UNVEILING SPEECH EMOTIONAL SPECTRUM THROUGH SOUND USING CONVOLUTIONAL NEURAL NETWORKS

A PROJECT REPORT

Submitted by

ASLAM SUJATH A	**830120104002**
PRADEEP P	**830120104020**
SELVAKUMAR E	**830120104026**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

GOVERNMENT COLLEGE OF ENGINEERING SRIRANGAM

ANNA UNIVERSITY: CHENNAI 600 025

MAY 2024

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"Unveiling Speech Emotional Spectrum Through Sound Using Convolutional Neural Networks"** is the bonafide work of **"ASLAMSUJATH A (830120104002), PRADEEP P (830120104020), SELVAKUMAR E (830120104026)"** who carried out the project work under my supervision.

SIGNATURE
Prof. P.VANITHA MUTHU, M.Tech.,
HEAD OF THE DEPARTMENT
Dept of Computer Science & Engineering,
Government College of Engg. Srirangam,
Tiruchirappalli-620012.

SIGNATURE
Prof. H. SHEIK MOHIDEEN, M.E.,
SUPERVISOR
Assistant Professor,
Dept of Computer Science & Engineering,
Government College of Engg. Srirangam,
Tiruchirappalli-620012.

Submitted for the Project work (CS8811) viva-voce examination held on _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We first of all thank GOD ALMIGHTY for giving us a golden opportunity to express our individual and technical skills in the field of Computer Science and Engineering.

We express our gratitude to our beloved Principal, **Dr. R. MALAYALAMURTHI, M.E., Ph.D.,** for giving us a chance to complete our higher education in one of the reputed government institutions running under her magnificent leadership.

We are extremely thankful to our Head of the Department, **Prof. P. VANITHA MUTHU, M.Tech.,** Associate Professor, Department of Computer Science and Engineering, for her support, motivation, inspiration and tireless encouragement.

We are immensely pleased to thank our project guide, **Prof. H. SHEIK MOHIDEEN, M.E.,** Assistant Professor, Department of Computer Science and Engineering for her valuable comments and suggestions.

We also express our thanks to our Project Coordinator, **Dr. S.ANNIE JOICE, M.E., M.B.A., Ph.D.,** Assistant Professor, Department of Computer Science and Engineering, for ideas given by him and for his constant inspiration throughout our project period.

It is a great opportunity to express our sincere thanks to our parents, friends and all the people who have contributed to the successful completion of our project work through their support, encouragement and guidance.

ASLAM SUJATH A
PRADEEP P
SELVAKUMAR E

ABSTRACT

Research in Speech Emotion Recognition (SER) has garnered significant attention, particularly in the realm of Human-Computer Interaction (HCI) with a focus on personal assistants and assistive robots. Human speech emotion recognition is the task of automatically detecting the emotional content conveyed by a person's speech, typically through analysis of acoustic features such as pitch, amplitude, and spectral content. This method involves analyzing subtle tones and pitches in speech, utilizing aural cues to classify human emotions like calm, happy, sad, anger, fear, surprise, and disgust. Annotated datasets like RAVDESS facilitate this research, containing recordings of actors expressing various emotions. Deep learning techniques, especially convolutional neural networks (CNNs), are emerging as powerful tools for processing emotional speech signals. CNNs automatically learn hierarchical representations from raw data, making them adept at capturing complex patterns in audio signals. This approach enhances human-technology interactions by enabling machines to recognize and respond to human emotions conveyed through language. Thus, the integration of SER into HCI research contributes to improving interactive computer systems' design.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

HCI	**H**uman-**C**omputer **I**nteraction
ML	**M**achine **L**earning
DL	**D**eep **L**earning
CNN	**C**onvolutional **N**eural **N**etwork
SVM	**S**upport **V**ector **M**achine
MLP	**M**ulti-**L**ayer **P**erceptron
RAVDESS	**R**yerson **A**udio-**V**isual **D**atabase Of **E**motional **S**peech And **S**ong.
MFCC	**M**el **F**requency **C**epstral **C**oefficients
ZCR	**Z**ero **C**rossing **R**ate
HTML	**H**yper **T**ext **M**arkup **L**anguage
CSS	**C**ascading **S**tyle **S**heet
TP	**T**rue **P**ositive
TN	**T**rue **N**egative
FP	**F**alse **P**ositive
FN	**F**alse **N**egative

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Human speech emotion recognition is the process of using computational techniques to identify the emotional content in spoken language. It involves analyzing various acoustic features of speech, such as chroma, MFCC, ZCR and using machine learning algorithms to classify the emotional state of the speaker. The ability to recognize human emotions from speech has numerous practical applications, including improving human-computer interaction, developing more effective speech- based therapies for individuals with emotional disorders, and enhancing the accuracy of lie detection techniques. The importance of this project lies in its potential to improve applications ranging from virtual assistants to emotion recognition technology s technology becomes more integrated into our daily lives, the capability of machines to comprehend and respond to human emotions will grow in significance. As such, ongoing research in this field is focused on developing more sophisticated algorithms that can better capture the complexity and variability of human emotional expression in speech.

1.1.1 Machine Learning

Machine learning (ML) is a subset of artificial intelligence (AI) that focuses on the development of algorithms and statistical models that enable computers to perform specific tasks without being explicitly programmed. Instead of relying on explicit instructions, ML algorithms learn from data, iteratively improving their performance over time. These algorithms and models learn from data, identifying patterns and making predictions or decisions based on that data.

- **Supervised Learning**

Supervised learning involves training a model on labeled data, where each input is paired with a corresponding output, with the aim of learning a mapping between the two to make predictions or decisions about new, unseen data. In speech emotion recognition, supervised learning entails training models with labeled speech samples and their corresponding emotions, enabling the system to predict the emotion of new, unseen speech data based on learned patterns.

- **Unsupervised Learning**

Unsupervised learning involves training a model on unlabeled data to uncover underlying patterns or structures without explicit guidance. In the context of speech emotion recognition, unsupervised learning may involve analyzing speech data to group similar patterns together or reduce the dimensionality of the feature space.

- **Reinforcement Learning**

Reinforcement learning involves learning to make decisions through interaction with an environment, receiving feedback to maximize cumulative reward over time. In speech emotion recognition, reinforcement learning might be used to develop systems that adjust their behavior based on user feedback, improving their ability to recognize and respond to emotions conveyed in speech.

Machine learning has a wide range of applications, including image and speech recognition, natural language processing, recommendation systems, autonomous vehicles, and medical diagnosis, among many others. It's a rapidly evolving field with continuous advancements in algorithms, techniques, and applications.

1.1.2 DEEP LEARNING

Deep learning is a subset of machine learning that focuses on algorithms inspired by the structure and function of the human brain's neural networks. These algorithms, known as artificial neural networks (ANNs), consist of multiple layers of interconnected nodes, or neurons, that process and learn from data to perform tasks such as classification, regression, and pattern recognition. Deep learning frameworks like TensorFlow, PyTorch, and Keras provide researchers and developers with powerful tools for building, training, and deploying deep learning models for SER. These frameworks offer high-level APIs and pre-trained model architectures, enabling rapid prototyping and experimentation.

1.1.3 CNN IN SPEECH EMOTION RECOGNITION

Convolutional neural networks (CNNs) are a valuable tool for speech emotion recognition (SER) due to their ability to effectively analyze the intricate patterns found in audio signals. Originally designed for image processing, CNNs have been successfully adapted to handle continuous data like audio, showcasing impressive performance in extracting both basic and advanced features crucial for emotion recognition. In the realm of SER, CNNs employ hierarchical feature learning to pinpoint variations in pitch, intensity, prosody, and other acoustic attributes that signify different emotional states. Moreover, CNN's parameter sharing approach aids in reducing the number of trainable parameters and enhancing the model's adaptability to diverse emotional expressions.

1.1.4 Applications of Deep learning

- **Computer Vision**

Deep learning is extensively used in computer vision tasks such as object detection, image classification, and segmentation. Applications include autonomous vehicles, facial recognition systems, medical image analysis, and surveillance systems.

- **Speech Recognition**

Deep learning models, particularly recurrent neural networks (RNNs) and convolutional neural networks (CNNs), are used for speech recognition tasks. Applications range from voice-controlled devices and virtual assistants to automated transcription services and language translation.

- **Healthcare**

Deep learning is transforming healthcare with applications in medical image analysis, disease diagnosis, drug discovery, and personalized treatment. Deep learning models can analyze medical images such as MRI scans, X-rays, and pathology slides to assist radiologists and clinicians in diagnosing diseases.

- **Robotics**

Deep learning is applied in robotics for tasks such as object manipulation, grasping, navigation, and human-robot interaction. Deep neural networks enable robots to perceive their environment, learn from experience, and adapt to changing conditions.

- **Gaming**

Deep learning is used in the gaming industry for tasks such as character animation, behavior modeling, and game testing. Deep neural networks can generate realistic graphics, control non-player characters (NPCs), and improve player experience through adaptive gameplay.

- **Cybersecurity**

Deep learning is employed in cybersecurity for tasks such as malware detection, intrusion detection, and network security analysis. Deep learning models can analyze network traffic, identify anomalous patterns, and detect cyber threats in real-time.

1.2 Motivation

Emotion recognition from speech is a crucial aspect of human communication, and it has important applications in various fields such as healthcare, education, and customer service. By developing accurate and reliable SER system, improves human-human and human-computer interactions, enhance emotional intelligence, and provide better care and services to people. In educational settings, SER-driven systems can adapt learning experiences based on student's emotional states and engagement levels. Emotion-aware tutoring systems could provide personalized feedback and support, ultimately improving learning outcomes and student satisfaction. SER projects contribute to advancing our understanding of human emotions and the underlying mechanisms of emotional expression. By analyzing large datasets of emotional speech samples, and deep learning model that uncover valuable insights into how emotions are conveyed through speech and develop more accurate models for emotion recognition.

1.3 Contribution

Speech Emotion Recognition (SER) project, utilizing Convolutional Neural Networks (CNNs), signifies a breakthrough in decoding emotions directly from speech signals. The CNN-based model showcased heightened accuracy and adaptability, proving effective in practical scenarios. The system effectively integrates audio features like Mel frequency cepstral coefficients (MFCC), zero crossing rate (ZCR), and chroma features to classify emotions. The RAVDESS dataset provides a diverse set of labeled emotional speech samples that are vital for training and evaluating SER systems. In navigating the complex landscape of human emotion, the voice emotion recognition system will play a key role in driving more meaningful and responsive interactions in the ever-evolving field of artificial intelligence

CHAPTER 2

LITERATURE SURVEY

2.1 SPEECH EMOTION RECOGNITION USING MLP CLASSIFIER

Author: Akshath Kumar B.H, Dr. Shivakumar G S

Published Year: 2021

Language is human's most important communication and speech is basic medium of communication. Emotion plays a crucial role in social interaction Emotion varies from person to person, same person has different emotions all together has different way to express it. The machine will convert the human speech signals into waveform and process its routine at last it will display the emotion. The data is speech sample and the characteristics are extracted from the speech sample using librosa package. RAVDESS dataset which are used as an experimental dataset. It provides an accuracy of 68%.

Disadvantage

- Using MLP into SER is sometimes computationally expensive.
- Does not able to understand sudden changes in pattern.

2.2 DECISION TREE AND SVM MODEL WITH FISHER FEATURE SELECTION FOR SER

Author: Linhui Sun, Sheng Fu, Fu Wang

Published Year: 2019

A speech emotion recognition method based on the decision tree support vector machine (SVM) model with Fisher feature selection. At the stage of feature selection, Fisher criterion is used to filter out the feature parameters of higher distinguish ability. At the emotion classification stage, an algorithm is proposed to determine the structure of decision tree. The decision tree SVM can realize the two-step classification of the first rough classification and the fine classification.

Thus, the redundant parameters are eliminated and the performance of emotion recognition is improved. In this method, the decision tree SVM framework is firstly established by calculating the confusion degree of emotion, and then the features with higher distinguish ability are selected for each SVM of the decision tree according to Fisher criterion. Finally, speech emotion recognition is realized based on this model. The decision tree SVM with Fisher feature selection on CASIA Chinese emotion speech corpus and Berlin speech corpus are constructed to validate the effectiveness of our framework. The experimental results show that the average emotion recognition rate based on the proposed method is 9% higher than traditional SVM classification method on CASIA, and 8.26% higher on Berlin speech corpus. It is verified that the proposed method can effectively reduce the emotional confusion and improve the emotion recognition rate.

Disadvantage

- Confusion on Fear and Sad emotion still remains relatively high for CASIA
- It is necessary to carry out targeted research on the two emotions types to search for more effective feature parameters in future study.

2.3 SPEECH BASED HUMAN EMOTION RECOGNITION USING MFCC

Author: Ye Sim Ulgen Sonmez, Asaf Varol

Published Year: 2019

Speech Emotion Recognition (SER) can be defined as extraction of the emotional state of the speaker from his or her speech signal. There are few universal emotions- including Neutral, Anger, Happiness, Sadness in which any intelligent system with finite computational resources can be trained to identify or synthesize as required. In this work spectral and prosodic features are used for speech emotion recognition because both of these features contain the emotional information. Mel-frequency cepstral coefficients (MFCC) is one of the spectral features. Fundamental frequency, loudness, pitch and speech intensity and glottal

parameters are the prosodic features which are used to model different emotions. The potential features are extracted from each utterance for the computational mapping between emotions and speech patterns. Pitch can be detected from the selected features, using which gender can be classified. Support Vector Machine (SVM), is used to classify the gender in this work. Radial Basis Function and Back Propagation Network is used to recognize the emotions based on the selected features, and proved that radial basis function produce more accurate results for emotion recognition than the back propagation network.

Disadvantage

- MFCCs, while effective for capturing spectral features in speech, may lack nuance in emotional expression and sensitivity to semantic content, and can be affected by noise and preprocessing parameters, combined features and complementary approaches for improved emotion recognition accuracy.
- Limited Coverage of Emotional States.
- Comparison with State-of-the-Art.

2.4 SVM SCHEME FOR SPEECH EMOTION RECOGNITION USING MFCC FEATURE

Author: Milton. A, Sharmy Roy. S, Tamil Selvi.S

Published Year: 2013

The 3-stage Support Vector Machine classifier to classify seven different emotions present in the Berlin Emotional Database. For the purpose of classification, MFCC features from all the 535 files present in the database are extracted. Nine statistical measurements are performed over these features from each frame of a sentence. The linear and RBF kernels are employed in hierarchical SVM with RBF sigma value equal to one. For training and testing of data, 10- fold cross-validation is used. Performance analysis is done by using the confusion matrix and the accuracy obtained is 68%. This author achieved average accuracy of

Three Stage SVM 68% and compared with SVM using Radial Basis Function (RBF), Linear Kernel i.e.55.4%, 65% and 68%. Out of these three classifiers highest accuracy is obtained in a SVM 68% accuracy is achieved.

Disadvantage

- Only MFCC features are considered higher accuracy can be obtained using the combination of more features.
- Dimensionality of Feature Space.
- Kernel Selection and Parameter Tuning

2.5 SPEECH EMOTION RECOGNITION USING SUPPORT VECTOR MACHINE

Author: Yashpalsing Chavhan, M. L. Dhore, Pallavi Yesaware.

Published Year: 2010

Automatic Speech Emotion Recognition (SER) is a current research topic in the field of Human Computer Interaction (HCI) with wide range of applications. The speech features such as, Mel Frequency cepstrum coefficients (MFCC) and Mel Energy Spectrum Dynamic Coefficients (MEDC) are extracted from speech utterance. The Support Vector Machine (SVM) is used as classifier to classify different emotional states such as anger, happiness, sadness, neutral, fear, from Berlin emotional database. The LIBSVM is used for classification of emotions.

Disadvantage

- Small database is used only 493 samples.
- Limited Feature Representation.
- SVM performance can be sensitive to hyperparameter tuning.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The existing system in speech emotion recognition employs machine learning algorithms like Multilayer Perceptron (MLP) and Support Vector Machine (SVM) for classification. It leverages feature extraction techniques such as Mel-Frequency Cepstral Coefficients (MFCCs) and prosodic features from speech signals to train and evaluate models. Through careful preprocessing, feature selection, and model optimization, the system achieves accurate classification of emotional states in speech.

3.1.1 SVM

SVM is a supervised learning algorithm used for classification tasks. It aims to find the hyperplane that best separates data points of different classes with the maximum margin. SVMs can face challenges in achieving optimal performance due to the intricate nature of parameter tuning, scalability concerns with larger datasets, inherent difficulties in managing imbalanced data distributions, limited interpretability of model decisions, and complexities associated with capturing nonlinear relationships effectively.

3.1.2 MLP

MLP is a type of artificial neural network composed of multiple layers of nodes (neurons), including an input layer, one or more hidden layers, and an output layer. MLPs' effectiveness can also be hindered by their susceptibility to overfitting, sensitivity to hyperparameters' selection, challenges with the vanishing gradient problem in deeper architectures, risks of converging to local optima, and dependencies on substantial, high-quality datasets for robust training.

3.1.3 Limitations of Existing system

Limited Ability to Capture Spatial Information:

- SVMs and MLPs typically operate on flattened feature vectors, which may not fully capture the spatial relationships present in data sample.

Scalability:

- SVMs and MLPs can become computationally expensive and memory-intensive, especially with large-scale datasets or deep architectures.

Challenges in Modeling Complex Audio:

- In cases where audio data exhibits intricate patterns and dependencies, MLPs and SVMs might not have the capacity to capture and generalize well, leading to lower accuracy on challenging tasks.

3.2 PROPOSED SYSTEM

The proposed system for speech emotion recognition (SER) is a combination of the RAVDESS dataset and a convolutional neural network (CNN) architecture. The system effectively integrates Mel frequency cepstral coefficients (MFCC), zero crossing rate (ZCR), and chroma features to classify emotions. The RAVDESS dataset provides a diverse set of labeled emotional speech samples that are vital for training and evaluating SER systems. The MFCC extracts a compressed representation of spectral characteristics, the ZCR captures temporal dynamics, and the chroma function encodes tonal content. These features are concatenated into the CNN's input feature vector, making both spectral and temporal information available. The CNN architecture comprises convolutional, pooling, and fully connected layers that are trained on labeled datasets using supervised learning techniques. Techniques such as dropout regularization and early stopping prevent overfitting during training. The model's performance on different test sets is evaluated using metrics such as accuracy and the F1 score. Fine-tuning and optimization strategies further refine the model, resulting in a

system that can be deployed in real-world applications. This integrated approach provides a robust solution for SER systems with potential applications in the fields of human- computer interaction and emotional computing.

3.2.1 CNN MODEL

Convolutional Neural Networks (CNNs) are widely utilized in speech emotion recognition (SER) for their ability to effectively capture spatial and temporal patterns in data. By automatically learning hierarchical representations of speech spectrograms, CNNs extract discriminative features relevant to emotion recognition while exhibiting robustness to variations in speech utterances and emotional expressions. Through hierarchical feature extraction and end-to-end learning, CNNs capture basic acoustic features and abstract emotional representations, facilitating accurate emotion classification. Additionally, the use of 3D CNNs enables direct processing of temporal sequences, enhancing the model's ability to capture temporal dynamics and long-range dependencies in speech signals.

3.2.2 Advantages of proposed system

- **Integration of Diverse Features**

  By combining Mel frequency cepstral coefficients (MFCC), zero crossing rate (ZCR), and chroma features, the system captures both spectral and temporal characteristics of emotional speech signals comprehensively. This holistic approach enhances the model's ability to discern subtle emotional cues.

- **Utilization of Labeled Dataset**

Using the RAVDESS dataset, which offers a diverse set of labeled emotional speech samples, ensures robust training and evaluation of the SER system. Access to labeled data is crucial for supervised learning techniques to effectively learn and classify emotions.

- **Efficient Feature Extraction**

The use of MFCC, ZCR, and chroma features allows for efficient extraction of relevant information from speech signals. MFCC compresses spectral characteristics, ZCR captures temporal dynamics, and chroma encoding captures tonal content, collectively providing a rich feature representation.

- **Prevention of Overfitting**

Incorporating techniques such as dropout regularization and early stopping during training helps prevent overfitting, ensuring that the model generalizes well to unseen data. This enhances the system's robustness and performance on real-world datasets.

- **Scalability and Optimization**

The system's architecture and optimization strategies allow for scalability, making it suitable for processing large-scale datasets efficiently. Fine-tuning and optimization further refine the model's performance, improving its effectiveness in classifying emotions accurately.

3.2.3 Applications

In today's world, the proposed system for SER has diverse applications across various domains, offering opportunities for improving human-machine interactions, enhancing user experiences, and gaining valuable insights into emotional responses in different contexts

- **Human-Computer Interaction (HCI)**

Integration into HCI systems to enhance user experience by enabling devices to adapt responses based on detected emotions in speech input. For example, virtual assistants can tailor their responses or actions to better meet the user's emotional needs.

- **Sentiment Analysis Tools**

Deployment in sentiment analysis tools for analyzing customer feedback, social media content, or audio reviews. Businesses can gain insights into customer sentiment and tailor their products or services accordingly.

- **Emotional Computing**

Utilization in emotional computing applications for monitoring and analyzing emotional states in various contexts, such as healthcare, education, or entertainment. For instance, emotion-aware tutoring systems can adapt teaching strategies based on student emotional responses.

- **Educational Technology**

Integration into educational technology platforms for assessing student engagement and emotional responses during online learning sessions. Adaptive learning systems can adjust content delivery based on detected emotional states to enhance learning outcomes.

- **Healthcare and Well-being**

Implementation in healthcare applications for assessing emotional states in patients' speech, such as in telemedicine or mental health monitoring. The system can aid in early detection of emotional distress or mood disorders.

- **Affective Computing Systems**

Integration into affective computing systems for recognizing and responding to emotional cues in human-machine interactions. This includes applications in robotics, virtual reality, and gaming, where machines can perceive and respond to users' emotions in real-time.

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE DESIGN

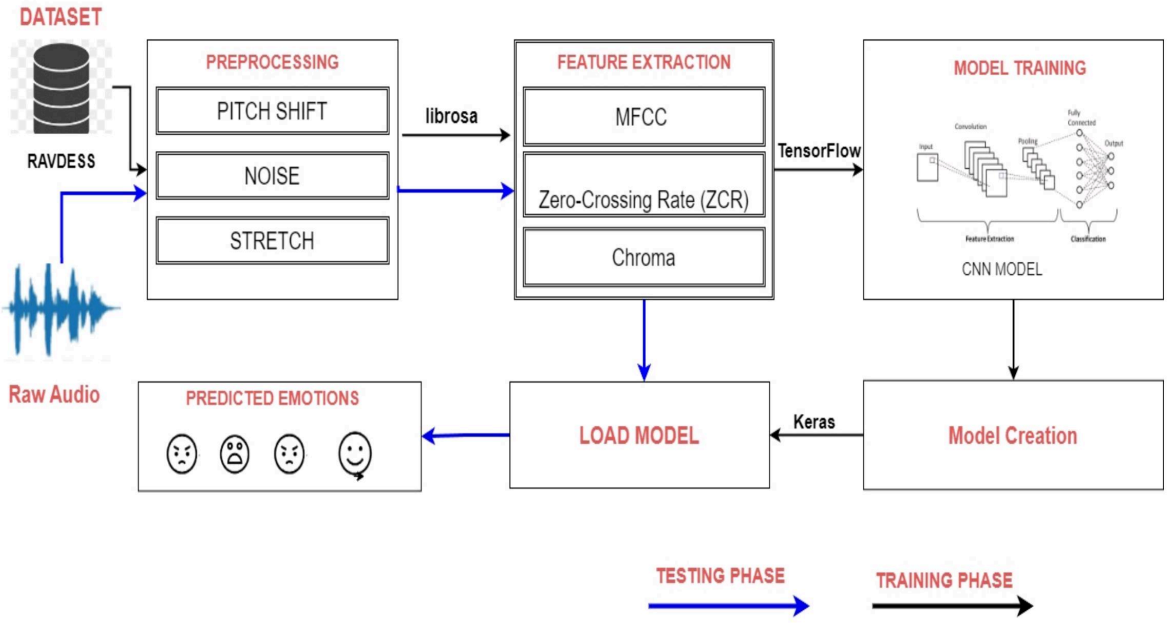


Fig. 4.1 System Architecture

The architecture for speech emotion recognition consists of two distinct phases: training and testing shown in Fig 3.2.1 During the training phase, machine learning models are trained using labeled datasets, and audio samples are preprocessed and feature-extracted using techniques such as MFCC, ZCR, and chroma features. Supervised learning techniques such as convolutional neural networks (CNN) are utilized to optimize model parameters with algorithms such as gradient descent. Performance evaluation is carried out using a validation dataset. During the testing phase, the trained model is used in real-time to analyze

incoming audio streams from both dataset audio and real application or system. The input audio samples undergo preprocessing and feature extraction, similar to the training phase. The extracted features are then used to make predictions regarding the emotional content of speech. After preprocessing, the model predicts the emotion from the speech data. The predicted emotion is then mapped to a human-readable label using a predefined mapping. These predictions can inform decisions, provide feedback, and trigger actions within your application or system.

4.2 UML DIAGRAM

Use case diagrams are usually referred to as behavior diagram. The Unified Modelling Language (UML) is a general-purpose, developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.

4.2.1 Use Case Diagram

Use case diagrams are usually referred to as behavior diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors). A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. It provides a high-level view of the functionalities or capabilities of the system and the ways in which actors interact with it to achieve specific goals or tasks. Use case diagrams are typically used during the early stages of software development to capture and communicate requirements. Use case diagrams help stakeholders, including developers, designers, and clients, to understand the functional requirements of a system, clarify the scope of the system, identify actors and their roles, and prioritize features based on user goals and interactions. They serve as a foundation for further detailed analysis, design, and implementation of the system.

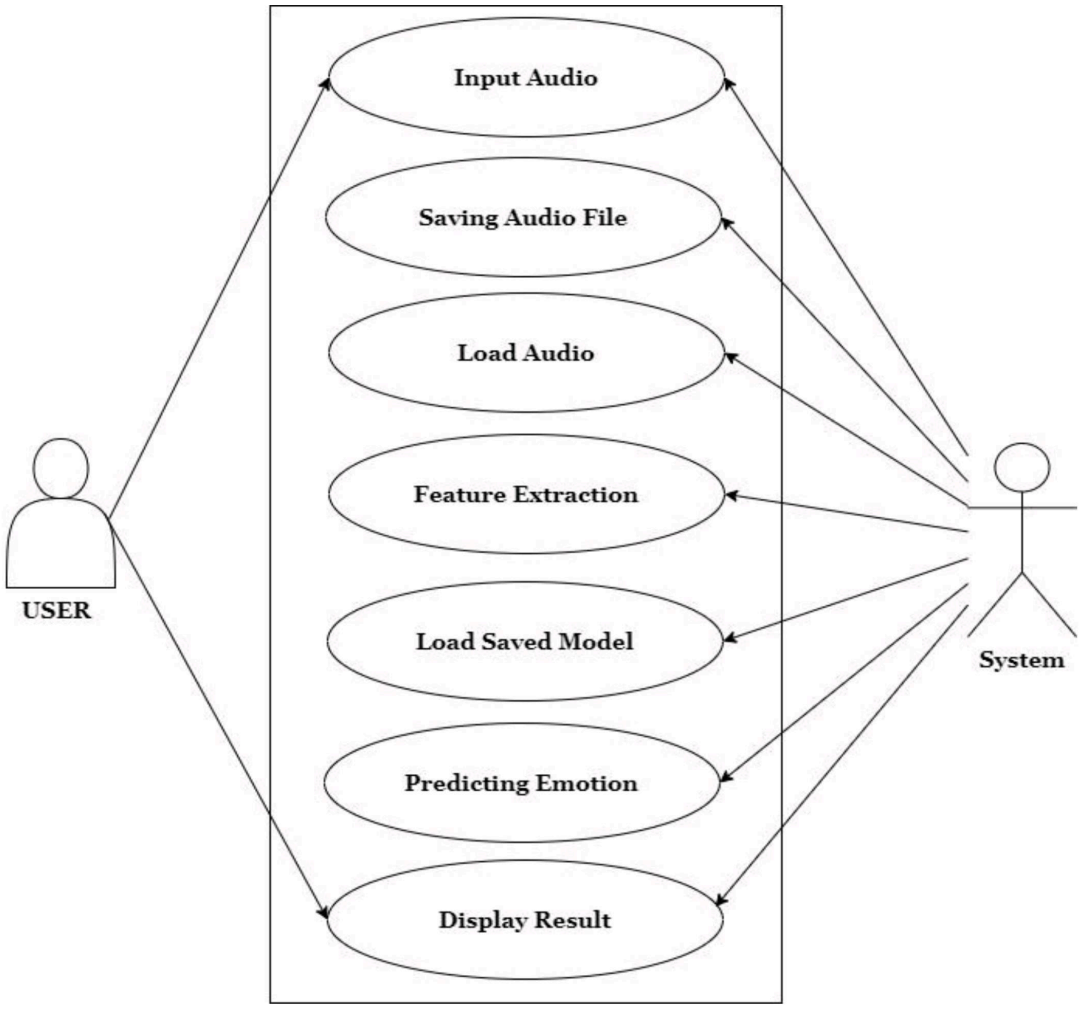


Fig. 4.2.1 Use Case Diagram

4.2.2 Class Diagram

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling.

In the diagram, classes are represented with boxes that contain three compartments:

- The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized.
- The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase.
- The bottom compartment contains the operations the class can execute.They are also left-aligned and the first letter is lowercase.

In the design of a system, a number of classes are identified and grouped together in a class diagram that helps to determine the static relations between them. With detailed modeling, the classes of the conceptual design are often split into a number of subclasses. In order to further describe the behavior of systems, these class diagrams can be complemented by a state diagram or UML state machine.

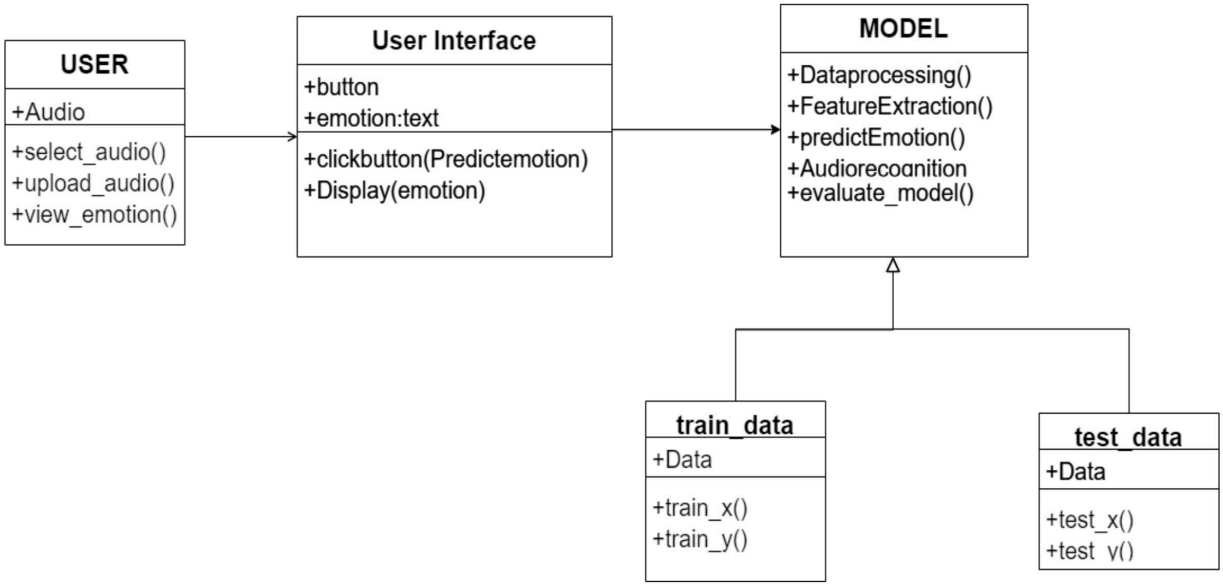


Fig. 4.2.2 Class Diagram

4.2.3 Activity Diagram

An activity diagram visually presents a series of actions or flow of control ina system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modeling. They can also describe the steps in a use case diagram. Activities modeled can be sequential and concurrent. The basic purpose of activity diagrams is similar to the other four diagrams. It captures the dynamic behavior of the system. An activity diagram is used to show message flow from one activity to another.

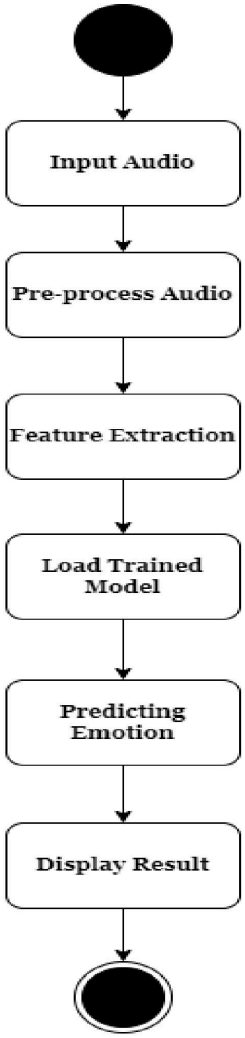


Fig. 4.2.3 Activity Diagram

4.2.4 Sequence Diagram

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

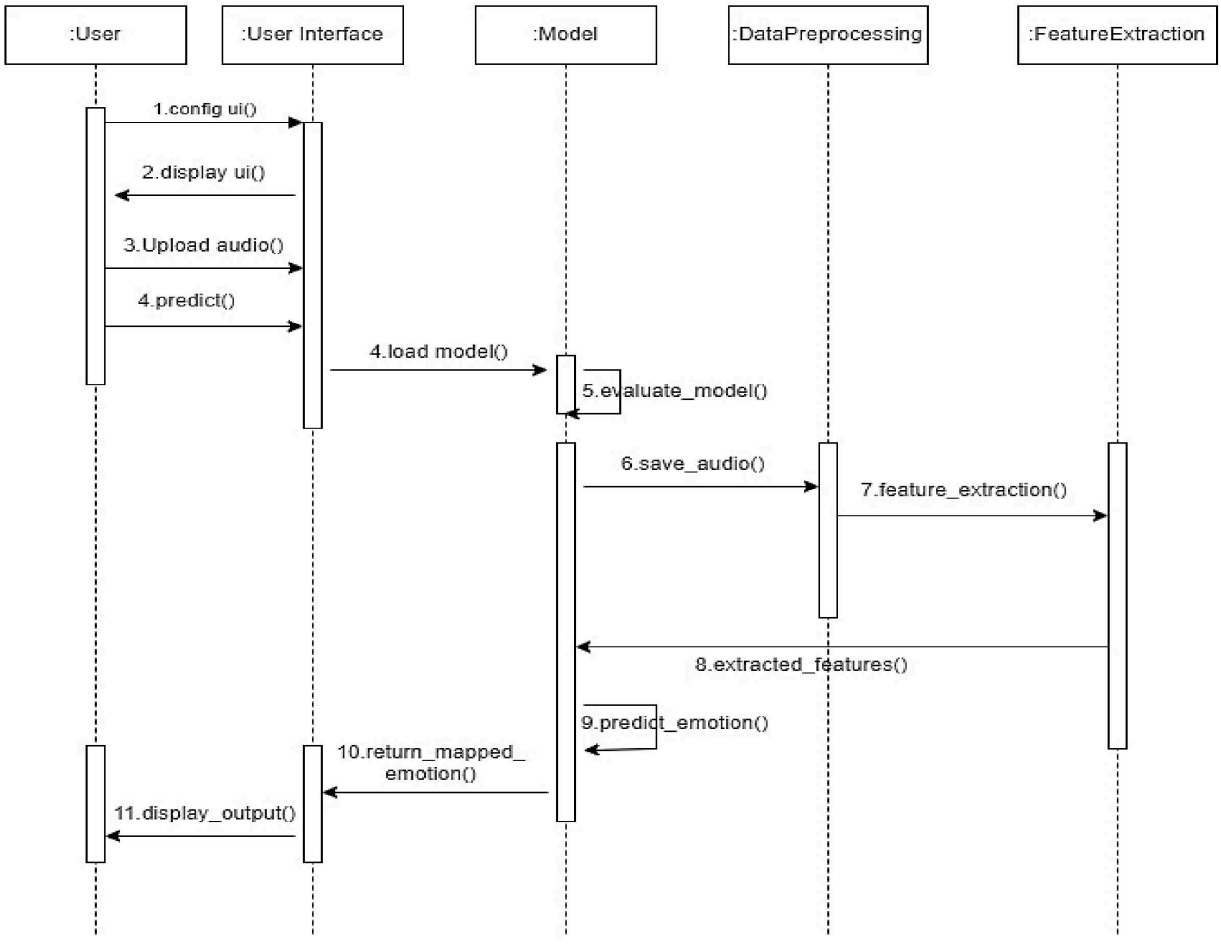


Fig. 4.2.4 Sequence Diagram

4.2.5 Component Diagram

The Unified Modelling Language, a component diagram depicts how components are wired together to form larger components or software systems. They are used to illustrate the structure of arbitrarily complex systems. In the Unified Modelling Language, a component diagram depicts how components are wired together to form larger components or software systems. They are used to illustrate the structure of arbitrarily complex systems.

A component is something required to execute a stereotype function. Examples of stereotypes in components include executable, documents, database tables, files, and library files. Components are wired together by using an assembly connector to connect the required interface of one component with the provided interface of another component.

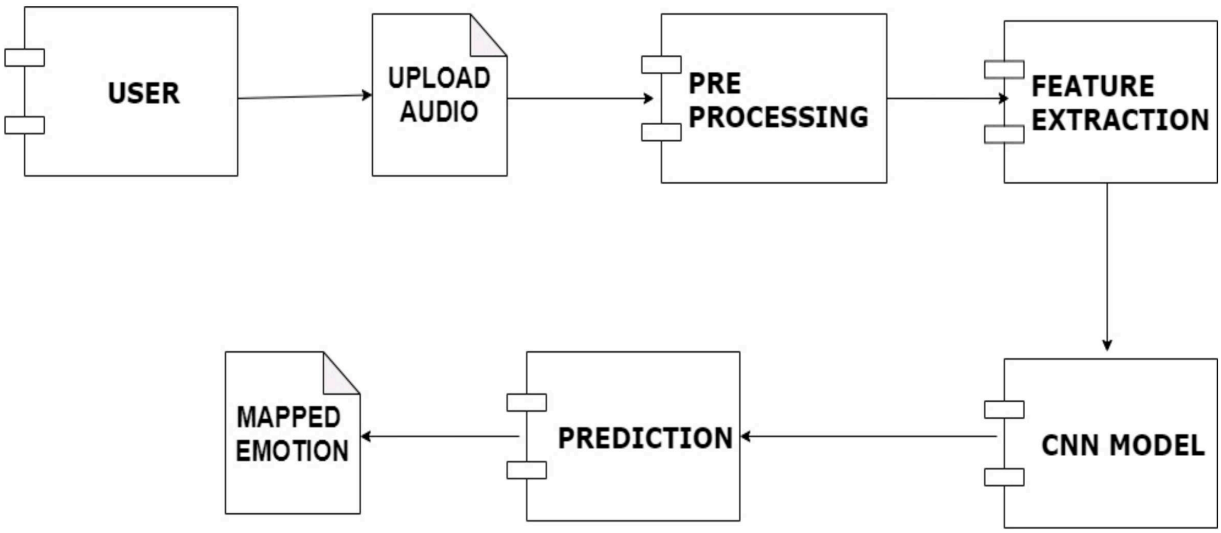


Fig. 4.2.5 Component Diagram

4.2.6 Deployment Diagram

Deployment diagram is a structure diagram which shows architecture of the system as deployment (distribution) of software artifacts to deployment targets. Artifacts represent concrete elements in the physical world that are the result of a development process. A deployment diagram in the Unified Modelling Language models the physical deployment of artifacts on nodes.

The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub-node, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.

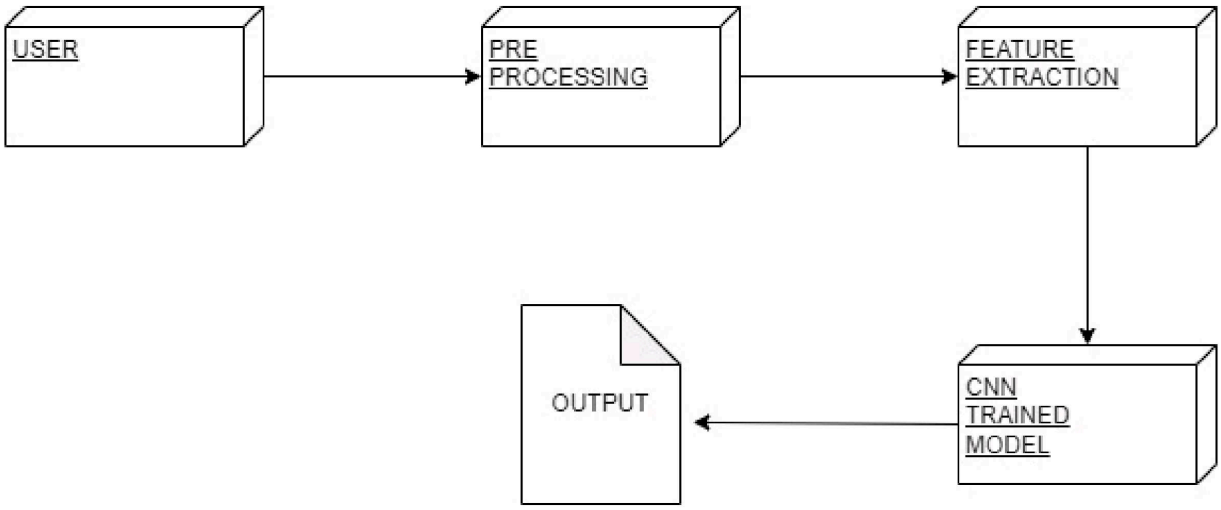


Fig. 4.2.6 Deployment Diagram

4.3 DATA FLOW DIAGRAM

The DFD is also called a bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.

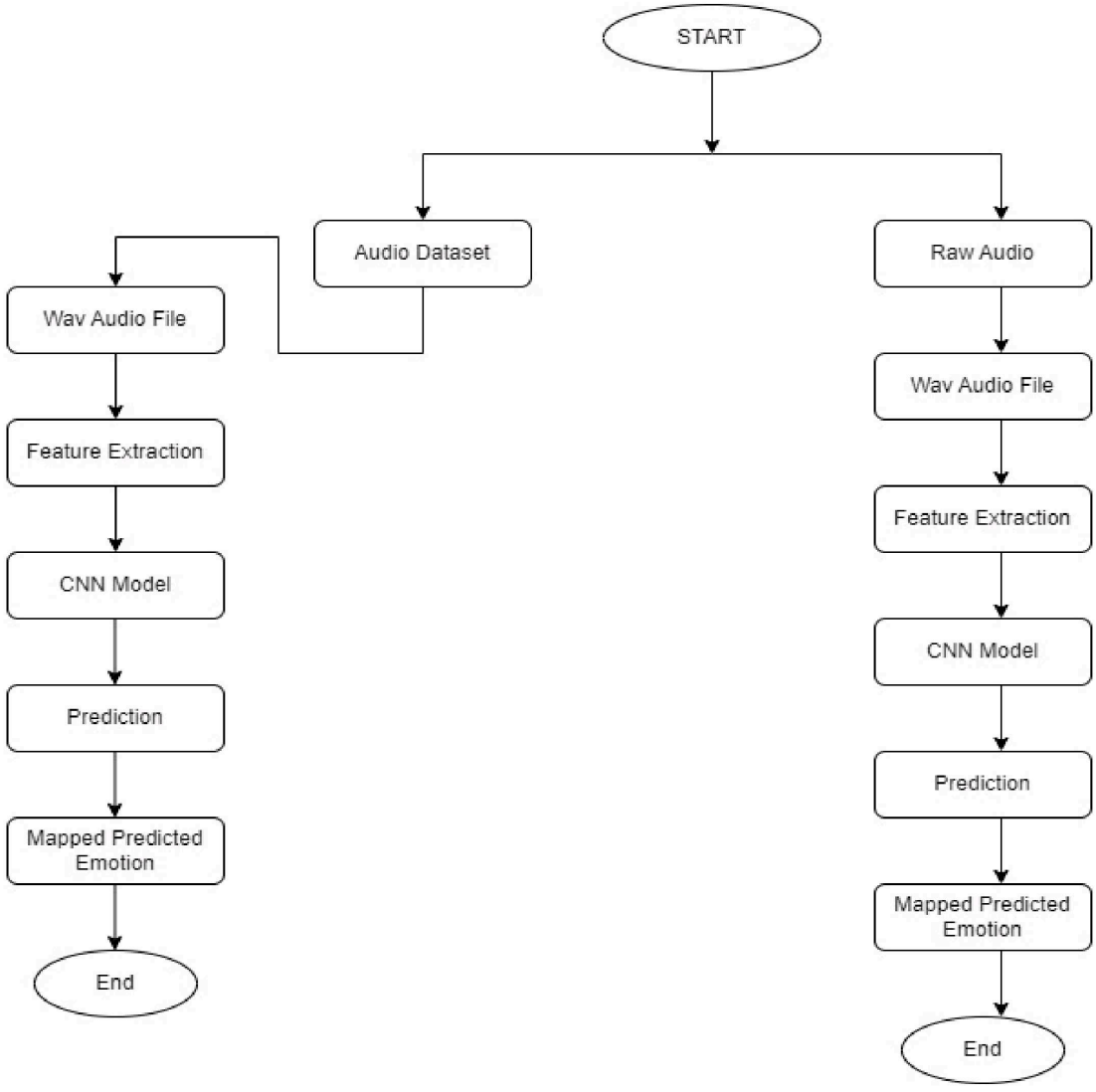


Fig. 4.3 Data Flow Diagram

CHAPTER 5

SYSTEM SPECIFICATION

5.1 HARDWARE SPECIFICATION

- Processor : AMD Ryzen 3
- RAM : 8GB
- Hard disk : 8GB and more

5.2 SOFTWARE SPECIFICATION

- Language : Python3, HTML, CSS.
- IDE : VS Code, Jupyter Notebook
- Framework : Flask

CHAPTER 6

SOFTWARE DESCRIPTION

6.1 PYTHON

Python is a high-level programming language that is commonly used in human speech emotion recognition projects. It is a versatile language that offers a wide range of libraries and frameworks for speech processing, machine learning, and data analysis.

Features of Python

- Easy to learn: Python is known for its simplicity and ease of use. Its syntax is easy to read and write, making it an ideal choice for beginners.
- Rich Library Ecosystem: Python offers a rich library ecosystem for speech processing, machine learning, and data analysis, such as Speech Recognition, PyAudio, Praat, TensorFlow, Keras, and Scikit-Learn.
- Cross-Platform Compatibility: Python code can run on different operating systems such as Windows, Linux, and macOS, making it a versatile choice for human speech emotion recognition projects.

Overall, Python is a popular and effective choice for human speech emotion recognition projects due to its ease of use, rich library ecosystem, and strong community support.

6.2 Librosa

Librosa is a popular Python library for audio and music analysis that can be used in Speech Emotion Recognition (SER) projects. It provides tools for audio feature extraction, manipulation, and visualization, making it widely used in various applications such as music information retrieval, sound classification, speech emotion recognition, and audio signal processing.

Applications of Librosa

- Music Information Retrieval (MIR): Librosa is extensively used in MIR tasks such as music genre classification, chord recognition, and music transcription. It provides tools for feature extraction, spectrogram computation, and onset detection, facilitating the analysis of audio signals.
- Audio Signal Processing: Librosa offers functionalities for audio signal processing tasks like time-frequency analysis, spectral decomposition, and filtering. It enables researchers and practitioners to analyze and manipulate audio signals for tasks such as speech processing, sound event detection, and audio scene analysis.
- Sound Visualization: Librosa enables the visualization of audio signals, spectrograms, chromagrams, and other audio representations. It helps researchers and practitioners visualize and interpret audio data, aiding in the analysis and understanding of complex audio phenomena.

6.3 NumPy

NumPy is a Python library for scientific computing that provides tools for working with arrays and matrices. It is a fundamental library for many scientific and data analysis tasks in Python, including Speech Emotion Recognition (SER) projects.

Applications of Numpy

- Numerical Operations: NumPy facilitates efficient array manipulation and mathematical functions for numerical operations, making it essential for handling large datasets.
- Data Analysis: Widely used alongside libraries like Pandas, NumPy enables data manipulation, cleaning, and visualization tasks, serving as the backbone for tabular data operations.

- Signal Processing: NumPy provides tools for processing signals and images efficiently, enabling tasks such as filtering, convolution, and spectral analysis in fields like digital signal processing and computer vision.

6.4 Pandas

Pandas is used for data preprocessing, feature engineering, analysis, and integration with machine learning pipelines. It is widely used for data manipulation, data cleaning, and data analysis tasks, making it one of the most popular libraries in the Python ecosystem for working with structured data. It efficiently manages and manipulates large datasets, facilitating the extraction of meaningful features from audio data and enabling rapid experimentation with SER models.

Application of pandas

- Data Organization: Pandas can be used to organize metadata related to speech samples, such as speaker IDs, recording timestamps, and emotional labels. It allows for efficient indexing, filtering, and sorting of speech data, enabling researchers to manage large datasets effectively.
- Data Augmentation: Pandas can be employed to generate augmented speech data by applying transformations such as time stretching, pitch shifting, and adding noise. By creating variations of existing speech samples with different emotional expressions, Pandas helps in expanding the diversity of the training dataset, improving the robustness of SER models.
- Model Evaluation: Pandas facilitates the analysis of model predictions and evaluation metrics such as accuracy, precision, recall, and F1-score. It allows for easy comparison of predicted emotions with ground truth labels, enabling researchers to assess the performance of SER models and identify areas for improvement.

6.5 TensorFlow

TensorFlow is a leading machine learning framework by Google. In SER, it's used to build, train, and deploy deep learning models efficiently. With TensorFlow, researchers can develop custom neural network architectures tailored to SER tasks, train models on large datasets, and deploy them in various environments. Its integration with Python libraries facilitates seamless workflow from data preprocessing to model deployment.

Applications of Tensorflow

- Model Development: TensorFlow provides a flexible and scalable platform for developing deep learning models for SER. Researchers and practitioners can use TensorFlow to design and train various neural network architectures, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformers, tailored to the task of emotion recognition from speech.
- Feature Extraction: TensorFlow can be employed to extract deep learned features from raw speech data using pre-trained models such as convolutional or recurrent neural networks. These deep features capture high-level representations of speech signals, enabling more effective emotion recognition.
- Model Optimization: TensorFlow provides tools for model optimization and deployment, including techniques for model compression, quantization, and optimization for inference on resource-constrained devices. This is particularly useful for deploying SER models on edge devices or mobile platforms.

6.6 FLASK

Flask is a lightweight and popular web framework for Python, designed to build web applications quickly with minimal code. It offers routing, templating, and easy handling of HTTP requests and responses. Flask's flexibility and modular design make it a preferred choice for small to medium-sized web projects and prototypes.

Features of Flask

- Lightweight and flexible: Flask is a lightweight framework that does not impose any particular way of doing things, making it easy to customize and adapt to different projects.
- Simple and easy to use: Flask has a simple and intuitive syntax that is easy to understand and use, even for beginners.
- Modular design: Flask is designed to be modular, with different components that can be used independently or together, depending on the needs of the project.
- Extensible: Flask can be extended with third-party extensions and plugins, which can add additional functionality to the framework.

Applications of Flask

- Building RESTful APIs: Flask is a popular choice for building RESTful APIs, with support for JSON, XML, and other data formats.
- Creating web applications: Flask can be used to create web applications of all types, including blogs, e-commerce sites, and social networks.
- Developing microservices: Flask's lightweight and modular design make it well-suited for developing microservices, which can be combined to create larger applications.

6.7 HTML

HTML stands for Hypertext Markup Language. It is the standard markup language used for creating and structuring content on the World Wide Web (WWW). HTML uses a system of tags and attributes to define the structure and presentation of web pages. These tags are interpreted by web browsers to display text, images, videos, links, and other multimedia elements on webpages.

Features of HTML

- Structure and formatting: HTML provides a way to structure and format web content, allowing developers to create headings, paragraphs, lists, and other elements that make it easy to read and navigate.
- Hypertext links: HTML allows developers to create hypertext links, which allow users to navigate between web pages and other online documents.
- Multimedia support: HTML supports the inclusion of multimedia content, such as images, videos, and audio files.
- Accessibility: HTML includes features that make it accessible to users with disabilities, such as alt tags for images and other multimedia content.

Applications of HTML

- Creating and designing web pages: HTML is used to create and design web pages of all types, including blogs, e-commerce sites, and social networks.
- Developing mobile applications: HTML is used in conjunction with other technologies, such as CSS and JavaScript, to develop mobile applications that run on a variety of platforms.
- Responsive Web Design: HTML plays a crucial role in implementing responsive web design.HTML elements can be styled and organized using CSS (Cascading Style Sheets) to create flexible and responsive layouts.

6.8 CSS

CSS stands for Cascading Style Sheets. It is a style sheet language used to control the presentation and layout of HTML documents. CSS allows web developers to apply styles and formatting to web pages, including fonts, colors, margins, padding, and positioning of elements.

Features of CSS

- Separation of content and presentation: CSS allows developers to separate the presentation of a web page from its content, making it easier to maintain and update.
- Style inheritance: CSS allows developers to apply styles to multiple elements at once, making it easy to create consistent styles across a website.
- Responsive design: CSS allows developers to create responsive designs that adjust to different screen sizes and devices.
- Advanced styling options: CSS includes a wide range of styling options, including color, font, layout, and animation.

Applications of CSS

- Styling web pages: CSS is used to style web pages of all types, including blogs, e-commerce sites, and social networks.
- Creating responsive designs: CSS is used to create responsive designs that adjust to different screen sizes and devices, making it easy to create mobile-friendly websites.
- Building web-based applications: CSS is used in conjunction with other technologies, such as HTML and JavaScript, to build web-based applications that are interactive and dynamic.

CHAPTER 7

SYSTEM IMPLEMENTATION

7.1 MODULES DESCRIPTION

7.1.1 Data Collection

In advancing speech emotion recognition (SER), this system focuses on leveraging the potential of the Ryerson Audio-Visual Database of Emotional Speech (RAVDESS). This part of RAVDESS contains 1440 files: 60 trials per actor x 24 actors = 1440. RAVDESS is made up of 24 professional actors (12 women, 12 men) who pronounce his two lexically matching statements in Neutral North American accents. Verbal emotions include expressions of happy, sad, anger, fear, surprise, neutral and disgust. Each expression is generated with two emotional intensities (usually strong), and an additional neutral expression is added. This inherent diversity allows our model to capture a wide range of emotional nuances, contributing to the effectiveness of understanding and classifying emotions in language. It aims to improve the model's generalization ability across different emotional contexts and ultimately improve its performance in real-world applications.

Emotions	Count of audio samples
Anger	192
Happy	192
Disgust	192
Angry	192
Fear	192
Sad	192
Neutral	288

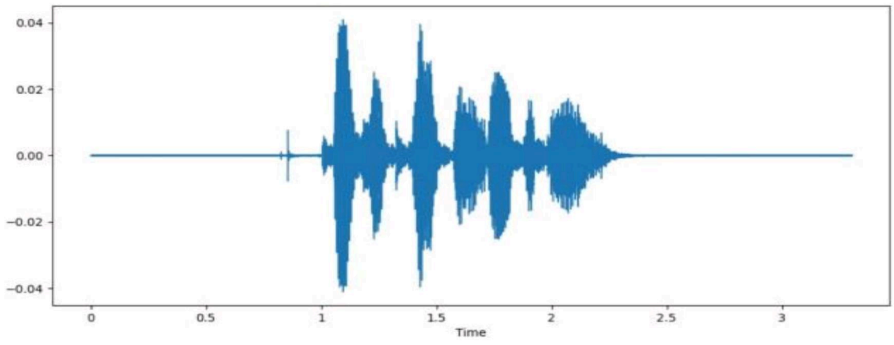
Table 7.1.1: RAVDESS audio samples

7.1.2 Data Preprocessing

Data augmentation is an important step in using the RAVDESS dataset to improve the diversity and robustness of speech emotion recognition (SER) datasets. Using the Librosa library implementation, it applies various transformations to the audio data, such as pitch shifting, time stretching, and adding noise, to improve the generalization of the model. Different acoustic conditions and emotional expressions can be simulated, allowing the model to better generalize to unseen instances.

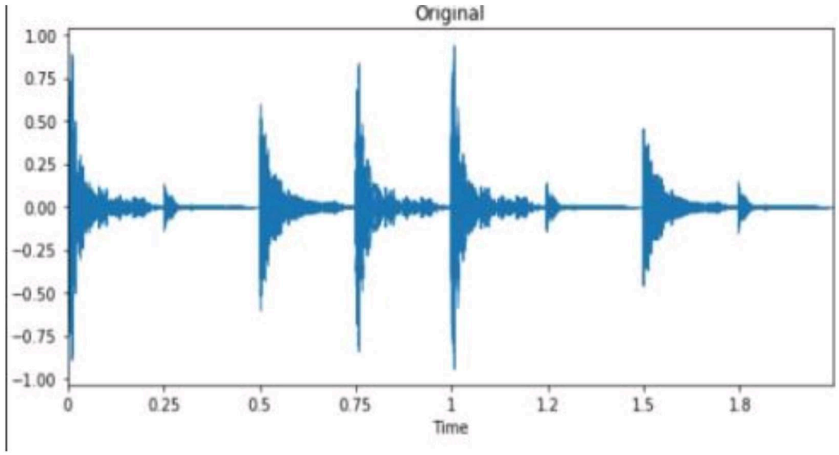
I. Pitch shifting

Pitch shifting is a fundamental technique in speech emotion recognition (SER) that allows the frequency characteristics of an audio signal to be changed while preserving its temporal characteristics. In SER, pitch shifting serves as a data augmentation method to increase the diversity of the training data set. The SER model detects pitch variations by applying a pitch shift to the original speech sample, allowing it to better generalize to different speech features and emotional expressions. This technique adjusts the audio signal's frequency up or down without affecting its duration. Pitch shifting implemented in Python through signal processing libraries such as Librosa allows researchers to simulate changes in vocal tone, contributing to a more comprehensive and robust training dataset for the SER model.



II. Time stretching:

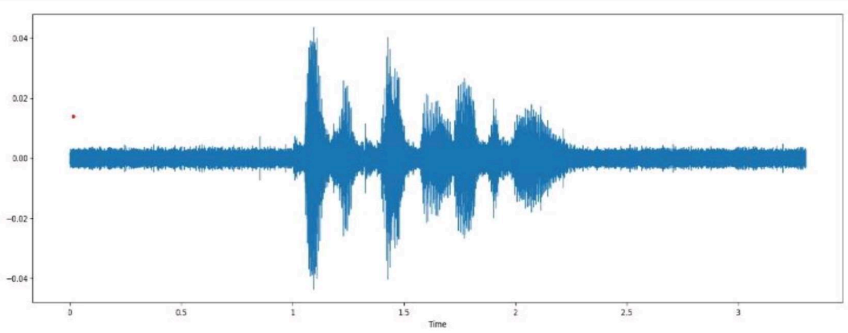
Time stretching facilitates the modification of the duration of speech signals while preserving their pitch or frequency content. Within SER, time stretching serves as a crucial data augmentation method, enriching the training dataset's variability. By adjusting speech samples' duration, time stretching allows SER models to learn from a broader spectrum of temporal patterns and speech rhythms, ultimately enhancing their capacity to identify emotional expressions across diverse speech speeds and styles. This technique is typically implemented through signal processing libraries like Librosa in Python.



III. Adding Noise:

Adding noise is a common technique in audio signal processing that introduces random fluctuations to an audio signal. This acts as a data augmentation method that increases the robustness of the training dataset. By injecting random noise into audio samples, researchers can simulate real-world environmental conditions and acoustic distortions, making SER models more robust to noise in real-world applications. This technique superimposes random noise, typically generated from a Gaussian distribution, onto the original audio signal. The added noise level can be controlled to achieve the desired

amplification level. SER models gain accuracy and generalization ability by adding noise and learning how to distinguish emotional data from background and other noise.

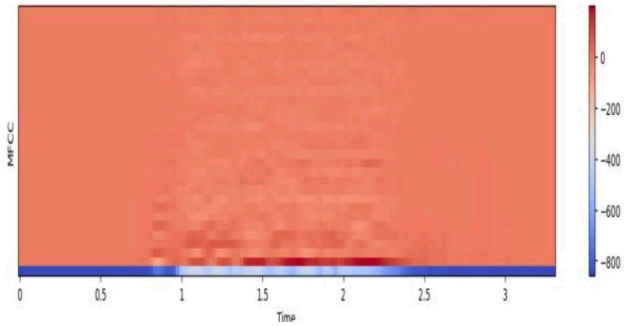


7.1.3 Feature Extraction:

A variety of Python libraries, including librosa, Numpy, Pandas, and Matplotlib, are used for feature extraction from audio recordings because of their user-friendly interface and extensive capabilities. For feature extraction from audio samples, load the audio file using librosa and extract various features such as Mel Frequency Cepstral Coefficient (MFCC), Zero Crossing Rate (ZCR), Chroma, Root Mean Square Energy (RMSE).

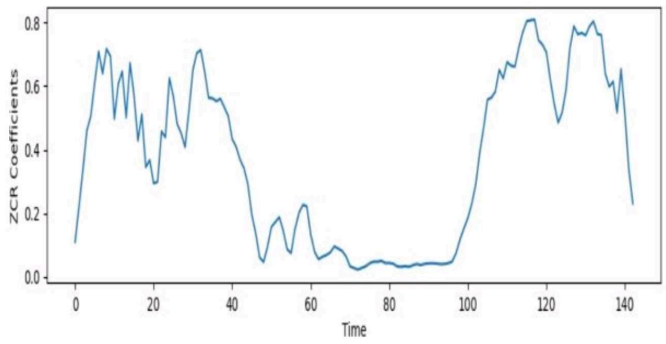
I. Mel-frequency Cepstral coefficients (MFCC):

The MFCC represents the short-term power spectrum of a sound, captures the spectral envelope of an audio signal, and is often used as a feature in speech and audio processing tasks for capturing the acoustic signal.



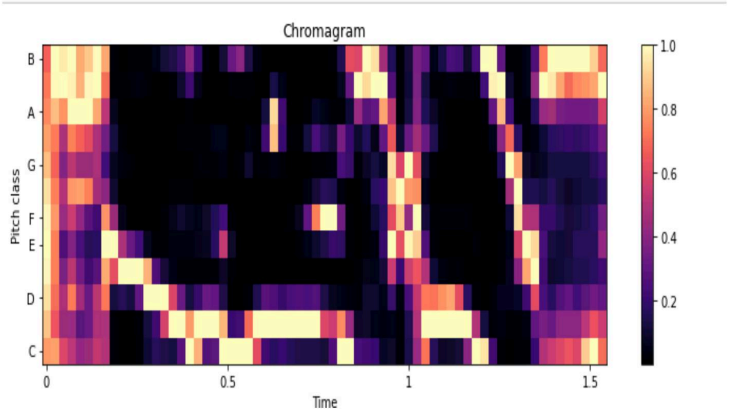
II. Zero Cross Rate (ZCR):

ZCR measures how quickly the sign of an audio signal changes and provides information about the frequency content and periodicity of the signal. This is useful for tasks such as speech onset recognition and rhythmic analysis.



III. Chroma:

The chroma function describes the distribution of pitch classes in an audio signal that is invariant to changes in timbre or octave, and is useful for tasks involving analysis of harmonic content, such as musical genre classification and chord recognition.



The extracted features were converted into a Pandas Data Frame in tabular format to facilitate further analysis.

	0	1	2	3	4	5	6	7	8	9	...	2257	2258	2259	2260	2261
0	0.028809	0.042969	0.057129	0.052734	0.065430	0.062988	0.064941	0.063965	0.049316	0.044922	...	8.507217	8.082827	8.828318	8.973294	2.93084
1	0.046387	0.061523	0.076660	0.054688	0.068359	0.065918	0.069824	0.077637	0.087402	0.132812	...	8.507217	8.082827	8.828318	8.973294	2.93084
2	0.030273	0.043457	0.056152	0.057129	0.062500	0.062012	0.065918	0.066895	0.070801	0.069336	...	8.507217	8.082827	8.828318	8.973294	2.93084
3	0.035156	0.048340	0.061035	0.056152	0.061523	0.061035	0.064941	0.067871	0.078613	0.087891	...	8.507217	8.082827	8.828318	8.973294	2.93084
4	0.023926	0.036133	0.049805	0.053711	0.060547	0.066895	0.077637	0.087891	0.092285	0.089844	...	8.507217	8.082827	8.828318	8.973294	2.93084
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
11195	0.016113	0.044922	0.064453	0.074707	0.081055	0.061035	0.050781	0.036133	0.031250	0.033691	...	8.507217	8.082827	8.828318	8.973294	2.93084
11196	0.020996	0.029297	0.038086	0.029785	0.032715	0.032715	0.035156	0.035156	0.031738	0.029785	...	8.507217	8.082827	8.828318	8.973294	2.93084
11197	0.025879	0.034180	0.043945	0.032715	0.034668	0.043457	0.046875	0.046875	0.042480	0.031738	...	8.507217	8.082827	8.828318	8.973294	2.93084
11198	0.022461	0.028809	0.035156	0.034180	0.041016	0.060547	0.075195	0.069336	0.059570	0.040527	...	8.507217	8.082827	8.828318	8.973294	2.93084
11199	0.026367	0.032715	0.040039	0.032227	0.043945	0.069336	0.095703	0.091797	0.078125	0.053223	...	8.507217	8.082827	8.828318	8.973294	2.93084

11200 rows × 2267 columns

7.2 MODEL TRAINING USING CNN

7.2.1 Model Configuration

Convolutional neural network (CNN) architectures tailored for audio emotion recognition are highly integrated with different components, each of which plays a critical role in accurately classifying emotions from audio signals. At the core of a CNN, an input layer is used to receive a concatenated feature vector extracted from an audio sample, such as mel- frequency cepstral coefficients (MFCC), zero-crossing rate (ZCR), and chroma features. These feature vectors provide a comprehensive representation of the spectral and temporal speech characteristics of the audio signal. Successive convolutional layers form the backbone of the CNN and perform feature extraction by convolving the input feature map with a learnable filter. These filters capture different patterns and structures in the input data and help the model detect subtle differences in emotional expressions. Activation functions such as Rectified Linear Units (Relu) introduce nonlinearity to the model, allowing it to learn complex mappings between input features and emotion classes. The pooling layer plays an important role in down sampling the feature map created by the convolutional layer, reducing overfitting and improving computational efficiency by preserving the most salient features while reducing the spatial dimension. The fully connected layer serves as

the final stage of the CNN. The extracted features are integrated and classified into emotion classes. This layer uses the hierarchical representation learned from previous layers to make informed predictions. Regularization techniques such as dropout and batch normalization prevent overfitting during training and improve the model's generalization performance on unseen data. Optimization algorithms such as stochastic gradient descent (SGD) using Adam and Momentum efficiently update model parameters to minimize the loss function, allowing CNNs to accurately classify emotions based on audio signals. The output layer uses softmax activations to compute the probability distribution between emotion classes and provides a measure of the model's confidence in its predictions.

7.2.2 Model Compilation

- **Optimizer Algorithm (Adam):** Adam adjusts model parameters based on gradient magnitudes, combining RMSProp and Momentum.
- **Learning Rate (0.001):** Step size for weight updates during training, affecting convergence speed and model performance.
- **Batch Size (32):** Number of training examples processed together, influencing memory usage and convergence speed.
- **Number of Epochs (50 to 100):** Total passes through the dataset during training, impacting model convergence and generalization.
- **Dropout (0.5 or 50%):** Regularization technique randomly dropping input units during training to prevent overfitting.
- **Loss Function (Categorical Cross Entropy):** Measures model performance by comparing predicted and actual outputs in classification tasks.
- **Activation Functions:** Introduce non-linearity to neural networks, enabling them to learn complex patterns and produce meaningful outputs. (ReLU for Hidden Layer, Softmax for Output Layer).

7.2.3 Model Training

- Train the model on the training dataset using the compiled configuration. During training, the model learns to classify audios into different emotional classes by adjusting the parameters of the additional layers.
- Iterate over batches of training data, passing them through the model, computing the loss, and updating the model parameters using backpropagation.

7.2.4 Model Evaluation

- Evaluate the trained model's performance on the testing dataset to assess its generalization ability. Compute metrics such as accuracy, precision, recall, and F1-score to measure how well the model classifies speech emotions.
- Analyze the model's predictions, including any misclassifications, to identify areas for improvement and potential biases.

7.3 PREDICTION USING MAPPED EMOTION

To predict the audio emotion with CNN model using mapped emotion, we start by accessing a pre-trained CNN model previously saved on disk. Through functions like load_model() in TensorFlow or pickle.load() in Python, we bring this model back into memory. Next, we prepare audios of different emotions for analysis. This involves preprocessing the audio to ensure they're in a format suitable for CNN model using functions like preprocess_input(). Once the audio are preprocessed, they're inputted into the CNN model for inference. This step utilizes the model's learned features to make predictions about different emotions. The model assesses each audio and predicts the probabilities of different emotion classes using functions like predict(). By analyzing these probabilities, we can classify the emotion in terms of human readable form based on mapped emotion and severity with a high level of accuracy.

CHAPTER 8

RESULTS AND DISCUSSION

8.1 EVALUATION METRICS

Assessing the effectiveness of statistical, machine learning, and deep learning models requires employing a range of evaluation criteria. These indicators are critical in measuring the effectiveness of a suggested model in a study. Evaluation metrics including as accuracy, precision, recall, and the F1-score are important for determining a model's prediction or classification efficacy.

8.1.1 ACCURACY

Accuracy, expressed as a percentage, signifies the proportion of images that are accurately predicted among all predictions made. Equation defines accuracy as:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

The accuracy score compares the model's total number of correct predictions (TP + TN) against its total number of predictions. The initials TP, TN, FP, and FN represent "true positive," "true negative," "false positive," and "false negative" accordingly.

8.1.2 PRECISION

The precision value, indicating the ratio of true positive outcomes among the predicted positive instances, is calculated using this equation:

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

8.1.3 RECALL

The ratio of true positive findings to the total number of True Positive and False Negative samples (TP + FN). The accuracy of positive predictions given by recall value. The following equation will be defined by

$$Recall\ or\ TPR = \frac{TP}{TP + FN}$$

8.1.4 F1-SCORE

F1-Score combines the precision and recall of a classifier into a single metric by taking their harmonic mean. It is primarily used to compare the performance of multiclassification. It calculated as follows

$$F1\ Score = 2 \times \frac{Recall \times Precision}{Recall + Precision}$$

	Precision	**Recall**	**F1-Score**
Angry	0.96	0.94	0.95
Disgust	0.96	0.94	0.95
Fear	0.97	0.93	0.95
Happy	0.92	0.96	0.94
Neutral	0.95	0.99	0.97
Surprise	0.98	0.97	0.97
Sad	0.98	0.95	0.97

Table 8.1: Evaluation metrics for emotions

8.2 RESULT ANALYSIS

In the project focused on speech emotion recognition, the CNN model achieving maximum accuracy of over 95.05% through the evaluation metrics using the Training and Testing Graph for Accuracy and loss 8.2.1, classification report

for CNN figure 8.2.2. In contrast, the MLP model reached an 80% accuracy rate, followed by SVM with 75%.The CNN model's superior accuracy in discerning speech emotions highlights its effectiveness in this domain, offering promising prospects for applications requiring emotion detection from audio inputs. Classification report for MLP figure 8.2.3, SVM figure 8.2.4 describes the model evaluation metrics performance of each model Figure 8.2.5.

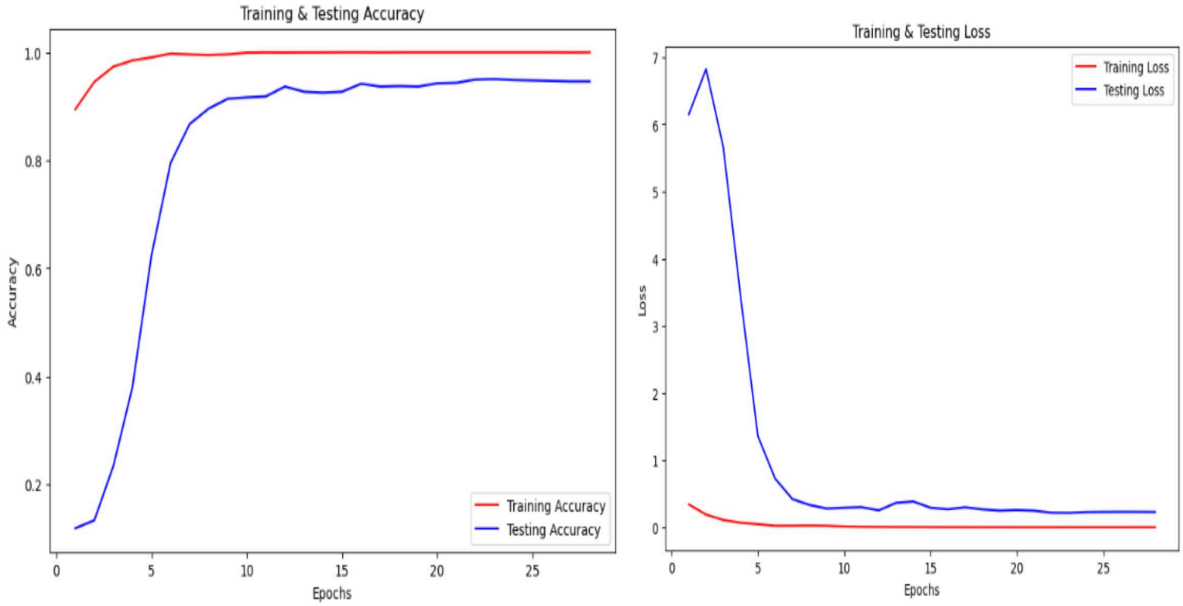


Fig 8.2.1 Training & Testing Accuracy and Loss graph for CNN

	precision	recall	f1-score	support
angry	0.95	0.93	0.94	139
disgust	0.96	0.94	0.95	162
fear	0.96	0.94	0.95	138
happy	0.93	0.91	0.92	157
neutral	0.93	0.99	0.96	246
sad	0.97	0.96	0.97	163
surprise	0.97	0.95	0.96	147
accuracy			0.95	1152
macro avg	0.95	0.95	0.95	1152
weighted avg	0.95	0.95	0.95	1152

Fig 8.2.2 Classification Report for CNN

	precision	recall	f1-score	support
angry	0.94	0.86	0.90	90
happy	0.81	0.80	0.80	94
neutral	0.70	0.73	0.71	44
sad	0.81	0.86	0.83	101
accuracy			0.82	329
macro avg	0.81	0.81	0.81	329
weighted avg	0.83	0.82	0.82	329

Fig 8.2.3 Classification Report for MLP

	precision	recall	f1-score	support
angry	0.78	0.79	0.78	90
happy	0.73	0.70	0.71	94
neutral	0.65	0.73	0.69	44
sad	0.80	0.77	0.78	101
accuracy			0.75	329
macro avg	0.74	0.75	0.74	329
weighted avg	0.75	0.75	0.75	329

Fig 8.2.4 Classification Report for SVM

Table 8.2.5

Model evaluation metrics performance

MODEL	RECALL	PRECISION	F1-SCORE	ACCURACY
CNN	0.952	0.94	0.95	0.95
MLP	0.81	0.81	0.81	0.82
SVM	0.74	0.75	0.74	0.75

CHAPTER 9

CONCLUSION & FUTURE ENHANCEMENTS

9.1 CONCLUSION

Speech Emotion Recognition (SER) is revolutionizing human-computer interaction by decoding emotional nuances in speech through advanced audio analysis and machine learning. Its applications span healthcare, education, customer service, and entertainment. Powered by Python libraries like librosa and NumPy, SER extracts key features such as MFCC and ZCR to capture subtle vocal patterns. In healthcare, emotion-aware virtual assistants promise to transform mental health support. SER-driven personalized learning enhances educational experiences, while sentiment analysis in customer service ensures empathetic interactions. In entertainment, SER creates immersive experiences, and in market research, it guides targeted strategies. The future of SER is bright, driven by advancements in machine learning and affective computing, fostering deeper human-technology connections.

9.2 FUTURE ENHANCEMENTS

Speech emotion recognition (SER) includes several important directions, including consideration of multimodal approaches that integrate audio, video, and text data to improve emotion recognition accuracy. Deep learning architectures such as recurrent neural networks (RNNs) and transformer models promise improved SER performance. Transfer learning and domain adaptation techniques facilitate the generalization of the SER model to new domains or languages with limited labelled data. Real-time emotion recognition systems, contextual emotion recognition models, and advances in emotion understanding and generation are also important areas of focus. Additionally, consideration of ethical and social implications such as privacy, bias, and fairness are critical for the responsible

development and use of his SER technology in a variety of applications. Through these research directions, the future of SER aims to advance the state of the art and enable more empathetic and emotionally intelligent human-computer interactions.

APPENDICES

APPENDIX 1

SCREENSHOTS

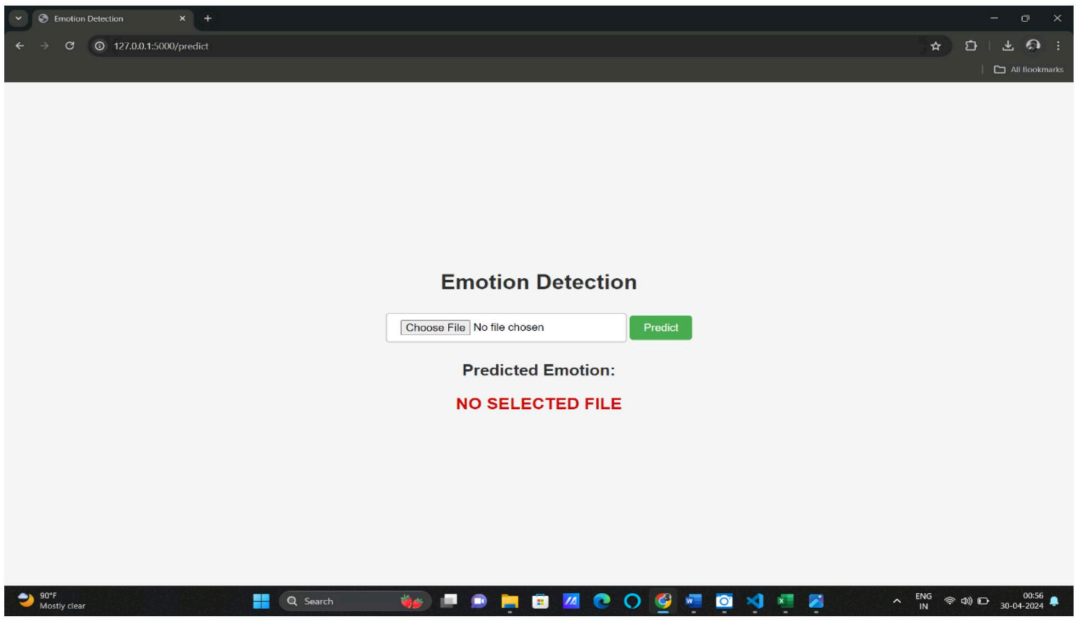


Fig. A.1.1 User Interface

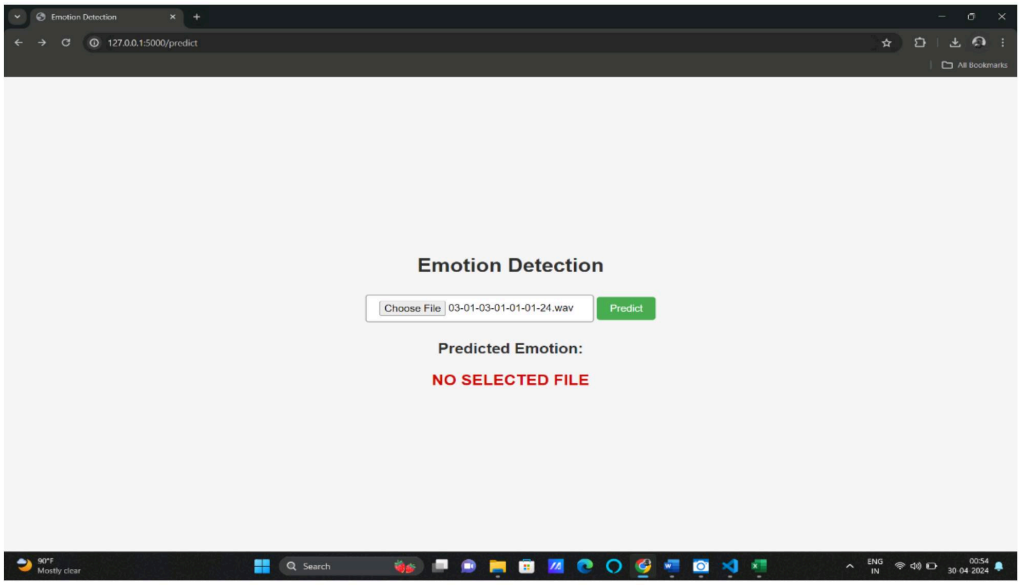


Fig. A.1.2 Upload the audio input file

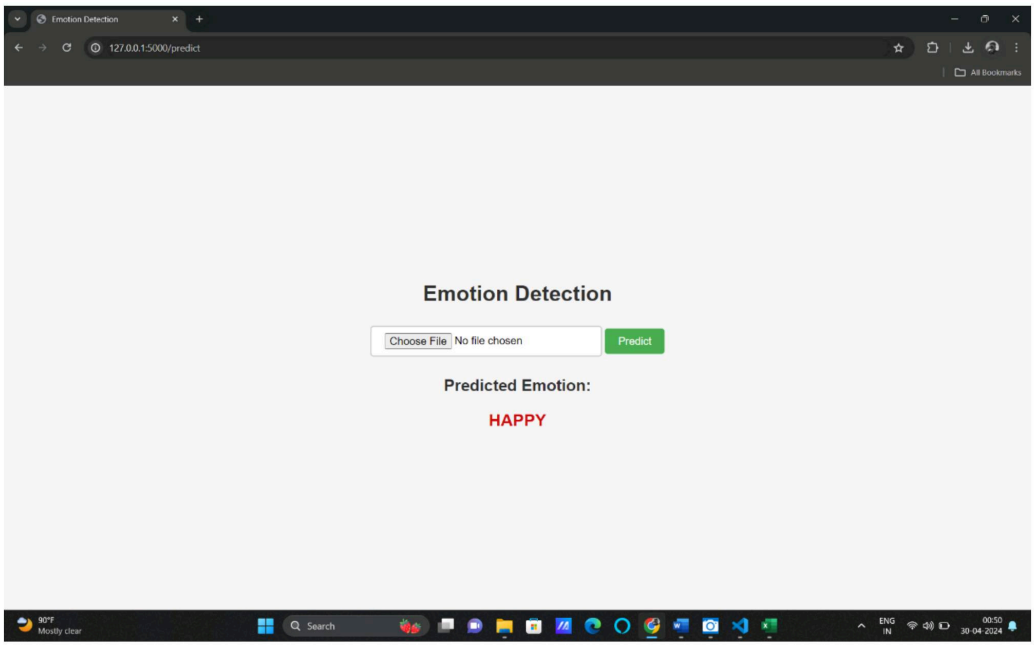


Fig. A.1.3 Predicted emotion for Happy audio input

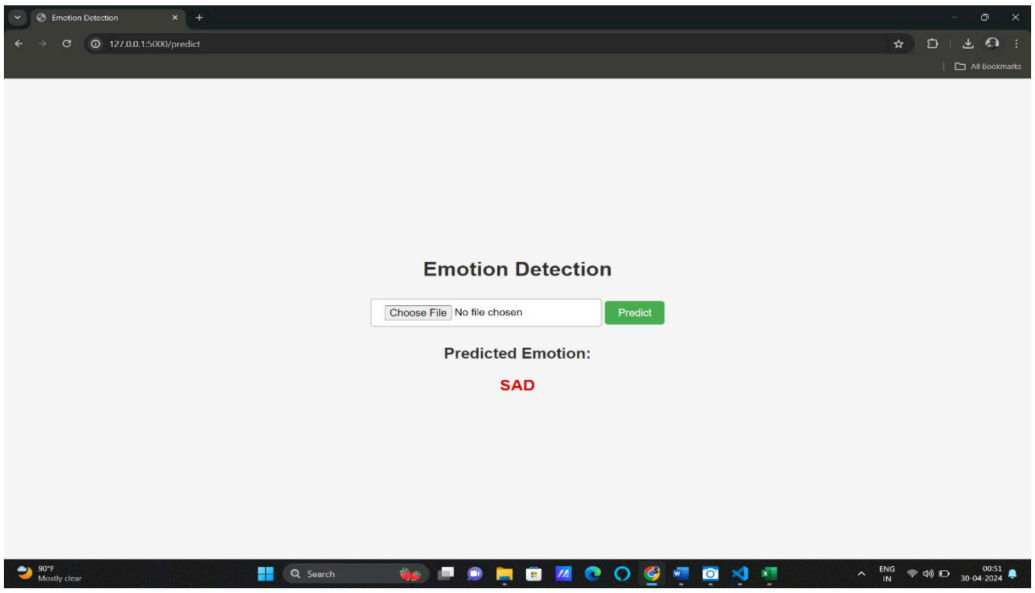


Fig. A.1.4 Predicted emotion for Sad audio input

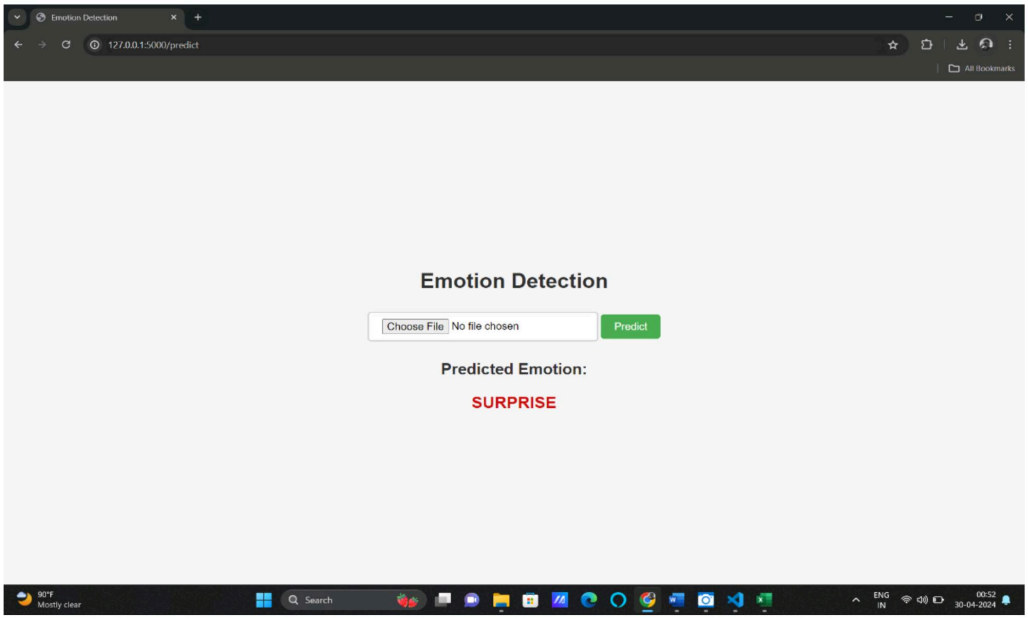


Fig. A.1.5 Predicted emotion for Surprise audio input

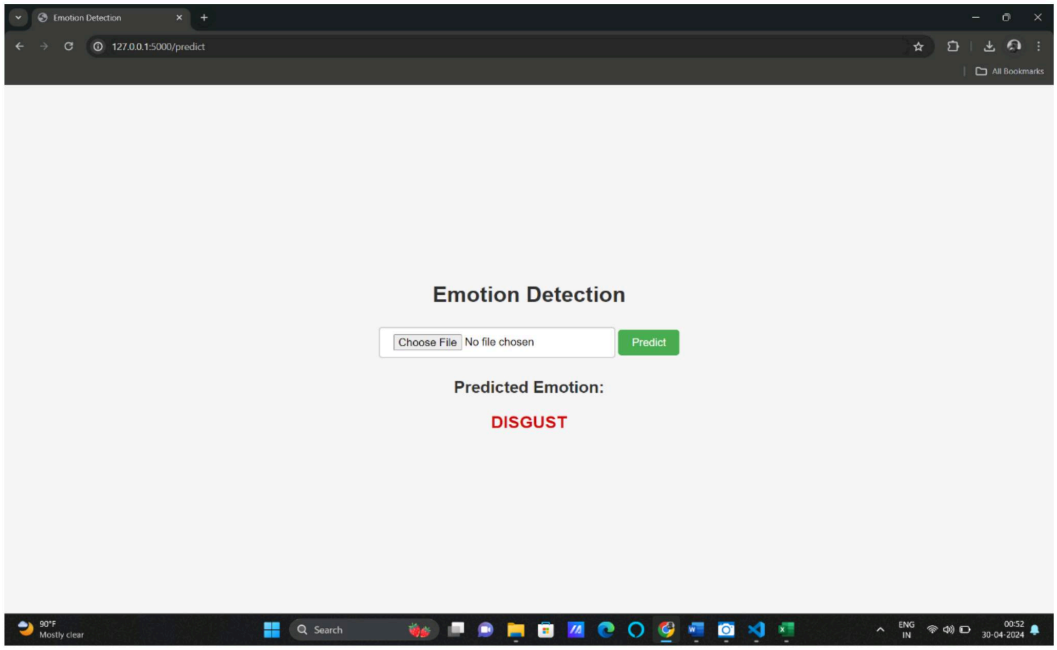


Fig. A.1.6 Predicted emotion for Disgust audio input

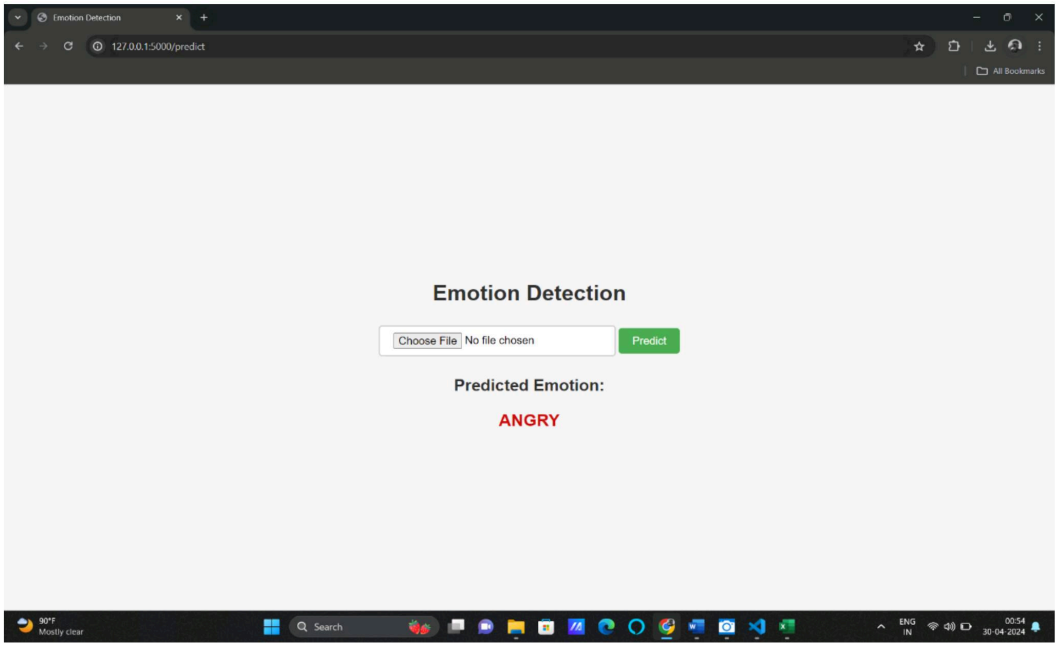


Fig. A.1.7 Predicted emotion for Angry audio input

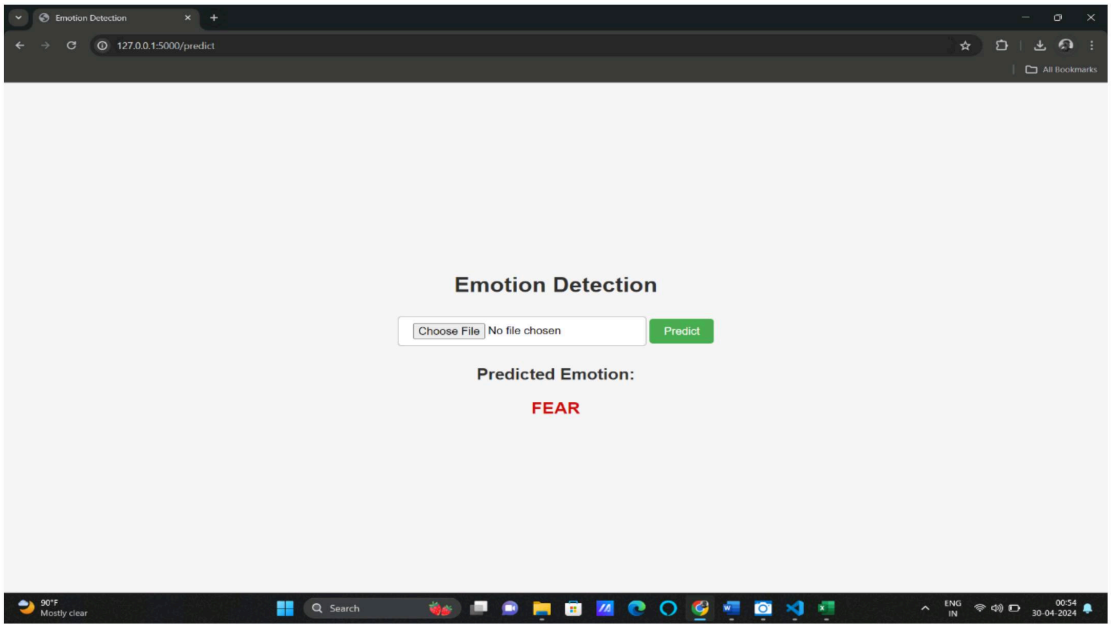


Fig. A.1.8 Predicted emotion for Fear audio input

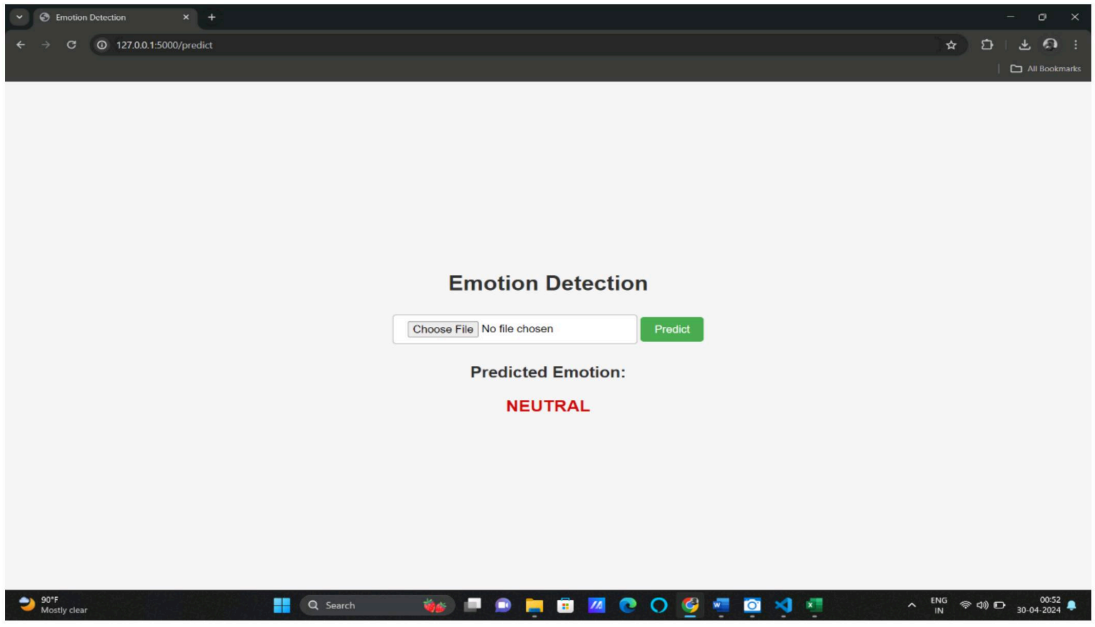


Fig. A.1.9 Predicted emotion for Neutral audio input

In [1]:

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

import os
import sys

# librosa is a Python library for analyzing audio and music. It can be used to extract the data from the audio files we will
import librosa
import librosa.display
import seaborn as sns
import matplotlib.pyplot as plt

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

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



import warnings
if not sys.warnoptions:
    warnings.simplefilter("ignore")
warnings.filterwarnings("ignore", category=DeprecationWarning)
import tensorflow as tf
print ("Done")
```

Fig. A.1.10 Import libraries

In [9]:

```
# NOISE
def noise(data):
    noise_amp = 0.035*np.random.uniform()*np.amax(data)
    data = data + noise_amp*np.random.normal(size=data.shape[0])
    return data

# STRETCH
def stretch(data, rate=0.8):
    return librosa.effects.time_stretch(data, rate)
# SHIFT
def shift(data):
    shift_range = int(np.random.uniform(low=-5, high = 5)*1000)
    return np.roll(data, shift_range)
# PITCH
def pitch(data, sampling_rate, pitch_factor=0.7):
    return librosa.effects.pitch_shift(y=data, sr=sampling_rate, n_steps=pitch_factor)
```

Fig. A.1.11 Data preprocessing

```python
def chroma(data, sr, frame_length=2048, hop_length=512, flatten=True):
    chroma_result = librosa.feature.chroma_stft(y=data, sr=sr, n_fft=frame_length, hop_length=hop_length)
    return np.squeeze(chroma_result.T) if not flatten else np.ravel(chroma_result.T)

def mfcc(data, sr, frame_length=2048, hop_length=512, flatten=True):
    mfcc_result = librosa.feature.mfcc(y=data, sr=sr, n_fft=frame_length, hop_length=hop_length)
    return np.squeeze(mfcc_result.T) if not flatten else np.ravel(mfcc_result.T)

def extract_features(data, sr=22050, frame_length=2048, hop_length=512):
    result = np.array([])

    result = np.hstack((result,
                        zcr(data, frame_length, hop_length),
                        chroma(data, sr, frame_length, hop_length),
                        mfcc(data, sr, frame_length, hop_length)
                       ))
    return result

def get_features(path, duration=2.5, offset=0.6):
    data, sr = librosa.load(path, duration=duration, offset=offset)
    aud = extract_features(data)
    audio = np.array(aud)

    # Assuming you have defined functions for noise and pitch
    noised_audio = noise(data)
    aud2 = extract_features(noised_audio)
    audio = np.vstack((audio, aud2))

    pitched_audio = pitch(data, sr)
    aud3 = extract_features(pitched_audio)
    audio = np.vstack((audio, aud3))

    pitched_audio1 = pitch(data, sr)
    pitched_noised_audio = noise(pitched_audio1)
    aud4 = extract_features(pitched_noised_audio)
    audio = np.vstack((audio, aud4))

    return audio
```

Fig. A.1.12 Feature Extraction

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 3564, 512)         3072

 batch_normalization (Batch  (None, 3564, 512)         2048
 Normalization)

 max_pooling1d (MaxPooling1  (None, 1782, 512)         0
 D)

 conv1d_1 (Conv1D)           (None, 1782, 512)         1311232

 batch_normalization_1 (Bat  (None, 1782, 512)         2048
 chNormalization)

 max_pooling1d_1 (MaxPoolin  (None, 891, 512)          0
 g1D)

 dropout (Dropout)           (None, 891, 512)          0

 conv1d_2 (Conv1D)           (None, 891, 256)          655616

 batch_normalization_2 (Bat  (None, 891, 256)          1024
 chNormalization)

 max_pooling1d_2 (MaxPoolin  (None, 446, 256)          0
 g1D)

 conv1d_3 (Conv1D)           (None, 446, 256)          196864

 batch_normalization_3 (Bat  (None, 446, 256)          1024
 chNormalization)

 max_pooling1d_3 (MaxPoolin  (None, 223, 256)          0
 g1D)

 dropout_1 (Dropout)         (None, 223, 256)          0

 conv1d_4 (Conv1D)           (None, 223, 128)          98432

 batch_normalization_4 (Bat  (None, 223, 128)          512
```

Fig. A.1.13 Model Training

APPENDIX 2

SOURCE CODE

App.py

```
from flask import Flask, render_template, request
from werkzeug.utils import secure_filename
import os
import numpy as np
from tensorflow.keras.models import load_model
import librosa
from sklearn.preprocessing import StandardScaler
import pickle

app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = r'E:\front end\UPLOAD_FOLDER'
# Load the pre-trained model and preprocessing objects
loaded_model = load_model('best_model1_weights.h5')
scaler2 = pickle.load(open('scaler2.pickle', 'rb'))
encoder2 = pickle.load(open('encoder2.pickle', 'rb'))

def zcr(data, frame_length, hop_length):
    zcr = librosa.feature.zero_crossing_rate(data, frame_length=frame_length, hop_length=hop_length)
    return np.squeeze(zcr)

def chroma(data, sr, frame_length=2048, hop_length=512, flatten=True):
    chroma_result = librosa.feature.chroma_stft(y=data, sr=sr, n_fft=frame_length, hop_length=hop_length)
```

```
    return np.squeeze(chroma_result.T) if not flatten else np.ravel(chroma_result.T)

def mfcc(data, sr, frame_length=2048, hop_length=512, flatten=True):
    mfcc_result = librosa.feature.mfcc(y=data, sr=sr, n_fft=frame_length,
hop_length=hop_length)
    return np.squeeze(mfcc_result.T) if not flatten else np.ravel(mfcc_result.T)

def extract_features(data, sr=22050, frame_length=2048, hop_length=512):
    result = np.array([])

    result = np.hstack((result,
                        zcr(data, frame_length, hop_length),
                        chroma(data, sr, frame_length, hop_length),
                        mfcc(data, sr, frame_length, hop_length)
                       ))
    return result

def predict_emotion(audio_path):
    data, sr = librosa.load(audio_path, duration=2.5, offset=0.6)
    features = extract_features(data)
    result = np.array(features)
    result = np.reshape(result, newshape=(1, 3564))
    i_result = scaler2.transform(result)
    final_result = np.expand_dims(i_result, axis=2)
    predictions = loaded_model.predict(final_result)
    predicted_emotion = encoder2.inverse_transform(predictions)
    return predicted_emotion[0][0]
```

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

@app.route('/predict', methods=['POST'])
def predict():
    if 'file' not in request.files:
        return render_template('index.html', prediction_text='No file part')
    file = request.files['file']
    if file.filename == '':
        return render_template('index.html', prediction_text='No selected file')
    if file:
        filename = secure_filename(file.filename)
        file_path = os.path.join(app.config['UPLOAD_FOLDER'], filename)
        file.save(file_path)
        if filename.endswith('.wav'):
            prediction = predict_emotion(file_path)
        else:
            prediction = "Unsupported file format"
        return render_template('index.html', prediction_text=prediction)
if __name__ == "__main__":
    app.run(debug=True)
```

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Emotion Detection</title>
<style>
    body {
        font-family: Arial, sans-serif;
        margin: 0;
        padding: 0;
        background-color: #f5f5f5;
        display: flex;
        justify-content: center;
        align-items: center;
        height: 100vh;
    }
    #content {
        text-align: center;
    }
    h1 {
        color: #333;
        margin-top: 50px;
    }
    form {
        margin-top: 30px;
    }
    input[type="file"] {
        padding: 10px 20px;
        border: 2px solid #ccc;
```

```
    border-radius: 5px;
    background-color: #fff;
    cursor: pointer;
    font-size: 16px;
    transition: border-color 0.3s;
}
input[type="file"]:hover {
    border-color: #aaa;
}
button[type="submit"] {
    padding: 10px 20px;
    border: none;
    border-radius: 5px;
    background-color: #4CAF50;
    color: white;
    cursor: pointer;
    font-size: 16px;
    transition: background-color 0.3s;
}

button[type="submit"]:hover {
    background-color: #45a049;
}

#prediction_result {
    margin-top: 30px;
}
```

```
        h2 {
            color: #333;
            font-size: 24px;
            margin-bottom: 10px;
        }
        p {
            color: #cf0e0e;
            font-size: 25px;
            font-weight: bold; /* Added font-weight property */
            text-transform: uppercase; /* Convert text to uppercase */
        }
    </style>
</head>
<body>
    <div id="content">
        <h1>Emotion Detection</h1>
        <form action="/predict" method="post" enctype="multipart/form-data">
            <input type="file" name="file">
            <button type="submit">Predict</button>
        </form>
        <div id="prediction_result">
        {% if prediction_text %}
        <h2>Predicted Emotion:</h2>
        <p>{{ prediction_text }}
            <!-- {% if prediction_text == 'Calm' %}
                <span class="emoji">&#128526;</span> >
            {% elif prediction_text == 'Happy' %}
```

```
            <span class="emoji">&#128512;</span>
        {% elif prediction_text == 'Sad' %}
            <span class="emoji">&#128546;</span>
        {% elif prediction_text == 'Angry' %}
            <span class="emoji">&#128545;</span>
        {% elif prediction_text == 'Fear' %}
            <span class="emoji">&#128561;</span>
        {% elif prediction_text == 'Surprise' %}
            <span class="emoji">&#128562;</span>
            <span class="emoji">&#128555;</span>
        {% else %}
            <span class="emoji">&#128533;</span>
        {% endif %}  -->
      </p>
      {% endif %}
   </div>
</body>

</html>
```

REFERENCES

[1] A. Milton, S. Sharmy Roy, S. Tamil Selvi, "SVM Scheme for Speech Emotion Recognition using MFCC Feature," International Journal of Computer Applications, vol. 69, no. 9, May 2013.

[2] Akshath Kumar B.H, Nagaraja N Poojary, Dr. Shivakumar G S, "Speech Emotion Recognition Using MLP Classifier," International Journal of Scientific Research in Computer Science, Engineering and Information Technology, vol. 7, no. 4, 2021.

[3] Fu Wang, Linhui Sun, Sheng Fu, "Decision tree and SVM model with Fisher feature selection for speech emotion recognition," EURASIP Journal on Audio, Speech, and Music Processing, 2019.

[4] G. Liu, W. He and B. Jin, "Feature Fusion of Speech Emotion Recognition Based on Deep Learning," International Conference on Network Infrastructure and Digital Content (IC-NIDC), pp. 193-197, Guiyang, China, August 22–24, 2018.

[5] H. O. Nasereddin and A. R. Omari, "Classification Techniques for Automatic Speech Recognition (ASR) Algorithms used with Real Time Speech Translation," Computing Conference, London, UK, pp. 200-207, 2017.

[6] K. Tarunika, R. B. Pradeeba and P. Aruna, "Applying Machine Learning Techniques for Speech Emotion Recognition," 9th International Conference On Computing, Communication and Networking Technologies.

[7] Masato Akagi, Reda Elbarougy, "Feature Selection Method or Real-time Speech Emotion Recognition," International Committee for Coordination and Standardization of Speech Databases and Assessment Technique, November 2017.

[8] S. Prasomphan," Improvement of Speech Emotion Recognition with Neural Network Classifier by Using Speech Spectrogram", 2015, International Conference on Systems, Signals and Image Processing (IWSSIP), London, UK, 10-12 September, pp. 72–76, 2015.

[9] T. Özseven, "Investigation of the Effect of Spectrogram Images and Different Texture Analysis Methods on Speech Emotion Recognition," Applied Acoustics, pp. 70–77, 2018.

[10] Ye Sim Ülgen Sonmez, Asaf Varol, "New Trends in Speech Emotion Recognition," Institute Of Electrical And Electronics Engineers, June 2019.

[11] Akagi, M. Xiao, H. Elbarougy, R. Hamada, Y. Li, J. "Toward affective speech-to-speech translation: Strategy for emotional speech recognition and synthesis in multiple languages" Proceedings of International Conference (APSIPA2014 ASC), Chiang Mai, December 2014.

[12] Shimizu, T., Ashikari, Y., Sumita, E., Zhang, J.S., and Nakamura, S., "NICT/ATR Chinese-Japanese-English Speech-to-Speech Translation System," Tsinghua Science and Technology, 13, 4, 540 544, 2008.

[13] R. Jenke, A. Peer, and M. Buss, ''Feature extraction and selection for emotion recognition from EEG,'' IEEE Trans. Affect. Comput. , vol. 5, no. 3, pp. 327-339, Jul./Sep. 2014.

[14] R. Elbarougy and M. Akagi, "Speech Emotion Recognition System Based on a Dimensional Approach Using a Three-Layered Model," Proc. Int. Conf. APSIPA ASC, 2012.

[15] Elbarougy, R. and Akagi, M. "Improving Speech Emotion Dimensions Estimation Using a Three-Layer Model for Human Perception," Journal of Acoustical Science and Technology, 35, 2, 86-98, March, 2014.

[16] Kecman, V., "Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models, " MIT Press, (2001).

[17] D. Bitouk, V. Ragini, and N. Ani, “Class-level spectral features for emotion recognition,” Journal of Speech Communication, vol. 52, no. 7-8, pp. 613-625, 2010.

[18] M. Grimm, and K. Kroschel, and E. Mower, and S. Narayanan, “ Primitives-based evaluation and estimation of emotions in speech," Speech Communication, 49, 787-800 (2007).

[19] H. Gunes, B. Schuller, M. Pantic, and R. Cowie, “Emotion representation, analysis and synthesis in continuous space: A survey,” in Proc. IEEE Int. Conf. Autom. Face Gesture Recog., 2011, pp. 827-834.

[20] M. Schroder, and R. Cowie, and E. D.-Cowie, M. Westerdijk, and S. Gielen, “Acoustic Correlates of Emotion Dimensions in View of Speech Synthesis,” Proc. Eurospeech 2001 , pp. 87-90 (2001).

[21] H.P. Espinosa, C.A.Reyes-Garca, L.V.Pineda, “Acoustic feature selection and classification of emotions in speech using a 3D continuous emotion model,” Biomedical Signal Processing and Control , 7(1), 79-87 (2012).

[22] Kecman, V., “Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models, ” MIT Press, (2001).

[23] Lacrimioara GRAMA, Liana TUNS, Corneliu RUSU, “On the Optimization of SVM Kernel Parameters for Improving Audio Classification Accuracy”,14th International Conference on Engineering of Modern Electric Systems (EMES), 2017.

[24] Yashpalsing Chavhan, M. L. Dhore, Pallavi Yesaware, "Speech Emotion Recognition Using Support Vector Machine", International Journal of Computer Applications, vol. 1, pp.6-9,February 2010.