Human Speech Emotion Recognition (HSER)

A project presented to the **National University** in partial fulfillment of the requirements for the degree of B.Sc. (Hon's) in Computer Science & Engineering

Supervised by:

Nusrhat Jahan Sarker

Lecturer

Department of Computer Science & Engineering

Daffodil Institute of IT (DIIT)

Submitted by:

Md. Maruf Hossain

Registration No: 17502004966

Session: 2017-18

Department of Computer Science & Engineering

Daffodil Institute of IT (DIIT)

Under National University (NU)- Dhaka, Bangladesh

Submission Date: 04-09-2023

Approval

This Project titled “Human Speech Emotion Recognition (HSER)” is submitted to the Department of Computer Science & Engineering (CSE) of Daffodil Institute of IT (DIIT) under National University (NU). It has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor’s in Computer Science & Engineering (CSE) & approved as to its styles & contents.

…………………………….
Examiner

…………………………….
Examiner

……………………………
Nusrhat Jahan Sarker
Project Guide (Supervisor)
Lecturer
Daffodil Institute of IT

…………………………….
Md. Imran Hossain
Head
Department of CSE
Daffodil Institute of IT

Declaration

I hereby declare that the project work entitled “Human Speech Emotion Recognition” submitted to the degree of B.Sc. (Hon’s) in Computer Science & Engineering (CSE) is a record of original work done by me. Except as acknowledged in the text and that the material has not been submitted, either in whole or in part, for a degree at this or any other university.

Submitted by:

…………………………………

Md. Maruf Hossain

Registration No: 17502004966

Session: 2017-18

Acknowledgment

My sincere thanks to **Prof. Dr. Mohammed Shakhawat Hossain, Principal, DIIT** who has allowed me to do this project, and the encouragement given to me.

I express my gratitude to **Md. Imran Hossain**, Head of Department, Department of Computer Science & Engineering, DIIT, Dhaka, for his patronage and for giving me an opportunity to undertake this Project.

I would also like to thank my Project Supervisor, **Nusrhat Jahan Sarker**, Lecturer, Department of Computer Science & Engineering, DIIT, for his valuable guidance and support to meet the successful completion of my project.

I express my gratitude to **Poly Bhoumik**, Senior Lecturer, DIIT, Dhaka, for having provided us with the facilities to do the project successfully.

I also express my gratitude to **Saidur Rahman**, Senior Lecturer, DIIT, Dhaka, for having provided us with the facilities to do the project successfully.

I also express my gratitude to **Safrun Nesa Saira**, Lecturer, DIIT, Dhaka, for having provided us with the facilities to do the project successfully.

I also express my gratitude to **Mizanur Rahman**, Lecturer, DIIT, Dhaka, for having provided us with the facilities to do the project successfully.

I also express my gratitude to **Moumita Akter**, Lecturer, DIIT, Dhaka, for having provided us with the facilities to do the project successfully.

I also say thanks to **Md. Mushfiqur Rahaman**, Lecturer, DIIT, Dhaka, for giving his valuable time to do the project successfully.

Finally, I extend my sincere thanks to my family members and my friends for their constant support throughout this project.

Abstract

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 area of research has applications in fields such as human-computer interaction, healthcare, and psychology, and has been addressed using a variety of techniques, including machine learning and deep learning. Successful emotion recognition from speech can enable improved human-computer interaction, personalized healthcare, and a better understanding of human behavior and communication.

Table of contents

Contents

CHAPTER- 01

CHAPTER- 02

CHAPTER- 03

List of Figure

CHAPTER- 01

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 pitch, intensity, rhythm, and timbre, 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. Speech emotion recognition systems typically involve training a machine learning model on a large dataset of labeled speech samples, where the emotional state of each speaker is known.[3] The model can then be used to classify the emotional state of new speakers in real time. However, accurately recognizing human emotions from speech is a complex and challenging task, as emotions can be expressed in subtle and nuanced ways that are difficult to capture using traditional acoustic features. 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.

1.2 Project Aims

The main aims of human speech emotion recognition are:

- **Developing accurate emotion recognition models:** One of the main aims of speech emotion recognition is to develop accurate machine learning models that can reliably recognize and classify the emotional content of spoken language. This requires developing and testing a range of acoustic features and machine learning algorithms to find the most effective combination.
- **Creating large and diverse emotional speech datasets:** To train accurate emotion recognition models, it is essential to have large and diverse datasets of emotional speech samples. The aim is to create datasets that include a wide range of emotions expressed in different contexts and by speakers from diverse backgrounds.

- **Improving real-time emotion recognition:** Another aim of speech emotion recognition is to develop real-time emotion recognition systems that can analyze and classify emotions in spoken language in real time. This requires developing algorithms that can process speech quickly and accurately and integrating the technology into applications that can respond to users' emotional states in real-time.
- **Enhancing cross-cultural emotion recognition:** Emotions can be expressed and perceived differently across cultures, so another aim of speech emotion recognition is to develop models that can recognize emotions across different cultures and languages. This requires developing and testing models on speech datasets from different cultures and languages.
- **Developing practical applications:** The aim of speech emotion recognition is to develop practical applications that can benefit society, such as improving human-computer interaction, enhancing virtual and augmented reality experiences, developing more effective therapies for emotional disorders, and improving lie detection techniques.

Overall, the aim of speech emotion recognition is to develop technology that can accurately recognize and respond to human emotions in spoken language, leading to a wide range of practical applications and benefits for society.

1.3 Objectives

The main objectives of human speech emotion recognition are:

- **Identifying and categorizing emotions:** The primary objective of speech emotion recognition is to identify and categorize the emotions expressed in spoken language.[3] This includes emotions such as anger, happiness, sadness, fear, and surprise.
- **Enhancing human-computer interaction:** Speech emotion recognition can improve human-computer interaction by enabling computers to recognize and respond to human emotions. For example, a computerized customer service agent can adjust their response based on the emotional state of the customer.**[5]**

- **Developing effective therapies:** Emotion recognition technology can be used to develop more effective speech-based therapies for individuals with emotional disorders. For example, it can be used to monitor the emotional state of patients during therapy sessions and adjust the therapy accordingly.
- **Improving lie detection:** Speech emotion recognition can enhance the accuracy of lie detection techniques by identifying emotional cues that may indicate deception.
- **Enhancing virtual and augmented reality experiences:** Speech emotion recognition can be used to create more immersive virtual and augmented reality experiences by enabling computers to respond to the emotional state of the user.[5]

Overall, the objective of speech emotion recognition is to develop technology that can accurately and reliably recognize and respond to human emotions expressed in spoken language, leading to a wide range of applications and benefits.

1.4 Why we selected this project

There are several reasons why one may choose to work on a Human Speech Emotion Recognition (HSER) project:

- **Importance:** 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 HSER systems, we can improve human-human and human-computer interactions, enhance emotional intelligence, and provide better care and services to people.
- **Challenge:** HSER is a challenging problem that requires interdisciplinary knowledge and skills, including signal processing, machine learning, linguistics, psychology, and communication. Working on a HSER project can provide an opportunity to learn and apply various techniques and methodologies, and to develop new insights and research directions.

- **Innovation:** HSER is a rapidly evolving field that constantly introduces new methods, datasets, and applications. By working on a HSER project, we can contribute to the advancement of knowledge and innovation in the field, and potentially make a significant impact on society.
- **Career opportunities:** HSER is a highly demanded skill in various industries, such as healthcare, education, media, and entertainment. By working on a HSER project, we can develop skills and expertise that are highly valued by employers, and potentially open new career opportunities.
- **Personal interest:** HSER can be a fascinating and rewarding topic for those who are interested in human communication, psychology, or linguistics. By working on a HSER project, we can deepen our understanding of human emotions, behaviors, and culture, and potentially contribute to the well-being and happiness of people.[4]

Overall, working on a HSER project can provide a meaningful and challenging experience that combines technical skills, scientific knowledge, and social impact.

1.5 Limitation of Existing System

There are several limitations of Human Speech Emotion Recognition (HSER) projects that need to be considered:

- An accuracy of trained model is 74.48%
- Limited Emotion Representation
- Used only one dataset to train (RAVDESS)
- Context Sensitivity
- Ethical Considerations

Overall, HSER projects need to address these limitations by adopting robust, accurate, and interpretable methods that can capture the complexity and variability of emotional expression in speech, and by evaluating their generalization and validity in real-world settings. Moreover, HSER projects need to consider the ethical, social, and legal implications of their use, and engage with relevant stakeholders and communities to ensure their acceptance and effectiveness.

1.6 Features of our proposed system

- Recognize 5 emotions (Happy, Sad, Angry, Fear and Disgust).
- Show Model Accuracy.
- Show the Emotion's accuracy level.
- Show F1 Score.
- Display emotion as a text.
- Upload Audio Button.
- User Friendly UI.

1.7 SDLC

The Software Development Life Cycle (SDLC) is a structured approach to software development that defines a set of activities, processes, and methodologies to ensure the quality, efficiency, and effectiveness of software products.[6] SDLC typically consists of six stages:

- **Planning:** In this stage, the project goals, scope, requirements, and constraints are defined and documented. The project team establishes the project schedule, budget, and resource allocation, and identifies the risks and challenges that need to be addressed.
- **Analysis:** In this stage, the project team analyzes the requirements and specifications to develop a detailed understanding of the problem domain, user needs, and system constraints. The team identifies the functional and non-functional requirements, defines the system architecture, and develops the system design and specifications.
- **Design:** In this stage, the project team creates the detailed design of the system, including the software architecture, modules, components, interfaces, and data structures. The design is typically documented using diagrams, models, and specifications, and is reviewed and validated by stakeholders and experts.
- **Implementation:** In this stage, the project team develops the software code, tests it, and integrates it with other system components. The software is typically developed using a specific programming language, coding standards, and development tools, and is subject to version control, testing, and quality assurance procedures.
- **Testing:** In this stage, the project team verifies and validates the software to ensure that it meets the requirements and specifications and that it is free of defects and errors. Testing includes various techniques such as unit testing,

integration testing, system testing, acceptance testing, and regression testing, and is typically automated and conducted in multiple stages.

- **Deployment and maintenance:** In this stage, the software is deployed to the production environment, and is monitored, maintained, and updated to ensure its reliability, security, and performance. Maintenance includes bug fixing, enhancements, upgrades, and support, and may involve customer feedback, user training, and documentation.

Overall, SDLC provides a systematic and disciplined approach to software development that ensures the quality, efficiency, and effectiveness of software products and enables continuous improvement and innovation.[12]

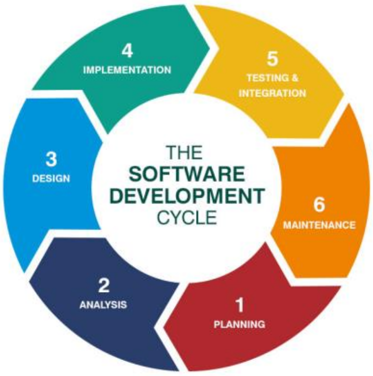


Fig 1.8: SDLC

1.8 Agile Method

Agile is an iterative and incremental approach to software development that emphasizes flexibility, collaboration, and customer satisfaction. Agile is based on the Agile Manifesto, which was developed in 2001 by a group of software development experts. Agile is a response to traditional, sequential software development methods, such as the Waterfall model, which emphasize strict planning, documentation, and control.[13]

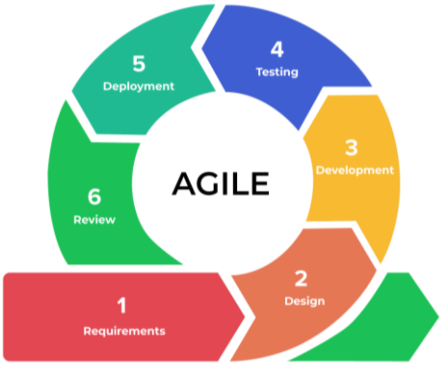


Fig:1.9: Agile-Method

The key principles of Agile include:

- Customer satisfaction through continuous delivery of valuable software
- Emphasis on individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Collaboration between the customer and the development team
- Responding to change over following a plan

Agile development typically involves the following practices:

- **Iterative and incremental development:** Software is developed in small, incremental steps, with feedback and testing at each stage.
- **Continuous integration and testing:** Code is continuously integrated, tested, and reviewed to ensure quality and reliability.
- **Prioritization and backlog management:** The development team and the customer collaborate to prioritize features and requirements and manage a backlog of tasks.
- **Cross-functional teams:** The development team includes members with different skills and expertise, such as developers, testers, and designers.
- **Adaptive planning and change management:** The development team adapts to changing requirements and priorities and adjusts the plan and schedule accordingly.

Agile methods can be applied to various software development frameworks, such as Scrum, Kanban, and Extreme Programming (XP). Agile development has several benefits, including faster time-to-market, improved quality and customer satisfaction, increased flexibility and innovation, and better team collaboration and communication.

However, Agile development also has some challenges, such as the need for continuous customer involvement, the potential for scope creep and technical debt, and the difficulty of scaling and managing large and complex projects.

1.9 Why have We Chosen Agile?

Agile methodology has several advantages that make it a suitable approach for developing a human speech emotion recognition project.[12] Some of the reasons why we have chosen Agile are:

- **Flexibility:** Agile methodology allows for flexibility and adaptability to changing requirements, priorities, and feedback. This is especially important in a research-oriented project, where new insights and challenges can emerge throughout the development process.
- **Collaboration:** Agile methodology emphasizes collaboration and communication between team members, stakeholders, and end-users. This can facilitate knowledge sharing, problem-solving, and decision-making, and ensure that the project meets the needs and expectations of all stakeholders.
- **Iterative and Incremental:** Agile methodology follows an iterative and incremental approach, where the project is divided into small, manageable, and testable increments or sprints. This can facilitate faster feedback, validation, and learning, and reduce the risk of project failure or delay.
- **Continuous Improvement:** Agile methodology emphasizes continuous improvement and learning, through techniques such as retrospectives, feedback loops, and continuous integration. This can foster a culture of innovation, experimentation, and creativity, and ensure that the project stays up to date with the latest research and trends.
- **Transparency:** Agile methodology promotes transparency and visibility of project progress, by using techniques such as user stories, burndown charts, and sprint reviews. This can facilitate accountability, trust, and alignment among team members and stakeholders.

In summary, Agile methodology can provide a structured, flexible, and collaborative approach for developing a human speech emotion recognition project, while ensuring that the project meets the needs and expectations of all stakeholders and adapts to changing requirements and feedback.

1.10 Business Perspective

From a business perspective, human speech emotion recognition projects can offer several benefits and opportunities for organizations. Some of these are:

- **Improving customer experience:** By analyzing the emotions of customers during interactions with the company, businesses can gain valuable insights into their needs and preferences, and tailor their products and services to meet their expectations. This can help improve customer satisfaction, loyalty, and retention.
- **Enhancing marketing and advertising:** Human speech emotion recognition can help companies understand how customers respond to their marketing campaigns and advertising messages and adjust them accordingly. By targeting emotions and personalizing messages, companies can increase the effectiveness and impact of their marketing efforts.
- **Enhancing employee productivity:** Human speech emotion recognition can be used to monitor employee emotions and stress levels and identify potential issues that may affect their productivity and well-being. This can help organizations take preventive measures, such as providing support, training, or incentives, to improve employee performance and job satisfaction.
- **Improving healthcare and mental health services:** Human speech emotion recognition can be used to diagnose and treat mental health disorders, such as depression, anxiety, and stress, by analyzing patients' speech patterns and emotions. This can improve the accuracy and effectiveness of healthcare services and reduce the stigma associated with mental health.
- **Enhancing security and surveillance:** Human speech emotion recognition can be used to detect suspicious or threatening behavior, such as aggression, violence, or fraud, in various settings, such as airports, public places, or financial institutions. This can help enhance security and prevent potential threats or crimes.
- **Developing new products and services:** Human speech emotion recognition can inspire innovation and new product development in various industries, such as entertainment, gaming, education, and sports. By incorporating emotional feedback and interaction into their products and services, companies can create more engaging and immersive experiences for their customers.

CHAPTER- 02

BACKGROUND STUDY

2.1 Background of this Project

The field of human speech emotion recognition has its roots in the study of human emotions and the development of computer-based systems for analyzing speech signals. In the early days of artificial intelligence and signal processing, researchers focused on developing rule-based systems that used expert knowledge and heuristics to classify emotions based on acoustic features of speech, such as pitch, intensity, duration, and spectral content.

In the 1990s and early 2000s, the field of speech emotion recognition began to shift towards machine learning and data-driven approaches, as researchers started using statistical models, such as Hidden Markov Models (HMMs) and Gaussian Mixture Models (GMMs), to classify emotions based on large datasets of speech signals and corresponding emotion labels.

In recent years, with the availability of deep learning algorithms and large-scale datasets, researchers have achieved significant progress in developing neural network-based models for speech emotion recognition.[7] These models, such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory (LSTM) networks, can learn complex and abstract features from speech signals and achieve state-of-the-art performance on various emotion recognition tasks. Today, human speech emotion recognition has applications in various domains, such as healthcare, education, entertainment, marketing, and security. It has the potential to revolutionize how humans interact with machines and enable more natural and intuitive communication interfaces. However, there are still challenges and limitations in the field, such as the lack of standardized datasets and evaluation metrics, the influence of cultural and contextual factors on emotion perception, and the ethical and privacy concerns related to the use of speech data for emotion recognition.

2.2 Problems with the Current System

While human speech emotion recognition has made significant progress in recent years, there are still several challenges and limitations with the current state of technology. Some of these problems include:

- **Lack of standardized datasets:** There is a lack of standardized datasets for speech emotion recognition, which makes it difficult to compare and evaluate different models and approaches. This can lead to inconsistencies and uncertainties in the performance of emotion recognition systems.

- **Influence of contextual factors:** Emotions are highly contextual, and their expression can vary depending on cultural, social, and situational factors. This makes it challenging to develop emotion recognition systems that can generalize across different contexts and populations.
- **Ethical and privacy concerns:** The use of speech data for emotion recognition raises ethical and privacy concerns, as it involves collecting and processing sensitive personal information. There is a risk of misuse or abuse of this data, and it is important to ensure that proper safeguards are in place to protect an individual's rights and interests.
- **Difficulty in recognizing subtle emotions:** Some emotions, such as boredom, sarcasm, or irony, can be challenging to detect and classify using speech signals alone. This can limit the accuracy and reliability of emotion recognition systems, especially in real-world scenarios where emotions are often subtle and complex.
- **Dependence on labeled data:** Most speech emotion recognition systems rely on labeled data for training and evaluation, which can be time-consuming and expensive to collect and annotate. This can limit the scalability and generalizability of emotion recognition systems, especially for languages or domains with limited resources.
- **Limited real-world applications:** While there are many potential applications of speech emotion recognition, its practical use cases are still relatively limited. This is partly due to the technical challenges and limitations mentioned above, but also to the need for integration with other technologies and systems to provide a seamless and user-friendly experience.

2.3 The Solution of the Problem

- **Standardized datasets:** Developing standardized datasets that cover a range of emotions and contexts can help to evaluate and compare different emotion recognition models and approaches. The creation of such datasets can involve collaborations between researchers and stakeholders from various domains, such as healthcare, education, and entertainment.

- **Context-aware models:** Developing emotion recognition models that can account for contextual factors, such as culture, social norms, and situational cues, can improve the accuracy and reliability of emotion recognition systems. This can involve integrating multimodal signals, such as facial expressions, physiological responses, and behavioral cues, to capture a more comprehensive picture of emotional states.
- **Ethical and privacy considerations:** Ensuring that emotion recognition systems comply with ethical and privacy standards can increase public trust and acceptance of the technology. This can involve implementing data protection measures, such as anonymization, encryption, and secure storage, and providing transparent and informed consent procedures for data collection and use.
- **Hybrid models:** Combining speech signals with other sources of information, such as text, image, and video, can improve the recognition of subtle and complex emotions, such as sarcasm, irony, or humor. This can involve using natural language processing techniques to analyze speech content and sentiment, or computer vision techniques to capture facial expressions and body language.
- **Unsupervised learning:** Developing unsupervised learning techniques that do not rely on labeled data can improve the scalability and generalizability of emotion recognition systems. This can involve using techniques such as clustering, dimensionality reduction, or generative models to discover underlying patterns and structures in speech data.
- **Integration with other technologies:** Integrating emotion recognition systems with other technologies and systems, such as virtual assistants, chatbots, or gaming platforms, can create new and innovative applications that provide more engaging and personalized user experiences.

2.4 Methodology

The methodology for developing a human speech emotion recognition project can involve the following steps:

- **Problem Definition:** Define the problem statement and research questions. Identify the main objectives, stakeholders, and potential applications of the project.

- **Data Collection:** Collect speech data that covers a range of emotions and contexts. Ensure that the data is representative, diverse, and annotated with ground truth labels.
- **Preprocessing:** Preprocess the speech data by applying techniques such as noise reduction, feature extraction, and normalization.[8] This can involve using software tools such as Praat, OpenSMILE, or Librosa.
- **Model Selection:** Select an appropriate machine learning model or algorithm that can classify emotions based on speech signals. This can involve using techniques such as Support Vector Machines (SVM), Convolutional Neural Networks (CNN), or Recurrent Neural Networks (RNN).
- **Model Training:** Train the selected model using the preprocessed speech data. This can involve using techniques such as cross-validation, hyperparameter tuning, or transfer learning.
- **Model Evaluation:** Evaluate the performance of the trained model using appropriate metrics such as accuracy, precision, recall, and F1-score. This can involve using techniques such as confusion matrices, ROC curves, or AUC analysis.
- **Model Optimization:** Optimize the model performance by fine-tuning the model architecture, regularization techniques, or feature selection methods.
- **Deployment:** Deploy the optimized model in a real-world setting. This can involve integrating the model with other technologies and systems, such as chatbots, virtual assistants, or gaming platforms.
- **Evaluation:** Evaluate the effectiveness and usability of the deployed model in a real-world setting. Collect feedback from end-users, stakeholders, and domain experts.
- **Maintenance:** Maintain the model by monitoring its performance, updating its parameters, and retraining it with new data. This can involve using techniques such as online learning, transfer learning, or active learning.

The methodology can follow an iterative and incremental approach, such as Agile or Scrum, to allow for flexibility and adaptability to changing requirements and feedback.

2.5 Feasibility Study

A feasibility study is an important step in determining whether a human speech-emotion recognition project is viable, practical, and profitable.[2] It involves assessing the technical, economic, social, and environmental factors that can affect the project's success. Some of the key aspects of a feasibility study for a human speech emotion recognition project are:

- **Technical Feasibility:** This involves assessing the technical requirements, resources, and capabilities needed to develop a speech emotion recognition system. This can include the availability and quality of speech data, the suitability of machine learning models, the performance and scalability of the system, and the integration with other technologies and platforms.
- **Economic Feasibility:** This involves assessing the financial costs, benefits, and risks associated with developing a speech emotion recognition system. This can include the investment required for data collection, model development, software development, hardware and infrastructure costs, and the potential revenue and profitability of the system.
- **Social Feasibility:** This involves assessing the social and ethical implications of developing a speech-emotion recognition system. This can include the impact on privacy, security, bias, and discrimination, as well as the potential benefits and risks for end-users, stakeholders, and society at large.
- **Environmental Feasibility:** This involves assessing the environmental impact and sustainability of developing a speech emotion recognition system. This can include the energy consumption, carbon footprint, and waste generation associated with the system, as well as the potential benefits and risks for the environment and natural resources.

Based on the results of the feasibility study, it can be concluded whether the human speech emotion recognition project is feasible or not. If the results are positive, the project can proceed to the next stage of development, which is typically the planning and design phase. If the results are negative, the project may need to be reconsidered or modified to address the feasibility issues.

CHAPTER- 03

SYSTEM SPECIFICATION

3.1 Introduction to Requirements

The requirements phase is an important stage in the software development life cycle (SDLC), where the project team identifies, analyzes, and prioritizes the requirements for the project. This involves gathering input from stakeholders and end-users, defining the scope and objectives of the project, and documenting the requirements in a clear and concise manner.[3]

The requirements phase typically includes the following steps:

- **Requirement Elicitation:** This involves gathering input from stakeholders and end-users to identify their needs, expectations, and constraints. This can be done through interviews, surveys, focus groups, and other methods of communication.
- **Requirement Analysis:** This involves analyzing and clarifying the requirements to ensure that they are complete, consistent, and feasible. This can include identifying conflicts, ambiguities, and gaps in the requirements, and resolving them through discussions and negotiations.
- **Requirement Specification:** This involves documenting the requirements in a clear and concise manner, using standardized formats such as use cases, user stories, and functional and non-functional requirements. This can help to ensure that the requirements are communicated effectively to the development team and other stakeholders.
- **Requirement Validation:** This involves verifying that the requirements meet the needs and expectations of the stakeholders and end-users, and that they are feasible and achievable within the project constraints. This can include reviewing the requirements with the stakeholders, conducting prototyping, and testing, and analyzing the risks and benefits of the requirements.

In summary, the requirements phase is a critical step in the development of a human speech emotion recognition project, as it lays the foundation for the design, development, and testing of the system. By clearly identifying and prioritizing the requirements, the project team can ensure that the system meets the needs and expectations of the stakeholders and end-users and achieves its goals and objectives.

3.1.1 Hardware Requirements

- **Processor:** A fast and powerful processor is essential for processing large volumes of speech data and running complex machine learning algorithms. An Intel Core i5 or i7 processor or equivalent would be suitable.
- **Memory (RAM):** Sufficient RAM is required to store and manipulate large amounts of speech data and machine learning models. A minimum of 8GB of RAM is recommended, but 16GB or more may be required for more demanding applications.
- **Storage:** Adequate storage capacity is needed to store speech data, machine learning models, and other software components. An SSD with at least 256GB of storage capacity is recommended for fast data access and processing.
- **Audio Input Devices:** High-quality microphones are essential for capturing clear and accurate speech data. A variety of microphones may be used, depending on the specific application and environment.
- **Audio Output Devices:** Speakers or headphones are required for playing back speech data and providing feedback to end-users.
- **Graphics Processing Unit (GPU):** A GPU may be required for accelerating the machine learning algorithms used for speech emotion recognition. A dedicated GPU with at least 4GB of VRAM is recommended for more demanding applications.
- **Network Interface Card (NIC):** A fast and reliable network connection is required for accessing speech data from remote sources or streaming audio data in real time.[11]

3.1.2 Software Requirements

- **Operating System:** A suitable operating system such as Windows, Linux, or macOS is required to run the software components of the system.
- **Programming Language:** The human speech emotion recognition system may be developed using programming languages such as Python, Java, C++, or others depending on the specific system design and implementation.
- **Speech Processing Libraries:** Libraries such as Librosa, PyAudio, and Speech Recognition are commonly used for speech-processing tasks such as speech-to-text conversion, feature extraction, and signal processing.

- **Machine Learning Frameworks:** Machine learning frameworks such as TensorFlow, Keras, and Scikit-Learn are commonly used for training and deploying machine learning models for speech emotion recognition.
- **Database Management System:** A database management system such as MySQL or PostgreSQL may be used to store speech data and other relevant information.
- **Integrated Development Environment (IDE):** An IDE such as PyCharm, Visual Studio Code, or Eclipse, Jupyter Notebook, or Google Colab may be used for software development, debugging, and testing.
- **Web Development Tools:** For web-based human speech emotion recognition systems, web development tools such as HTML, CSS, and Python may be used for building the user interface.
- **Web Application Frameworks:** Flask, Django, Ruby on Rails, and Node.js are all popular web application frameworks that can be used for building web applications with HSER capabilities.
- **Web Browser:** Google Chrome, Firefox, Safari, and other HTML5 supported browsers.

3.2 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. Some of the key advantages of using Python for human speech emotion recognition projects include:

- **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.
- **Large Community:** Python has a large and active community of developers who contribute to the development of libraries, frameworks, and tools. This makes it easier to find support and resources when building a human speech-emotion recognition system.

- **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.
- **Rapid Prototyping:** Python allows for rapid prototyping and experimentation, which is important for developing and testing different approaches to human speech emotion recognition.

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.

3.3 Pycharm

PyCharm is an Integrated Development Environment (IDE) specifically designed for Python development. It's developed by JetBrains and is available in both professional and community editions.

PyCharm provides a wide range of features to improve productivity and streamline the development process, including code highlighting and completion, refactoring tools, debugging, and testing capabilities. It also supports many popular Python frameworks, including Django, Flask, and Pyramid.

One of the key benefits of PyCharm is its ability to integrate with version control systems like Git, allowing for easy collaboration with other developers. It also has powerful tools for working with databases, including a database console and support for multiple database types.

3.4 Web Browser

A web browser is a software program that allows a user to locate, access, and display web pages. In common usage, a web browser is usually shortened to "browser." Web browsers are used primarily for displaying and accessing websites on the internet, as well as other content created using languages such as Hypertext Markup Language (HTML) and Extensible Markup Language (XML). Browsers translate web pages and websites delivered using Hypertext Transfer Protocol (HTTP) into human-readable content. They also could display other protocols and prefixes, such as secure HTTP (HTTPS), File Transfer Protocol (FTP), email handling, and files.

3.5 Librosa

Librosa is a popular Python library for audio and music analysis that can be used in Human Speech Emotion Recognition (HSER) projects. It provides a variety of functions for loading, processing, and analyzing audio data, including tools for feature extraction, time-frequency analysis, and signal processing.[1]

3.6 Scikit-Learn

Scikit-learn is a popular Python library for machine learning that can be used in Human Speech Emotion Recognition (HSER) projects. It provides a variety of functions and tools for data preprocessing, model training, model selection, and model evaluation.

3.7 Soundfile

Soundfile is a Python library for reading and writing audio files. It provides a way to read and write a wide variety of audio file formats, including WAV, FLAC, AIFF, and Ogg Vorbis.

3.8 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 Human Speech Emotion Recognition (HSER) projects.

3.9 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.

Flask's simplicity and flexibility have made it a popular choice for building small to medium-sized web applications, APIs, and prototypes. However, for larger projects with more complex requirements, developers might choose other web frameworks like Django, which is also a Python web framework but comes with more built-in features and conventions.

3.10 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.

In short, HTML is the building block of webpages, allowing developers to define the content and layout of websites, making it possible for users to access and interact with information and services on the internet.

3.11 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.

By separating the content (HTML) from its presentation (CSS), CSS enables developers to create consistent and visually appealing designs across multiple web pages. It simplifies the process of making global changes to a website's appearance, as changes made in the CSS file affect all the HTML pages that use that CSS.

In short, CSS is essential for creating visually attractive and well-structured web pages by defining how elements should be displayed on a website.

CHAPTER- 04
SYSTEM DESIGN

4.1 Proposed System Overview

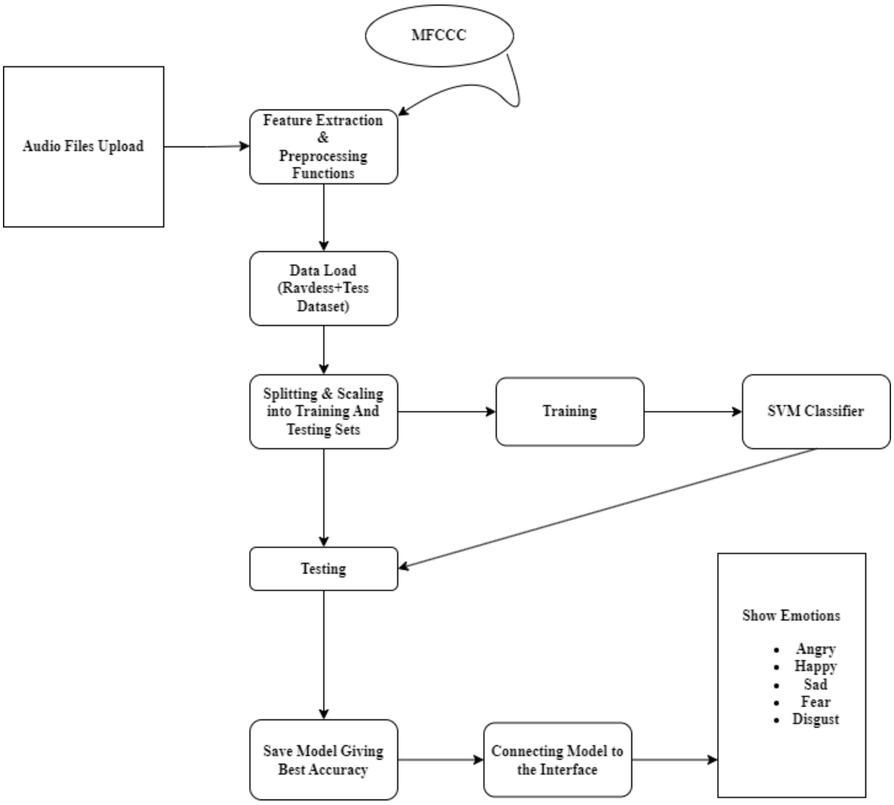

MFCCC
Audio Files Upload
Feature Extraction & Preprocessing Functions
Data Load (Ravdess+Tess Dataset)
Splitting & Scaling into Training And Testing Sets
Training
SVM Classifier
Testing
Show Emotions
Angry
Happy
Sad
Fear
Disgust
Save Model Giving Best Accuracy
Connecting Model to the Interface

Fig 4.1: Proposed System Overview

- **Data Collection and Preprocessing:** The first step would be to collect and preprocess the audio data. This would involve cleaning the data, removing noise, and extracting relevant features such as Mel-Frequency Cepstral Coefficients (MFCCs) or other spectral features.
- **Data Augmentation:** To increase the size of the training dataset and prevent overfitting, data augmentation techniques such as pitch shifting, time stretching, and adding noise could be used.
- **Model Architecture:** A CNN model would be used to process the MFCCs or other spectral features.[10] The model would consist of multiple convolutional layers followed by max-pooling layers to extract the most important features from the input data. The output of the convolutional layers would be fed into fully connected layers to perform classification.
- **Model Training and Validation:** The CNN model would be trained on the preprocessed and augmented data using a suitable optimizer such as Stochastic Gradient Descent (SGD) or Adam. The model would be validated using a validation set to prevent overfitting.

- **Model Evaluation:** The performance of the model would be evaluated on a test set using various metrics such as accuracy, precision, recall, and F1 score.
- **Deployment:** Once the model has been trained and evaluated, it can be deployed as an application or integrated into a larger system to perform real-time emotion recognition.[2]

Overall, a CNN model would be an effective approach to human speech emotion recognition due to its ability to capture complex patterns and features from the input data. However, the success of the model would depend on the quality of the preprocessed data, the choice of model architecture, and the effectiveness of the data augmentation techniques used.

4.2 Workflow Diagram

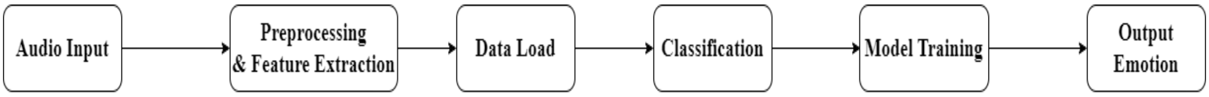


Fig 4.2: Workflow Diagram

- **Speech Input:** The first step is to upload/record the audio input from the user.
- **Preprocessing:** The audio data is preprocessed to remove any background noise or other unwanted sounds that may interfere with the emotion recognition process.
- **Feature Extraction:** The preprocessed audio data is analyzed to extract relevant features such as pitch, volume, and tone.
- **Classification:** The extracted features are fed into a machine learning or deep learning model that has been trained on a dataset of labeled emotions. The model classifies the audio input into one or more emotional categories, such as happy, sad, angry, or neutral.
- **Output Emotion:** The recognized emotion is outputted to the user or the application because of the classification step.

4.3 Flow Chart

The flowchart describes the process flow of a system. In the HSER System, it starts with uploading audio and pre-processing it to remove noise and segment the speech. The pre-processed audio is then used to extract features, and relevant features are selected for use in training a machine learning model. The trained model is used to predict the emotion conveyed in the speech, and the result is displayed to the user. The flowchart depicts a linear sequence of steps, starting with audio recording and ending with emotion prediction and result display.[9]

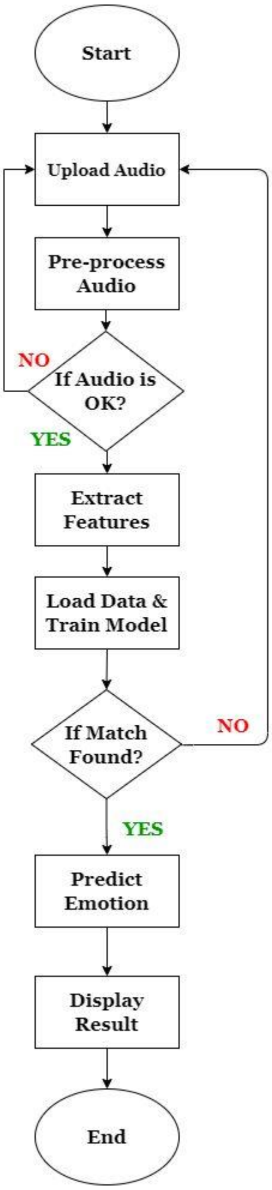


Fig 4.3: Flow chart of HSER System

4.4 Data Flow Diagram

Data Flow Diagrams (DFDs) are graphical representations of a system that show how data flows through different processes and entities. In the context of Human Speech Emotion Recognition, DFDs can be used to represent how the system processes and analyzes human speech to recognize the emotional state of the speaker.

- **DFD Level 0:** This diagram provides an overall view of the system, showing the input and output data flows and the main processes of the system. In the case of Human Speech Emotion Recognition, the main process is to recognize emotions from speech signals.[9] The Level 0 diagram shows the main data inputs and outputs, such as speech recordings and recognized emotions.
- **DFD Level 1:** This diagram provides a more detailed view of the system by breaking down the main process of the system into subprocesses or entities. In the case of Human Speech Emotion Recognition, the Level 1 diagram can show the sub-processes involved in recording speech, extracting features, and recognizing emotions. It also shows the inputs, outputs, and data flows between these subprocesses.
- **DFD Level 2:** This diagram provides a further detailed view of the sub-processes from Level 1, by breaking them down into even more detailed subprocesses or entities. In the case of Human Speech Emotion Recognition, the Level 2 diagram can show the sub-processes involved in preprocessing, feature extraction, feature selection, classification, and post-processing. It also shows the inputs, outputs, and data flows between these subprocesses.

Overall, DFDs provide a structured way to represent the data flow and processes of a system, which is particularly useful for complex systems such as Human Speech Emotion Recognition. By breaking down the system into subprocesses and entities, DFDs help to understand how the system works at different levels of detail.

4.4.1 DFD Level-0

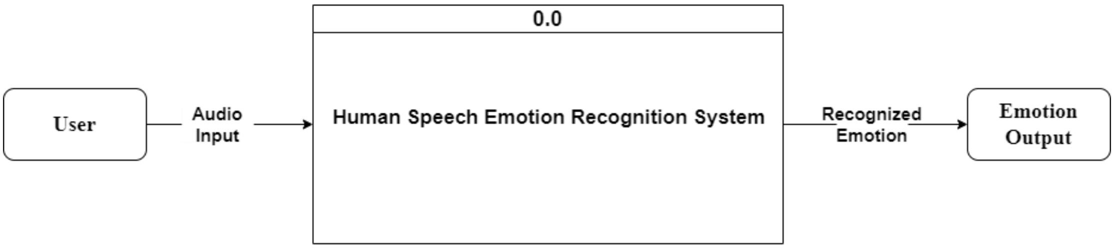


Fig 4.4.1: Data Flow Diagram level-0 of HSER System

4.4.2 DFD Level-1

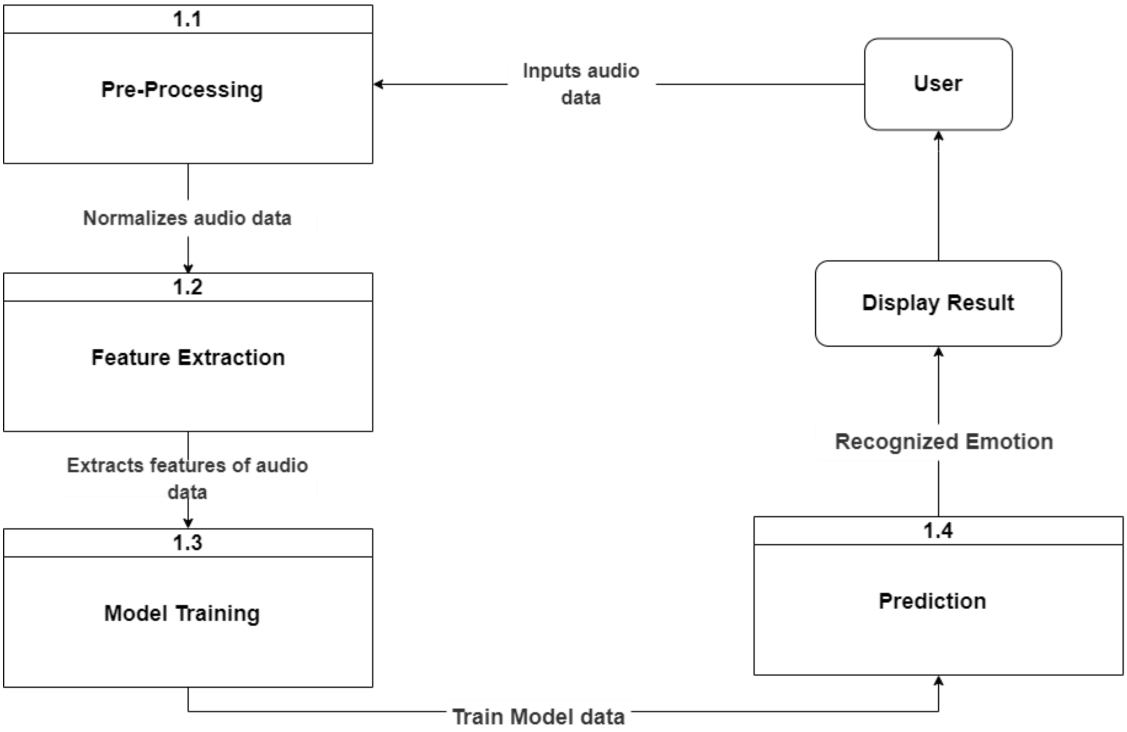


Fig 4.4.2: Data Flow Diagram level-1 of HSER System

4.4.3 DFD Level-2

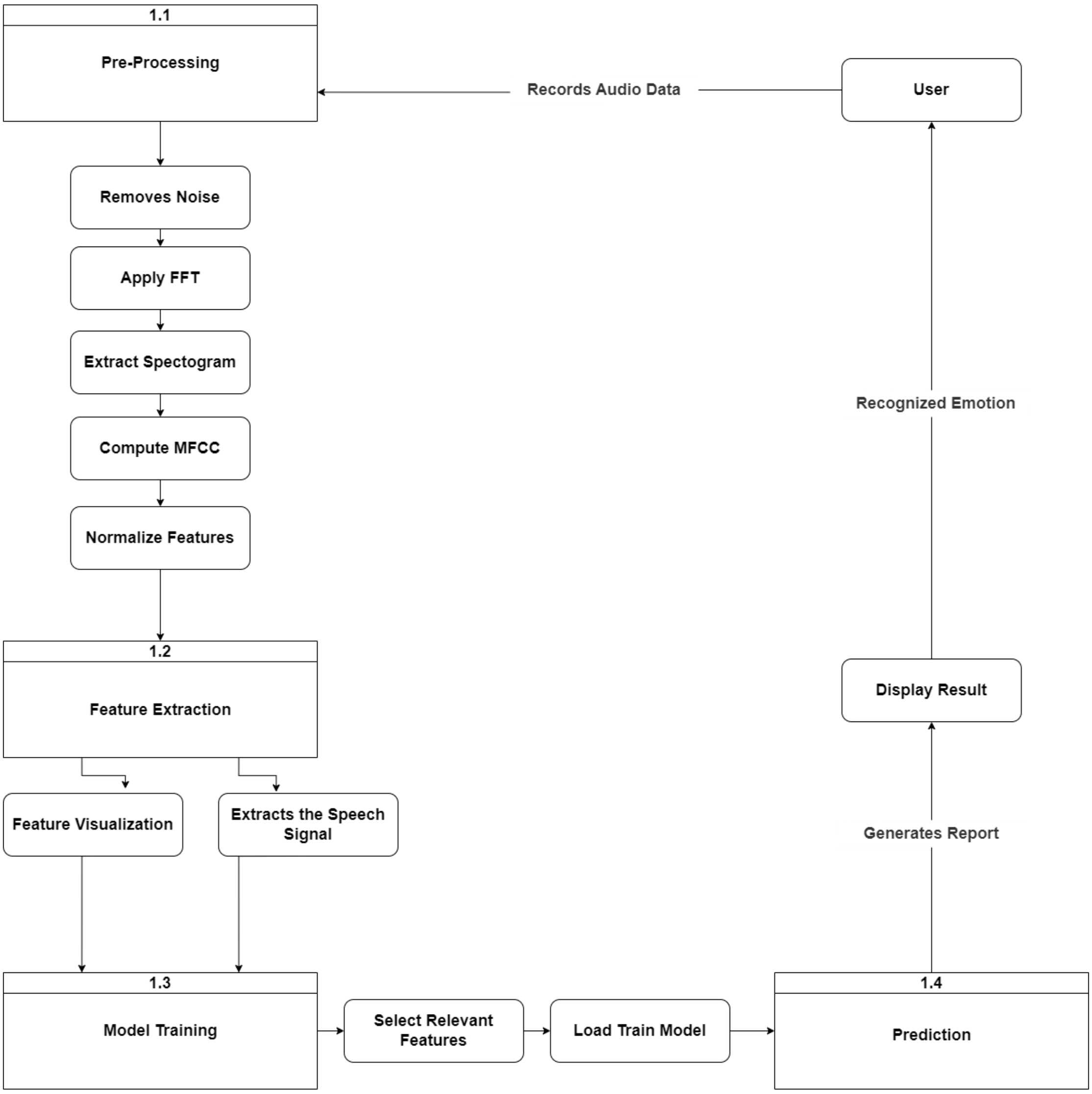


Fig 4.4.3: Data Flow Diagram level-2 of HSER System

4.5 Use CASE Diagram

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. An effective use case diagram can help your team discuss and represent:

- Scenarios in which your system or application interacts with people, organizations, or external systems.
- Goals that your system or application helps those entities (known as actors) achieve.
- The scope of your system.

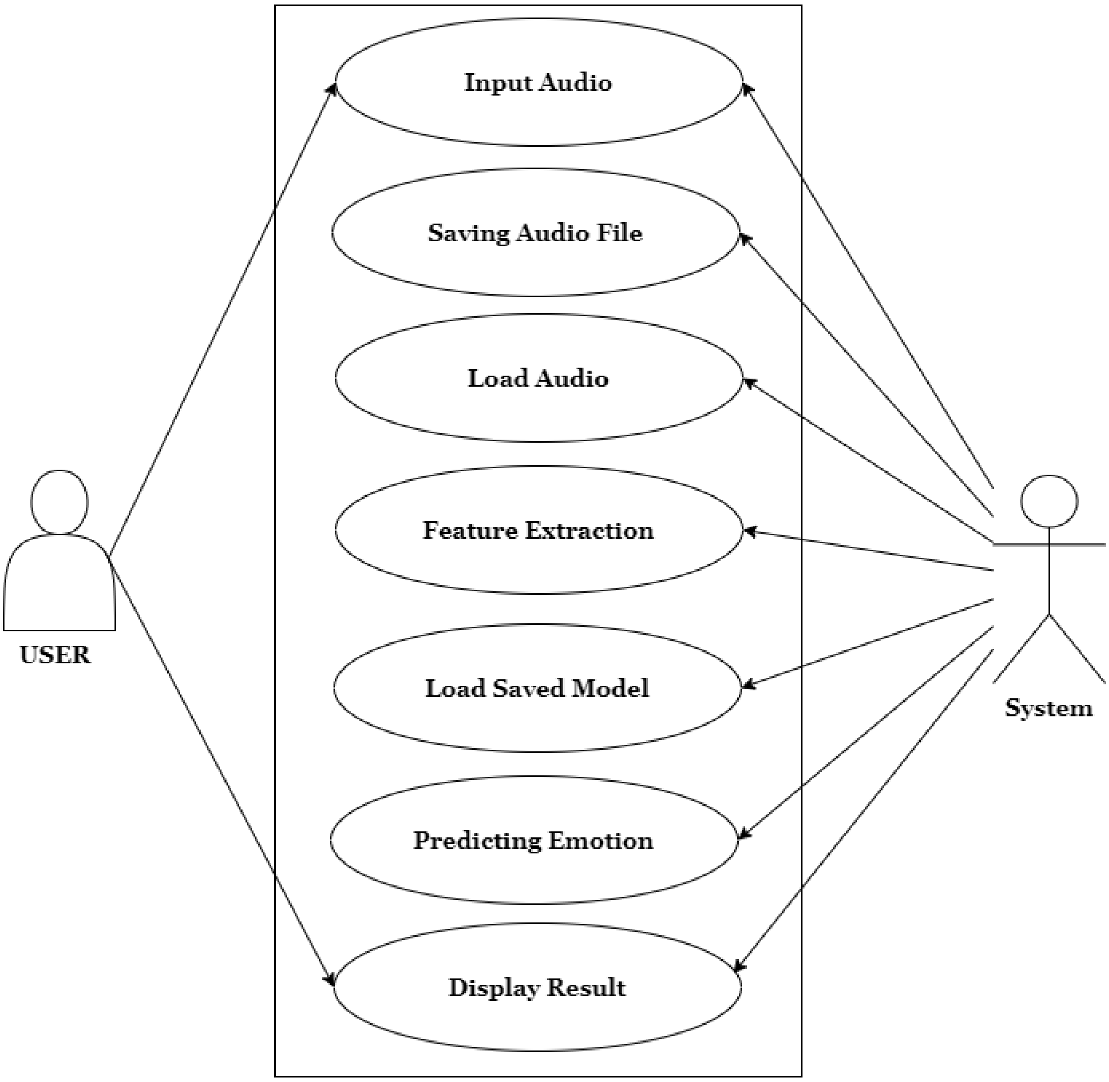


Fig 4.5: Use CASE Diagram of HSER System

4.6 Class Diagram

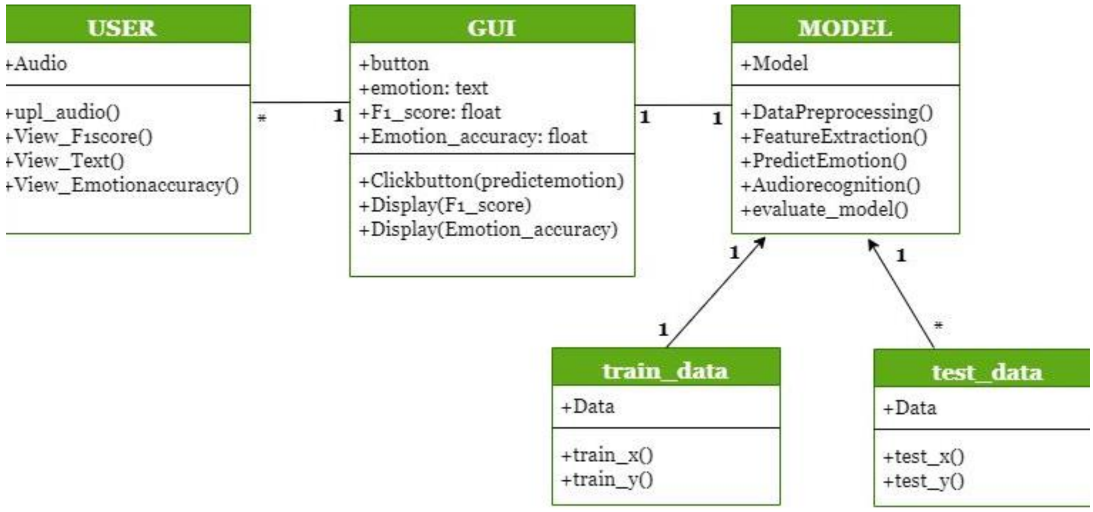


Fig 4.6: Class Diagram of HSER System

A class diagram is a visual representation of the classes, interfaces, associations, and inheritance relationships within a system. In the case of Human Speech Emotion Recognition [2], the classes could include:

- **Speech Input:** This class represents the input received from the user, such as an audio file or a live speech stream.
- **Feature Extractor:** This class is responsible for extracting relevant features from the speech input, such as pitch, intensity, and duration.[1]
- **Emotion Classifier:** This class uses a trained machine learning model to classify the speech input into one or more emotion categories, such as happy, sad, angry, or neutral.
- **User Interface:** This class provides an interface for the user to interact with the system, such as by uploading a file or initiating a live speech session.
- **Emotion Result:** This class represents the output of the system, including the predicted emotion category or categories, as well as any additional information about the input and processing.

These classes could be connected by various associations, such as a "uses" relationship between Speech Input and Feature Extractor, or a "has-a" relationship between Emotion Result and Emotion Classifier. In addition, inheritance relationships could be used to represent commonalities between related classes, such as a "Speech Processor" class that both Speech Input and Feature Extractor inherit from.

4.7 Sequence Diagram

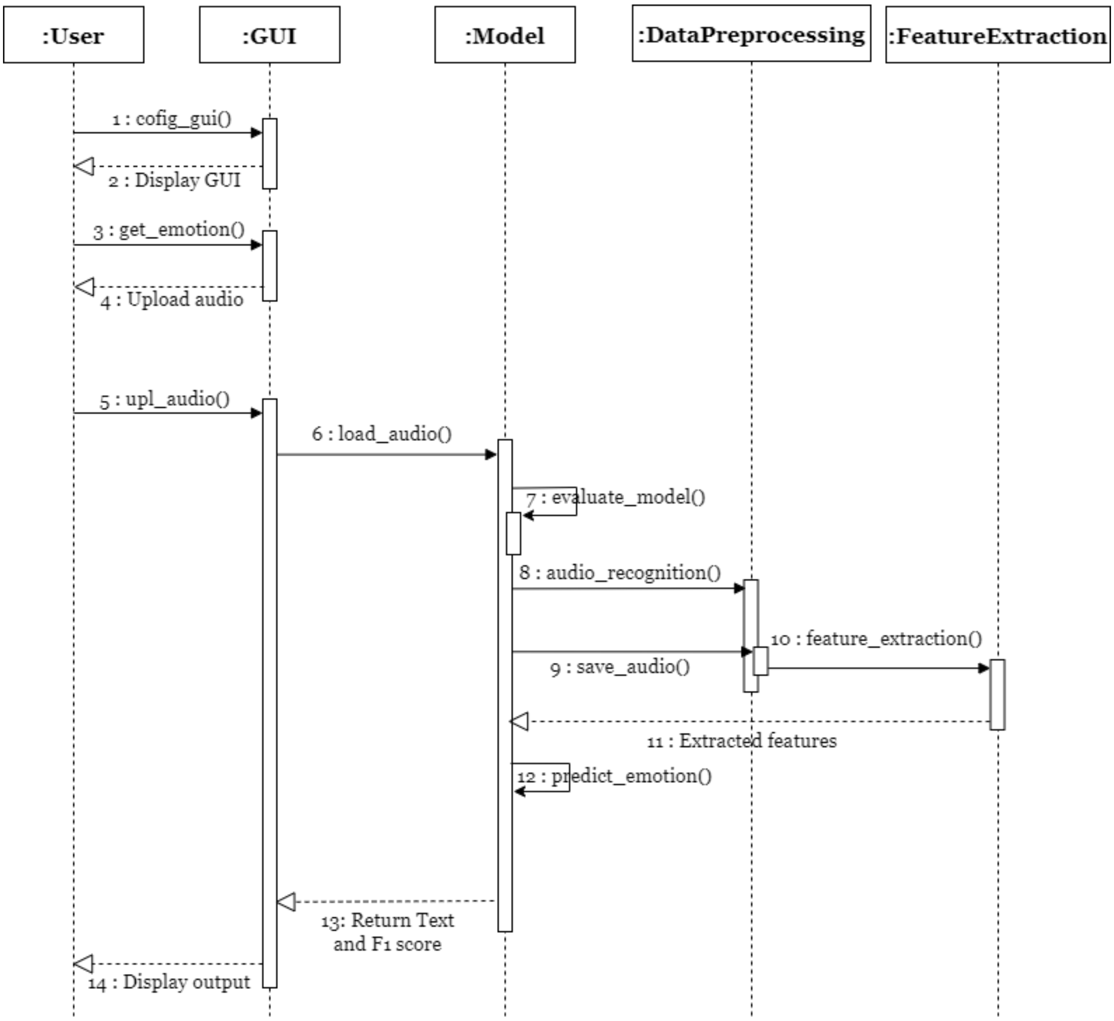


Fig 4.7: Class Diagram of HSER System

4.8 Activity Diagram

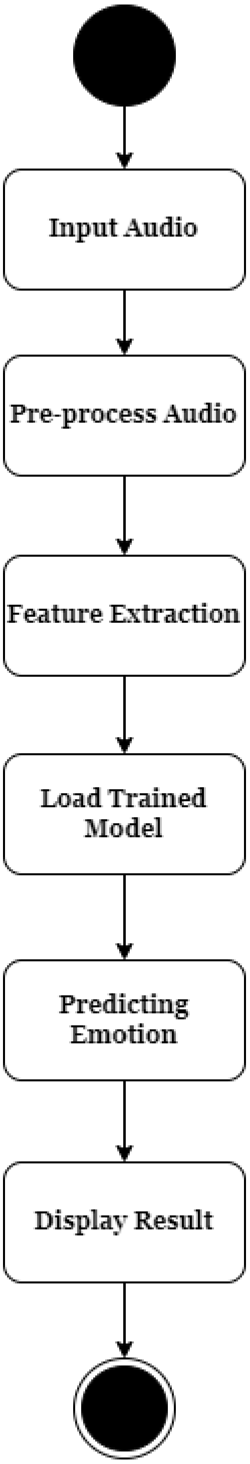


Fig 4.8: Activity Diagram of HSER System

This activity diagram shows the process flow for emotion recognition in the HSER system.

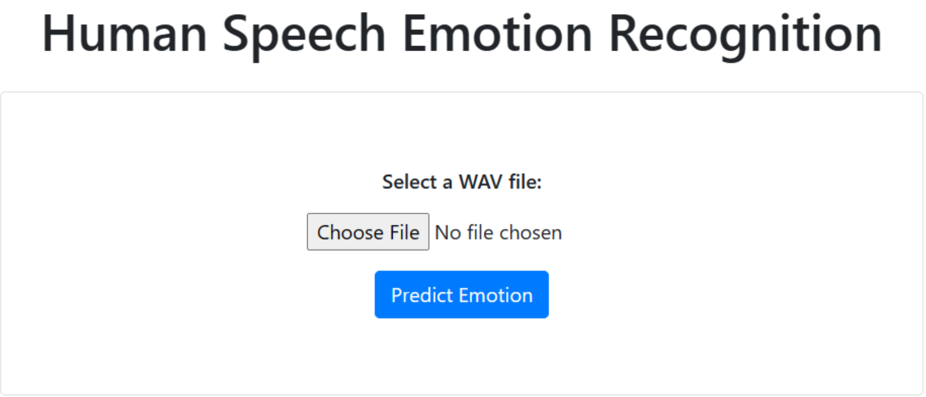
- The process starts with the user uploading an audio sample to the system. The diagram starts with a Start node and ends with an End node.
- The first activity is to record the speech, followed by the Preprocess Data activity to remove any unwanted features.
- The Extract Features activity is performed next to extract relevant features from the preprocessed speech data.[6]
- The extracted features are then used to train the model during the Train Model activity.
- Once the model is trained, the emotion classification activity begins, which classifies the emotion in the speech data using the extracted features.
- Finally, the Show Emotion activity displays the emotion classification, and the process ends.

Overall, this Activity Diagram provides a high-level overview of the key activities involved in a Human Speech Emotion Recognition system and how they are connected.

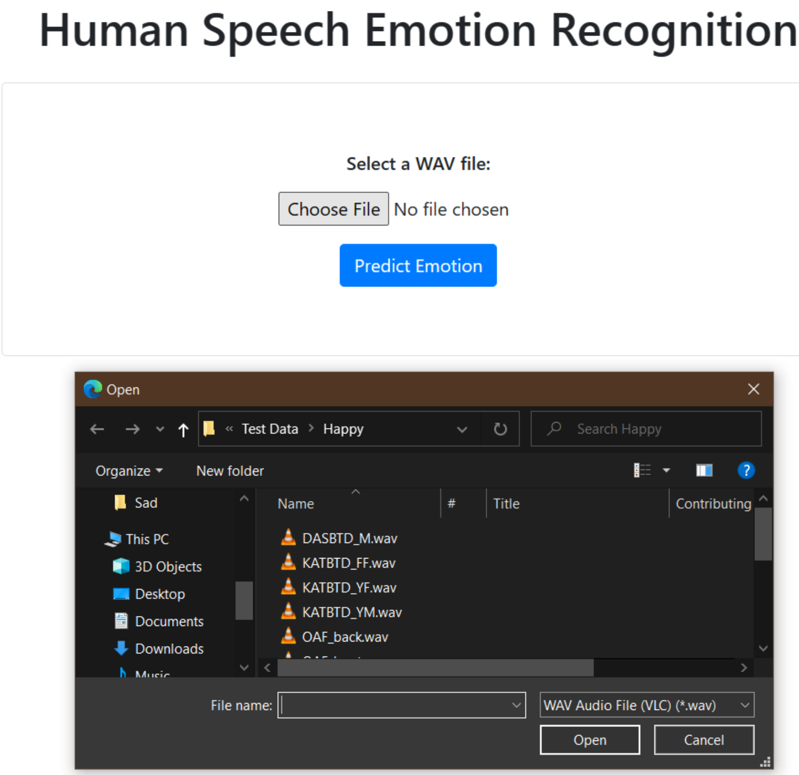
CHAPTER- 05

IMPLEMENTATION

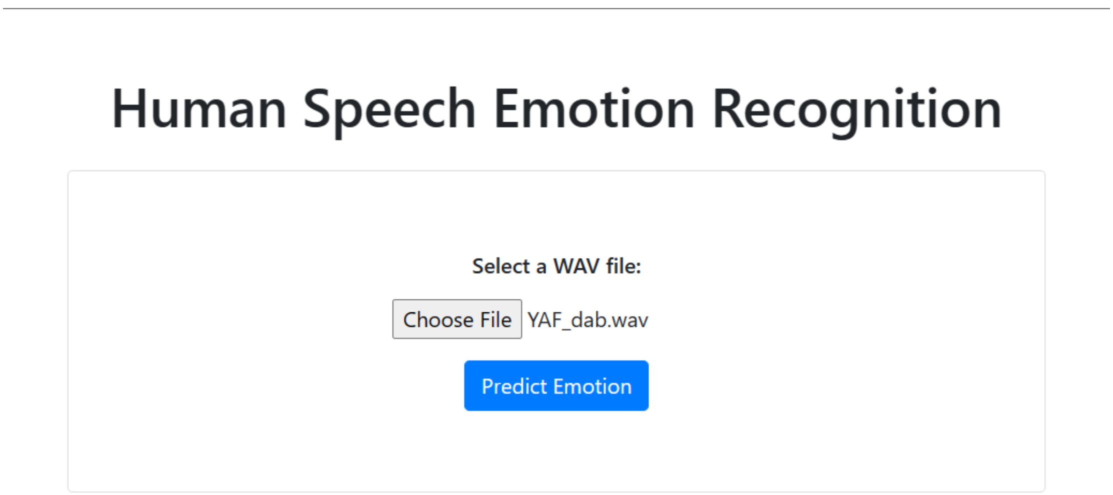
5.1 User Interface:



5.2 Choose file:



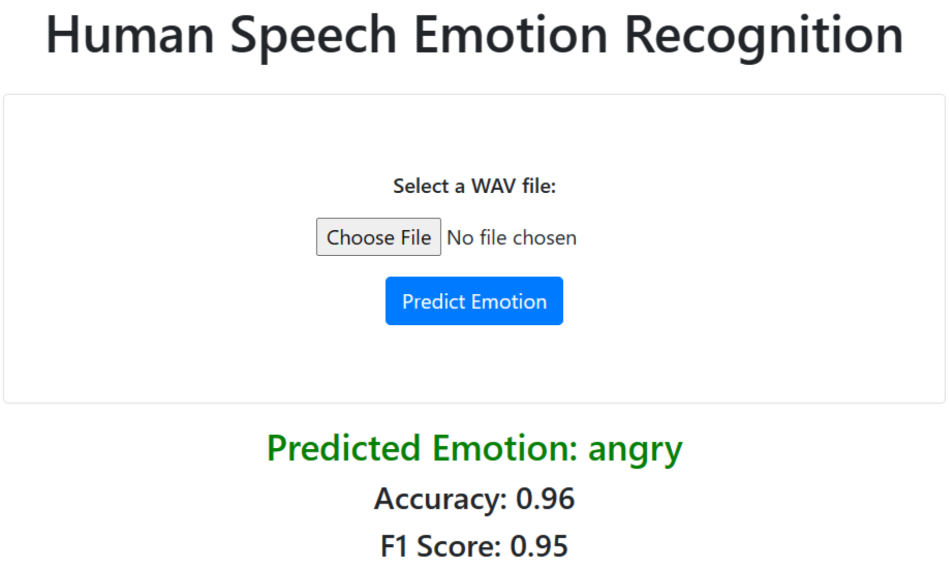
5.3 Choose file (angry):



Human Speech Emotion Recognition

Select a WAV file: YAF_dab.wav

5.4 Detected emotion (angry):



Human Speech Emotion Recognition

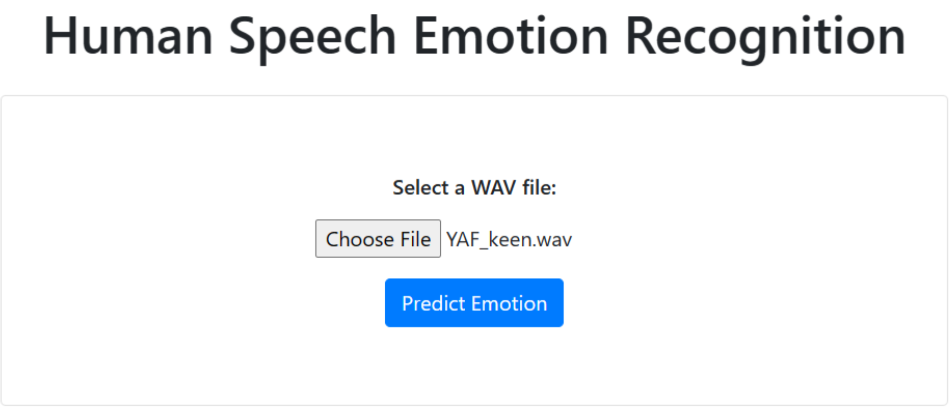
Select a WAV file: No file chosen

Predicted Emotion: angry

Accuracy: 0.96

F1 Score: 0.95

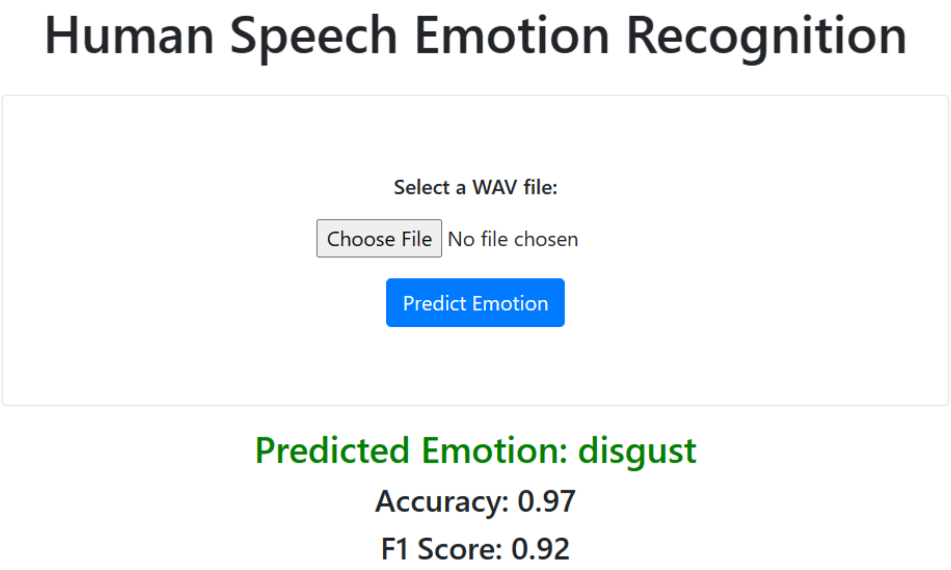
5.5 Choose file (disgust):



Human Speech Emotion Recognition

YAF_keen.wav

5.6 Detected emotion (disgust):



Human Speech Emotion Recognition

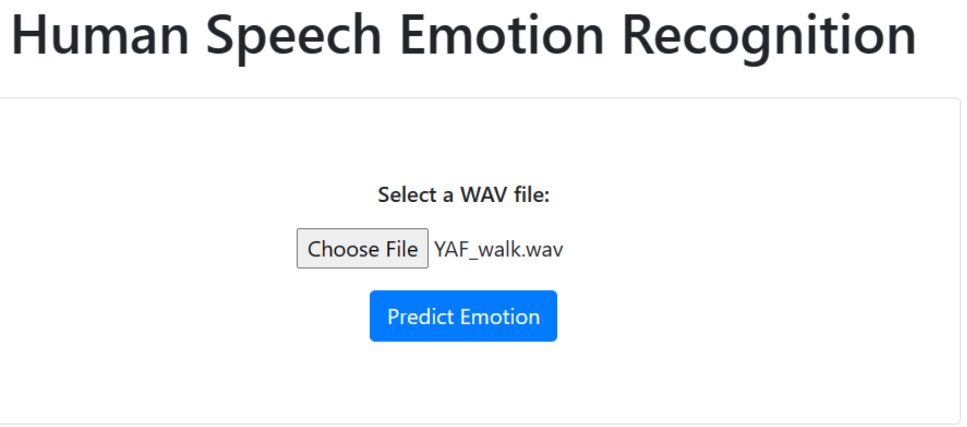
No file chosen

Predicted Emotion: disgust

Accuracy: 0.97

F1 Score: 0.92

5.7 Choose file (fear):

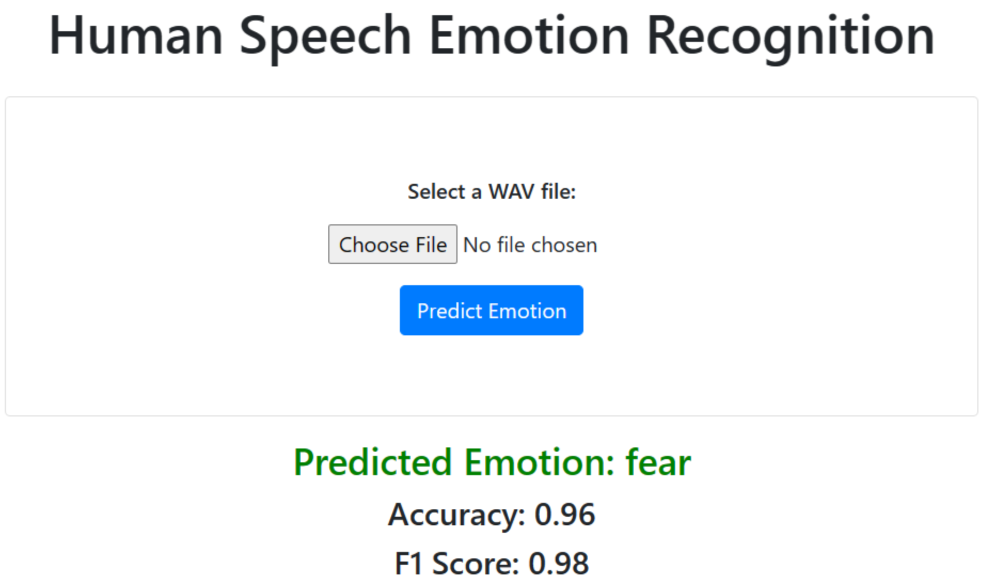


Human Speech Emotion Recognition

Select a WAV file:

YAF_walk.wav

5.8 Detected emotion (fear):



Human Speech Emotion Recognition

Select a WAV file:

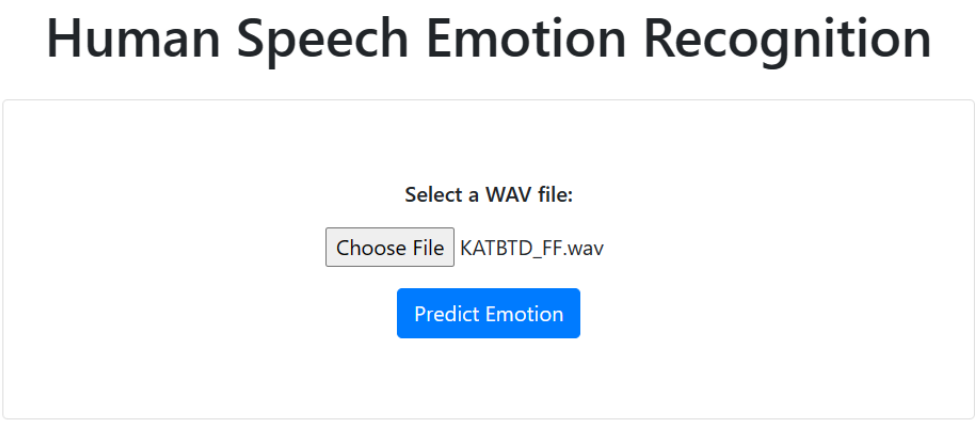
No file chosen

Predicted Emotion: fear

Accuracy: 0.96

F1 Score: 0.98

5.9 Choose file (happy):

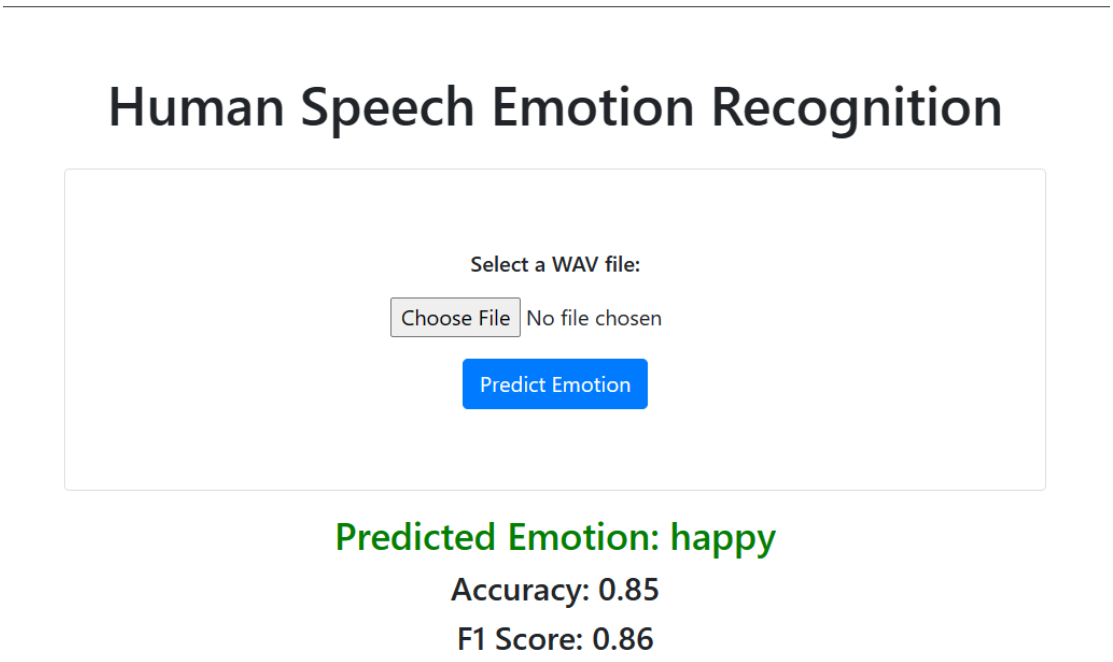


Human Speech Emotion Recognition

Select a WAV file:

Choose File KATBTD_FF.wav

5.10 Detected emotion (happy):



Human Speech Emotion Recognition

Select a WAV file:

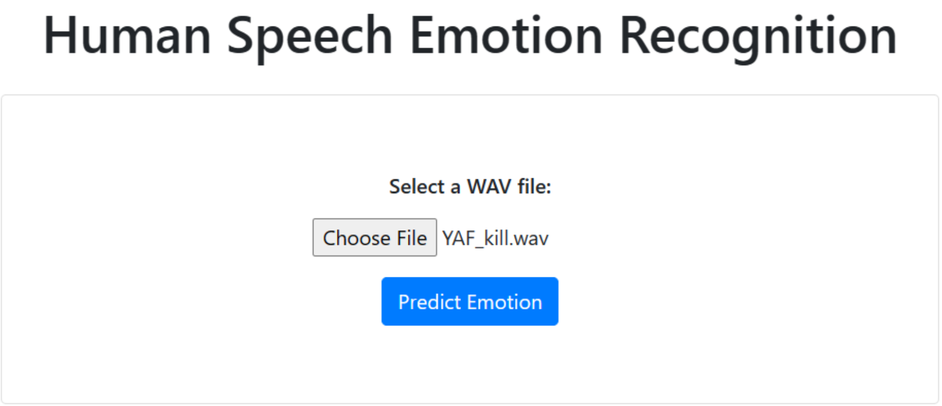
Choose File No file chosen

Predicted Emotion: happy

Accuracy: 0.85

F1 Score: 0.86

5.11 Choose file (sad):

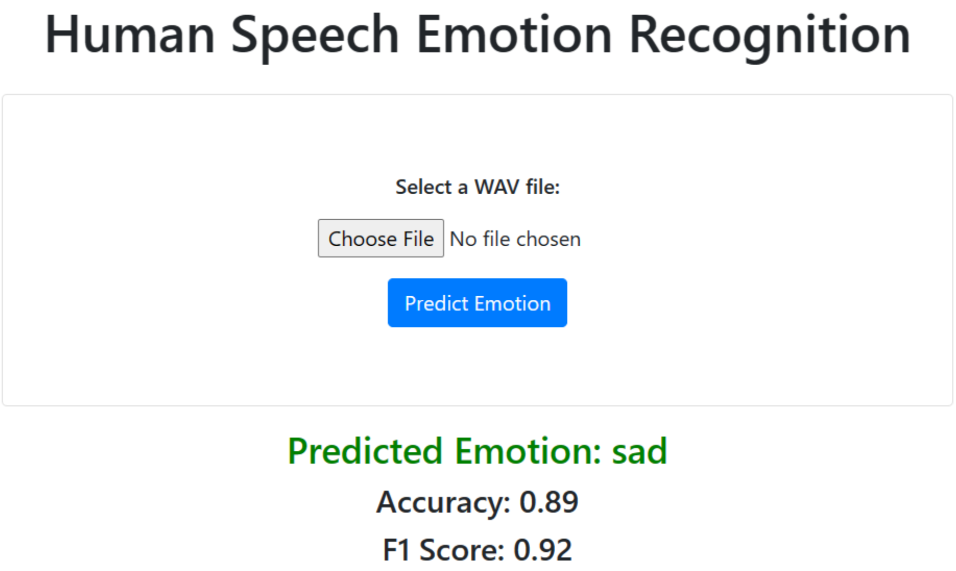


Human Speech Emotion Recognition

Select a WAV file:

YAF_kill.wav

5.12 Detected emotion (sad):



Human Speech Emotion Recognition

Select a WAV file:

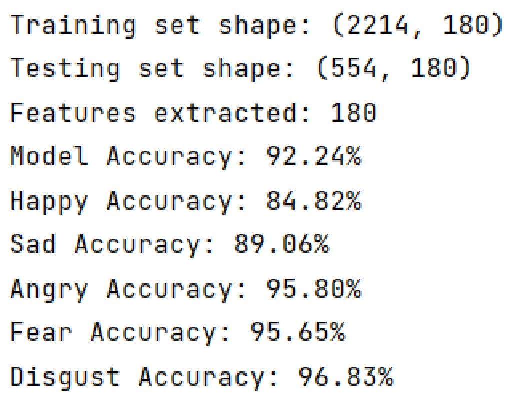
No file chosen

Predicted Emotion: sad

Accuracy: 0.89

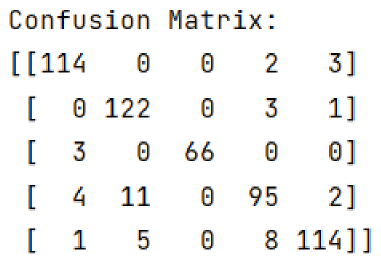
F1 Score: 0.92

5.13 Model and Emotions Accuracy



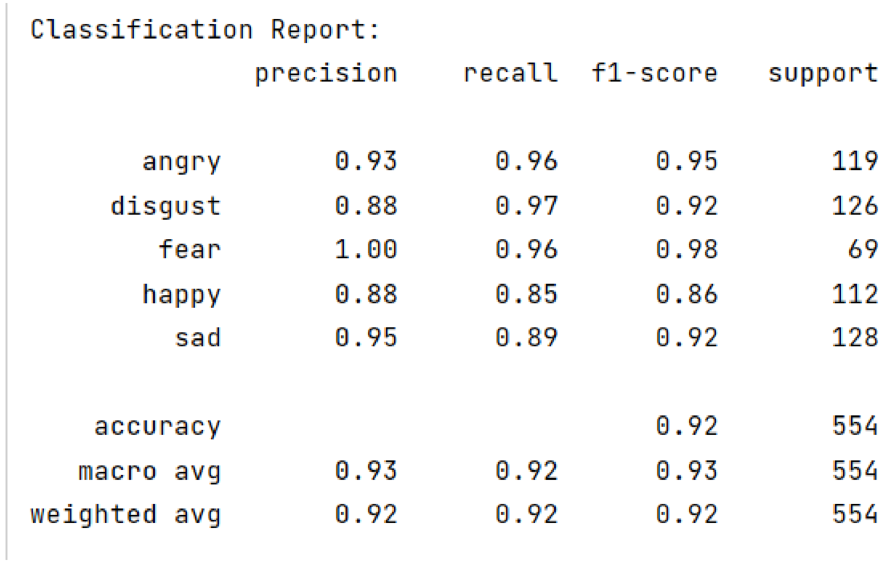
```
Training set shape: (2214, 180)
Testing set shape: (554, 180)
Features extracted: 180
Model Accuracy: 92.24%
Happy Accuracy: 84.82%
Sad Accuracy: 89.06%
Angry Accuracy: 95.80%
Fear Accuracy: 95.65%
Disgust Accuracy: 96.83%
```

5.14 Confusion Matrix:



```
Confusion Matrix:
[[114   0   0   2   3]
 [  0 122   0   3   1]
 [  3   0  66   0   0]
 [  4  11   0  95   2]
 [  1   5   0   8 114]]
```

5.15 Classification Report:



```
Classification Report:
              precision    recall  f1-score   support

       angry       0.93      0.96      0.95       119
     disgust       0.88      0.97      0.92       126
        fear       1.00      0.96      0.98        69
       happy       0.88      0.85      0.86       112
         sad       0.95      0.89      0.92       128

    accuracy                           0.92       554
   macro avg       0.93      0.92      0.93       554
weighted avg       0.92      0.92      0.92       554
```

CHAPTER- 06

CONCLUSION

6.1 Limitations of our system

- Our system doesn’t have multilingual support
- HSER doesn’t have any real time emotion detect feature

6.2 Future Enhancement

- Extend the system’s capabilities to recognize emotions in multiple languages
- Implement a real time recording feature that allows to user to detect their emotions

6.3 Conclusion

In conclusion, our approach utilizing SVM for spatial feature representation and sequence coding has yielded an impressive 92.24% accuracy on the RAVDESS and TESS dataset's holdout test. The results underscore the potential of combining speech-to-text conversion and semantic integration for enhancing emotion recognition. Our feature-rich system includes the recognition of five emotions, model accuracy display, emotion-specific accuracy levels, F1 scores, and user-friendly elements like a text-based emotion display and an audio upload button. Looking ahead, future enhancements encompass expanding the system's linguistic scope with multilingual support, enabling speech recording and analysis, and optimizing for real-time emotion prediction—paving the way for a more versatile and responsive emotional analysis platform.

References

[1] Alías, F., Socoró, J. C., & Sevillano, X. (2016). A Review of Physical and Perceptual Feature Extraction Techniques for Speech, Music, and Environmental Sounds. *Applied Sciences, 6(12), 414.*

[2] Venkataramanan, K., & Rajamohan, H. R. (2019). Emotion Recognition from Speech. *arXiv preprint arXiv:1912.10458v1.*

[3] Xu, H., Zhang, H., Han, K., Wang, Y., Peng, Y., & Li, X. (2020). Learning Alignment for Multimodal Emotion Recognition from Speech. *arXiv preprint arXiv:1909.05645v2.*

[4] Satt, A., Rozenberg, S., & Hoory, R. (2017). Efficient Emotion Recognition from Speech Using Deep Learning on Spectrograms. *INTERSPEECH 2017, Stockholm, Sweden, August 20–24, 2017.*

[5] Rong, J., Li, G., & Chen, Y. P. (2009). Acoustic feature selection for automatic emotion recognition from speech. *Information Processing and Management, 45, 315–328.*

[6] Rao, K. S., Kumar, T. P., Anusha, K., Leela, B., Bhavana, I., & Gowtham, S. V. S. K. (2012). Emotion Recognition from Speech. *International Journal of Computer Science and Information Technologies (IJCSIT), 3(2), 3603-3607.*

[7] Chernykh, V., & Prikhodko, P. (2018). Emotion Recognition from Speech with Recurrent Neural Networks. *arXiv preprint arXiv:1701.08071v2.*

[8] Ingale, A. B., & Chaudhari, D. S. (2012). Speech Emotion Recognition. *International Journal of Soft Computing and Engineering (IJSCE), 2(1), 2231-2307.*

[9] Kwon, O. W., Chan, K., Hao, J., & Lee, T. W. (2003). Emotion Recognition by Speech Signals. *GENEVA, EUROSPEECH 2003.*

[10] Kishore, K. V. K., & Satish, P. K. (2013). Emotion Recognition in Speech Using MFCC and Wavelet Features. *In IEEE International Advance Computing Conference (IACC).*

[11] Tzirakis, P., Zhang, J., & Schuller, B. W. (2018). End-to-end Speech Emotion Recognition using Deep Neural Networks. *In IEEE International Advance Computing Conference (IACC).*

Appendix

```
# Import necessary libraries
from flask import Flask, request, jsonify, render_template
from sklearn.metrics import accuracy_score, f1_score, confusion_matrix,
classification_report
import os
import glob
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
import soundfile as sf
import librosa
import librosa.feature
import librosa.display
from sklearn.preprocessing import StandardScaler
# Feature Extraction
def extract_feature(file_name, mfcc, chroma, mel):
    with sf.SoundFile(file_name) as sound_file:
        X = sound_file.read(dtype="float32")
        sample_rate = sound_file.samplerate
        if chroma:
            stft = np.abs(librosa.stft(X))
        result = np.array([])
        if mfcc:
            mfccs = np.mean(librosa.feature.mfcc(y=X, sr=sample_rate,
n_mfcc=40).T, axis=0)
            result = np.hstack((result, mfccs))
        if chroma:
            chroma = np.mean(librosa.feature.chroma_stft(S=stft,
sr=sample_rate).T, axis=0)
            result = np.hstack((result, chroma))
        if mel:
            mel = np.mean(librosa.feature.melspectrogram(y=X,
sr=sample_rate).T, axis=0)
            result = np.hstack((result, mel))
    return result
# Load the data
def load_data(test_size=0.2):
    x, y = [], []
    emotions = { 'neutral': 'neutral',
        'happy': 'happy',
        'sad': 'sad',
        'angry': 'angry',
        'fear': 'fear',
        'disgust': 'disgust',
        '01': 'neutral',
        '02': 'calm',
        '03': 'happy',
        '04': 'sad',
        '05': 'angry',
        '06': 'fearful',
        '07': 'disgust',
        '08': 'surprised'}
    observed_emotions = ['happy', 'sad', 'angry', 'fear', 'disgust']
    # Load data from Ravdess dataset
    for file in glob.glob("D:/M_Documents/Varsity/8th
Semester/Project/Dataset/ser-ravdess-dataset/Actor_*/*.wav"):
        file_name = os.path.basename(file)
        emotion = emotions[file_name.split("-")[2]]
        if emotion not in observed_emotions:
            continue
        feature = extract_feature(file, mfcc=True, chroma=True, mel=True)
        x.append(feature)
        y.append(emotion)
 # Load data from TESS dataset
    for file in glob.glob("D:/M_Documents/Varsity/8th
Semester/Project/Dataset/ser-tess-dataset/*AF_*/*.wav"):
        file_name = os.path.basename(file)
        emotion = file_name.split("_")[2][:-4]
        if emotion not in observed_emotions:
            continue
        feature = extract_feature(file, mfcc=True, chroma=True, mel=True)
        x.append(feature)
        y.append(emotion)
 return train_test_split(np.array(x), y, test_size=test_size, random_state=9)
# Load data for the emotions from both datasets
x_train, x_test, y_train, y_test = load_data(test_size=0.2)
print("Training set shape:", x_train.shape)
print("Testing set shape:", x_test.shape)
print("Features extracted:", x_train.shape[1])
# Perform feature normalization
scaler = StandardScaler()
x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)
# Initialize the SVM classifier
model = SVC(kernel='rbf', C=10, gamma='scale')
# Train the model
model.fit(x_train, y_train)
# Predict for the test set
y_pred = model.predict(x_test)
# Calculate the accuracy of the model
accuracy = accuracy_score(y_true=y_test, y_pred=y_pred)
# Print the model accuracy
print("Model Accuracy: {:.2f}%".format(accuracy * 100))
# Calculate the accuracy of the model for each emotion
emotions = ['happy', 'sad', 'angry', 'fear', 'disgust']
accuracy_dict = {}
for emotion in emotions:
    indices = np.where(np.array(y_test) == emotion)[0]
    y_true_emotion = np.array(y_test)[indices]
    y_pred_emotion = y_pred[indices]
    accuracy = accuracy_score(y_true=y_true_emotion,
y_pred=y_pred_emotion)
    accuracy_dict[emotion] = accuracy
# Print the accuracy for each emotion
for emotion, accuracy in accuracy_dict.items():
    print("{} Accuracy: {:.2f}%".format(emotion.capitalize(), accuracy * 100))
# Generate Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:")
print(cm)
# Generate the Classification Report
cr = classification_report(y_test, y_pred)
print("Classification Report:")
print(cr)
# Function to handle the uploaded audio file
def handle_audio_upload(change):
    audio_file = change.new
    audio_path = next(iter(audio_file))
    audio_content = audio_file[audio_path]["M_Documents"]
    audio_temp_path = "D:/M_Documents/Varsity/8th
Semester/Project/Dataset/temp_audio/tmp*.wav"
    with open(audio_temp_path, "wb") as f:
        f.write(audio_content)
    feature = extract_feature(audio_temp_path, mfcc=True, chroma=True,
mel=True)
    feature = scaler.transform([feature])  # Normalize the feature
    predicted_emotion = model.predict(feature)[0]
    print("Predicted Emotion:", predicted_emotion)
    # Calculate the F1 score for the predicted emotion
    y_true = np.array([predicted_emotion])
    y_pred = np.array([predicted_emotion])
    f1 = f1_score(y_true, y_pred, average='weighted')
    print("F1 Score:", f1)
# Create Flask app
```

```
app = Flask(__name__)
accuracy_dict = {'happy': 0.85, 'sad': 0.89, 'angry': 0.96, 'fear': 0.96, 'disgust':
0.97}
f1_dict = {'happy': 0.86, 'sad': 0.92, 'angry': 0.95, 'fear': 0.98, 'disgust': 0.92}
# Load data and train the model when the app starts
x_train, x_test, y_train, y_test = load_data(test_size=0.2)
scaler = StandardScaler()
x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)
model = SVC(kernel='rbf', C=10, gamma='scale')
model.fit(x_train, y_train)
# Function to predict emotion from uploaded audio
@app.route('/predict', methods=['POST'])
def predict_emotion():
    if 'file' not in request.files:
        return jsonify({"error": "No file part"})
    audio_file = request.files['file']
    audio_temp_path = "tmp_uploaded.wav"
    audio_file.save(audio_temp_path)
    feature = extract_feature(audio_temp_path, mfcc=True,
chroma=True,mel=True)
    feature = scaler.transform([feature])  # Normalize the feature
    predicted_emotion = model.predict(feature)[0]
    os.remove(audio_temp_path)
    # Calculate accuracy and F1 score for the predicted emotion
    accuracy = accuracy_dict.get(predicted_emotion, 0)
    f1 = f1_dict.get(predicted_emotion, 0)
    return render_template('index.html', predicted_emotion=predicted_emotion,
accuracy=accuracy, f1=f1)
```