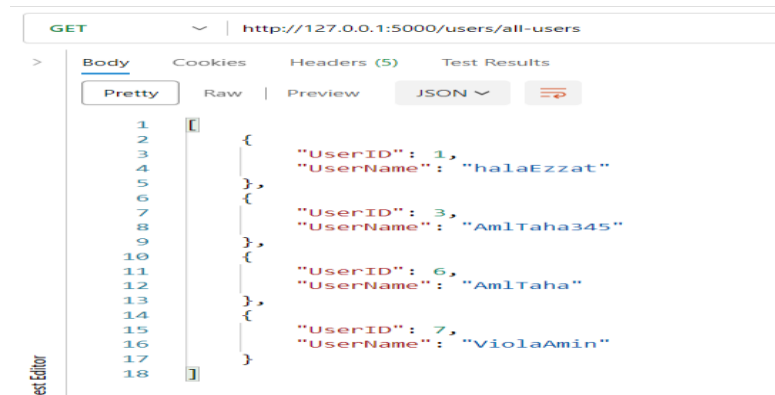
Pretty Raw Preview JSON ▼ ≡

```

1 {
2   "emotion": "fear",
3   "message": "File and prediction result saved successfully."
4 }

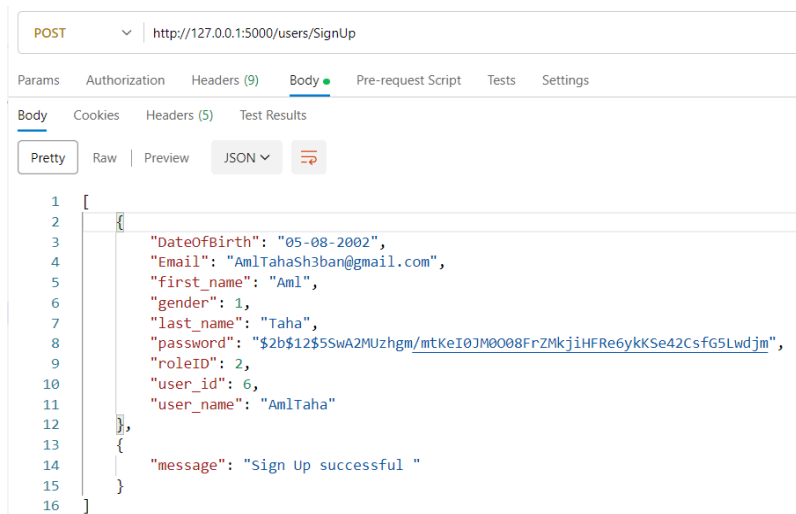
```

- Retrieve all users

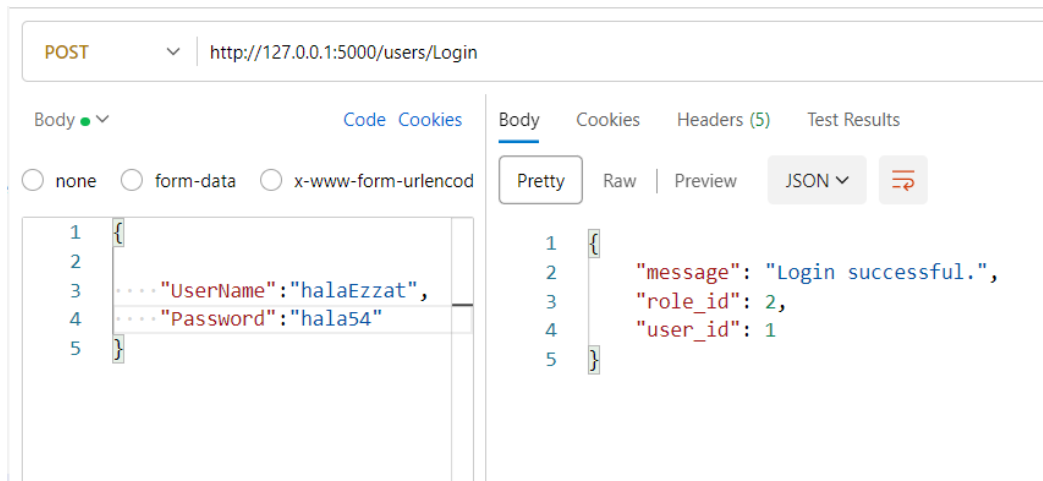


For user

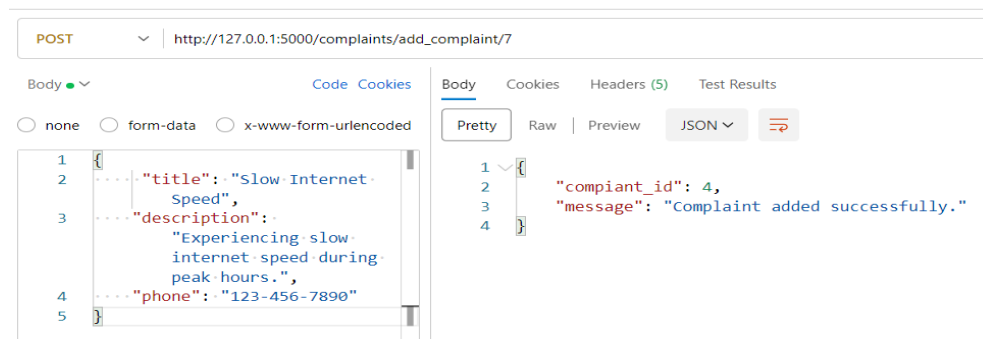
- Create New Account



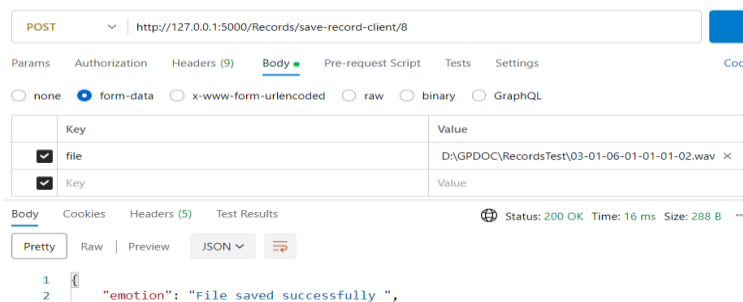
- **Login**



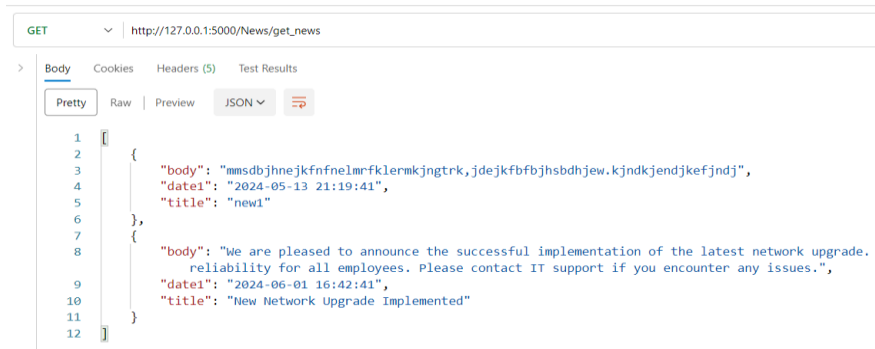
- **Send Complaint**



- **Send Recording**



- **View News**



Chapter 6

Conclusion & Future work

6.1 Conclusion

In summary, this project brings together state-of-the-art technology and user-friendly design to create an emotion detection application for speech analysis. By leveraging Convolutional Neural Networks trained on diverse datasets, it achieves impressive accuracy in identifying emotions. The integration of this model with a Flask-powered backend and intuitive Android interfaces ensures seamless interactions for both clients and employees. This application exemplifies the potential of advanced natural language processing techniques to enhance user experiences and address real-world needs effectively.

6.2 Future work

- **Real-Time Recording:** Implement real-time voice recording functionality, allowing users to capture and analyze emotions as they speak. This would enhance the user experience by providing immediate feedback and could have applications in live conversations, interviews, and customer service interactions.
- **Actor Role Addition:** Introduce a new role in the system, the "Manager," who has the authority to rate the number of positive emotions (such as happiness) identified in employee interactions. Based on this evaluation, the manager can provide feedback to the employee, assess their performance, and potentially offer bonuses or incentives as recognition for exemplary work.
- **Performance Evaluation and Bonus System:** Develop a feature that allows managers to assess employee performance based on emotion analysis results. By tracking the frequency and intensity of positive emotions expressed by employees during interactions with clients, managers can gain insights into their effectiveness and impact. This data can inform performance evaluations and bonus allocation, fostering motivation and recognition within the workforce.
- **Data Logging and Dataset Expansion:** Enhance the application's data logging capabilities to store not only the audio files but also the corresponding predicted emotions. This aggregated dataset of recorded interactions and associated emotions can serve as a valuable resource for further research and training of emotion detection models. By continually expanding and refining the dataset, we can improve the accuracy and robustness of our emotion recognition system.
- **Integration with Additional Platforms and Devices:** Explore opportunities to integrate the application with other platforms and devices, such as smart speakers, wearable devices, and video conferencing software. By extending the reach of our emotion detection capabilities across various channels and contexts, we can enhance communication and user experiences in diverse settings.

References

- [1] J. Zhao, X. Mao and L. Chen, "Speech emotion recognition using deep 1D & 2D CNN LSTM networks," *Biomedical Signal Processing and Control*, vol. 47, p. 312 – 323, 2019.
- [2] H. M. Fayek, M. Lech and L. Cavedon, "Evaluating deep learning architectures for Speech Emotion Recognition," *Neural Networks*, vol. 92, p. 60 – 68, 2017.
- [3] M. Neumann and N. T. Vu, "Attentive convolutional neural network based speech emotion recognition: A study on the impact of input features, signal length, and acted speech," 2017.
- [4] A. Aftab, A. Morsali, S. Ghaemmaghami and B. Champagne, "LIGHT-SERNET: A LIGHTWEIGHT FULLY CONVOLUTIONAL NEURAL NETWORK FOR SPEECH EMOTION RECOGNITION," 2022.
- [5] M. Z. Uddin and E. G. Nilsson, "Emotion recognition using speech and neural structured learning to facilitate edge intelligence," *Engineering Applications of Artificial Intelligence*, vol. 94, 2020.
- [6] T. Anvarjon, Mustaqeem and S. Kwon, "Deep-net: A lightweight cnn-based speech emotion recognition system using deep frequency features," *Sensors (Switzerland)*, vol. 20, p. 1 – 16, 2020.
- [7] Mustaqeem, M. Sajjad and S. Kwon, "Clustering-Based Speech Emotion Recognition by Incorporating Learned Features and Deep BiLSTM," *IEEE Access*, vol. 8, p. 79861 – 79875, 2020.
- [8] Y. Yang, Q. Wu, M. Qiu, Y. Wang and X. Chen, "Emotion Recognition from Multi-Channel EEG through Parallel Convolutional Recurrent Neural Network," 2018.
- [9] J. X. Chen, P. W. Zhang, Z. J. Mao, Y. F. Huang, D. M. Jiang and Y. N. Zhang, "Accurate EEG-Based Emotion Recognition on Combined Features Using Deep Convolutional Neural Networks," *IEEE Access*, vol. 7, p. 44317 – 44328, 2019.
- [10] D. Hazarika, S. Poria, R. Mihalcea, E. Cambria and R. Zimmermann, "ICoN: Interactive conversational memory network for multimodal emotion detection," 2018.
- [11] J. Deng, X. Xu, Z. Zhang, S. Fruhholz, B. Schuller, J. Deng, X. Xu, Z. Zhang, S. Fruhholz and B. Schuller, "Semisupervised Autoencoders for Speech Emotion Recognition," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 26, p. 31–43, January 2018.
- [12] S. Parthasarathy and C. Busso, "Semi-Supervised Speech Emotion Recognition With Ladder Networks," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 28, p. 2697–2709, October 2020.

- [13] Z. Zhang, E. Coutinho, J. Deng and B. Schuller, "Cooperative Learning and Its Application to Emotion Recognition from Speech," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 23, p. 115–126, January 2015.
- [14] R. Lotfian and C. Busso, "Curriculum Learning for Speech Emotion Recognition From Crowdsourced Labels," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 27, p. 815–826, April 2019.
- [15] Y. Zhou, X. Liang, Y. Gu, Y. Yin and L. Yao, "Multi-Classfier Interactive Learning for Ambiguous Speech Emotion Recognition," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 30, p. 695–705, January 2022.
- [16] W. Chen, X. Xing, X. Xu, J. Pang and L. Du, "SpeechFormer++: A Hierarchical Efficient Framework for Paralinguistic Speech Processing," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 31, p. 775–788, January 2023.
- [17] Z. Lian, B. Liu and J. Tao, "CTNet: Conversational Transformer Network for Emotion Recognition," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 29, p. 985–1000, January 2021.
- [18] P. Wei and Y. Zhao, "A Novel Speech Emotion Recognition Algorithm Based on Wavelet Kernel Sparse Classifier in Stacked Deep Auto-Encoder Model," *Personal Ubiquitous Comput.*, vol. 23, p. 521–529, July 2019.
- [19] W. Fan, X. Xu, B. Cai and X. Xing, "ISNet: Individual Standardization Network for Speech Emotion Recognition," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 30, p. 1803–1814, May 2022.
- [20] T. Afouras, J. S. Chung, A. Senior, O. Vinyals and A. Zisserman, "Deep Audio-Visual Speech Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, p. 8717 – 8727, 2022.
- [21] M. S. Hossain and G. Muhammad, "Emotion recognition using deep learning approach from audio–visual emotional big data," *Information Fusion*, vol. 49, p. 69 – 78, 2019.
- [22] U. Kumaran, S. Radha Rammohan, S. M. Nagarajan and A. Prathik, "Fusion of mel and gammatone frequency cepstral coefficients for speech emotion recognition using deep C-RNN," *International Journal of Speech Technology*, vol. 24, p. 303 – 314, 2021.
- [23] D. Li, L. Sun, X. Xu, Z. Wang, J. Zhang and W. Du, "BLSTM and CNN Stacking Architecture for Speech Emotion Recognition," *Neural Processing Letters*, vol. 53, p. 4097 – 4115, 2021.
- [24] H. Meng, T. Yan, F. Yuan and H. Wei, "Speech Emotion Recognition from 3D Log-Mel Spectrograms with Deep Learning Network," *IEEE Access*, vol. 7, p. 125868 – 125881, 2019.

- [25] Z. Peng, Y. Lu, S. Pan and Y. Liu, "Efficient speech emotion recognition using multi-scale cnn and attention," 2021.
- [26] G. Trigeorgis, F. Ringeval, R. Brueckner, E. Marchi, M. A. Nicolaou, B. Schuller and S. Zafeiriou, "Adieu features? End-to-end speech emotion recognition using a deep convolutional recurrent network," 2016.
- [27] S. Zhang, X. Tao, Y. Chuang and X. Zhao, "Learning deep multimodal affective features for spontaneous speech emotion recognition," *Speech Communication*, vol. 127, p. 73 – 81, 2021.
- [28] L. F. Parra-Gallego and J. R. Orozco-Arroyave, "Classification of emotions and evaluation of customer satisfaction from speech in real world acoustic environments," *Digital Signal Processing: A Review Journal*, vol. 120, 2022.
- [29] Mustaqeem and S. Kwon, "MLT-DNet: Speech emotion recognition using 1D dilated CNN based on multi-learning trick approach," *Expert Systems with Applications*, vol. 167, 2021.
- [30] Mustaqeem and S. Kwon, "A CNN-assisted enhanced audio signal processing for speech emotion recognition," *Sensors (Switzerland)*, vol. 20, 2020.
- [31] Z.-T. Liu, A. Rehman, M. Wu, W.-H. Cao and M. Hao, "Speech emotion recognition based on formant characteristics feature extraction and phoneme type convergence," *Information Sciences*, vol. 563, pp. 309-325, 2021.
- [32] D. Li, J. Liu, Z. Yang, L. Sun and Z. Wang, "Speech emotion recognition using recurrent neural networks with directional self-attention," *Expert Systems with Applications*, vol. 173, 2021.
- [33] M. Lech, M. Stolar, C. Best and R. Bolia, "Real-Time Speech Emotion Recognition Using a Pre-trained Image Classification Network: Effects of Bandwidth Reduction and Companding," *Frontiers in Computer Science*, vol. 2, 2020.
- [34] A. Koduru, H. B. Valiveti and A. K. Budati, "Feature extraction algorithms to improve the speech emotion recognition rate," *International Journal of Speech Technology*, vol. 23, pp. 45-55, 2020.
- [35] S. Kanwal and S. Asghar, "Speech Emotion Recognition Using Clustering Based GA-Optimized Feature Set," *IEEE Access*, vol. 9, pp. 125830-125842, 2021.
- [36] D. Issa, M. Fatih Demirci and A. Yazici, "Speech emotion recognition with deep convolutional neural networks," *Biomedical Signal Processing and Control*, vol. 59, 2020.
- [37] M. Farooq, F. Hussain, N. K. Baloch, F. R. Raja, H. Yu and Y. B. Zikria, "Impact of feature selection algorithm on speech emotion recognition using deep convolutional neural network," *Sensors (Switzerland)*, vol. 20, pp. 1-18, 2020.

- [38] L. Chen, W. Su, Y. Feng, M. Wu, J. She and K. Hirota, "Two-layer fuzzy multiple random forest for speech emotion recognition in human-robot interaction," *Information Sciences*, vol. 509, pp. 150-163, 2020.
- [39] B. B. Al-onazi, M. A. Nauman, R. Jahangir, M. M. Malik, E. H. Alkhamash and A. M. Elshewey, "Transformer-Based Multilingual Speech Emotion Recognition Using Data Augmentation and Feature Fusion," *Applied Sciences (Switzerland)*, vol. 12, 2022.
- [40] L. Abdel-Hamid, "Egyptian Arabic speech emotion recognition using prosodic, spectral and wavelet features," *Speech Communication*, vol. 122, pp. 19-30, 2020.
- [41] C. PARLAK and Y. ALTUN, "A novel filter bank design for speech emotion recognition," 2022.
- [42] L. Kerkeni, Y. Serrestou, M. Mbarki, K. Raoof, M. A. Mahjoub and C. Cleder, *Automatic speech emotion recognition using machine learning*, IntechOpen, 2019.
- [43] B. J. Abbaschian, D. Sierra-Sosa and A. Elmaghraby, "Deep learning techniques for speech emotion recognition, from databases to models," *Sensors (Switzerland)*, vol. 21, p. 1 – 27, 2021.
- [44] Mustaqeem, M. Sajjad and S. Kwon, "Clustering-Based Speech Emotion Recognition by Incorporating Learned Features and Deep BiLSTM," *IEEE Access*, vol. 8, p. 79861 – 79875, 2020.
- [45] M. A. Mohammed, K. H. Abdulkareem, S. A. Mostafa, M. K. A. Ghani, M. S. Maashi, B. Garcia-Zapirain, I. Oleagordia, H. Alhakami and F. T. Al-Dhief, "Voice pathology detection and classification using convolutional neural network model," *Applied Sciences (Switzerland)*, vol. 10, 2020.
- [46] T. Anvarjon, Mustaqeem and S. Kwon, "Deep-net: A lightweight cnn-based speech emotion recognition system using deep frequency features," *Sensors (Switzerland)*, vol. 20, p. 1 – 16, 2020.
- [47] Z. Song, "English speech recognition based on deep learning with multiple features," *Computing*, vol. 102, p. 663 – 682, 2020.
- [48] K. Tarunika, R. B. Pradeeba and P. Aruna, "Applying machine learning techniques for speech emotion recognition," 2018.
- [49] Mustaqeem, M. Sajjad and S. Kwon, "Clustering-Based Speech Emotion Recognition by Incorporating Learned Features and Deep BiLSTM," *IEEE Access*, vol. 8, pp. 79861-79875, 2020.