

SPEECH TO TEXT WITH EMOJI

TONG KAH PAU

UNIVERSITI TUNKU ABDUL RAHMAN

SPEECH TO TEXT WITH EMOJI

TONG KAH PAU

**A project report submitted in partial fulfilment of the
requirements for the award of Bachelor of Science
(Honours) Software Engineering**

**Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman**

May 2023

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature :



Name : Tong Kah Pau


ID No. : 1901229

Date : 2nd April 2023

APPROVAL FOR SUBMISSION

I certify that this project report entitled **SPEECH TO TEXT WITH EMOJIS** was prepared by **TONG KAH PAU** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Science (Hons) Software Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature : 

Supervisor : Dr Pua Chang Hong

Date : 2nd April 2023

Signature : *D.gunavathi*

Co-Supervisor : Ms Gunavathi a/p Duraisamy

Date : 2nd April 2023

The copyright of this report belongs to the author under the terms of the copyright Act 1987 as qualified by Intellectual Property Policy of Universiti Tunku Abdul Rahman. Due acknowledgement shall always be made of the use of any material contained in, or derived from, this report.

© 2023, Tong Kah Pau. All right reserved.

ACKNOWLEDGEMENTS

I would like to express my gratitude to everyone who helped this project be completed successfully. I would like to convey my thanks to Dr. Pua Chang Hong, my supervisor, and Ms Gunavathi a/p Duraisamy, my co supervisor, for their important assistance, patience, and counsel during the project's growth.

In addition, I want to thank my supportive parents and friends who assisted me with the project and gave me moral support and both mental and physical assistance.

ABSTRACT

Speech transcription technology has been a significant life changer in the media, entertainment, and education fields. Transcript services have greatly simplified the work of record-keeping, research, and note-taking without the inconvenience of manually transcribing protracted audio or video segments for hours at a time. However, reading plain text alone from the transcription cannot convey the messenger's emotion compared to listening to it. Humans are wired to experience many basic emotions. These fundamental emotions assist us in understanding, connecting, and communicating with others. Thus, emoticons deliver the user's emotions in the message. With our current technology, we can add emoticons to our text by choosing it manually or by saying the emoticons' names using speech recognition technology. However, this may cause some hassle and other problems. In this project, an artificial intelligence-based mobile application for emotional voice transcription was proposed to solve the difficulties of improving digital communication, increasing equality for disabled persons, and boosting attentiveness in online courses. The objectives of this project encompass examining the feasibility of voice recognition for emotion detection, develop an emotional voice recognition system that accurately measures various speech features to display appropriate emojis and create a speech-to-text solution that transcribes text with emojis at a rate comparable to the user's speech rate and emotional portrayal. Furthermore, prototyping methodology was chosen as the project approach. It consists of a requirement analysis phase, followed by a five steps repeatable cycle: design, model training, prototyping, review, and refinement, and finally, the development, test, and release phase. In conclusion, the final prototype achieved a processing speed of 10-15 seconds, a speech transcript accuracy of 99.5%, and an emotion identification accuracy of 80.3% via incremental upgrades and adjustments through the prototype and development phase. Although there were future enhancements and improvements, such as a customised voice profile, multilingual assistance, transcription sharing and system architecture change to the client and server side, the project is considered successful where all objectives are fulfilled.

TABLE OF CONTENTS

DECLARATION	i
APPROVAL FOR SUBMISSION	ii
ACKNOWLEDGEMENTS	iv
ABSTRACT	v
TABLE OF CONTENTS	vi
LIST OF TABLES	xi
LIST OF FIGURES	xiv
LIST OF APPENDICES	xix

CHAPTER

1	INTRODUCTION	1
	1.1 Introduction	1
	1.2 Problem Statement	2
	1.2.1 Efficiency of digital communications	2
	1.2.2 Equality between normal people and disabled people	3
	1.2.3 Poor concentration in online class	3
	1.3 Project Objectives	4
	1.4 Project Solution	4
	1.5 Project Approach	5
	1.6 Project Scope	6
	1.6.1 Target Users	6
	1.6.2 Modules	7
	1.7 Limitation	7
2	LITERATURE REVIEW	8
	2.1 Introduction	8

2.2	Voice Recognition system for emotion detection	8
2.2.1	Voice Feature Extraction	10
2.2.2	Approaches on Speech Emotion Recognition (SER)	12
2.2.3	Results comparison among methods	14
2.2.4	Conclusion	18
2.3	Evaluation on current speech transcription methodologies	19
2.3.1	Comparing Google Cloud Speech with Real-time transcription by certified SLPs and TTs	20
2.3.2	Human Evaluation on speech transcription technology	21
2.3.3	Parameters affecting Speech to Text methodologies	23
2.3.4	Conclusion	25
2.4	Application of artificial intelligence techniques on mobile applications	26
2.4.1	Conclusion	30
2	METHODOLOGY AND WORK PLAN	31
3.1	Introduction	31
3.2	Prototyping Development Methodology	31
3.2.1	Requirement Gathering Phase	31
3.2.2	Design Phase	32
3.2.3	Iteration Process	33
3.2.4	Implementation Phase	35
3.2.5	Evaluation Phase	36
3.2.6	Refinement Phase	37
3.3	AI Learning Model Methodology	37
3.3.1	Data Acquisition	38
3.3.2	Pre-processing	39
3.3.3	Splitting and Balancing the Dataset	40
3.3.4	Building and Training the Model	41
3.3.5	Evaluating Performance	42

	3.3.6	Tuning Hyperparameters	42
	3.3.7	Deployment of software	43
3.4		Speech Transcription Methodologies	43
	3.4.1	Automatic Speech Recognition (ASR)	44
	3.4.2	Deep Learning based model	44
	3.4.3	Conclusion	45
3.5		Emoji Recognition Methodologies	45
	3.5.1	Long Short-Term Memory (LSTM) networks	46
	3.5.2	Multi-Layer Perceptron (MLP) Classifier Networks	46
	3.5.3	CNN-LSTM Networks	47
	3.5.4	Conclusion	48
3.6		Mobile Application Development	48
	3.6.1	AI model deployment	48
	3.6.2	Module Integration Methodology	50
3.7		Workplan	51
	3.7.1	Work Breakdown Structure	51
	3.7.2	Gantt Chart	53
4		PROJECT SPECIFICATIONS	55
	4.1	Introduction	55
	4.2	Requirement Specification	55
	4.2.1	Functional Requirements	55
	4.2.2	Non-Functional Requirements	55
	4.3	Use Cases	56
	4.3.1	Use Case diagram	56
	4.3.2	Use Case Description	57
5		SYSTEM DESIGN	63
	5.1	Introduction	63
	5.2	System Architecture Design	63
	5.2.1	Backend flow design	63

5.3	User Interface design	65
6	IMPLEMENTATION	67
6.1	Introduction	67
6.2	Tools and Technologies	67
6.3	Backend Development	68
6.3.1	Speech to Text module	68
6.3.2	Emotion Recognition module	75
6.3.3	Emoji Selection Implementation	76
6.4	Frontend Development	77
6.4.1	User Interface and Interaction Implementation	77
6.4.2	Integration of Backend Modules	78
6.5	Versioning and Iterative Improvements	82
6.5.1	Version 1: Initial Prototype	82
6.5.2	Version 2: Additional feature and enhanced Speech to Text module	82
6.5.3	Version 3: Enhanced Emotion Prediction and new framework	82
6.5.4	Conclusion	83
7	EVALUATION AND TESTING	84
7.1	Introduction	84
7.2	Speech to Text models performances	84
7.3	Emotion Features and Visualisation evaluation	85
7.4	Emotion Recognition models performance	91
7.4.1	LSTM	91
7.4.2	MLP Classifier	94
7.4.3	CNN-LSTM Model	97
7.4.4	Conclusion	101
7.5	Version Comparison and Selection	101
7.6	Testing	103
7.6.1	Testing Objectives	103

	7.6.2	Unit Testing	103
	7.6.3	Integration Testing	113
	7.6.4	User Acceptance Test	118
	7.6.5	Usability Testing	122
8		CONCLUSION	125
	8.1	Introduction	125
	8.2	Fulfilment of Objectives	125
	8.3	Suggestions and Recommendations	126
8		REFERENCES	128
8		APPENDICES	131

LIST OF TABLES

Table 2.1: Acoustic Characteristics of Emotions	12
Table 2.2: Confusion matrix of the RBF LIBSVM classifier (Gender Independent)	15
Table 2.3: Confusion matrix of the the RBF LIBSVM classifier (Male)	15
Table 2.4: Confusion matrix of the RBF LIBSVM classifier (Female)	15
Table 2.5: Confusion matrix of the Polynomial LIBSVM classifier (Gender Independent)	16
Table 2.6: Confusion matrix of the Polynomial LIBSVM classifier (Male)	16
Table 2.7: Confusion matrix of Polynomial LIBSVM classifier (Female)	16
Table 2.8: Results from using MLR classifier based on Berlin and Spanish databases	17
Table 2.9: Results from using the SVM classifier based on databases from Berlin and Spain	17
Table 2.10: Recognition results using RNN classifier based on Berlin and Spanish databases	18
Table 2.11: Confusion matrix when applying MFCC and MS features, relying on the Spanish database	18
Table 2.12: Word error rate descriptive statistics by transcription technique	20
Table 2.13: One excerpt, transcribed three ways	22
Table 2.14: Effects of transcript due to external interference	23

Table 2.15: Preliminary results with SCLITE scoring	23
Table 2.16: TV shows included in the RTVE2020 dataset	24
Table 2.17: Total Word Error Rate (WER) of ASR systems on each RTVE2020 test set TV programme	25
Table 2.18: An overview of the data-driven activities and methods used to different context-aware mobile services and systems	30
Table 4.1: Use Case Description of Transcribe Speech	57
Table 4.2: Use Case Description of Transcribe Speech by recording real-time	58
Table 4.3: Use Case Description of Transcribe Speech by uploading pre-recorded speech	59
Table 4.4: Use Case Description of View Transcript Text with Emoji	60
Table 4.5: Use Case Description of Play Recording	61
Table 4.6: Use Case Description of Pause Recording	62
Table 7.1: Comparison of Speech to text model performances	84
Table 7.2: Confusion matrix of the emotion recognised	97
Table 7.3: Comparison on the different versions of the app	101
Table 7.4: Unit Test Case for record audio.	103
Table 7.5: Unit Test Case for upload pre-recorded speech.	105
Table 7.6: Unit Test Case for play and pause recording.	106
Table 7.7: Unit Test Case for transcript text	107

Table 7.8: Unit Test Case for Emoji recognition	108
Table 7.9: List of Integration Test Cases	113
Table 7.10: List of lines from different movies	115
Table 7.11: List of Task Scenario with Test Cases	119
Table 7.12: List of Test Speeches used for user acceptance test.	121
Table 7.13: System Usability Scale Scores	123

LIST OF FIGURES

Figure 1.1: Survey on digital communications	2
Figure 1.2: Survey on students finding difficulties in online learning	3
Figure 1.3: Speech to Text with emojis solution system overview	5
Figure 1.4: Prototyping Development Methodology	6
Figure 2.1: A high-level overview of the Speech Emotion Recognition System	9
Figure 2.2: MFCC extraction schema	10
Figure 2.3: Process for computing the ST representation	11
Figure 2.4: A fundamental RNN notion and the unfolding in time of the computation involved in its forward computation	13
Figure 2.5: LRC algorithm	13
Figure 2.6: Cross-level interaction plot.	21
Figure 2.7: Software development lifecycle in our DL-based software	26
Figure 2.8: DL-based Software System Operation Architecture	27
Figure 2.9: The CIAUI Framework	28
Figure 2.10: Users' interest trends through time, where the x-axis shows timestamp data and the y-axis indicates a popularity score ranging from 0 (min) to 100 (max)	29
Figure 3.1: Basic workflow of the AI learning model	38
Figure 3.2: Automatic Speech Recognition (ASR) flow diagram	44

Figure 3.3: General concept diagram of LSTM model	46
Figure 3.4: MLP Classifier Structure Diagram	47
Figure 3.5: CNN-LSTM Structure Diagram	48
Figure 3.6: Project 1 Gant Chart	53
Figure 3.7: Project II Gant Chart	54
Figure 4.1: Use Case Diagram	56
Figure 5.1: Single Tier Architecture design flow on Speech to Text with Emoji mobile app	64
Figure 5.2: Speech to Text with Emoji app UI design before transcribing speech.	65
Figure 5.3: Speech to Text with Emoji app UI design after transcribing speech.	66
Figure 6.1: Google's real-time speech recognition, powered by Android's built-in SpeechRecognizer API	69
Figure 6.2: Wav2Vec implementation code snippet 1	70
Figure 6.3: Wav2Vec implementation code snippet 2	70
Figure 6.4: Wav2Vec implementation code snippet 3	70
Figure 6.5: Wav2Vec implementation code snippet 4	70
Figure 6.6: AssemblyAI code snippets 1	71
Figure 6.7: AssemblyAI code snippets 2	71
Figure 6.8: AssemblyAI code snippets 3	72

Figure 6.9: AssemblyAI code snippets 4	72
Figure 6.10: AssemblyAI code snippets 5	72
Figure 6.11: AssemblyAI code snippets 6	73
Figure 6.12: DeepSpeech Algorithm	74
Figure 6.13: List of emojis with its ASCII code	76
Figure 6.14: Emoji Selection implementation in the program	77
Figure 6.15: Side to side view of the XML codes and the UI design	77
Figure 6.16: Interaction listener code	78
Figure 6.17: Multithreading architecture	79
Figure 6.18: UI display when the backend modules are working	79
Figure 6.19: AsyncTask class implementation	80
Figure 6.20: doInBackground function	80
Figure 6.21: onPostExecute function	80
Figure 6.22: Permission section for the mobile application	81
Figure 6.23: Asking permission UI design	81
Figure 7.1: Comparison of Speech to text model performances graph	85
Figure 7.2: Fear speech data waveform	85
Figure 7.3: Fear speech data spectrogram	86
Figure 7.4: Angry speech data waveform	86

Figure 7.5: Angry speech data spectrogram	86
Figure 7.6: Disgust speech data waveform	87
Figure 7.7: Disgust speech data spectrogram	87
Figure 7.8: Neutral speech data waveform	87
Figure 7.9: Neutral speech data spectrogram	88
Figure 7.10: Sad speech data waveform	88
Figure 7.11: Sad speech data spectrogram	88
Figure 7.12: Pleasant surprise speech data waveform	89
Figure 7.13: Pleasant surprise speech data spectrogram	89
Figure 7.14: Happy speech data waveform	89
Figure 7.15: Happy speech data spectrogram	90
Figure 7.16: Summary of LSTM Model	91
Figure 7.17: Number of speech dataset for each emotion	92
Figure 7.19: Training results of LSTM Model part 1	92
Figure 7.20: Training results of LSTM Model part 2	93
Figure 7.21: Relationship between train and validation accuracy of LSTM model	93
Figure 7.22: Relationship between the training loss and validation loss of LSTM model	93
Figure 7.23: MLP Classifier architecture code implementation	94

Figure 7.24: Last 5 training/validation results epoch of MLP model	95
Figure 7.25: Relationship between train and validation accuracy of MLP model	96
Figure 7.26: Relationship between the training loss and validation loss of MLP model	96
Figure 7.27: CNN-LSTM model architecture summary	98
Figure 7.28: Last 5 training/validation results epoch of CNN-LSTM model	99
Figure 7.29: Relationship between train and validation accuracy of CNN-LSTM model	99
Figure 7.30: Relationship between the training loss and validation loss of CNN-LSTM model	100

LIST OF APPENDICES

APPENDIX A: User Satisfaction Survey Sample	130
APPENDIX B: User Acceptance Test Sample	133
APPENDIX C: User Acceptance Test and System Usability Test Results	136

CHAPTER 1

INTRODUCTION

1.1 Introduction

All human relationships are built on based communication. Communication allows individuals to share their ideas and feelings while also allowing us to comprehend the emotions and thoughts of others. We establish love or contempt for other individuals via communication which forms good or bad connections.

There are many ways of communicating but talking is the primary natural method we always use. Over the years, speech has evolved into various forms as we start to incorporate different types of slang and phrases from the internet. Thankfully with speech recognition technology, a speech transcript, also known as a subtitle or caption, has been created to help users understand what the speaker is talking about. Furthermore, text transcript can improve accessibility; in scenarios where the user cannot listen to the audio and does not understand the language while watching the video, they can always read the subtitles to understand the contents of the video. This implementation can be widely seen in the entertainment field, where video clips are on major video-sharing platforms such as YouTube and movies. Speech transcripts have also been a significant part of academic research when students find it hard to keep up with the lecturer's talking speed and slang.

However, reading plain text alone from the transcription cannot convey the message's emotions compared to listening to it. Humans are wired to experience many basic emotions, including rage, fear, pleasure, sorrow, enthusiasm, and disgust. These fundamental emotions assist us in understanding, connecting, and communicating with others. They also assist us in connecting with ourselves. (Mamorsky et al., 2019). Thus, emoticons are created to deliver the user's message entirely.

With our current technology, we can add emoticons to our text, either choosing it manually or by saying the emoticons' names using speech recognition technology. However, this might be time-consuming and inconvenient when searching for the right emoji. In this project, a solution is proposed to integrate speech transcription technology and speech emotion recognition to develop a speech to text with emojis solution. By adding emojis to the speech transcript, users who have trouble listening can further understand the context of the message and solve other relevant problems.

1.2 Problem Statement

Several problems that can be solved by implementing emotional speech transcription have been researched and found. The main problems are the efficiency of digital communication, the equality between ordinary and disabled people, and poor concentration in an online class.

1.2.1 Efficiency of digital communications

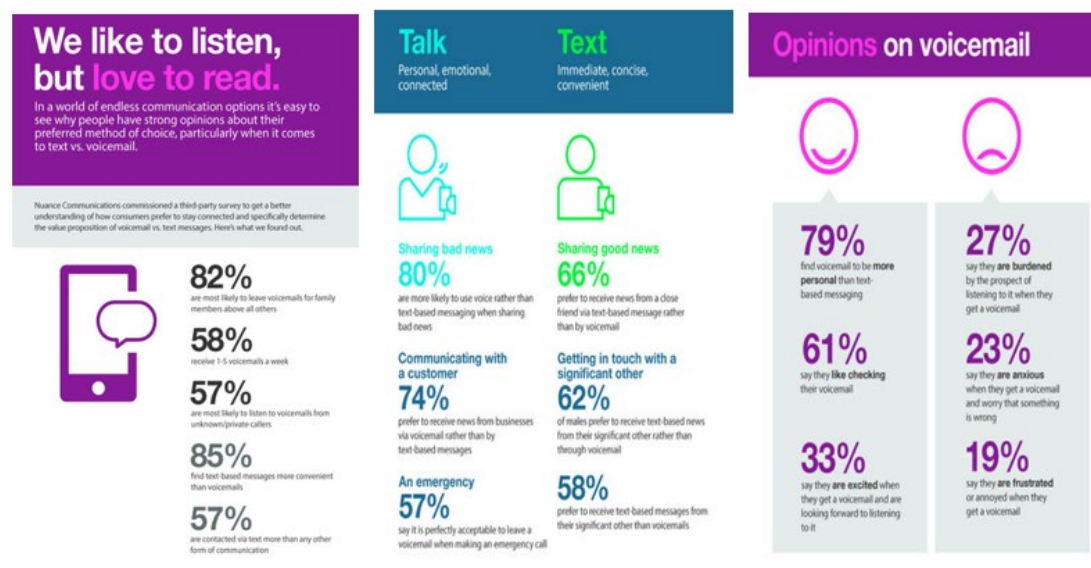


Figure 1.1: Survey on digital communications (www.businesswire.com, 2014)

According to Figure 1.1, a survey conducted by Nuance Communications, Inc. and Research Now, the poll found that customers want to guarantee that their text and voice communications are received quickly every time. Given a chance, 62% of consumers would switch to voice-to-text messaging to keep connected on the road. The poll looked at message-sending and receiving preferences. 80% of individuals were more inclined to utilise voicemail when delivering critical news, such as a funeral or illness. However, these important, personal messages sometimes went undetected for hours. 79% of respondents said getting voicemails was more personal than texting, yet texting remained the dominating medium due to its speed and ease. 66% of respondents prefer to get news from a close friend by text rather than voicemail, and 85% find text messages handier than voicemails (www.businesswire.com, 2014).

1.2.2 Equality between normal people and disabled people

The first major problem of having only speech transcription is that the whole meaning of the message cannot be delivered to the audience. This would not be a massive problem for any average human being as we can hear sounds to interpret the emotions behind the transcription. However, people who have disabilities which are related to hearing, also known as deafness, will have trouble interpreting the transcription. According to the World Health Organization, over 5% of the world's population, or 430 million individuals, need hearing therapy (432 million adults and 34 million children). By 2050, it is estimated that one in ten individuals will have debilitating hearing loss. Hearing loss higher than 35 dB in the better hearing ear is considered a disability. Interestingly, 80% of deaf people reside in low- and middle-income nations. Over 25% of 60-year-olds have debilitating hearing loss (World Health Organization, 2021). Therefore, we should think about equality and find a solution to enhance text messaging experience for disabled people.

1.2.3 Poor concentration in online class

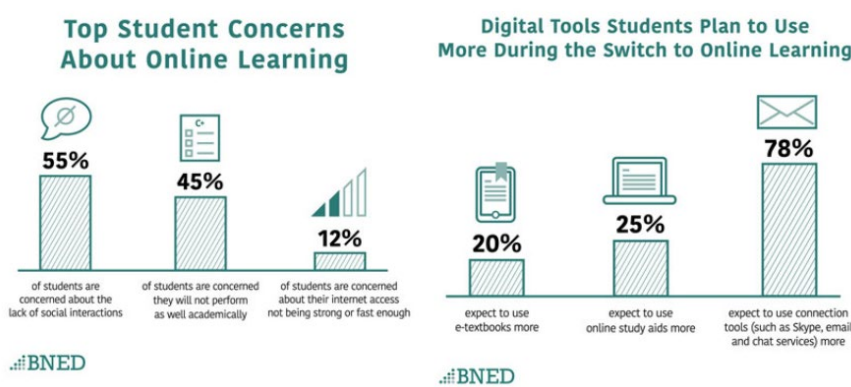


Figure 1.2: Survey on students finding difficulties in online learning (News, 2020)

Online or remote learning is the only option to keep the educational system operating during the coronavirus pandemic. This has created a range of difficulties for pupils. During the week of March 23rd, 2020, Barnes & Noble Education polled 432 college students nationwide. According to Figure 1.2, more than half (64%) of students expressed anxiety about keeping attention and discipline. They also voiced their doubt on their ability to sustain the motivation to complete their assignments remotely. However, according to the poll, 60% of students claim to be technically ready to

transfer to online courses, while the remaining 40% are less specific and claim they need more time to get used to the change. Since they believe they learn best when interacting with their peers, more than half of students (55%) expressed worry about the absence of social contacts, and 45% expressed anxiety that they would not do as well academically in such a situation. Fewer pupils (12%), concerned that their internet connection is weak or slow, are worried about technology. Emoji solutions for speech-to-text in the classroom might amuse the kids (News, 2020). Speaking is often significantly quicker than typing. To avoid disrupting the lesson, students may voice their queries while the technology converts them to text. Emojis are used in every phrase to give the class a little fun value.

1.3 Project Objectives

The primary objective of this project is to create an artificial intelligent mobile application which can interpret the speech of the user and convert it to text with emoji. The objectives of the project are:

- To investigate the feasibility of the voice recognition system for emotion detection
- To develop an emotional voice recognition system which can measure the highest accuracy of the number of decibels, pitch/octave, speech rate, timbre, speech pattern and common phrases from a user's speech to display the correct emojis on screen.
- To develop a speech to text with emoji solutions which the transcript text with emojis rate is as similar as the speech rate and the emotions portrait by the user.

1.4 Project Solution

The proposed solution is to develop a speech to text with emojis solution which is done by integrating two main components; the speech to text module and speech recognition to detect emotion module. It aims to solve the encountered problem stated above. After asking permission to allow access to the device's microphone, the microphone will first capture the user's speech. The speech will then be processed by the software and display out the speech of the user and the emotion of the speech at the end of the sentence. The methodology and implementation of it for the project solution shall be shown in Chapter 3 and 4 respectively. A simple overview on how the Speech to Text with Emojis System works is presented in Figure 1.3.

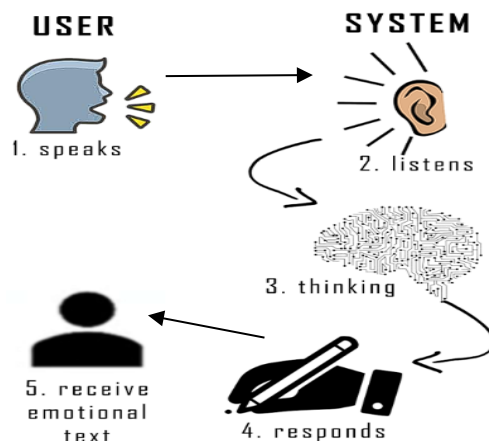


Figure 1.3: Speech to Text with emojis solution system overview

1.5 Project Approach

The development process utilised for this project was prototyping. This is a suitable methodology for my "Speech to text with emojis" project because my primary test input is the user's voice, and having different user tests exposes the app to different types of voice and emotions, which may lead to the further tuning of the model to increase accuracy. The time required to process may vary depending on the user text and the chat length. Obtaining these data early may accelerate the development process and enhance the application's overall performance. Hence, the complete app can be developed within a short amount of time.

Prototyping's iterative nature enables us to swiftly test and tweak the design to ensure that it fulfils the demands of our consumers. This may result in speedier development and a more agile development process, which is especially useful when there is significant ambiguity regarding the requirements and design.

The Prototyping technique is divided into five primary steps, which include requirements gathering, design, implementation, assessment, and refining. During the requirements gathering phase, the developer will collect and record the software system or application's functional and non-functional needs. During the design phase, the developer will create the prototype containing the necessary features, functionalities, and user interface. The prototype was constructed and tested throughout the implementation phase. During the assessment phase, the team assessed the prototype against the requirements and design objectives and collected user input. During the refinement phase, the team improved the prototype's design and functionality based on feedback and assessment, and they repeated the assessment

process to confirm that the new design fit the criteria. Figure 1.4 shows the flow of how the prototyping development methodology works from start to finish.

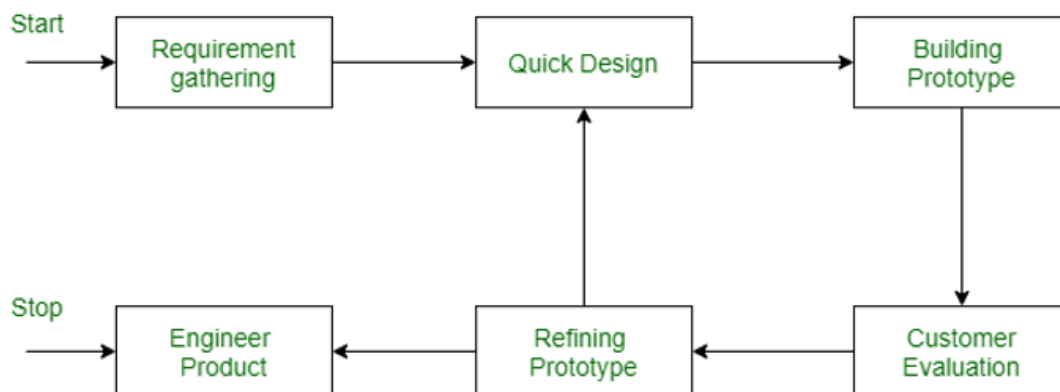


Figure 1.4: Prototyping Development Methodology

1.6 Project Scope

1.6.1 Target Users

The "Speech to text with emojis" mobile app is aimed towards a wide variety of users, including disabled, deaf persons, students, and the general public.

First, disabled deaf people face difficulties in interpreting voice emotions as they cannot hear properly. Traditional plain-text communications are sometimes insufficient to express a message's entire content and feelings. However, the "Speech to text with emojis" software may be a viable answer to this problem. Users can now comprehend the message's tone and emotions more efficiently and accurately by utilising emojis that complement the text.

Next, the software may make connecting with lecturers and classmates simpler and more enjoyable for students, particularly those taking online programmes. Students may better comprehend the topic and interact with others using speech-to-text technologies and emoticons. This may assist in improving the effectiveness and engagement of online learning.

On top of that, the app makes communication more effective for individuals in general. Listening to extended audio messages may be time-consuming, yet reading simple text messages may need more emotions and tone intended by the sender. Users may save time and ensure that the feelings and style of the news are not lost by utilising the "Speech to text with emojis" software to swiftly read transcribed messages with emojis that match the content.

1.6.2 Modules

1.6.2.1 Speech Recognition module

This module would be in charge of turning spoken words into text. Chapter 3 Methodology and Work Plan.

1.6.2.2 Emotion Recognition module

This module would be responsible for analysing and recognizing the speech's emotion recorded or uploaded by the user.

1.6.2.3 Emoji Selection module

This module will select the most appropriate emoji after given the emotion from the emotion recognition module to convey the meaning and emotions behind the text.

1.6.2.4 Text with Emoji integration module

With two heavy modules executing concurrently, this module is responsible for ensuring that each module integrates well and not stress the mobile system by separating them as threads, getting their results as strings respectively and displaying them to user.

1.7 Limitation

One of the limitations of the project is the insufficient amount of collected speech datasets for the deep learning training purpose. In a real-world scenario, there are infinite possibilities on how an emotional speech is conveyed. Therefore, a higher quality of datasets is better than a higher quantity of datasets. This will eliminate the possibility of exhaustive testing. Next, to obtain a high-quality transcript from speech with emoji recognition software, the recorded audio must be clear and understandable. This implies no background noise, proper pronunciation, no accents, and only one person speaking at a time. Moreover, the hardware specifications of my machine are also a limitation in the project. A machine with a better CPU, RAM and GPU can process and process the codes faster, hence more time is needed to train the model. Lastly, the most challenging part of the emotional voice recognition system is the human nature of sarcasm. This may confuse or trick the AI in the real intention of the user's speech. Hence, there will be some lack of accuracy in the emotion recognition model.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this section, several articles are referenced to further analyse the chosen field of studies. An overview, a summary, and an assessment of the present level of knowledge of three field of studies related to Speech to Text with Emojis will be presented. This chapter will be separated into three different sections (2.2, 2.3, 2.4) to present each field of study:

Field of studies:

2.2 Voice Recognition system for emotion detection.

2.3 Evaluation on current speech transcription methods

2.4 Application of artificial intelligence techniques on mobile applications

2.2 Voice Recognition system for emotion detection

There are many techniques to identify emotions, but the two basic ones are listening to one's speech and observing one's behaviours. In this research, we will concentrate on speech emotion recognition. Regardless of the semantic content of speech, the challenge of speech emotion recognition (SER) is to recognise emotional features in speech. Although humans can complete this task as a part of verbal communication, it is still not yet possible to automate it using programmable devices. Research into automatic emotion recognition systems intends to provide practical, real-time methods of recognising the emotions of a wide variety of human-machine interface users, including mobile phone users, contact centre employees and customers, drivers, pilots, and other professionals. It has been determined that the key to making robots seem and act like people is to give them emotions (André et al., 2004). As a result, emotionally intelligent robots may behave appropriately and display exciting personalities. In certain circumstances, computer-generated characters that can speak with a high degree of authenticity and persuasion by appealing to human emotions may take the place of actual people. However, robots must be able to comprehend spoken emotions. Only with this ability is it feasible to establish a genuine conversation based on mutual understanding and trust between humans and machines.

The goal of speech emotion detection is to automatically detect a person's emotional state from his or her voice. It is based on an in-depth investigation of the speech signal generating process, extracting specific characteristics containing emotional information from the speaker's voice and using suitable pattern recognition algorithms to determine emotional states. Like other pattern recognition systems, the general voice emotion identification system has four key modules: speech input, feature extraction, SVM-based classification, and emotion output (Joshi and Kaur, 2013).

The typical design for a SER system is represented in Figure 2.1 as three steps:

- i. A speech processing system extracts certain acceptable quantities from the signal, such as pitch or energy.
- ii. These quantities are summarised into a smaller set of features.
- iii. A classifier learns how to correlate the features to the emotions in a supervised way using example data.

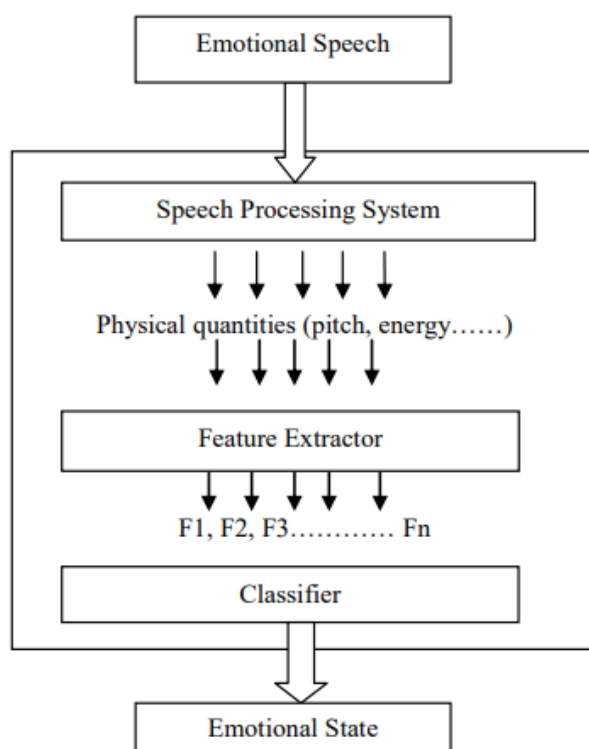


Figure 2.1: A high-level overview of the Speech Emotion Recognition System (Joshi and Kaur, 2013).

2.2.1 Voice Feature Extraction

The speech signal contains many elements that are emotional characteristics. Selecting the right features to use is one of the most challenging aspects of emotion detection. The extraction of voice characteristics from speech signals is a crucial stage in the SER system for choosing appropriate features that convey emotional information. Energy, formant, pitch and specific spectrum features like Mel-Frequency Cepstrum Coefficients (MFCC), Modulation spectral components and Linear Prediction Coefficients (LPC) recovered in recent research are only a few of the well-known characteristics. In a 2018 research, Modulation spectral features and MFCC were used by Kerkeni et al. to identify emotional traits.

2.2.1.1 MFCC Features

Mel-Frequency the most popular way to characterise the spectral characteristics of a voice signal is via a cepstrum coefficient. These are the best since they consider how sensitive to frequencies human perception is. The energy spectrum and Fourier transform were calculated for each frame and then projected onto the Mel-frequency scale. The top 12 DCT coefficients of the Mel log energies' discrete cosine transform (DCT) the MFCC values that were applied throughout the classification procedure. The usual MFCC calculation technique is shown in Figure 2.2. (Kerkeni et al., 2018).



Figure 2.2: MFCC extraction schema (Srinivasan et al., 2014).

From 16 kHz speech samples, this research recovered the first 12 order MFCC coefficients. Then, for each order coefficient, which holds for all frames of an utterance, they determined the mean, standard deviation, Kurtosis, and Skewness. The dimensions of each MFCC feature vector are 60. (Kerkeni et al., 2018).

2.2.1.2 Modulation Spectral Features

A long-term spectro-temporal representation influenced by auditory perception is used to derive spectral modulation components (MSFs). These characteristics are generated by simulating the human auditory system's spectro-temporal (ST) processing, which takes regular acoustic frequency and modulation frequency into account. Figure 2.3 displays the steps for calculating the ST representation. First, an auditory filter bank splits the spoken stream into segments to get the ST representation. The modulation signals are then produced by computing the Hilbert envelopes of the critical-band outputs. The modulation filter bank is then used to analyse frequency using the Hilbert envelopes. The suggested characteristics are referred to as spectral modulation features (MSFs), and the modulation spectra are the spectrum contents of the modulation signals (Wu, Falk, and Chan, 2011). Finally, the energy of the decomposed envelope signals is estimated as a function of both the modulation frequency and the regular acoustic frequency to get the ST representation. The energy provides each spectral band's feature meanings over all frames. The ST format is used in this study to estimate 95 MSFs (Kerkeni et al., 2018).

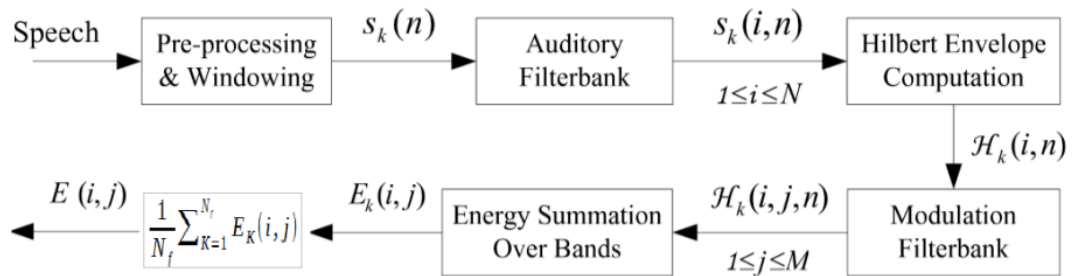


Figure 2.3: Process for computing the ST representation (Wu, Falk, and Chan, 2011).

2.2.1.3 Prosodic Features

Contrarily, Sudhkar and Anil used prosodic characteristics to extract the speech dataset's features. There are two categories: long-term features and short-term characteristics. Short-term features include traits that last for a short time, such as formants, pitch, and vigour. The statistical approach to digitised speech signals is one of the long-term properties. Two often used long-term qualities are the mean and standard deviation. The categorisation procedure is better the more significant the characteristic that is used. After extracting speech traits, only those with crucial emotional information are picked. Following that, these properties are transformed into

n-dimensional feature vectors. Pitch, intensity, speaking pace, and variance are critical prosodic traits for recognising different emotions in speech. The acoustic characteristics of various speech moods are shown in Table 2.1. The data provided in Table 2.1 below was gathered using the Paart programme (Sudhkar and Anil, 2015).

Table 2.1: Acoustic Characteristics of Emotions (Sudhkar and Anil, 2015).

Characteristics	Happy	Anger	Enquiry	Fear	Surprise
Emotion					
Pitch Mean	High	Very high	High	Very high	Very high
Pitch Range	High	High	High	High	High
Pitch Variance	High	Very high	High	Very high	Very high
Pitch Contour	Incline	Decline	Moderate	Incline	Incline
Speaking Rate	High	High	Medium	High	High

2.2.2 Approaches on Speech Emotion Recognition (SER)

In the past, machine learning (ML) included obtaining feature parameters from unprocessed data (e.g., speech, images, video, ECG, EEG). With the help of the features, a model is trained to provide the required output labels. A frequent difficulty with this approach is choosing the qualities. There is a widespread lack of knowledge on the traits that can result in the most efficient data sorting into several groupings (or classes). Some insights may come from testing a wide range of features, combining several features into a single feature vector, or using different feature selection techniques. Additionally, the effectiveness of categorisation may be significantly impacted by the quality of the hand-crafted features provided.

The problem of optimal feature selection was easily solved with the advent of deep neural networks (DNN) classifiers. The intended method is to utilise an end-to-end network to accept raw data as input and output a class label. There is no need to choose the appropriate parameters for categorisation or to compute hand-crafted characteristics. Instead, everything is handled by the network. In particular, the network parameters (i.e., weights and bias values provided to network nodes) are set to act as features during the training phase, effectively categorising the input. This method has far higher requirements for labelled data samples than conventional classification algorithms. As a result, we will compare several systems for categorising speech emotion.

2.2.2.1 Recurrent Neural Networks

Speech data categorized as time series data may be learned using Recurrent Neural Networks (RNN). While RNN models are effective at understanding temporal correlations, they struggle with the vanishing gradient problem, which worsens as the duration of the training sequence increases. To address this issue, LSTM (Long Short-Term Memory) RNNs is created so that memory cells could be employed to retain information and may leverage long-range relationships in the data (Kerkeni et al., 2018).

Figure 2.4 depicts a simple RNN implementation approach.

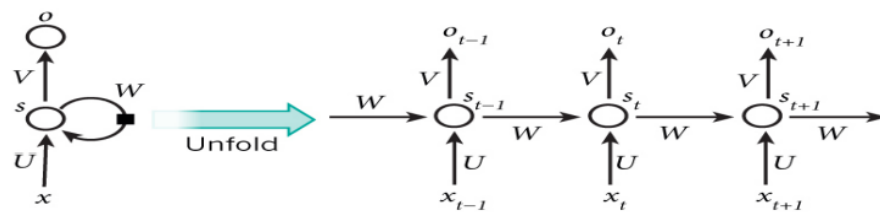


Figure 2.4: A fundamental RNN notion and the unfolding in time of the computation involved in its forward computation (Lim, Jang, and Lee, 2016).

2.2.2.2 Multivariate Linear Regression Classification

Multivariate Linear Regression (MLR) is a simple and efficient machine learning method calculation that can be utilised for both regression and classification tasks. The LRC algorithm mentioned in algorithm 1 has been somewhat changed (Naseem, Togneri and Bennamoun, 2010). In step 3, Kerkeni computed the absolute value of the difference between the original and predicted response vectors ($|y - y_i|$), rather than the euclidean distance between them ($\|y - y_i\|$):

Algorithm 1 : Linear Regression Classification (LRC)

Inputs: Class models $X_i \in \mathbb{R}^{q \times p_i}$, $i = 1, 2, \dots, N$ and a test speech vector $y \in \mathbb{R}^{q \times 1}$

Output: Class of y

1. $\hat{\beta}_i \in \mathbb{R}^{p_i \times 1}$ is evaluated against each class model, $\hat{\beta}_i = (X_i^T X_i)^{(-1)} X_i^T y$, $i = 1, 2, \dots, N$
 2. \hat{y}_i is computed for each $\hat{\beta}_i$, $\hat{y}_i = X_i \hat{\beta}_i$, $i = 1, 2, \dots, N$;
 3. Distance calculation between original and predicted response variables
 $d_i(y) = |y - y_i|$, $i = 1, 2, \dots, N$;
 4. Decision is made in favor of the class with the minimum distance $d_i(y)$
-

Figure 2.5: LRC algorithm (Naseem, Togneri and Bennamoun, 2010).

2.2.2.3 Support Vector Machine

In machine learning, a Support Vector Machine (SVM) is a binary classifier in general, but it may also be employed as a multiclass classifier. LIBSVM is a famous SVM classification and regression tool created by C. J. Lin. The Radial Basis Function (RBF) kernel is utilised in the training phase. The benefit of utilising an RBF kernel is that it limits training data to inside set bounds. Unlike the linear kernel, the RBF kernel can handle scenarios where the connection between class labels and attributes is nonlinear since it nonlinearly translates samples into a higher dimensional space. The RBF kernel has fewer numerical challenges than the polynomial kernel (Chavhan, Dhore and Yesaware, 2010).

2.2.3 Results comparison among methods

Chavhan, Dhore and Yesaware experimented with detecting speech emotion using a support vector machine. They use the speech data from the Berlin Emotion database, which includes 406 voice files representing five emotion classes. Anger, sadness, happiness, neutrality, and fear have been spoken utterances in 127, 62, 71, 79, and 67. The LIBSVM is trained using RBF and Polynomial kernel functions on MFCC and MEDC feature vectors. These feature vectors are tested using the LIBSVM. Experiment by adjusting the RBF kernel's cost values and the Polynomial kernel's degree values. Experiments are carried out that are both gender independent and gender dependent. Using the RBF kernel at cost value $c=4$, the recognition rate for gender-independent cases is 93.75%, 94.73% for male speakers, and 100% for female speeches. The gender-independent recognition rate for Polynomial kernel at degree $d=4$ is 96.25% for male and 100% for female speakers. Tables 2.2, 2.3, and 2.4 illustrate the confusion matrices employing RBF kernels that are gender independent, male, and female. Tables 2.5, 2.6, and 2.7 illustrate confusion matrices with Polynomial kernels that are gender independent, male, and female.

Table 2.2: Confusion matrix of the RBF LIBSVM classifier (Gender Independent)
(Chavhan, Dhore and Yesaware, 2010).

Emotion	Emotion Recognition (%)				
	Angry	Sad	Happy	Neutral	Fear
Angry	100	0	0	0	0
Sad	0	100	0	0	0
Happy	0	0	100	0	0
Neutral	0	6.25	0	93.75	0
Fear	0	0	30.76	0	69.24

Table 2.3: Confusion matrix of the the RBF LIBSVM classifier (Male) (Chavhan,
Dhore and Yesaware, 2010).

Emotion	Emotion Recognition (%)				
	Angry	Sad	Happy	Neutral	Fear
Angry	100	0	0	0	0
Sad	0	100	0	0	0
Happy	16.66	0	83.34	0	0
Neutral	0	0	0	100	0
Fear	0	0	0	14.85	85.15

Table 2.4: Confusion matrix of the RBF LIBSVM classifier (Female) (Chavhan,
Dhore and Yesaware, 2010).

Emotion	Emotion Recognition (%)				
	Angry	Sad	Happy	Neutral	Fear
Angry	100	0	0	0	0
Sad	0	100	0	0	0
Happy	0	0	100	0	0
Neutral	0	0	0	100	0
Fear	0	0	0	0	100

Table 2.5: Confusion matrix of the Polynomial LIBSVM classifier (Gender Independent) (Chavhan, Dhore and Yesaware, 2010).

Emotion	Emotion Recognition (%)				
	Angry	Sad	Happy	Neutral	Fear
Angry	100	0	0	0	0
Sad	0	100	0	0	0
Happy	0	0	100	0	0
Neutral	0	0	0	100	0
Fear	7.69	0	15.18	0	76.92

Table 2.6: Confusion matrix of the Polynomial LIBSVM classifier (Male) (Chavhan, Dhore and Yesaware, 2010).

Emotion	Emotion Recognition (%)				
	Angry	Sad	Happy	Neutral	Fear
Angry	100	0	0	0	0
Sad	0	100	0	0	0
Happy	0	0	100	0	0
Neutral	0	0	0	100	0
Fear	0	0	14.28	0	85.72

Table 2.7: Confusion matrix of Polynomial LIBSVM classifier (Female) (Chavhan, Dhore and Yesaware, 2010).

Emotion	Emotion Recognition (%)				
	Angry	Sad	Happy	Neutral	Fear
Angry	100	0	0	0	0
Sad	0	100	0	0	0
Happy	0	0	100	0	0
Neutral	0	0	0	100	0
Fear	0	0	0	0	100

In conclusion, the work was done by Chavhan, Dhore and Yesaware. In the case of LIBSVM employing RBF and Polynomial kernels, it has been discovered that altering the parameters of a kernel function might provide superior results.

In contrast, a similar experiment conducted by Kerkeni employed a support vector machine and other methods such as Multivariate Linear Regression Classification and Recurrent Neural Networks.

They used the Berlin database terms of databases. There is 535 utterances total, delivered by ten actors (five women and five men), in 7 simulated moods such as boredom, anger, fear, disgust, sadness, joy, and neutral. Additionally, they took advantage of the INTER1SP Spanish emotional database, which included quotes from two trained actors (one female and one male speaker). The six basic emotions plus neutral were recorded twice in the Spanish corpus (Spa), and they have access to anger, grief, joy, fear, disgust, surprise, and neutral/normal. Once, there were four neutral versions recorded (soft, loud, slow, and quick). To get a better and more accurate recognition rate and to compare with the Berlin database mentioned above, this research narrowed its focus to just seven fundamental emotions from the Spanish Dataset (Kerkeni et al., 2018). Below are the experiment's findings.

Table 2.8: Results from using MLR classifier based on Berlin and Spanish databases (Kerkeni et al., 2018)

database	Features		A	E	F	L	N	T	W	Rate (%)
Berlin	MS	avg	41,79	29,86	42,92	75,40	54,84	85,64	78,10	60,70
		σ	10,97	9,86	9,07	10,85	6,63	13,37	8,40	2,50
	MFCC	avg	54,48	61,77	46,56	52,05	64,61	80,54	92,67	67,10
		σ	19,22	16,82	9,07	10,69	8,47	14,72	7,17	3,96
	MFCC+MS	avg	83,63	67,18	56,05	79,43	75,20	87,59	78,92	75,90
		σ	9,40	26,43	15,63	14,65	7,55	11,39	7,50	3,63
			A	D	F	J	N	S	T	Rate (%)
Spanish	MS	avg	61,61	53,08	72,42	54,20	90,97	61,59	68,16	70,60
		σ	3,70	4,03	4,29	4,67	2,14	3,90	4,62	1,37
	MFCC	avg	70,33	52,59	79,18	48,16	96,47	78,00	73,70	76,08
		σ	5,22	6,27	2,45	4,51	0,78	4,24	3,53	1,44
	MFCC+MS	avg	77,46	76,31	83,39	66,56	97,14	80,96	84,99	82,41
		σ	3,26	2,93	2,47	3,68	1,19	4,81	4,95	4,14

Table 2.9: Results from using the SVM classifier based on databases from Berlin and Spain (Kerkeni et al., 2018).

database	Features		A	E	F	L	N	T	W	Rate (%)
Berlin	MS	avg	60,35	57,54	49,75	66,54	62,93	80,02	67,01	63,30
		σ	12,55	22,72	18,14	13,90	12,70	9,36	8,40	4,99
	MFCC	avg	62,76	51,37	44,72	39,25	49,40	66,26	72,20	56,60
		σ	16,78	9,03	10,15	14,58	15,12	15,59	7,97	4,88
	MFCC+MS	avg	55,04	49,82	44,61	71,60	55,68	70,11	65,42	59,50
		σ	12,81	22,16	14,56	15,58	16,30	12,57	10,01	5,76
			A	D	F	J	N	S	T	Rate (%)
Spanish	MS	avg	71,99	68,72	79,54	65,59	86,93	69,76	79,76	77,63
		σ	6,45	4,21	3,15	5,86	3,50	3,60	3,78	1,67
	MFCC	avg	81,54	80,67	80,18	68,92	68,69	67,12	86,65	70,69
		σ	5,56	4,92	8,61	18,57	22,18	29,23	4,07	12,66
	MFCC+MS	avg	76,41	85,39	69,76	76,03	53,31	64,40	84,59	68,11
		σ	6,65	3,80	3,10	2,50	23,70	2,25	3,27	11,55

Table 2.10: Recognition results using RNN classifier based on Berlin and Spanish databases (Kerkeni et al., 2018).

Dataset	Feature	Average (avg)	Standard deviation (σ)
Berlin	MS	66.32	5.93
	MFCC	69.55	3.91
	MFCC+MS	58.51	3.14
Spanish	MS	82.30	2.88
	MFCC	86.56	2.80
	MFCC+MS	90.05	1.64

Table 2.11: Confusion matrix when applying MFCC and MS features, relying on the Spanish database (Kerkeni et al., 2018).

Emotion	Anger	Disgust	Fear	Joy	Neutral	Surprise	Sadness	Rate (%)
Anger	131	14	3	23	8	2	0	72,38
Disgust	3	197	1	6	6	6	2	89,95
Fear	3	15	115	6	12	0	0	76,16
Joy	8	4	1	411	0	11	0	89,14
Neutral	9	14	9	4	144	1	1	79,12
surprise	1	4	0	18	0	133	0	85,26
Sadness	8	1	18	11	17	0	93	62,84
Precision (%)	80,37	79,12	78,23	85,80	77,00	86,92	96,87	

2.2.4 Conclusion

There are numerous unknowns about the best algorithm for categorising emotions. The rate of emotion identification varies based on the combination of emotional traits employed. The experts are currently arguing which characteristics impact emotion perception in speech. The best recognition rate in the article by Kerkeni et al. was 90.05%, attained by integrating the MFCC and MS features with the RNN model in the Spanish emotional database. Furthermore, greater precision may be attained by combining additional characteristics. Aside from that, the quest for solid feature representation and practical classification algorithms for automated speech emotion identification is part of the continuing study.

When identifying the emotions present in speech, techniques that rely on the Fourier transform, such as MFCC and MS, are the most commonly utilized methods. Their ubiquity and effectiveness do have a downside, however. Consequently, signal processing now has a minimal and constrained grasp of frequency. Frequencies are a

grouping of the several periodic signals' unique frequencies that make up a particular signal in the context of Fourier methods.

2.3 Evaluation on current speech transcription methodologies

Speech-to-text is a kind of voice recognition software that recognises and converts spoken words into text using computational linguistics. It is also known as computer voice recognition or speech recognition. Specific apps, tools, and devices can transcribe audio streams in real-time so that text may be shown and acted on.

From my research, the birth of speech transcription technology invention dates to the 18th century. First, studying the nature of sound waves, particularly the work of Leonhard Euler, who authored a dissertation *De Sono* (On Sound) in 1726, paved the way for speech synthesis. Following that, Wolfgang von Kempelen's speaking machine appeared in the late 18th century. Around the same period, Christian Gottlieb Kratzenstein began researching how sounds are produced in the human vocal tract and developed the "vowel organ," a network of resonators. All of them were mechanical devices that used reeds, bellows, and resonators, among other things.

Speech synthesis made significant advances in the twentieth century, following the standard twentieth-century progression from electric to electronic to digital computer models. Dennis H. Klatt (who created Stephen Hawking's voice) produced a notable historical overview essay, "Review of text to speech conversion for English", which details the significant advancements between the 1920s and the 1980s. Synthesis methods advanced dramatically throughout the 1950s and 1960s. Combining them with acceptable text analysis to drive voice synthesis began in the late 1960s (with excellent arguments for 1968), and numerous commercial implementations were available in the 1970s. The efforts of each scientist from different generations gave birth to the speech transcription technology we use today. From researching various articles related to speech transcription evaluation, we can understand what parameters affect the speech transcription and which methods may be the best. Now we shall evaluate the speech transcription modern research from various articles.

2.3.1 Comparing Google Cloud Speech with Real-time transcription by certified SLPs and TTs

In a paper by Fox et al., the accuracy of each transcription method was assessed against a reference corpus that served as the industry's gold standard (2021). Their study looked at the reliability and accuracy of two accelerated transcription techniques: automated speech recognition using Google Cloud Speech and real-time transcription by trained SLPs and TTs. The therapeutic value of each methodology was established by assessing the precision of scores derived from transcripts generated by each approach on a range of linguistic sample analysis (LSA) measures. There were seven qualified TTs and seven qualified SLPs present. From a total of 42 language samples, each participant was required to produce a set of six transcripts in real-time. Following that, Google Cloud Speech was used to transcribe the same 42 samples.

A weighted word error rate based on the clean (i.e., noncoded) reference corpus was used to assess the accuracy of each accelerated transcription technique (Google Cloud Speech and real-time). Analysis of the word error rate would either provide the average or the weighted word error rate since the word error rate is computed at the utterance level. It determines the total word mistake rate after considering each utterance's duration. Table 2.12, which focuses on the sample mean and standard deviation range, provides vital descriptive data on the weighted word error rate per transcribing procedure (Fox et al.,2021).

Table 2.12: Word error rate descriptive statistics by transcription technique (Fox et al ,2021).

Variable	<i>M</i>	<i>Mdn</i>	Min	Max	Range
ASR (<i>n</i> = 42)	0.30 (0.11)	0.30	0.08	0.51	0.43
S-RT (<i>n</i> = 42)	0.42 (0.19)	0.40	0.11	0.83	0.72
T-RT (<i>n</i> = 41)	0.43 (0.19)	0.45	0.10	0.74	0.64

Note. ASR = automatic speech recognition; S-RT = real-time transcription–speech-language pathologist; T-RT = real-time transcription–trained transcriber.

The final MLM model findings are shown in Figure 2.6. While sample size and narrator age were not significant predictors, transcription technique and speech rate had a significant cross-level interaction. This revealed that the narrator's rate affected the transcription quality. However, sample length and age did not seem to affect any transcription procedures' accuracy.

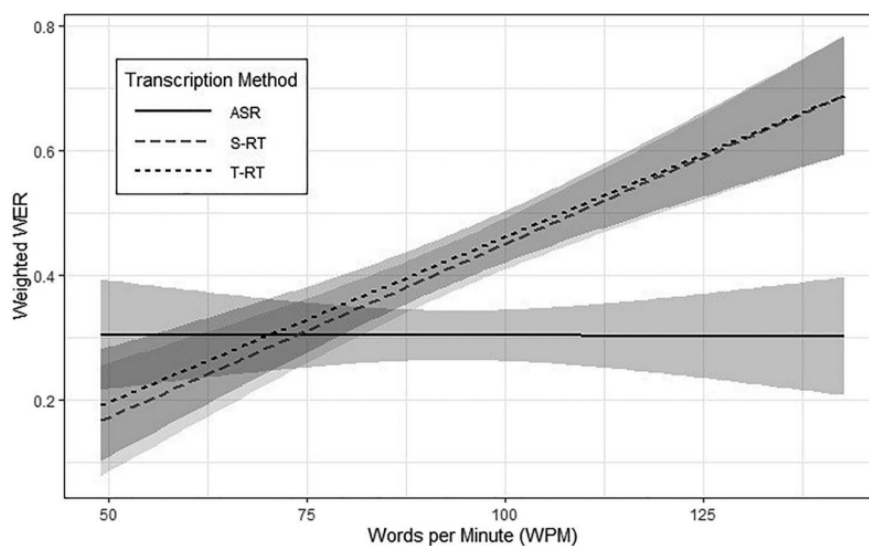


Figure 2.6: Cross-level interaction plot. (Fox et al ,2021).

Hence, Google Cloud Speech transcripts were much more accurate than any real-time transcription source (SLP or TT).

2.3.2 Human Evaluation on speech transcription technology

In contrast, Glen et al. concluded that technology assessments occasionally require gold-standard references, which are produced using a Careful Transcription Review (CTR) process that necessitates numerous quality control passes and, as a result, takes longer than a rapid transcription approach. The following components are required for an accurate transcript: a verbatim transcript; time alignment to the level of sentences or breath groups; consistent speaker identification; standard spelling and punctuation; annotation of events such as filled pauses, sounds, and proper nouns; dialect annotation, if applicable; and numerous human and automated quality control passes. The transcripts were painstakingly transcribed and scored using the SCLITE toolkit from NIST (Glenn et al., n.d.).

Based on Table 2.13 below, QTR (Quality Transcription and Review), QRTR (Quality Review and Transcription), and CTR (Careful Transcription Review) are all speech transcription techniques used to verify that the produced transcripts are accurate and comprehensive. Both QTR and QRTR entail a review phase, although they vary in the sequence in which transcription and review are performed. QTR entails transcribing the voice recording and assessing it for correctness and completeness. In contrast, QRTR entails utilising automatic speech recognition software to create an initial transcript and then improving it via human review. Conversely, CTR is a more conventional method that depends entirely on human labour to produce high-quality transcriptions. Transcribers use CTR to listen to and transcribe voice recordings while looking for faults and inconsistencies. Another individual then reviews the transcripts to verify correctness and completeness.

Table 2.13: One excerpt, transcribed three ways (Glenn et al., n.d.).

Method	Transcript text
QTR	well i don't know that i don't know that i'd score it as one better than the other i think every one of them to got a chance obama, edwards and senator clinton got a chance to provide a narrative of their own journey
QRTR	Well I don't know that uh I don't know that I would score it as one doing better than the other. I think that every one of them to got a chance Obama, Edwards and Senator Clinton got a chance to provide a narrative of their own faith journey.
CTR	Well I don't know that %uh – I don't know that I would score it as one doing better than the other. I think that every one of them to got a chance – %uh Obama, E- Edwards and Senator Clinton – got a chance to provide a narrative of their own faith journey.

Transcripts were created utilising exact time alignment wherever feasible. The file was manually subdivided in certain situations before being given to two separate, skilled transcribers for the first-pass transcript. In other circumstances, a finished transcript's temporal alignment was retrieved and assigned to a second transcriber. The SCLITE toolset from NIST was used to score all transcripts (Fiscus, 2006).

2.3.3 Parameters affecting Speech to Text methodologies

Using LDC's specialised transcription adjudication GUI, further extensive analysis was undertaken on most English-language fast transcripts. Annotators listened to each discrepancy and categorised it as a "transcriber mistake," "insignificant difference," or "judgement call," much like in the EARS research. Annotators may describe any acoustic circumstances or speaker characteristics that contributed to the dispute in detail. For example, in Table 2.14, a single speech had three disagreements: two judgement calls and one transcriber mistake (marked in bold in the example). Because of background noise and overlapping speech during adjudication, this transcript pair varies (Glenn et al., n.d.). Table 2.15 shows the statistics of transcription accuracy during different scenarios or situations with SCLITE scoring.

Table 2.14: Effects of transcript due to external interference (Glenn et al., n.d.).

Transcript	Decision	Details
[Right so the // So ((it would be))] little things like wires and stuff we should just check on ~E bay and order them up.	judgment call	background noise
Right so the little things like wires and stuff [we should just check on ~E bay and // we should just look up on E-bay and // (()) in the] order them up.	transcriber error	background noise
Right so the little things like wires and stuff we should just check on ~E bay and order [them up. // of the -]	judgment call	background noise, overlapping speech

Table 2.15: Preliminary results with SCLITE scoring (Glenn et al., n.d.).

Language	Genre	Careful Transcription WDR	Quick (Rich) Transcription WDR
English	CTS	4.1-4.5%	9.63% (5 pairs)
	Meeting	-	6.23% (4 pairs)
	Interview	n/a	3.84% (22 pairs)
	BN	1.3%	3.5% (6 pairs)
	BC	n/a	6.3% (6 pairs)
Chinese	BN	7.40% (23 pairs)	6.14% (18 pairs)
	BC	9.06% (24 pairs)	9.45% (4 pairs)
Arabic	BN	3.13% (14 pairs)	3.42% (16 pairs)
	BC	3.93% (12 pairs)	8.27% (18 pairs)

Furthermore, a study of voice transcription architectures in the Spanish Database was also conducted by Alvarez et al. The TV shows from various genres shown on Spanish public television (RTVE) from 2018 to 2019 are included in the database used for the speech transcription assessment. The audio in the database, which lasted 55 hours and 40 minutes, was entirely transcribed by humans to make actual references. These materials are in STM format with time-marked segments that specify the waveform's filename and channel number, the speaker, the start and end times, an optional subset label, and the segment's precise transcription. Table 2.16 (Alvarez et al., 2022) displays the diversity of TV programmes in the database.

Table 2.16: TV shows included in the RTVE2020 dataset (Álvarez et al.,2022).

TV Program	Duration	Description
Ese programa del que Ud. habla	01:58:36	A TV program that reviews daily political, cultural, social and sports news from the perspective of comedy.
Los desayunos de RTVE	10:58:34	The daily news, politics, interviews and debate program.
Neverfilms	00:11:41	A webseries that parody humorously trailers of series and movies well-known to the public.
Si fueras tú	00:51:14	Interactive series that tells the story of a young girl.
Bajo la red	00:59:01	A youth fiction series whose plot is about a chain of favours on the internet.
Comando actualidad	04:01:31	A show that presents a current topic through the choral gaze of several street reporters.
Boca norte	01:00:46	A story of young people who dance to the rhythm of trap.
Wake-up	00:57:28	A story that combines science fiction, a post-apocalyptic Madrid and lots of action inspired in video games.
Versión española	02:29:12	Program dedicated to the promotion of Spanish and Latin American cinema.
Aquí la tierra	10:26:02	A magazine that deals with the influence of climatology and meteorology both personally and globally.
Mercado central	08:39:47	A Spanish soap opera set in a today's Madrid market.
Vaya crack	05:06:00	A contest where contestants take multiple quiz designed to test their abilities in several disciplines.
Cómo nos reímos	02:51:42	A program dedicated to the great comedians and their work on RTVE programs.
Imprescindibles	03:12:31	A documentary series on the most outstanding figures of Spanish culture in the 20th century.
Millennium	01:56:11	Debate show for the spectators of today, accompanying them in the analysis of everyday events.
Total duration	55:40:16	

As can be seen from the descriptions of the programmes in Table 2.16, most TV shows include material with spontaneous speech, considerably increasing the complexity of mechanically transcribing this database. Next, the systems' overall Word Error Rate (WER) findings for each TV show in the RTVE2020 dataset are provided in Table 2.17.

Table 2.17: Total Word Error Rate (WER) of ASR systems on each RTVE2020 test set TV programme (Álvarez et al.,2022).

TV Program	Multistream CNN	CNN-TDNN-F	Q15×5	Q5×5	Wav2vec2.0
Ese programa del que Ud. habla	23.64	25.67	29.65	36.15	26.81
Los desayunos de RTVE	9.26	10.11	12.14	14.68	11.08
Neverfilms	19.81	24.21	29.03	37.82	28.05
Si fueras tú	24.57	29.31	36.76	46.73	36.43
Bajo la red	22.41	33.31	32.99	41.06	32.33
Comando actualidad	22.58	24.68	27.34	32.70	25.6
Boca norte	32.07	37.94	43.16	52.92	40.37
Wake-up	30.87	33.96	40.81	47.71	38.19
Versión española	16.14	18.10	19.15	25.66	18.06
Aquí la tierra	14.90	16.48	19.69	24.68	17.67
Mercado central	16.44	17.83	25.43	34.05	21.91
Vaya crack	19.22	19.96	28.43	30.16	20.80
Cómo nos reíamos	46.17	48.53	54.33	61.41	53.20
Imprescindibles	30.44	34.45	37.12	44.94	29.52
Millenium	16.02	15.98	17.30	18.82	17.57
Global	17.60	19.27	22.96	28.42	20.68

The systems consistently perform over the whole RTVE2020 dataset, as demonstrated in Table 2.17. The Multistream CNN-based system achieved the most remarkable results among the TV programmes, except Millenium, where the baseline CNN-TDNN-F system performed best. In contrast, Wav2vec2.0 fared just slightly better in Imprescindibles TV program. Comparatively, the Quartznet Q5 system has the worst overall performance. Regarding overall WER, the remaining systems stay consistent for most TV programmes. The Wav2vec2.0-based system outperforms the Quartznet Q155-based ASR engine in all but one scenario (Millenium). However, the margin of victory is just 0.27. The systems' behaviour regarding content profiles is generally consistent with expectations. Compared to other programmes with poor acoustics, overlapping or spontaneous speech, WER drastically decreases in programmes with more explicit communication.

2.3.4 Conclusion

In conclusion, rather than choosing the best speech transcription solutions, we should also look at the speech data used. Speech data used must be clean without any third-party interference. The speech transcription solutions must be sensitive enough to capture and predict the words the user is speaking to provide the most optimal results.

2.4 Application of artificial intelligence techniques on mobile applications

After reviewing research articles about emotional speech recognition and evaluating current speech transcription technology, we shall finally review the available techniques to integrate AI-dependent software programs and mobile apps into a mobile application. We shall also discuss how its various systems can implement AI into their mobile application. This is to fulfil the final product of my final year project program, the Speech to Text with emojis.

Castanyer, Martnez-Fernández, and Franch proposed a pipeline for creating a functional mobile application for traffic sign recognition in their original article. For merging DL modelling and developing DL-based applications, current DL frameworks are continually evolving. These frameworks provide a maximum of two modelling approaches for transferring processes. On the other side, there is model conversion. TensorFlow-Light for Android and CoreML for iOS apps are the two most well-known applications for these abilities. On the other hand, the Open Neural Network Exchange (.ONNX) file format, made available in 2017 by Facebook and Microsoft, is used for model export. Castanyer, Martnez-Fernández, and Franch (2002) claim that this file format enables the compression of serialised versions of trained networks.

The development environment must allow for writing to an ONNX files and CNN training. The operation-side component must be able to read the model, use it, and enable the use of the camera on a mobile device. In this effort, Unity serves as the operation-side platform, while PyTorch is the development-side platform. The two platforms are connected by use of the Unity Barracuda libraries. Additionally, as seen in Figure 2.7, PyTorch models undergo training on the Kaggle GPU, for free for 40 hours per week. (Castanyer, Martnez-Fernández, and Franch, 2021).

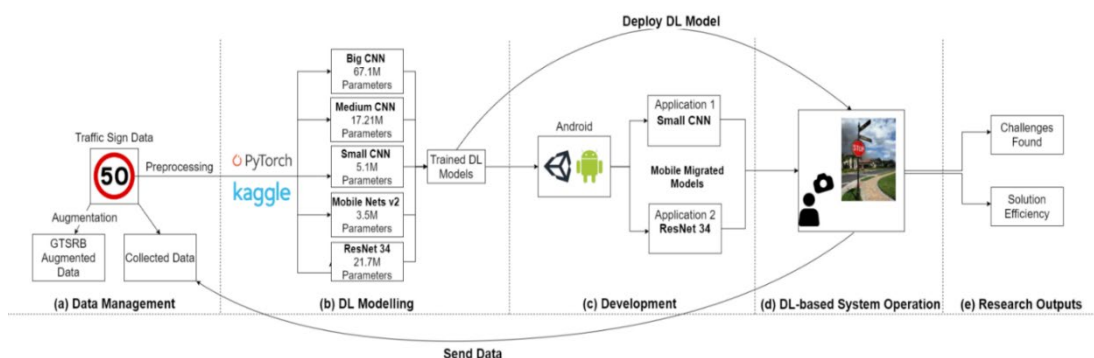


Figure 2.7: Software development lifecycle in our DL-based software (Castanyer, Martínez-Fernández and Franch, 2021).

Figure 2.8 depicts the architecture of the DL-based software system. They produced two applications with identical architectures that differ only in terms of their DL model. For example, the software that loads the SmallCNN which has the size of 68.84MB, but the application that loads the RN34 which has the size of 131MB. Without the models, the application package created by Unity3D which has the size of 46.73MB (Castanyer, Martnez-Fernández, and Franch, 2021). In contrast, they used mobile device storage for our actual study, manually annotated and uploaded it to the local desktop, and fed the model into the Kaggle GPU. Although not scalable, this approach is appropriate for research.

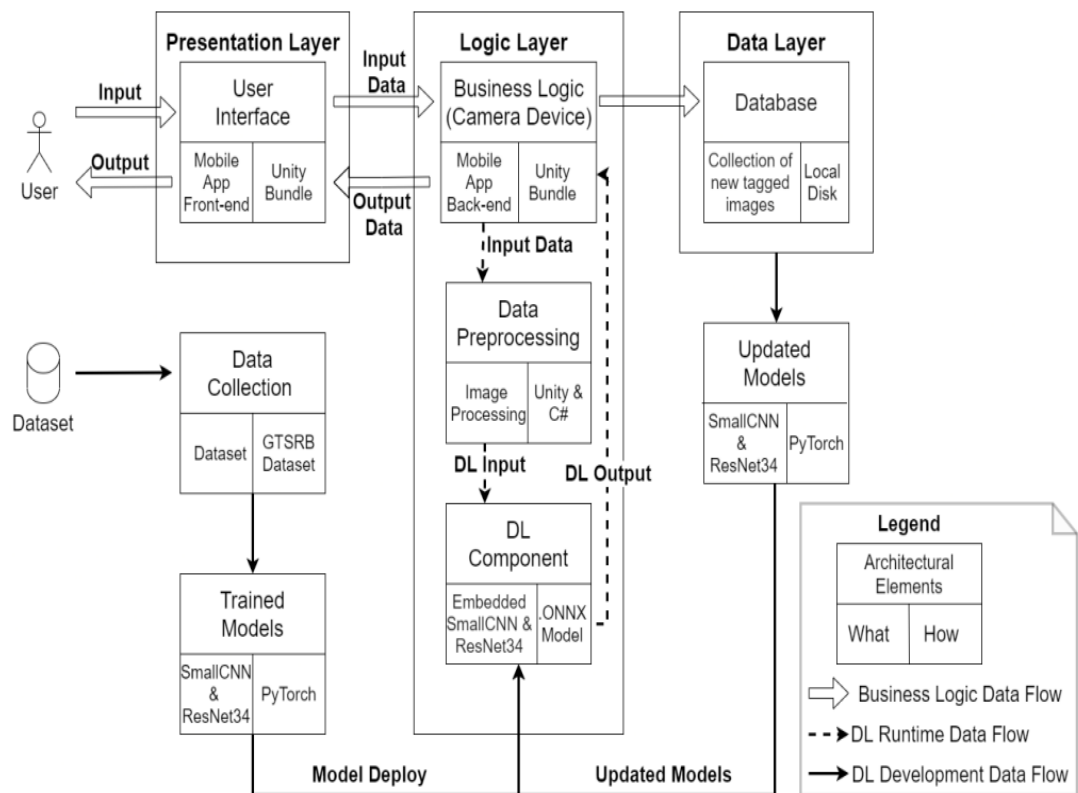


Figure 2.8: DL-based Software System Operation Architecture (Castanyer, Martínez-Fernández and Franch, 2021).

In contrast, Miraz, Ali, and Excell employ the Culturally Inclusive Adaptable User Interface (CIAUI) Framework for their study on an AI-based adaptive user interface for a mobile application, as illustrated in Figure 2.9. (Miraz, Ali and Excell, 2022).

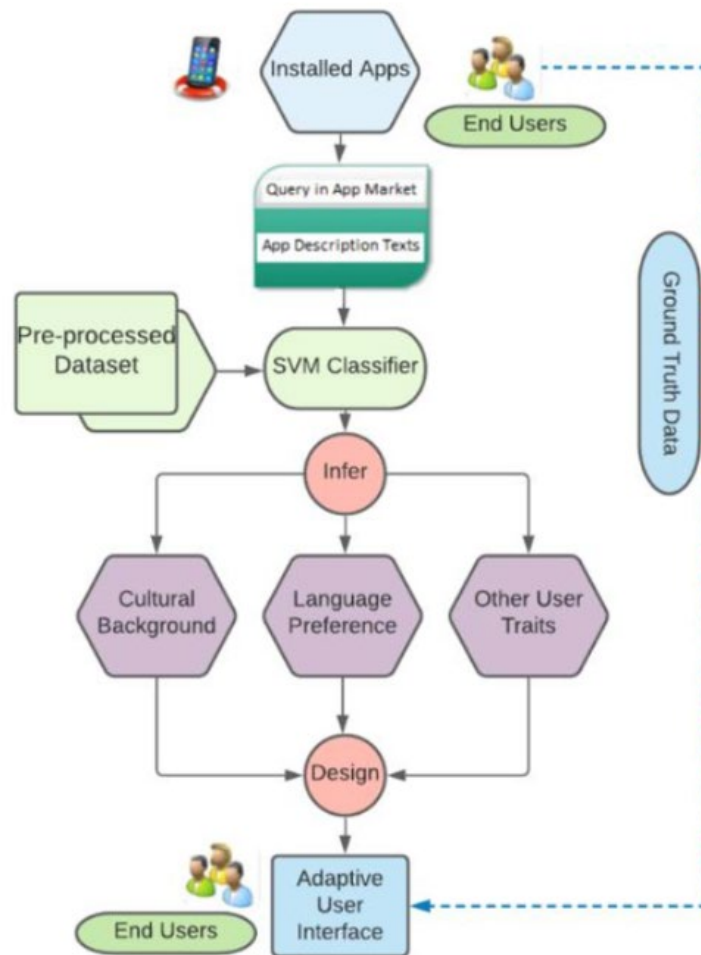


Figure 2.9: The CIAUI Framework (Miraz, Ali and Excell, 2022).

The prototype uses the list of installed applications to deduce the users' language choice and cultural affinity and then offers an altered UI tailored to those characteristics. The SVM classifiers' training parameters are used to make the prediction. The prototype produced thus supports user interface customization by predicting desired language choice and cultural affiliation based on a snapshot of the applications installed in the various devices. The recall and accuracy statistics of the SVM classifiers (machine learning algorithms) were examined using data (ground truth) from 253 culturally diverse consumers (Miraz, Ali and Excell, 2022).

Finally, Sarker et al. wrote an article about the importance creating AI based mobile applications and the proper AI techniques to implement for different kinds of mobile systems. The smartphone of today is also referred to as "a next-generation, multi-functional cell phone that supports data processing as well as increased wireless connection,". According to Google Trends data, consumers' interest in "Mobile

Phones" has increased over other platforms such as "Desktop Computer," "Laptop Computer," or "Tablet Computer" throughout the past five years from 2014 to 2019 shown in Figure 2.10 (Sarker et al., 2020).

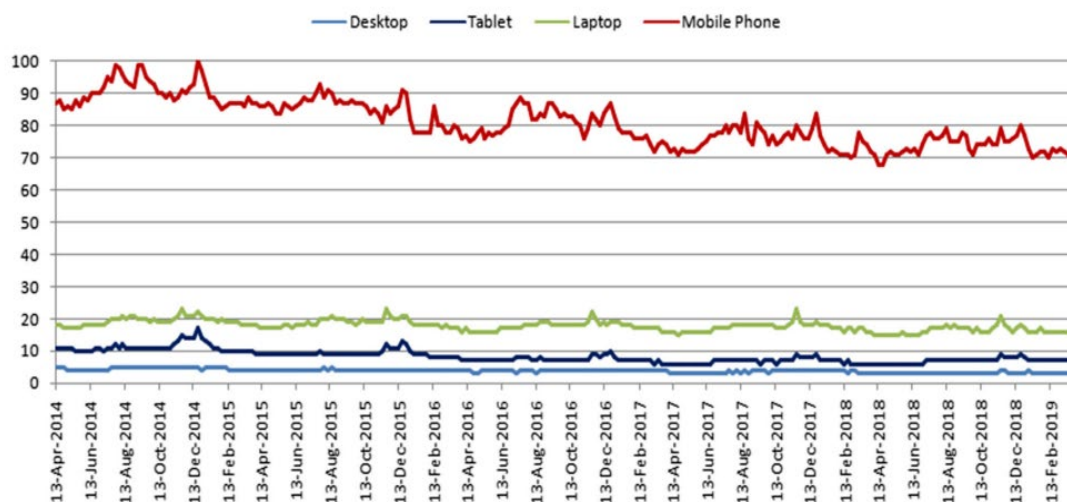


Figure 2.10: Users' interest trends through time, where the x-axis shows timestamp data and the y-axis indicates a popularity score ranging from 0 (min) to 100 (max) (Sarker et al., 2020).

According to user needs, several academics build various context-aware mobile services using association rules. Additionally, a clustering approach has been used in several studies for various research objectives. Deep learning has also been the subject of significant study for various mobile analytics goals. The research also heavily relies on context engineering techniques like context correlation and principal component analysis. Based on their inquiry's most common approaches and data-driven actions, they have summarised this study in Table 2.18.

Table 2.18: An overview of the data-driven activities and methods used to different context-aware mobile services and systems (Sarker et al., 2020).

Tasks and Approaches	Purposes
Clustering	Time-based segmentation, Time-series modeling
Association	Notification management, Usage modeling, Interruption management, Recency analysis
Context Engineering	Apps usage, principal component analysis, Context correlation analysis, Context Pre-modeling
Support Vector Machine	Instant messaging, transportation system, activity recognition, notification management, interruptibility prediction
K-Nearest Neighbor	Mobile search, recommender system, location prediction, activity recognition, interruptibility prediction
Logistic Regression	Activity recognition, user modeling, recommendation system, health analytics, interruptibility prediction
Naive Bayes	Phone call prediction, location prediction, interruption management
Decision Tree	Context-aware system, mobile service, interruption management, interruptibility prediction, phone call prediction
AdaBoost	Interruption management, interruptibility prediction, location prediction, recommender system
Random Forest	Call availability prediction, instant messaging, transportation system, activity recognition, interruptibility prediction
Neural Network	Smartphone power modeling, mobile credit card payment, mobile commerce, mobile learning, smartphone characterization

2.4.1 Conclusion

In conclusion, appropriate AI techniques must be used to create the AI-based mobile application. By doing so, the best optimal results shall be obtained. A successful, intelligent mobile system must have the appropriate AI-based modelling depending on the data qualities. Before the system can aid users with recommendations and decision-making, the complex algorithms must be trained using acquired data and information linked to the target application.

CHAPTER 3

METHODOLOGY AND WORK PLAN

3.1 Introduction

Chapter 3 presents the project timeline. Furthermore, this chapter also covers the overall approach and methodologies used to develop the system, including the system architecture and work plan. It also includes details on data collection and preprocessing, speech recognition and emoji prediction models, and mobile application development. More than one methodology has been used as in the course of developing the system challenges were faced hence different methodologies were used for certain areas of the system. To emphasise the project scope and schedule planning, the work breakdown structure (WBS) and Gantt chart were created and provided at the end of this chapter.

3.2 Prototyping Development Methodology

The prototyping development methodology is implemented as the system development methodology for this project.

3.2.1 Requirement Gathering Phase

Requirement gathering phase includes two subphases namely specification and knowledge acquisition.

3.2.1.1 Specification

The first step of this project is to identify the specifications. Next, we need to identify the target users for this project. In my case, the target users are the public interested in and find the speech-to-text with emoji solutions helpful. After identifying the target users, research on the problem statement was conducted to get more information and statistics on the need for a speech-to-text solution. The functional and non-functional requirements are also collected in this stage along with required features in the system. This phase will be further elaborated in Chapter 4 System Specification.

3.2.1.2 Knowledge Acquisition

This section focuses on acquiring project information from various sources, such as a literature review. Evaluating academic resources such as journal articles, websites, and books are called a "literature review." For this project, several articles are referenced and studied to further analyse the chosen field of studies. An overview, a summary, and an assessment of the present level of knowledge of three field of studies related to Speech to Text with Emojis were presented. Furthermore, significant features may be retrieved and included in the project by analysing the results, techniques, and the concepts of the field of studies.

Firstly, voice recognition system for emotion detection is studied. The process and basic outline of speech emotion recognition system was thoroughly studied. Moreover, the types of voice feature extraction were also reviewed to select the proper feature selection to train the model. Furthermore, the types of approach to train the emotional speech recognition model was also studied. Factors affecting clean speech data was also concerned. The correct approach would need to be implemented in my project to obtain the optimal results. Different implementations were compared and discussed.

Next, the current speech transcription methodologies were evaluated to observe the comparison and the efficiency of current speech transcription solutions. Several methods were compared and discussed to use the optimal method to be implemented in the project.

Finally, the different applications on integrating AI-based mobile applications were studied. Different kinds of mobile application system would require different kinds of AI learning methods. On top of that, the software to integrate AI into mobile apps were discussed including TensorFlow-Lite and PyTorch framework.

3.2.2 Design Phase

The design step was crucial because it laid the groundwork for the subsequent implementation phase. A prototype of the software system or application was produced during the design phase of the Prototyping technique. The required features, functionality, and user interface were determined and included in this prototype during the requirements-collecting process. The user experience was carefully studied, and the prototype was designed to be simple and intuitive. The system's performance,

scalability, and security requirements were also considered, ensuring that the prototype was created to match these standards. During this step, the use case diagram was created to demonstrate how the target users would interact with the constructed system. Furthermore, use case descriptions were created to offer additional data and elaboration on how the built system would react to the different user interactions. Following that, the system architecture and process flow diagram was created to represent the flows of the built systems. This shows how the back-end flow is executed when modules are run concurrently. This phase will be further elaborated in Chapter 5 System Design.

3.2.3 Iteration Process

There will be two iterations. Each iteration is built on newly created features. A new feature will be added with each iteration. There will be five phases: design, model training phase, prototype construction, user assessment, and review.

3.2.3.1 First Iteration

In this first iteration, prototyping development starts with requirements and features being understood. In my case, the main feature of my Speech to Text with Emojis mobile application is to convert speech into text with emojis. Hence, a lot of work are being done in the back-end development to train the model and integrate both modules in the system. Research and development are quickly executed to develop the first version of the system.

- i. **Design Phase**

Design the main features which must be implemented in the app. Diagrams will be prepared to show detailed flow between the modules working concurrently in the system.

- ii. **Model Training Phase**

This system development heavily focuses on back-end development such as model training hence the emotion recognition model will be trained in this phase. An inbuilt Android Speech Transcript API will be used hence no training will be needed.

- iii. **Prototype Construction Phase**

A prototype will be developed in this phase. This prototype will mainly focus on display the speech transcript with emojis on screen without considering other factors such as data fetching, loading time and performance. These steps will be achieved using Chaquopy framework in Android Studio.

iv. User Assessment Phase

5 random people from my target user group will be selected to review the prototype and will be asked to complete a brief survey to offer feedback on the prototype. Their behaviours and use of the prototype will be watched and documented.

v. Review Phase

The feedback received from participants will be examined and studied to identify potential changes. Once the prototype has been polished, go on to the second iteration.

3.2.3.2 Second Iteration

In the second iteration, two features for translating the audio such as real-time speech recording and prerecord audio uploading will be included. Users can have the flexibility to upload their speech or may speak in real-time to have their speech translated to text with emojis.

i. Design Phase

When the prior prototype has been refined, additional features are introduced to the second iteration. System logics are changed to suit the newly introduced features.

ii. Model Training Phase

When the prototype is being refined in the previous iteration, the emotion recognition model is tuned to increase the accuracy of the model.

iii. Prototype Construction Phase

This iteration will include new features created using the necessary development tools. Chaquopy framework is used to deploy the AI model to Android Studio.

iv. User Assessment Phase

5 random people from my target user group will be selected to review the prototype and will be asked to complete a brief survey to offer feedback on the

prototype. Their behaviours and use of the prototype will be watched and documented.

v. Review Phase

The feedback received from participants will be examined and studied to identify potential changes. Final system refinement is done and ready to be delivered.

3.2.3.3 Third Iteration

In the third iteration, TensorFlow lite framework is used instead of Chaquopy to make the mobile app more robust.

i. Design Phase

When the prior prototype has been refined, deployment methodology of AI model has changed.

ii. Model Training Phase

When the prototype is being refined in the previous iteration, the emotion recognition model is tuned to increase the accuracy of the model.

iii. Prototype Construction Phase

This iteration will include new back-end development created using the necessary development tools.

iv. User Assessment Phase

5 random people from my target user group will be selected to review the prototype and will be asked to complete a brief survey to offer feedback on the prototype. Their behaviours and use of the prototype will be watched and documented.

v. Review Phase

The feedback received from participants will be examined and studied to identify potential changes. Final system refinement is done and ready to be delivered.

3.2.4 Implementation Phase

The implementation phase was an essential aspect of the Prototyping technique since the design was transformed into a functional software system or application. The prototype was constructed and tested during the implementation phase. The prototype

was built and tested repeatedly, with each iteration building on the one before it. The prototype was tested to verify that it was stable and operated correctly, and any faults or errors discovered during testing were fixed. The prototype implementation heavily focuses on the backend development as all the features depends on the back-end code to work. This phase will be further elaborated in Chapter 6 Implementation.

Hence, the system documentation is completed, including a clear account of the entire creation of the mobile application project. Finally, PowerPoint transparencies are created to show the whole project's growth from project inception to the conclusion of the execution period. This phase's deliverables would be the plan report and the final implemented system.

3.2.5 Evaluation Phase

The prototype was evaluated against the specifications and design goals during the assessment phase. Users provided feedback. This input was crucial in identifying areas where the prototype might be improved. The prototype was thoroughly tested to verify that it fulfilled the requirements and expectations of the intended audience.

i. Unit Testing

In unit testing, test cases are designed with a few tests which each of the components or features must pass and some where they must fail. The two main components of the speech to text with emojis system is the speech transcription module and the emotional speech recognition system module.

ii. Integration Testing

Two or more components or functionalities are integrated to be evaluated as part of integration testing. Integration testing is crucial for checking for any interface flaws that could be present. It also examines how integrated parts interact with one another. In this phase, we shall integrate both our individual modules into one and test them.

iii. Usability Test

During the usability testing phase, 10 selected individuals from the target audience will evaluate the mobile app's user experience and ease of use. They will be given a set of tasks to perform, and their interactions with the app will be observed and recorded, paying close attention to their behaviour and any difficulties they encounter. Upon completion of the tasks, each user will be

given a brief survey to gather their comments and opinions on the mobile app's design, navigation, and overall user experience. The results will be analysed to identify areas of improvement, which will be addressed by the development team to enhance the app's usability.

iv. User Acceptance testing

User acceptance testing is the final phase of testing, conducted to ensure the application meets its requirements and functions effectively in a real-world setting. A group of 10 selected individuals will test the final product, following a checklist outlining the tasks they need to perform and the expected outcomes. This helps verify if the mobile app fulfills its intended criteria and satisfies the needs of end-users. After completing the tasks, each user will provide feedback through a brief survey to share their opinions on the app's performance and overall suitability. The results will be analyzed to identify any issues that were not detected during earlier testing stages, such as unit and integration testing, ensuring the application is ready for deployment.

This phase will be further elaborated with the evidence of the Speech to Text with emojis system being tested and evaluated in Chapter 7 Evaluation and Testing.

3.2.6 Refinement Phase

The refinement phase was crucial in ensuring that the prototype satisfied the demands of the target users and was suitable for final deployment. Feedback and evaluation enhanced the prototype's design and functionality throughout the refinement phase. The comments were carefully reviewed, and any adjustments implemented were consistent with the project's aims and objectives. Hence, the final version of the system is delivered alongside the final year documentation report and presentation slides.

3.3 AI Learning Model Methodology

Two AI learning modules are required to fulfil the requirements of the speech-to-text with emojis system, which is emotional speech recognition and speech recognition. These models are implemented in the emotion and words detection of the user's speech. The system will need to classify and predict the correct emojis displayed on the screen

with the speech the user speaks and transcript the speech. The basic workflow for developing the AI learning models is shown in Figure 3.1.

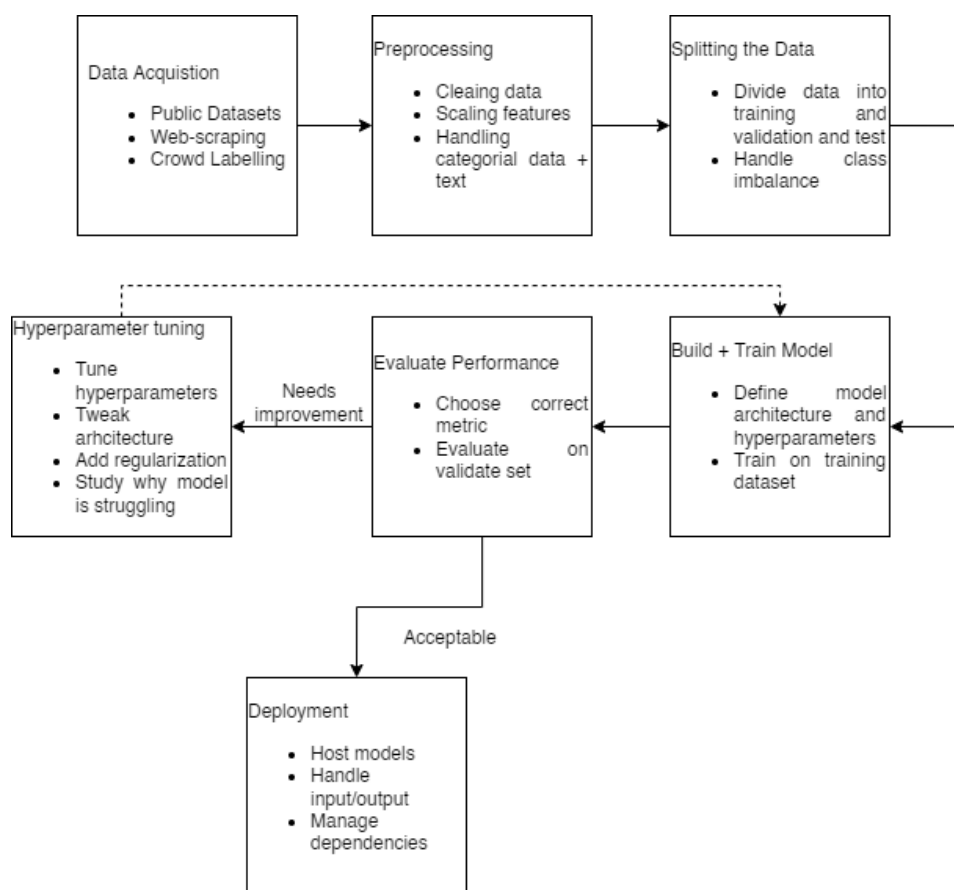


Figure 3.1: Basic workflow of the AI learning model

3.3.1 Data Acquisition

The most worrying issue is the lack of labelled data obtained in a deep learning project. Logically, our model will be more accurate with more tagged data. However, the ability to gather data can either make or break our solution. Data acquisition is the most critical and challenging part of deep learning. Thankfully, there are several methods to find relevant data sources.

Firstly, the most acceptable data sources are readily available to the public. For instance, Kaggle includes hundreds of enormous, labelled data sets. Launching a deep learning project has less overhead when using these chosen datasets. In some instances, businesses may already have a sizable dataset. A Relational Database Management System is often used to store these datasets (RDMS). In this case, we may use SQL queries to build our dataset.

Additionally, rich data streams from social media, online news, and search results are available for our deep learning applications. Web scraping, which extracts data from websites, is how we do this. We should consider ethical considerations, such as privacy and consent problems, when scraping and gathering data. One of the many Python web scraping tools is called BeautifulSoup. Many websites, like Reddit and Twitter, use Python application programming interfaces (APIs). Data from several applications may be gathered via APIs. While some APIs are paid services, others are available for free.

3.3.2 Pre-processing

After finishing the data acquisition process, we shall move on to the pre-processing stage. After we have created our dataset, we will need to pre-process it to extract functional characteristics for our deep learning models. Pre-processing is also an excellent chance to get better acquainted with our data. When preparing data for neural networks, we have three primary goals:

- i. Clean the data,
- ii. Manage category features and text
- iii. Scale the real-valued features using normalisation or standardisation approaches.

3.3.2.1 Cleaning the data

Our datasets often include noisy instances, additional features, missing data, and outliers. Testing for outliers, removing extraneous features, filling in missing data, and filtering out noisy samples are acceptable practices. In our case, we need to have clean speech recording data. The speech the user speaks must be clean without external disturbances like unwanted background noises (Codecademy, n.d.).

3.3.2.2 Scaling features

Our models will struggle with input characteristics with high values since we start neural networks with minimal weights to stabilise training. Hence, we often scale real-valued features in one of two ways: either by normalising them to be between 0 and 1 or by standardising them to have a mean of 0 and a variance of 1 (Codecademy, n.d.).

3.3.2.3 Handling categorical data

Numbers are what neural networks anticipate as inputs. All category data and language must thus be converted into numbers. Categorical variables are often handled by converting them to one-hot encodings or allocating a unique integer to each choice. However, before we go through a few more processing processes encoding our words as numbers, when dealing with raw text strings, seizing, and padding our data are two procedures (Codecademy, n.d.).

3.3.3 Splitting and Balancing the Dataset

After data processing, it is time to divide the dataset. Typically, training and validation datasets are created from our data. In certain circumstances, we produce a third holdout dataset known as the test set. When we do not do this, we often conflate the phrases "test" and "validation" sets. We use the training dataset to develop our model and the validation dataset to assess it. After choosing our model and fine-tuning our hyperparameters, we test our model on this dataset if a third holdout test set has been generated. With this third step, we can avoid selecting a collection of hyperparameters that happens to be effective with the data we selected for our validation set. The size of our divides and whether we will stratify our data are the two main factors to consider while dividing our dataset. We must fix imbalances in our training set once we have divided our data (Codecademy, n.d.).

3.3.3.1 Splitting the Data

10% to 30% of our data are often preserved for validation and testing. Allocating a higher fraction of data to the validation set is essential when we have a smaller corpus. This increases the likelihood that the distribution of our actual data in our validation dataset will be correct. Our data are divided into training and validation datasets using Scikit-learn's train test split function, specifying how much validation data is utilised (Codecademy, n.d.).

3.3.3.2 Stratified Train-Test Splits

Since it is highly possible that more occurrences of our minority classes will wind up in the training or validation set, extra caution must be used when separating a particularly unbalanced dataset for classification. Our validation measures will not

appropriately reflect how well our model can first categorise the minority class therefore. The model will thus overestimate the likelihood of the majority class in the second scenario. The answer is to utilise a stratified split, which guarantees an equal number of samples from each class in the training and validation sets. The train test split function will calculate the percentage of each class and guarantee that this ratio is the same in both training and validation data if we provide the stratified parameter to our array of labels (Codecademy, n.d.).

3.3.3.3 Handling Imbalanced Data

Unbalanced data, with some classes occurring much more often than others, presents a difficulty to deep learning systems. When neural networks are trained on variable data, the resulting model will be skewed heavily towards predicting the classes that are the majority. This is especially troublesome since we typically care more about identifying examples of the minority classes (Codecademy, n.d.).

Undersampling and oversampling are the two fundamental methods for dealing with varied training data. Both strategies should be utilised with extreme care, and it has been advised that we consult with a subject-matter expert first. We balance our data by undersampling and excluding samples from our majority class. We duplicate instances of our minority class during oversampling to increase their frequency. The artificial Minority Oversampling Technique often replaces traditional oversampling (SMOTE). To make the instances in our dataset equal to those in our minority class, the SMOTE algorithm generates fictitious examples. In virtually all cases, we balance the data in our training set while leaving the validation set alone. As soon as our train-test split is over, we must compensate for the imbalance by adding more training data (Codecademy, n.d.).

3.3.4 Building and Training the Model

After dividing our dataset, we may choose our layers and loss function. We also need to choose an appropriate number of concealed units for each tier. The number of layers you use and the size of each layer that works best depend entirely on your data and architectural setup. It is best to begin with, a couple of layers (2-6). Usually, we use between 32 and 512 hidden units for building each layer. As we go through the model, we also tend to reduce the size of hidden layers. Typically, we begin by using SGD

and Adam optimizers. It is customary to set an initial learning rate by default to 0.01 (Codecademy, n.d.).

3.3.5 Evaluating Performance

We assess the model's performance on our validation set after each training iteration. When we provide a validation set during training, Keras takes care of this automatically. We can predict how well our model will perform on brand-new, untested data based on how well it performed on the validation set. It is crucial to use the right measure when evaluating performance. The significance of accuracy (and even AUC) will be diminished if our data collection is severely unbalanced. We should probably think about measures like recall and accuracy in this situation. Another helpful statistic that combines recall and accuracy is the F1 score. A confusion matrix may be used to see which data points are and are not misclassified (Codecademy, n.d.).

3.3.6 Tuning Hyperparameters

The next step is to fix our initial hyperparameters using the standard methods. We experiment with various learning rates, batch sizes, architectures, and regularisation methods to train and evaluate our model. As we adjust our parameters, we should keep an eye on our metrics and loss, and search for clues as to why our model is having problems. For instance, unstable learning recommends that we alternate between smaller and bigger batch sizes. We have overfitted if there is a performance difference between the training and assessment sets. Therefore, we need either to employ regularisation or minimise the size of our model (like dropout). We are underfitting if we do poorly on the training and test sets. For example, we could need a bigger model or a different learning rate. Starting with a more basic model and progressively adjusting the hyperparameters until the model's performance during training and validation diverges, indicating overfitting of the data, is a common strategy. Regardless of the hyperparameters, our outcomes will significantly vary since neural network weights are randomly initialised. Therefore, running the same hyperparameter setup several times with various random seeds is one way to confirm the accuracy of our findings. We are prepared to apply our approach once we are happy with the outcomes. Our model may now be validated using any holdout test sets we developed

that were not dependent on our validation data. This is due to the holdout test set finally ensuring our model's effectiveness in missing data (Codecademy, n.d.).

3.3.7 Deployment of software

Finally, after a model has been trained, it must be made available to the public. This is especially true in professional settings where customers and employees will utilise our networks, or our tools and products will be used internally. Three primary considerations need to be made when putting a neural network into practice.

We must first consider the computational needs for running our models even when managing inputs from several users; using a neural network to analyse a single input requires significant computing. Therefore, it is crucial to hosting the Docker container where it can access necessary processing resources when installing a neural network model in a container. An excellent place to start is one of the cloud platforms, such as AWS, GCP, or Azure. These platforms provide adaptable hosting services for programmes that might grow to accommodate shifting needs. Finally, we can run the code and handle dependencies whenever we host our application. Finally, we can run the code and manage dependencies whenever we host our application using Docker Containers. To enable our application to run in any computer environment, we must pack up our code and all its dependencies using Docker containers, including the correct version of TensorFlow (Codecademy, n.d.).

3.4 Speech Transcription Methodologies

Speech transcription is an essential part of the project since it requires turning spoken language into written text, which is then combined with the emoji recognition module to display the transcript text with emoji.

Various approaches, including Automatic Speech Recognition (ASR) and Deep Learning, have been investigated to determine the system's most accurate and efficient technique. Each method has advantages and disadvantages, and the project's unique needs will determine the choice.

3.4.1 Automatic Speech Recognition (ASR)

ASR (Automatic Speech Recognition) is a technique that allows computers to recognise and transcribe spoken language into text. Virtual assistants, dictation software, and voice-activated gadgets all use ASR extensively.

ASR works by analysing the sound waves in spoken language to detect individual words and phrases and then converting these words into text using language models. The technology employs digital signal processing, statistical modelling, and machine learning algorithms to transcribe voice into text. Figure 3.1 shows the flow of ASR.

Based on Figure 3.1, the following are the fundamental phases in the ASR process:

1. Pre-processing: the incoming audio signal is examined for noise, distortion, and other undesirable effects.
2. Feature extraction: To reflect the spectral content of the speech signal, features such as Mel-frequency cepstral coefficients (MFCC) are retrieved from the pre-processed signal.
3. Acoustic modelling: a statistical model is developed to reflect the link between the retrieved data and the phonemes (the minor sound units in a language).
4. Language modelling: a language model is used to predict the most probable series of words given the acoustic model's sequence of phonemes.
5. Decoding: The linguistic and acoustic models are merged to provide the most probable transcription of the input voice signal.

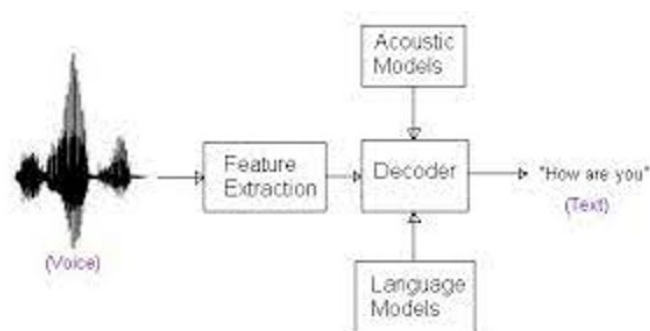


Figure 3.2: Automatic Speech Recognition (ASR) flow diagram

3.4.2 Deep Learning based model

Deep learning-based voice-to-text technology is a natural language processing (NLP) type that employs artificial neural networks to turn speech signals into text. It is a

machine-learning approach that is especially well-suited for voice recognition due to its ability to capture complex patterns and correlations in audio data.

Deep learning-based speech-to-text technology first converts an audio input into a feature representation that a neural network can analyse. This feature representation is often constructed using a method known as Mel-frequency cepstral coefficients (MFCCs), which break down the audio signal into a sequence of spectral features that capture the frequency content of the movement. The converted signal is then sent into a deep neural network, which has been trained to anticipate the matching text output given the input audio signal. The network often comprises numerous layers of artificial neurons coupled in a hierarchical form. During training, the network's weights are modified to minimise the projected and actual text output disparity.

3.4.3 Conclusion

Following experiments with two voice transcription technologies, ASR, and deep learning, it is possible to infer that both approaches have benefits and limits. While deep learning-based voice recognition has shown promising results in certain applications, ASR has proved to be more dependable, accurate in general and easier to be integrated in the Android application system. Both techniques will be implemented and discussed in Chapter 6 Implementation.

3.5 Emoji Recognition Methodologies

The recognition of emotions is critical to the Speech to Text with Emojis project. This module aims to recognise and categorise human emotions based on voice cues. This study will employ various methods to establish the optimal way for emotion recognition in voice data. Emotion identification in speech may be accomplished using multiple methods, including classic machine learning algorithms and deep learning approaches. Traditional machine learning algorithms include feature extraction methods and classifiers such Multi-Layer Perceptron (MLP) for training and prediction. Deep learning algorithms, on the other hand, may train feature representations from raw speech signals and generate predictions based on the learnt features, such as Long Short-Term Memory (LSTM). Each technique has benefits and limitations, and the approach chosen is determined by the nature of the issue and the available data.

3.5.1 Long Short-Term Memory (LSTM) networks

Delving into the broader domain of Deep Learning Methodologies, the Long Short-Term Memory networks (LSTM) model is employed for training and evaluating the emotional speech module. LSTM, a type of recurrent neural network (RNN), is adept at learning extended dependencies, especially in sequence prediction tasks. Unlike standalone data points like images, LSTM possesses feedback connections that allow it to process entire data sequences. This proves useful in applications such as machine translation and speech recognition. A distinct variant of RNN, known as LSTM, demonstrates remarkable performance in various problems.

A memory cell, referred to as the "cell state," maintains its status over time and serves a vital role in an LSTM model. The horizontal line at the top of the figure below symbolizes the cell state, which can be thought of as a conveyor belt that transports data unaltered. Gates in LSTMs manage the inflow and outflow of information from the cell state. These gates can permit or restrict data entry and exit from the cell. A sigmoid neural network layer and a pointwise multiplication operation support this mechanism. Figure 3.4 illustrates the overall concept of the LSTM model.

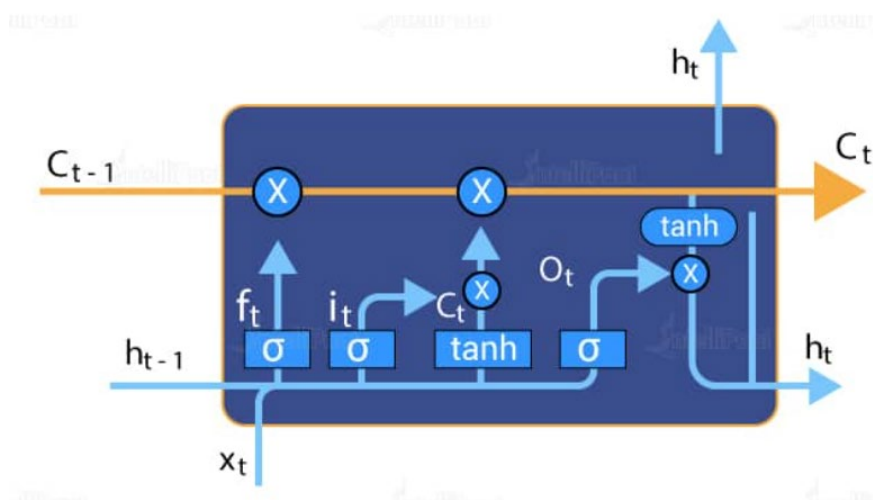


Figure 3.3: General concept diagram of LSTM model (Intellipaas Blog, 2020).

3.5.2 Multi-Layer Perceptron (MLP) Classifier Networks

Figure 3.4 shows an MLP Classifier neural network model comprising numerous layers of nodes completely coupled to one another. The input layer receives the raw feature values and is processed via one or more hidden layers before output. The

hidden layers discover underlying patterns in the input data, whereas the output layer generates the final classification result.

The MLP Classifier is used in emotional speech recognition to categorise speech samples into distinct emotional states based on characteristics collected from the speech signal. The MLP Classifier's input layer would accept feature values, including pitch, intensity, and spectral characteristics, which would then be processed via the hidden layers to learn the essential patterns in the data. Following that, the output layer would provide a classification result showing the emotional state of the speech sample. The MLP Classifier may be trained using a labelled dataset of emotional speech samples. Its accuracy in predicting the emotional state of unseen speech samples can be used to assess its performance.

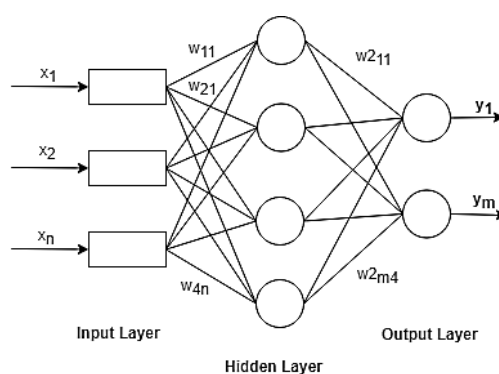


Figure 3.4: MLP Classifier Structure Diagram (Javed et al., 2020)

3.5.3 CNN-LSTM Networks

A CNN-LSTM network is a hybrid deep learning model that combines the capabilities of CNNs with Long Short-Term Memory (LSTM) networks. This combination makes it especially well-suited for tasks involving spatial and temporal information, such as speech-emotion identification. The CNN component of the network analyses the input data, such as spectrograms or other time-frequency representations of the speech stream, to discover local patterns and characteristics. The network's LSTM component analyses temporal information by modelling the data's order and context.

The CNN-LSTM network initially uses the CNN layers to find significant aspects in the speech signal that may include emotional information, such as pitch, intensity, or spectral characteristics, in the context of speech emotion identification. The parts are then processed by the LSTM layers, which capture the temporal dynamics of the speech and describe how emotions change over time. The CNN-

LSTM network can successfully learn the complicated links between speech signals and emotions by incorporating both spatial and temporal information, enhancing the overall performance of emotion identification systems. Figure 3.5 shows the general concept of the CNN-LSTM model.

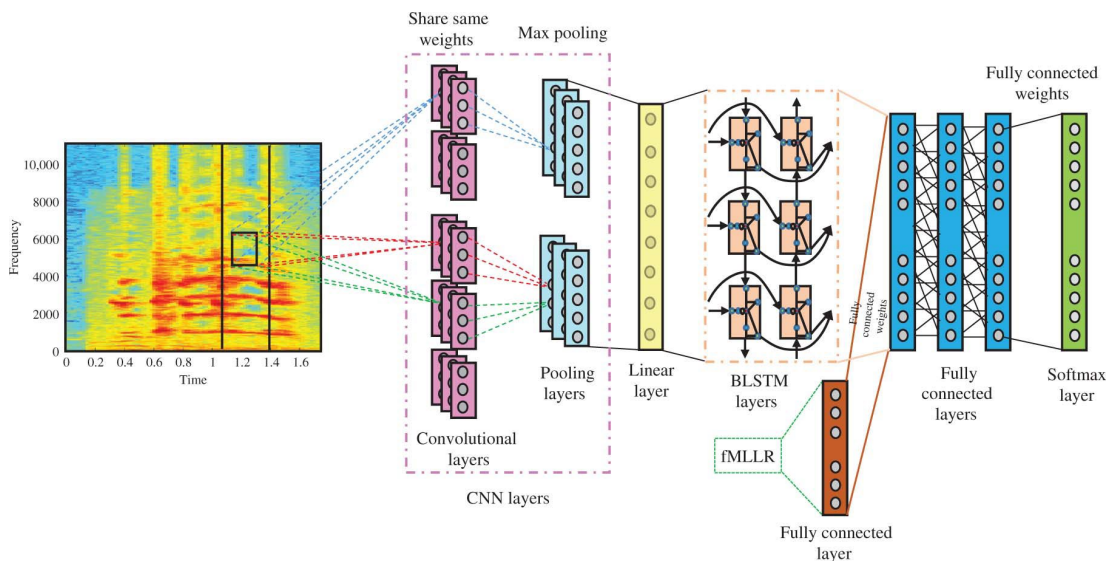


Figure 3.5: CNN-LSTM Structure Diagram (Passricha and Aggarwal, 2019)

3.5.4 Conclusion

Following experiments with two emotion detection technologies, MLF Classifier, and LSTM model, it is possible to infer that both approaches have benefits and limits. Both techniques will be implemented and discussed in Chapter 6.

3.6 Mobile Application Development

The development of the mobile application heavily focuses on the backend development such as deploying the speech transcript and emotion recognition modules to the Android mobile application. Lastly, the threading methodology is used to integrate both modules together.

3.6.1 AI model deployment

Two techniques were used in various iterations throughout the development of the Speech Text with Emojis system to enhance the final design as the project proceeded. The TensorFlow Lite framework was used to install the AI model first. This framework allows learned TensorFlow models to be converted into a lightweight format suited for deployment on mobile devices. The second way entailed running a Python script in

Android Studio using the Chaquopy framework to execute the model. Chaquopy is a plugin that enables Python code to be integrated into Android applications. These two approaches enabled the Speech to Text with Emojis system to be implemented on mobile devices, offering users an on-the-go and accessible way of speech recognition and emotion recognition conversion.

3.6.1.1 TensorFlow-Lite

TensorFlow Lite is a machine learning model deployment framework for mobile devices. It is a mobile version of TensorFlow, a Google open-source toolkit for constructing and training machine learning models. TensorFlow Lite allows developers to create and deploy AI models on mobile devices such as Android smartphones, tablets, and iOS devices. The framework facilitates the development of lightweight models that can operate on low-resource mobile devices. TensorFlow Lite also includes methods for optimising models for mobile devices, such as quantisation, which decreases the accuracy of the model's parameters to minimise the model's size and increase performance. Overall, TensorFlow Lite allows for integrating AI models into mobile applications, enabling new and unique use cases such as voice-to-text with emojis.

Multiple phases are involved in deploying AI models to an Android mobile app using the TensorFlow Lite framework. First, the AI model is trained on a high-performance computer using a deep learning framework such as TensorFlow or PyTorch. The model is optimised and translated to the TensorFlow Lite format using the TensorFlow Lite converter. The optimised model is then included with the Android app and the TensorFlow Lite interpreter. The interpreter oversees executing the optimised model on the mobile device. During the inference stage, the app uses the interpreter to feed input data, in my case, audio, to the model and receive the results (such as text transcript or emotion labels). The programme preprocesses the input data to suit the model's input criteria.

3.6.1.2 Chaquopy

Another popular platform for deploying AI models to Android mobile applications is Chaquopy. It allows developers to leverage their current Python code to construct

mobile applications by enabling them to incorporate Python scripts directly into Android apps. Chaquopy offers a Python API for interacting with Android-specific capabilities like sensors, location services, and cameras. Furthermore, it provides an interface for running Python scripts, allowing developers to use robust Python modules for data analysis and machine learning.

Chaquopy offers a robust and versatile framework for integrating AI models into Android mobile applications, allowing developers to design complex apps capable of real-time data processing and analysis. Chaquopy is used by integrating Python scripts and dependencies into an Android app, creating, and compiling the program, and then deploying it to an Android device or emulator. To begin, developers must write a Python script incorporating the AI model and required libraries. The Python script is then integrated into the app through Chaquopy's API. The Android Studio IDE is then used to build and compile the app. Finally, the app is tested and debugged on an Android device or emulator.

3.6.1.3 Conclusion

In conclusion, the TF Lite and Chaquopy frameworks have been used to deploy the AI models in different versions of the Speech to Text with Emojis system. Implementation of these methods will be discussed in Chapter 6. However, it is essential to note that the implementation process for the Chaquopy framework is much more straightforward than the TF Lite framework. The TF Lite framework is the better option as it uses less space and has faster processing time.

3.6.2 Module Integration Methodology

Multithreading is a method for improving application speed by enabling numerous threads to run simultaneously. Multithreading must be implemented to prevent crashing the mobile application while combining the speech transcript and emotion detection modules. Both modules need a lot of processing power and cannot execute on the main thread at the same time. As a result, they must be operated as distinct threads in the background. This enables both modules to run concurrently and smoothly. The modules will wait for each other to finish in the background. Once both modules have finished processing their inputs, they shall signal to the main thread that their outputs are ready to be released. The outputs of both modules may then be shown

as strings by the main thread, resulting in the final product - text with emojis - being displayed on the screen. Multithreading is used in this process to guarantee that the mobile application operates smoothly and efficiently, improving the user experience.

3.7 Workplan

Workplan helps to organise and schedule tasks so that no task will be delayed or missed out. My workplan for this project includes the work breakdown structure and Gantt chart.

3.7.1 Work Breakdown Structure

1.0 Planning and analysis

1.1 Project initiation

1.1.1 Background research on project

1.1.1.1 Determine project background.

1.1.1.2 Determine problem statement.

1.1.2 Determine project goal.

1.1.3 Determine project solution.

1.1.4 Determine project approach.

1.1.5 Determine project scope.

1.2 Requirement gathering

1.2.1 Data gathering

1.2.1.1 Consult supervisor

1.2.1.2 Analyse results

1.2.2 Literature review

1.2.2.1 Research on voice recognition system for emotion detection

1.2.2.2 Research on evaluation on current speech transcription methods

1.2.2.3 Research on application of artificial intelligence techniques on

mobile applications

1.3 Methodology

1.3.1 Experiment methodology

1.3.2 Choose methodology.

1.3.2 Work breakdown structure

1.3.3 Gantt Chart

2.0 Quick Design

2.1 Requirement specification

2.2 Develop UML

2.2.1 Use case diagrams

2.2.2 Use case descriptions

2.3 System Design

2.3.1 System architecture design

2.3.2 User Interface design

3.0 Prototype development

3.1 First iteration

3.1.1 Design

3.1.2 Train model

3.1.3 Build prototype

3.1.4 User evaluation

3.1.5 Review

3.2 Second iteration

3.2.1 Design

3.2.2 Train model

3.2.3 Build prototype

3.2.4 User evaluation

3.2.5 Review

3.3 Third iteration

3.3.1 Design

3.3.2 Train model

3.3.3 Build prototype

3.3.4 User evaluation

3.3.5 Review

4.0 Development phase

4.1 Evolve final prototype to final product.

5.0 Testing phase

5.1 Unit testing

5.2 Integration testing

5.3 Usability test

5.4 User acceptance test

6.0 Implementation phase

6.1 Proposal writing

6.2 Presentation

6.3 System deployment

3.7.2 Gantt Chart

The project timeline is planned for both Project I and Project II using Gantt Charts. Gantt Charts provide visualization of tasks scheduled overtime. These Gantt charts show the duration of each phase which is included in the work breakdown structure.

First, Figure 3.5 shows the project Gantt Chart for Project I. The issue was identified at the start of the project, and the user needs were understood. The first month was spent doing an in-depth literature review of available solutions. Next, individual components of the system, such as the speech transcription module and the emotional speech recognition training module, were developed. Next, a significant dataset must be collected and annotated to train the deep learning model in the following step. Finally, both modules are integrated by the end of Project 1. However, it only works on prerecorded audio and does not work on accurate time speech recording. Hence, this shall be continued during Project II.

No.	Project Activities	Planned Completion Date	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	W16	W17	
1.	-Project Planning -Problem Formulation -Objective and Scope	2022-07-08	█	█	█	█														
2.	-Proposed solution -Developing individual components of the system - System Methodology	2022-07-16				█	█	█												
3.	-Literature review	2022-07-27						█	█											
4.	-Proposal preparation -Proposal report -system prototype with functionality	2022-09-01								█	█	█	█	█	█	█				

Figure 3.6: Project 1 Gantt Chart

Next, Figure 3.6 shows the project Gantt Chart for Project II done in my final trimester. This is the continuity where we left off from Project I during the final trimester of my final year. A new methodology implementation is thought of in Project

II so that both prerecorded and real-time speech-recognising features can work. A new system's logic is developed and the methodologies from Project I have been altered to suit the new system logic. During the first three weeks of the project, the emphasis is on constructing the first version of the voice-to-text application with emojis using an open-source speech transcript solution. As the project enters weeks 3 and 4, the focus changes to upgrading the app by including a cloud-based voice transcriber in version 2, allowing quicker loading and audio processing times.

From weeks 4 to 5, the team will test several methods for the voice transcript module, including using AI models to improve system performance. Following this inquiry, weeks 5 to 8 will be devoted to thorough software testing and report writing to record the development process and discoveries.

As the project nears completion, weeks 9 to 12 will be focused on FYP2 report revisions, software upgrades, and producing the FYP2 poster to highlight the project's accomplishments. During the last phase, from weeks 12 to 14, the team will test and develop the system regularly while preparing for the final oral presentation, ensuring the project is well-documented, functional, and well-articulated.

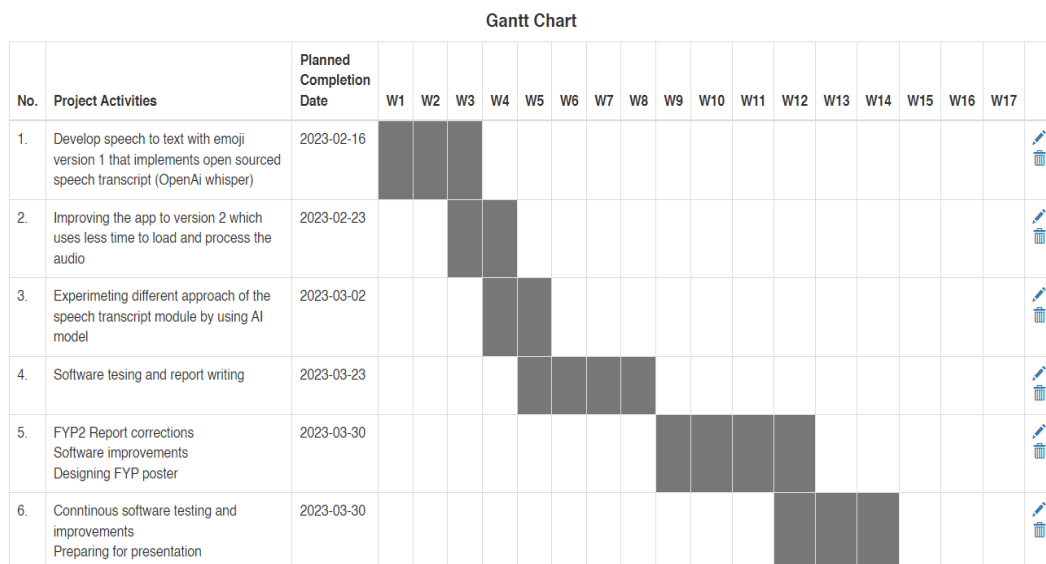


Figure 3.7: Project II Gantt Chart

CHAPTER 4

PROJECT SPECIFICATIONS

4.1 Introduction

This chapter specifies both functional and non-functional needs. In addition, use case illustrations and descriptions are shown.

4.2 Requirement Specification

4.2.1 Functional Requirements

Functional Requirements of the Speech to Text with emojis mobile app include:

1. The application shall allow users to utilize their mobile device's microphone for speech recording purposes.
2. The application shall allow users to upload pre-recorded speech.
3. The application shall allow users to view the accurately transcribe speech to text with emojis on the screen.
4. The application shall allow users to play their recorded speech or pre-recorded speech.
5. The application shall allow users to pause their recorded speech or pre-recorded speech once played.

4.2.2 Non-Functional Requirements

Non-Functional Requirements of the Speech to Text with emojis mobile app include:

1. The application shall have high accuracy and low latency in transcribing speech to text with emojis.
2. The application shall have a user-friendly and intuitive interface.
3. The application shall be able to handle a large volume of speech data without crashing or slowing down.
4. The application shall be secure and protect user data and privacy.
5. The application shall be reliable and available 24/7.
6. The application shall connect with phone microphone.
7. The application shall have access to phone external storage.

8. The application shall only be in English language.

4.3 Use Cases

4.3.1 Use Case diagram

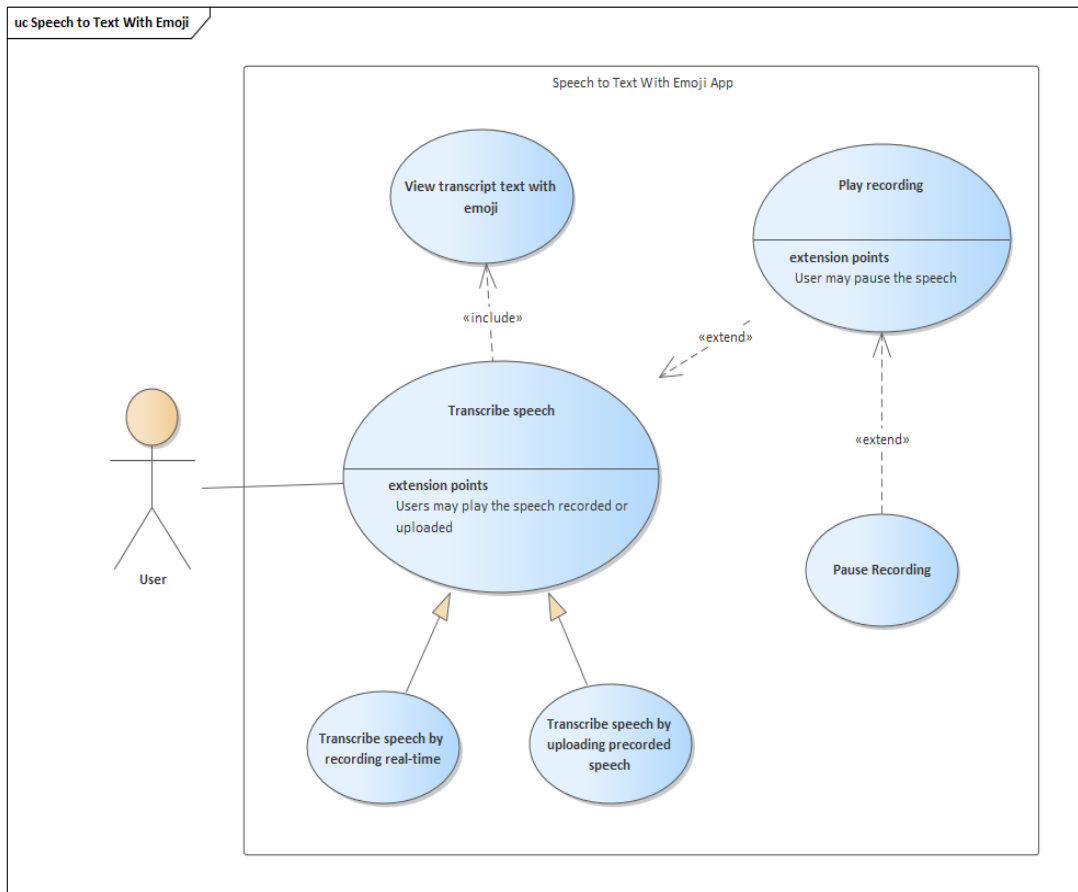


Figure 4.1: Use Case Diagram

4.3.2 Use Case Description

Table 4.1: Use Case Description of Transcribe Speech

Use Case Name: Transcribe Speech	ID: 1	Importance Level: High
Primary Actor: User	Use Case Type: Details, Essential	
Stakeholders and Interests: User- The end user of the mobile application aims to transcribe their own speech		
Brief Description: Transcribe speech use case describes how a user can transcribe their speech so it can be translated to text with emoji on screen.		
Trigger: User wants to transcribe their speech to translate it to text with emoji		
Relationships: Association: User Include: View transcript text with emoji Extend: Play Recording Generalisations: Transcribe speech by recording real-time and Transcribe speech by uploading pre-recorded speech.		
Normal Flow of Events: <ol style="list-style-type: none"> 1. User may choose to record their speech real-time or upload their pre-recorded speech from the system interface. 2. User chooses speech input. 3. The system transcribes the speech into text with emoji format. 		
SubFlows: -		
Alternate/ Exceptional Flows: <ol style="list-style-type: none"> 2.1 If user's mobile does not have a microphone or microphone is not working, display error message for microphone problem. 2.2 If user initially did not allow permission for the app to access the mobile's microphone, display error message to allow permission. 		

Table 4.2: Use Case Description of Transcribe Speech by recording real-time

Use Case Name: Transcribe Speech by recording real-time	ID: 2	Importance Level: High
Primary Actor: User	Use Case Type: Details, Essential	
<p>Stakeholders and Interests:</p> <p>User- The end user of the mobile application aims to transcribe their speech by capturing and transcribing it in real-time.</p>		
<p>Brief Description:</p> <p>Transcribe Speech by recording real-time use case describes how a user can transcribe their speech by recording their speech in real-time.</p>		
<p>Trigger: User wants to transcribe their speech to translate it to text with emoji by recording</p>		
<p>Relationships:</p> <p>Association: User</p> <p>Include: N/A</p> <p>Extend: N/A</p> <p>Generalisations: Transcribe Speech (parent)</p>		
<p>Normal Flow of Events:</p> <ol style="list-style-type: none"> 1. User chooses to transcribe their speech by recording real-time. 2. User holds on to the microphone button. 3. System prompts user to start speaking to the mobile phone's mic. 4. User releases the microphone button. 5. The system transcribes the speech into text with emoji format. 		
<p>SubFlows: -</p>		
<p>Alternate/ Exceptional Flows:</p> <p>2.1 If user's mobile does not have a microphone or microphone is not working, display error message for microphone problem.</p> <p>2.2 If user initially did not allow permission for the app to access the mobile's microphone, display error message to allow permission</p>		

Table 4.3: Use Case Description of Transcribe Speech by uploading pre-recorded speech

Use Case Name: Transcribe Speech by uploading pre-recorded speech	ID: 3	Importance Level: High
Primary Actor: User	Use Case Type: Details, Essential	
<p>Stakeholders and Interests:</p> <p>User- The end user of the mobile application aims to transcribe their speech by uploading their pre-recorded speech</p>		
<p>Brief Description:</p> <p>Transcribe Speech by uploading pre-recorded speech use case describes how a user can transcribe their speech by uploading their pre-recorded speech</p>		
<p>Trigger: User wants to transcribe their speech to translate it to text with emoji by uploading</p>		
<p>Relationships:</p> <p>Association: User</p> <p>Include: N/A</p> <p>Extend: N/A</p> <p>Generalisations: Transcribe Speech (parent)</p>		
<p>Normal Flow of Events:</p> <ol style="list-style-type: none"> 1. User chooses to transcribe their speech by uploading their pre-recorded speech. 2. User selects the upload button. 3. System prompts user to files directory. 4. User chooses the audio file to transcribe. 5. System prompts user back to main system interface. 6. The system transcribes the speech into text with emoji format. 		
<p>SubFlows: -</p>		
<p>Alternate/ Exceptional Flows:</p> <p>2.1 If user initially did not allow permission for the app to access the mobile's external storage, display error message to allow permission</p>		

Table 4.4: Use Case Description of View Transcript Text with Emoji

Use Case Name: View Transcript Text with Emoji	ID: 4	Importance Level: High
Primary Actor: User	Use Case Type: Details, Essential	
Stakeholders and Interests: User- The end user of the mobile application aims to transcribe their speech and view the transcript text with emoji on screen.		
Brief Description: View Transcript Text with Emoji use case describes how a user would like to transcribe their speech and view the transcript text with emoji.		
Trigger: User wants to see the transcript text with emoji		
Relationships: Association: User Include: N/A Extend: N/A		
Normal Flow of Events: <ol style="list-style-type: none"> 1. User may choose to record their speech real-time or upload their pre-recorded speech from the system interface. 2. User chooses speech input. 3. The system transcribes the speech into text with emoji format. 4. The system displays the transcript text with emoji on screen. 		
SubFlows: -		
Alternate/ Exceptional Flows: -		

Table 4.5: Use Case Description of Play Recording

Use Case Name: Play Recording	ID: 5	Importance Level: High
Primary Actor: User	Use Case Type: Details, Essential	
Stakeholders and Interests: User- The end user of the mobile application aims to play the speech they recorded or uploaded.		
Brief Description: View Transcript Text with Emoji use case describes how a user would like to play the speech they recorded or uploaded		
Trigger: User wants to play the recorded speech or uploaded speech		
Relationships: Association: User Include: N/A Extend: Pause Recording		
Normal Flow of Events: <ol style="list-style-type: none"> 1. User may choose to record their speech real-time or upload their pre-recorded speech from the system interface. 2. User chooses speech input. 3. The system transcribes the speech into text with emoji format. 4. The system displays the transcript text with emoji on screen. 5. User selects the play button. 6. The system shall play the speech recorded or uploaded. 		
SubFlows: -		
Alternate/ Exceptional Flows: -		

Table 4.6: Use Case Description of Pause Recording

Use Case Name: Pause Recording	ID: 6	Importance Level: High
Primary Actor: User	Use Case Type: Details, Essential	
Stakeholders and Interests: User- The end user of the mobile application aims to pause the speech they recorded or uploaded.		
Brief Description: View Transcript Text with Emoji use case describes how a user would like to pause the speech they recorded or uploaded		
Trigger: User wants to pause the recorded speech or uploaded speech		
Relationships: Association: User Include: N/A Extend: N/A		
Normal Flow of Events: <ol style="list-style-type: none"> 1. User may choose to record their speech real-time or upload their pre-recorded speech from the system interface. 2. User chooses speech input. 3. The system transcribes the speech into text with emoji format. 4. The system displays the transcript text with emoji on screen. 5. User selects the play button. 6. The system shall play the speech recorded or uploaded. 7. User selects the pause button. 8. The system shall pause the speech. 		
SubFlows: -		
Alternate/ Exceptional Flows: -		

CHAPTER 5

SYSTEM DESIGN

5.1 Introduction

This chapter includes the introduction of the architecture design of the system. Furthermore, the user interface design will also be shown.

5.2 System Architecture Design

A single-tier architecture, often known as a monolithic architecture, is a system design in which all an application's components and modules execute in the same environment, generally on the user's device. There is no separation between the client-side and server-side components in this design. All processing, data storage, and user interface rendering occur on the same system or device, allowing the programme to be self-contained and removing the need for external connectivity.

A single-tier architecture in the context of a mobile application implies operating all functions directly on the mobile device. This covers user interface rendering, user input processing, business logic execution, and data storage and retrieval management. A single-tier design has the benefit of simplifying development and reducing reliance on external services. This may result in shorter development cycles.

5.2.1 Backend flow design

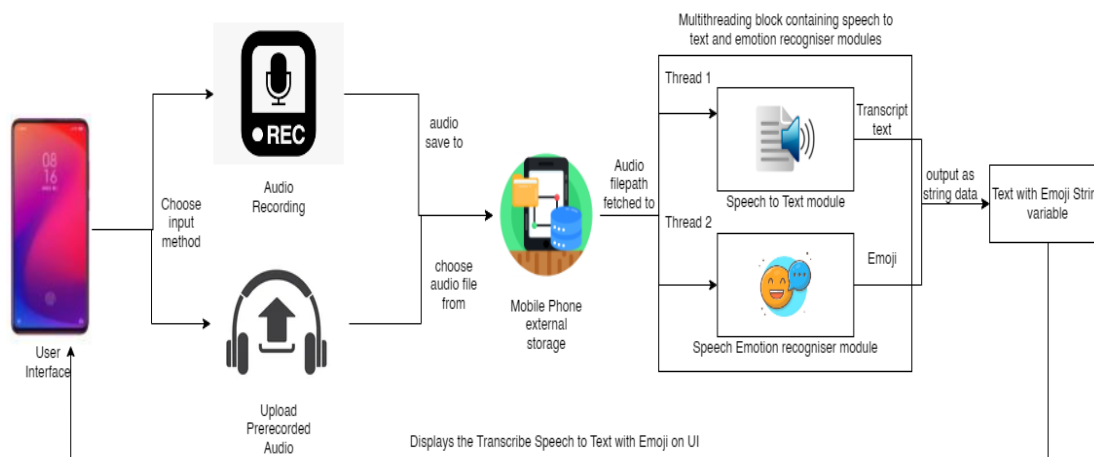


Figure 5.1: Single Tier Architecture design flow on Speech to Text with Emoji mobile app

Based on Figure 5.1 above depicts the data flow and interactions between multiple components on the user's device. The Speech to Text with Emoji mobile application's single-tier architecture allows all components to communicate inside the same environment on the user's mobile device. The procedure starts with the User Interface, where the user may either record their speech in real-time or upload their pre-recorded speech. If the user decides to record their speech in real-time, the user shall interact with the microphone button by holding it down to start and release the button to stop recording. When the recording stops, the Audio Recording component saves the recorded speech into the device's external storage and shares the file path with the Speech-to-Text and Emotion Recognition Module. In contrast, if the user chooses to upload their pre-recorded speech, the user shall interact with the upload button by selecting it. The user shall then choose the audio file to be transcribed from the phone's external storage and shares the file path with the Speech-to-Text and Emotion Recognition Module.

These two modules operate parallel, using multithreading to boost speed and responsiveness. The voice-to-Text Module uses a pre-trained deep learning model or a Python script to convert recorded voice to text. Simultaneously, the Emotion Recognition Module uses a similar method to analyse the speech to identify the corresponding emoji.

The Speech-to-Text and Emotion Recognition modules will wait for one another, once both finish their duties, they send their outputs as strings, the transcript text, and the emoji, to the Integration component. The Integration component mixes these outputs and display the result to the user through the User Interface.

5.3 User Interface design

As stated before, the app heavily focuses on backend development and there is not much going on in the front-end development. Only one screen is designed for this application. Figure 5.2 shows the UI design for the application before any function is selected. There are 2 available points of interaction, the play button is excluded as a point of interaction.

Points of interaction:

1. Microphone button (centre): User holds the button to start recording. Once the user releases the button, their speech shall be analysed and transcribe to text with emoji on screen.
2. Upload button (right): User selects the button, prompts to phone external storage. Once user chooses their desired audio file, it shall be analysed and transcribe to text with emoji on screen.
3. Play button (left): Cannot be select and disabled as there is nothing to be played.

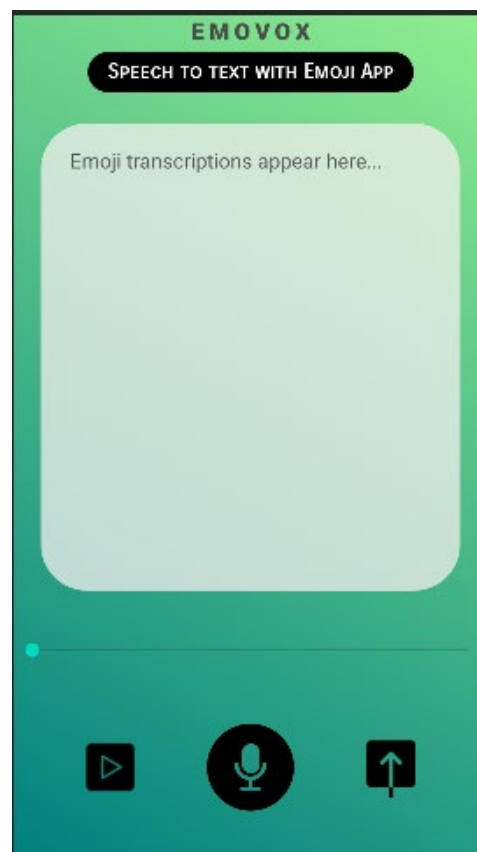


Figure 5.2: Speech to Text with Emoji app UI design before transcribing speech.

Figure 5.3 shows the UI design for the application after a speech is transcribed. There are 5 available points of interaction.

Points of interaction:

1. Microphone button (centre): User holds the button to start recording. Once the user releases the button, their speech shall be analysed and transcribe to text with emoji on screen. This shall rewrite the current text with emoji on screen.
2. Upload button (right): User selects the button, prompts to phone external storage. Once user chooses their desired audio file, it shall be analysed and transcribe to text with emoji on screen. This shall rewrite the current text with emoji on screen.
3. Play button (left): Available after speech has been transcribed. User selects the button and the transcribe speech is played.
4. Pause button (left after play button is selected): Available after speech is played. User selects the button and speech is paused.
5. Speech to text display block (middle centre): User may select it and change the text on it.

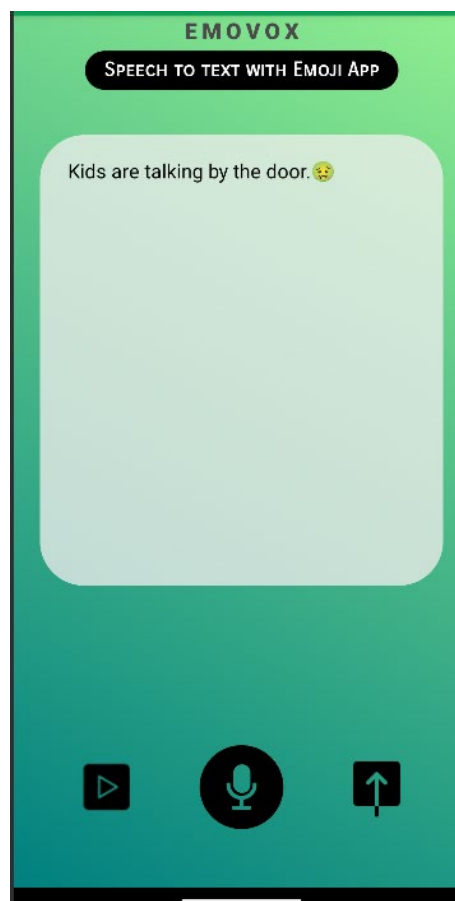


Figure 5.3: Speech to Text with Emoji app UI design after transcribing speech.

CHAPTER 6

IMPLEMENTATION

6.1 Introduction

This chapter includes the tools and technologies used to implement the methodology for the system. Furthermore, this chapter also further elaborate on the implementation of the backend development of speech-to-text and emotion recognition module. The front-end implementation will also be discussed. On top of that, different versions of the system that was developed by improvements from previous versions will also be shared.

6.2 Tools and Technologies

Throughout its development and execution, the Speech to Text with Emojis mobile application uses various tools and technologies.

Android Studio is the leading Integrated Development Environment (IDE) for developing and testing Android mobile applications. It provides a complete environment to develop the user interface, implement the application logic, and manage dependencies such as the Chaquopy framework and TensorFlow Lite.

The Chaquopy framework is used as an Android Studio plugin, allowing for the smooth integration of Python scripts into the Android application. This framework is conducive for running the Speech-to-Text and Emotion Recognition modules, which may depend on deep learning models written in Python.

Jupyter Notebook is a powerful tool for data exploration, model training, and assessment throughout development. It provides an interactive platform for working with Kaggle datasets, the primary source for data used in training and evaluating deep learning models for voice recognition and emotion analysis.

TensorFlow Lite is an alternative to Chaquopy for developing deep learning models in mobile applications. It enables the quick deployment of pre-trained models on Android smartphones, guaranteeing a seamless and responsive user experience during voice processing and emotion detection.

The programme uses multithreading technology to improve speed and responsiveness. The programme can successfully process data without stopping the

main thread responsible for handling user interactions and updating the User Interface by executing the Speech-to-Text and Emotion Recognition modules simultaneously on different threads.

Finally, Kaggle is critical in supplying the primary datasets required for training and verifying deep learning models. Kaggle allows building robust and accurate models for the Speech to Text with Emojis mobile application by providing a vast and diversified data collection.

6.3 Backend Development

This mobile application heavily focuses on the implementation and development of the backend development modules such as speech to text and emotion recognition modules. These modules run in the background to analyse speech to text with emoji.

6.3.1 Speech to Text module

Two methodologies of the speech to text module have been implemented, Automatic Speech Recognition (ASR) and Deep Learning based model. In which 4 tools from it have been tried to be implemented in the system such as Android inbuilt Google's real-time speech recognition, Facebook's Wav2Vec Hugging Face transformer, AssemblyAI cloud API and DeepSpeech.

6.3.1.1 Android inbuilt Google's real-time speech recognition implementation

Using Google's powerful voice recognition algorithms, Google's real-time speech recognition, powered by Android's built-in SpeechRecognizer API, delivers excellent accuracy and performance (Android Developers, 2019). These algorithms are built on deep learning models trained on massive quantities of data, enabling them to recognise spoken words and phrases with remarkable accuracy. Deep learning models analyse the speech and return the recognised text to the application. The approach entails continually recording and transferring the user's voice data to Google's servers for real-time processing. Figure 6.1 shows the UI of the Android's built-in SpeechRecognizer API.

There are, however, certain disadvantages to utilising Google's real-time voice recognition in the context of the voice to Text with Emoji smartphone application. One of the most significant restrictions is that the SpeechRecognizer API does not allow

direct access to the recorded audio data since the processing takes place on Google's servers. Writing a separate function for recording will not help as well as the API does not have a specific trigger when it has stop listening. The real-time speech recognition function must be integrated with the emotion identification module, which needs an audio file input. Consequently, developers may need to investigate other ways of capturing and processing voice to guarantee smooth integration with speech-to-text and emotion detection features.

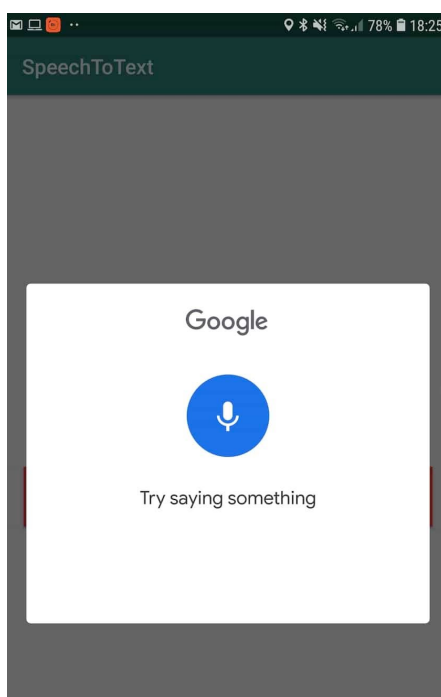


Figure 6.1: Google's real-time speech recognition, powered by Android's built-in SpeechRecognizer API

6.3.1.2 Wav2Vec implementation

To interpret user voice and create text, the voice to Text with Emoji mobile application uses Facebook's Wav2Vec library and the Hugging Face Transformers library (huggingface.co, n.d.). This solution is carried by using a Python script inside the Chaquopy framework, allowing for easy interaction with Android Studio.

The Wav2Vec algorithm implementation begins by loading the pre-trained Wav2Vec2 model and tokenizer using the configuration "facebook/wav2vec2-base-960h." When a user's voice is captured and stored as an audio file, the audio data is

loaded using the Librosa library, which ensures that the sample rate remains constant at 16,000 Hz. After that, the audio data is tokenized using the Wav2Vec2 tokenizer:

```
input_values = tokenizer(speech, return_tensors='pt').input_values
```

Figure 6.2: Wav2Vec implementation code snippet 1

The tokenized input values are sent into the Wav2Vec2 model, which analyses the audio data and produces logits:

```
logits = model(input_values).logits
```

Figure 6.3: Wav2Vec implementation code snippet 2

These logits are used to predict IDs by picking the token with the greatest probability:

```
#Store predicted id's
predicted_ids = torch.argmax(logits, dim = -1)
```

Figure 6.4: Wav2Vec implementation code snippet 3

Finally, the transcriptions are generated by decoding the expected IDs:

```
#decode the audio to generate text
transcriptions = tokenizer.decode(predicted_ids[0])
```

Figure 6.5: Wav2Vec implementation code snippet 4

Despite its efficiency, the Wav2Vec implementation has a few flaws. The large size of the pre-trained models is one of the primary problems, which might lead to increased app size and memory utilisation. Furthermore, executing the Wav2Vec model on-device may demand many computing resources, resulting in longer processing times and worse battery life.

6.3.1.3 AssemblyAI speech recognition API implementation

Although the specific architecture and training details are secret, the algorithm follows a broad framework comparable to other ASR systems. The AssemblyAI speech recognition API is implemented in the Speech to Text with Emoji smartphone

application as an alternate technique to transcribe the user's voice (AssemblyAI, n.d.). The AssemblyAI voice recognition API is implemented in the voice to Text with Emoji mobile application using several methods and constants that communicate with the API through HTTP requests. This solution is carried by using a Python script inside the Chaquopy framework, allowing for easy interaction with Android Studio. This implementation's primary methods are 'upload', 'transcribe', 'poll', 'get_transcription_result_url', and 'save_transcript'. Here's a thorough description of the procedure, along with code snippets:

1. Upload the audio file:

```
def upload(filename):
    def read_file(filename):
        with open(filename, 'rb') as f:
            while True:
                data = f.read(CHUNK_SIZE)
                if not data:
                    break
                yield data

    upload_response = requests.post(upload_endpoint, headers=headers_auth_only, data=read_file(filename))
    return upload_response.json()['upload url']
```

Figure 6.6: AssemblyAI code snippets 1

Based on Figure 6.6, the 'upload' method reads the audio file in chunks and uses a POST request to send it to AssemblyAI. For the uploaded audio, the API returns an 'upload_url'.

2. Transcribe the audio:

```
def transcribe(audio_url):
    transcript_request = {
        'audio_url': audio_url
    }

    transcript_response = requests.post(transcript_endpoint, json=transcript_request, headers=headers)
    return transcript_response.json()['id']
```

Figure 6.7: AssemblyAI code snippets 2

Based on Figure 6.7, the 'transcribe' function makes a transcription request to AssemblyAI through POST with the 'audio_url' parameter. The API produces a 'transcript_id' that may be used to monitor the transcription process.

3. Poll the API for transcription status:

```
def poll(transcript_id):
    polling_endpoint = transcript_endpoint + '/' + transcript_id
    polling_response = requests.get(polling_endpoint, headers=headers)
    return polling_response.json()
```

Figure 6.8: AssemblyAI code snippets 3

Based on Figure 6.8, by submitting a GET request to AssemblyAI with the 'transcript_id', the 'poll' function checks the transcription status. The API returns the transcription's current state.

4. Get the transcription result or error:

```
def get_transcription_result_url(url):
    transcribe_id = transcribe(url)
    while True:
        data = poll(transcribe_id)
        if data['status'] == 'completed':
            return data, None
        elif data['status'] == 'error':
            return data, data['error']
        print("processing audio")
```

Figure 6.9: AssemblyAI code snippets 4

Based on Figure 6.9, the 'get_transcription_result_url' method invokes the 'poll' function continuously until the transcription is finished or an error occurs. Depending on the conclusion, it returns either the transcription result or the error.

5. Save the transcript or handle errors:

```
def save_transcript(url):
    data, error = get_transcription_result_url(url)
    if data:
        return (data['text'])
        print('Transcript saved')
    elif error:
        return error
        print("Error!!!", error)
```

Figure 6.10: AssemblyAI code snippets 5

The 'save_transcript' method invokes the 'get_transcription_result_url' function and processes the result. It returns the transcribed text if it is successful; else, it returns the error message.

6. Main function to get the text from recorded audio:

```
def getTextFromRecordedAudio(filename):  
    audio_url = upload(filename)  
    text = save_transcript(audio_url)  
    return text
```

Figure 6.11: AssemblyAI code snippets 6

The 'getTextFromRecordedAudio' method connects the dots by first uploading the audio file with the 'upload' function and then storing the transcript with the 'save_transcript' function.

Despite the AssemblyAI voice recognition API's precision and efficiency, the implementation has several downsides, such as needing an active internet connection, depending on the API, and having possible latency or cost consequences.

6.3.1.4 DeepSpeech Algorithm Implementation

The TensorFlow Lite framework allows the DeepSpeech model to be implemented into an Android application (mozilla, 2019). To begin, the learned DeepSpeech model is transformed into a TensorFlow Lite model that can be utilised effectively on mobile devices. This is accomplished via the usage of the TensorFlow Lite converter. Once translated, the model may be incorporated into my Android application.

DeepSpeech is built on a deep learning architecture known as the Recurrent Neural Network (RNN), which is specially intended to handle sequential input such as audio signals. For regularisation, the model analyses the input audio characteristics via various layers of hidden units using activation functions such as ReLU (Rectified Linear Unit) and dropout. The RNN layer takes the processed information and models the audio signal's temporal relationships. The output is routed through further dense layers and a final linear layer after passing through the RNN layer, yielding logits for each time step and class. These logits are then used to forecast the most probable

characters at each time step, which are added together to form the final transcription. Figure 6.12 shows the DeepSpeech algorithm flow.

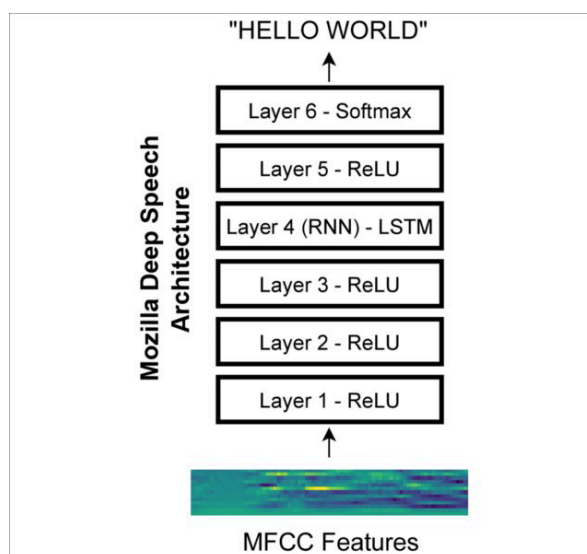


Figure 6.12: DeepSpeech Algorithm (Hashemnia et al.,2021)

Using the TensorFlow Lite framework, the framework's efficient inference capabilities is used to run the DeepSpeech model on Android devices with optimised speed and a small memory footprint. This enables mobile devices to do real-time speech-to-text conversion, allowing various applications such as voice assistants, transcription services, and more.

6.3.1.5 Conclusion

DeepSpeech is the greatest option for my particular use case because it combines open-source availability, on-device capability, and real-time processing capabilities. It can operate on various devices, from Raspberry Pi 4 to high-power GPU servers, as an offline, integrated speech-to-text engine, giving adaptability and flexibility. Furthermore, its real-time performance offers a consistent user experience in your application. Because it is open-source, I can customise and fine-tune the model as required, giving me greater control over the implementation while avoiding possible privacy problems connected with cloud-based services.

Due to time limits in this final year project, computing resource requirements, data requirements, and the skill required for effective implementation, creating, and training my own speech-to-text model may not be appropriate in for my instance.

Creating a bespoke model takes time and needs sophisticated hardware, a massive quantity of high-quality annotated voice data, and a thorough grasp of machine learning, natural language processing, and audio processing. Given my limited time to complete the project, using pre-trained, off-the-shelf models like DeepSpeech, Wav2Vec, or AssemblyAI may be more efficient and practical.

6.3.2 Emotion Recognition module

Two methodologies of the emotion recognition module have been implemented, Long Short-Term Memory (LSTM) Network Implementation and Multi-Layer Perceptron (MLP) Classifier Networks Implementation.

6.3.2.1 Long Short-Term Memory (LSTM) Network Implementation

The LSTM emotion recognition model was developed and trained before the prototype development of the application began. The LSTM was not implemented into the application as the overall accuracy results of the model could be better. The training results will be shown in Chapter 7. Hence, I researched and opted for another type of neural network with higher accuracy than LSTM.

6.3.2.2 Multi-Layer Perceptron (MLP) Classifier Network Implementation

With MLP Classifier having higher accuracy compared to LSTM, it is implemented in the application. The MLP Classifier emotion detection model is implemented in the Speech to Text with Emoji mobile application utilising the Chaquopy framework to run Python code on the Android platform. The Multilayer Perceptron (MLP) technique, a feedforward neural network, is used to train the model.

The trained model is stored as a serialised object and loaded during application execution using the pickle library. The voice signal is initially preprocessed in Python, utilising the librosa module to extract the MFCC features. The collected characteristics are then fed into the loaded MLP Classifier model, which predicts the associated emotion.

6.3.2.3 CNN-LSTM Network Implementation

With CNN-LSTM having higher accuracy compared to MLP Classifier, it is implemented in the application. The CNN-LSTM emotion detection model is

implemented in the Speech to Text with Emoji mobile application utilising the TensorFlow lite framework which directly executes the CNN-LSTM tflite model using tensor buffers. This network implementation showed promising results, and which will be shown in Chapter 7.

6.3.2.4 Conclusion

In conclusion, due to its lesser accuracy than the MLP Classifier, the LSTM emotion identification model was not used in the Speech to Text with Emoji mobile application; instead, the MLP Classifier was used to train the model and then deployed on the Android platform using the Chaquopy framework. The speech signal is preprocessed in Python, using the librosa package to extract MFCC characteristics and the MLP Classifier model predicting the related emotion. The trained model is serialised and loaded during programme execution through the pickle library.

6.3.3 Emoji Selection Implementation

The Speech to Text with Emoji mobile application picks the right emoji to complement the text after recognising the emotion of the speech. This is accomplished by matching the emotion label's name with the relevant ASCII code for the appropriate emoji. The code for the chosen emoji is then returned as a string mixed with the data from the transcribed text string. This procedure is carried out smoothly as part of the application's implementation to improve the user experience and give more context to the transcribed voice. Figure 6.13 below shows the list of emojis with its ASCII code and Figure 6.14 shows how the emoji selection is implemented in the program.

No	Code	Browser
1	U+1F600	
2	U+1F603	
3	U+1F604	
4	U+1F601	

Figure 6.13: List of emojis with its ASCII code (Full Emoji List, V12.0, 2019)

```

public String getEmoji(String audioPath) throws IOException {
    emotion = getEmotion(audioPath);

    if (emotion.equals("happy")) {
        emoji = getEmojiCode(0x1F604);
    } else if (emotion.equals("neutral")) {
        emoji = getEmojiCode(0x1F610);
    } else if (emotion.equals("calm")) {
        emoji = getEmojiCode(0x1F60C);
    } else if (emotion.equals("sad")) {
        emoji = getEmojiCode(0x1F62D);
    } else if (emotion.equals("angry")) {
        emoji = getEmojiCode(0x1F621);
    } else if (emotion.equals("fearful")) {
        emoji = getEmojiCode(0x1F623);
    } else if (emotion.equals("disgust")) {
        emoji = getEmojiCode(0x1F922);
    } else if (emotion.equals("surprised")) {
        emoji = getEmojiCode(0x1F62E);
    }

    return emoji;
}

```

Figure 6.14: Emoji Selection implementation in the program

6.4 Frontend Development

6.4.1 User Interface and Interaction Implementation

The Android XML layout files are used to implement the UI design. The Button view is used to implement the record and stop buttons. The TextView and ImageView views are used to implement the text and picture views, respectively. Figure 6.15 shows the side-to-side view of the XML codes and the UI design.

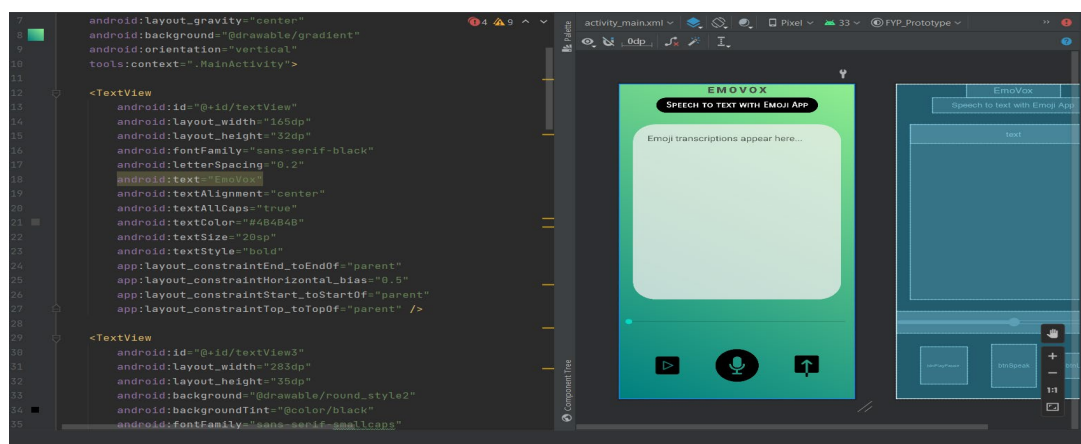


Figure 6.15: Side to side view of the XML codes and the UI design

Java code facilitates interaction between the app's UI and its backend. Figure 6.16 shows the interaction listeners for the user to interact with the app.

```
//add touch listener for audio recorder
btnSpeak.setOnTouchListener(new View.OnTouchListener() {...});

//Add click listener for play/pause button
btnPlayPause.setOnClickListener(new View.OnClickListener() {...});

//Add click listener for upload button
LISTENER FOR UPLOAD BUTTON
btnUpload.setOnClickListener(view -> {...});
```

Figure 6.16: Interaction listener code

6.4.2 Integration of Backend Modules

A multithreading technique is implemented to integrate the backend modules with the front end to allow smooth processing of the heavy task modules, preventing the application from crashing. On top of that, to let the recording, audio file saving and selection work, permission for the mobile phone microphone and external storage must be permitted by the user. This can be implemented by writing permission code in the Android Manifest file.

6.4.2.1 Multithreading

Multithreading is used in the Speech to Text with Emoji smartphone app to handle the heavy duties of speech-to-text and emotion identification in the background. Figure 6.17 depicts a multi-threaded programme design. It is made up of a single process with numerous processing processes. Each thread works separately and shares the same memory area, enabling activities to be executed in tandem. The illustration shows how multiple processes can access and modify the same data simultaneously, increasing efficiency and decreasing delay. However, multi-threaded computing requires accurate synchronisation to prevent data collisions and other concurrency problems. The image also demonstrates how the operating system plans and handles thread processing on available CPUs (Kissell, 2007).

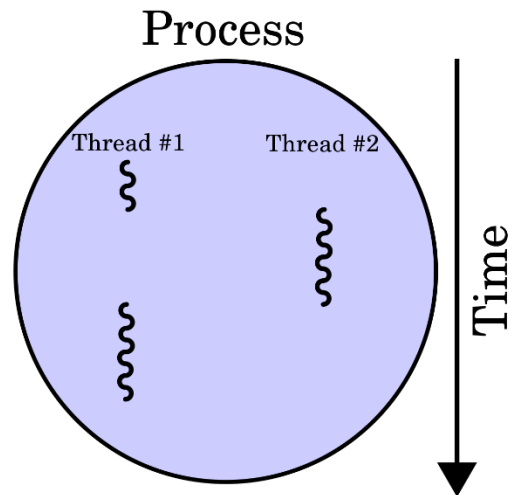


Figure 6.17: Multithreading architecture (Kissell, 2007)

Figure 6.18 shows the loading message that appears while the tasks are processing, informing the user that the programme is working on their voice input. This enables the operations to operate simultaneously, enhancing app speed and preventing freezing or crashing.

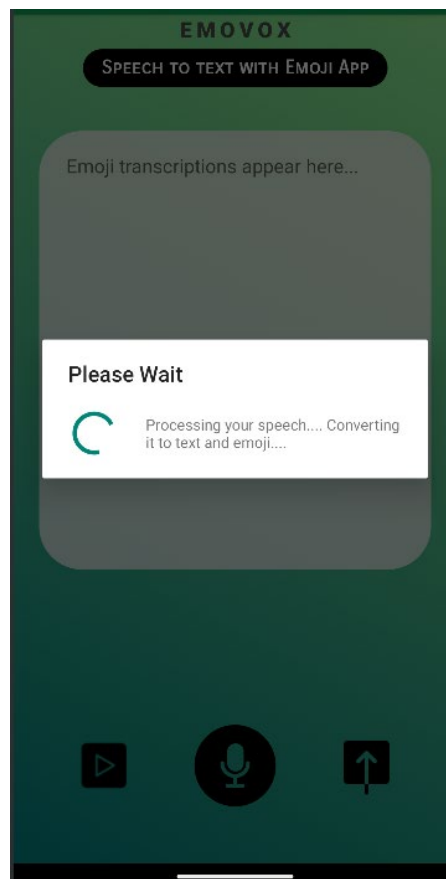


Figure 6.18: UI display when the backend modules are working

The AsyncTask class is used to implement multithreading in an Android app. According to Figure 6.19, a HeavyOperationsTask class extends AsyncTaskVoid, Void, String[]>. The three type arguments are doInBackground(), onProgressUpdate(), and onPostExecute(), in that order.

```
private class HeavyOperationsTask extends AsyncTask<Void, Void, String[]> {
```

Figure 6.19: AsyncTask class implementation

As show in Figure 6.20, the doInBackground() function does the hard lifting, in this instance invoking the getEmoji() and getTranscription() methods from the emotionRecogniser and speechTranscriptAPICall classes to detect emotion and transcribe speech, respectively.

```
@Override
protected String[] doInBackground(Void... voids) {
    String[] result = new String[2];
    try {
        result[0] = emotionRecogniser.getEmoji(filename);
    } catch (IOException e) {
        throw new RuntimeException(e);
    }
    result[1] = speechTranscriptAPICall.getTranscription(filename);
    return result;
}
```

Figure 6.20: doInBackground function

When doInBackground() provides a String[] array, it is offered to the onPostExecute() function, as seen in Figure 6.21. The transcript and emoji acquired from the String[] array are displayed in the front end using the TextView object text's setText() function. After then, the progress dialogue is closed.

```
@Override
protected void onPostExecute(String[] result) {
    String transcript = result[1];
    String emoji = result[0];
    text.setText(transcript + emoji);
    progressDialog.dismiss();
}
```

Figure 6.21: onPostExecute function

This approach guarantees that heavy processes execute in the background, preventing the UI from becoming unusable, while a progress dialogue displays to alert the user of continuing operations.

6.4.2.2 Permissions

Figure 6.22 depicts the permission section of the application's `AndroidManifest.xml` file. It provides the rights the application needs for users to engage with the front-end application. These rights are required for the user to interact with and utilise the front-end application's functionalities. The "RECORD_AUDIO" permission enables the programme to capture audio input from the microphone. The permission named "WRITE_EXTERNAL_STORAGE" enables the software to store information on external storage locations, such as an SD card of the emulator. The permission "MANAGE_EXTERNAL_STORAGE" enables the programme to handle access to external storage. The permission "READ_EXTERNAL_STORAGE" allows the programme to access data from external storage. The permission "STORAGE" is a combination of the permissions "WRITE_EXTERNAL_STORAGE" and "READ_EXTERNAL_STORAGE." Finally, the "INTERNET" permission enables the application to connect to the internet to perform API requests for speech-to-text and emotion detection. On top of that, Figure 6.23 shows the UI design when the system is asking for permission on accessing the external storage and microphone.

```
<uses-permission android:name="android.permission.RECORD_AUDIO"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.MANAGE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.STORAGE"/>
<uses-permission android:name="android.permission.INTERNET" />
```

Figure 6.22: Permission section for the mobile application

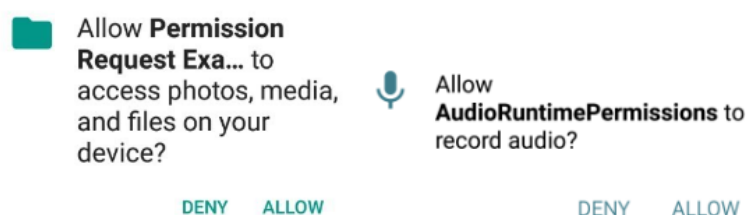


Figure 6.23: Asking permission UI design

6.5 Versioning and Iterative Improvements

Versioning and iterative improvements are critical components of software development because they enable developers to make incremental enhancements to a programme while ensuring that it fits the changing demands of consumers. In this scenario, the speech-to-text development of the emoji mobile application proceeded through three unique iterations, with each version building on the preceding version's triumphs and drawbacks.

6.5.1 Version 1: Initial Prototype

The early version of the app featured the fundamental capabilities of merging the voice-to-text and emotion-detection modules. The primary purpose of this version was to combine both modules to obtain a text with emoji; however, the programme only allowed pre-recorded audio and couldn't interface with Google's real-time speech-to-text service. The LSTM model was utilised in the emotion identification model, whereas the Wav2Vec model was employed in the speech-to-text model, which was quite sluggish.

6.5.2 Version 2: Additional feature and enhanced Speech to Text module

The app's version 2 was upgraded over the previous version since it included various new features and improved the speech-to-text module. The speech-to-text module was updated to utilise the Assembly AI API while emotion recognition module was updated to the MLP Classifier and was run using the Chaquopy framework, which lowered processing time and increased overall module performance. Additionally, multithreading was incorporated in the backend to boost speed, and two new features were added: play/pause the recording and real-time voice recording.

6.5.3 Version 3: Enhanced Emotion Prediction and new framework

Version 3 of the app was the most sophisticated, with an improved emotion prediction engine and a new architecture for the speech-to-text module. The emotion detection model was improved, and the speech-to-text module now uses a pre-trained profound speech model, DeepSpeech rather than the Assembly AI API. Moreover, the speech emotion recognition model was updated to CNN-LSTM to increase the module prediction accuracy. The app no longer used Chaquopy, instead using the TensorFlow

Lite framework to load and run AI models in Android Studio. Therefore, app storage and processing time were significantly reduced.

6.5.4 Conclusion

Finally, versioning and iterative changes were critical in developing a valuable and efficient voice-to-text emoji smartphone app. The final version, which included the pre-trained profound speech model, updated speech emotion recognition model and the TensorFlow Lite framework, proved to be the most successful in performance and accuracy.

CHAPTER 7

EVALUATION AND TESTING

7.1 Introduction

This chapter includes the test results on the speech to text models and Emotion recognition model performances. Furthermore, version comparison and selection will also be shown. Lastly, unit testing, integration testing and user testing will also be shown.

7.2 Speech to Text models performances

Wav2Vec Chaquopy, AssemblyAI API, and pre-trained model Tensorflow lite are among the three models tested.

A single 10 second audio file was utilised for each subchapter to evaluate the performance of the speech-to-text models. Each subchapter provides the evaluation findings and a complete analysis of each model's performance. The results are compared to the version of other models to decide which model works best in Table 7.1 and Figure 7.1. With lower processing time, it is decided that DeepSpeech is the best speech to text model.

Table 7.1: Comparison of Speech to text model performances

Model	Processing Time (s)	Accuracy (%)
Wav2Vec	30	99.5
AssemblyAI API	15	99.5
Pretrained DeepSpeech model	5	99.5

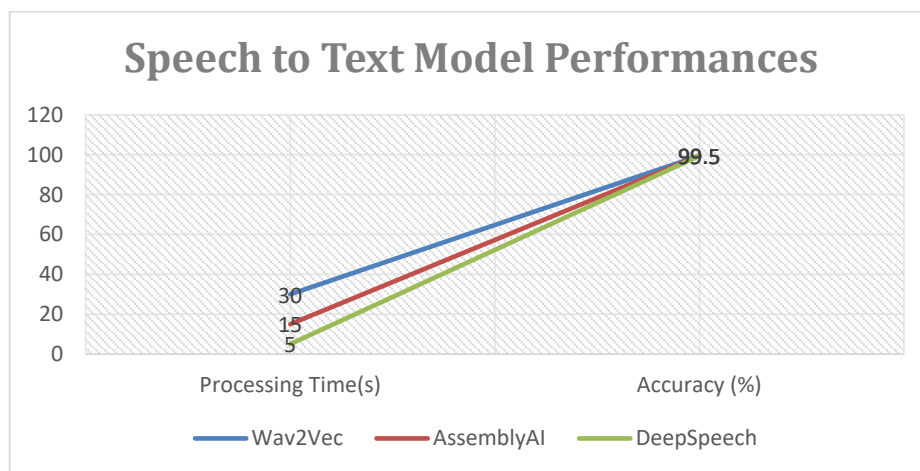


Figure 7.1: Comparison of Speech to text model performances graph

7.3 Emotion Features and Visualisation evaluation

Mel-frequency cepstral coefficients (MFCCs) are extracted from audio to serve as key features, allowing us to capture the spectral aspects of the signal, which is necessary for discriminating distinct emotions. Visualising these elements in spectrograms or waveforms will enable us to comprehend the data better and develop our approach to emotion detection by shedding light on the precise changes in the audio stream associated with each emotion. Furthermore, before feeding the data into the speech-to-text model, we perform preprocessing and feature engineering techniques such as normalisation and noise reduction. These procedures guarantee that the model concentrates on the most critical data and generalises well over a wide range of input signals. Figure 7.2 to 7.15 shows the 7 main emotions in spectrogram and waveforms that were fed into the deep learning model for training purposes.

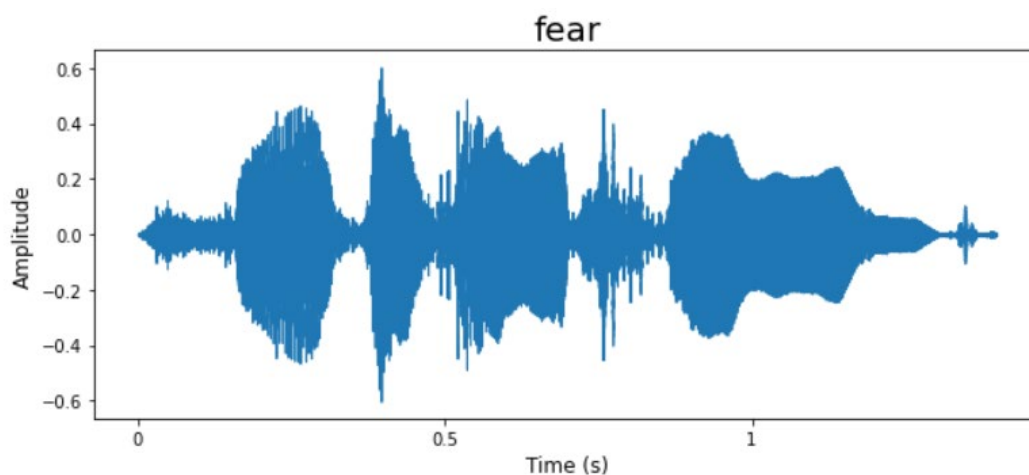


Figure 7.2: Fear speech data waveform

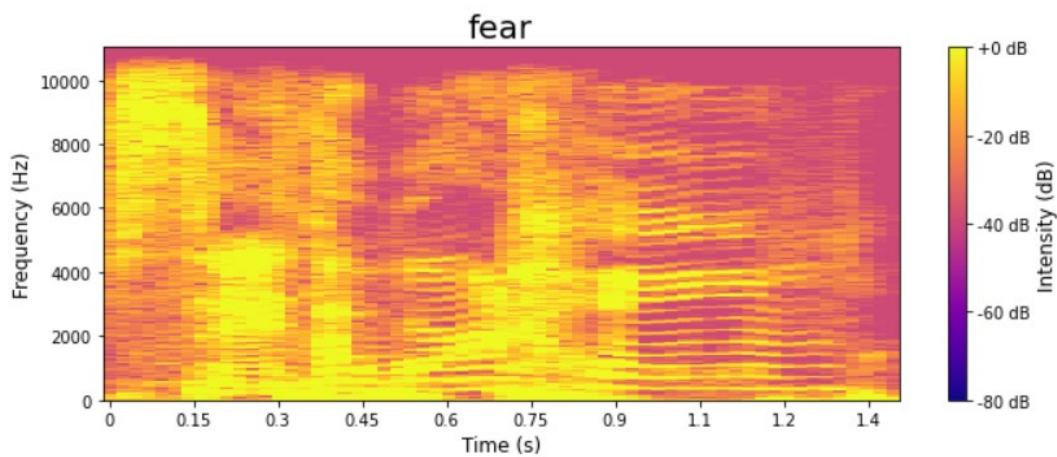


Figure 7.3: Fear speech data spectrogram

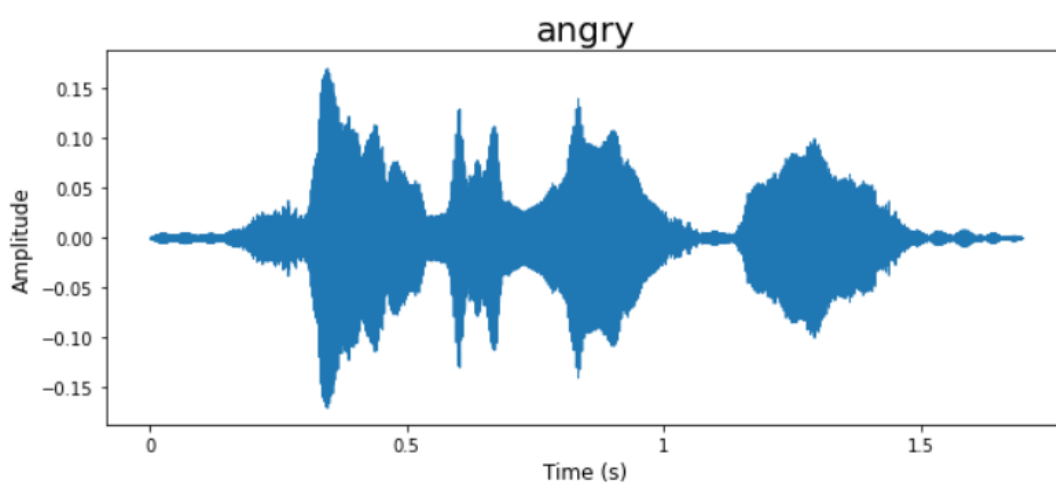


Figure 7.4: Angry speech data waveform

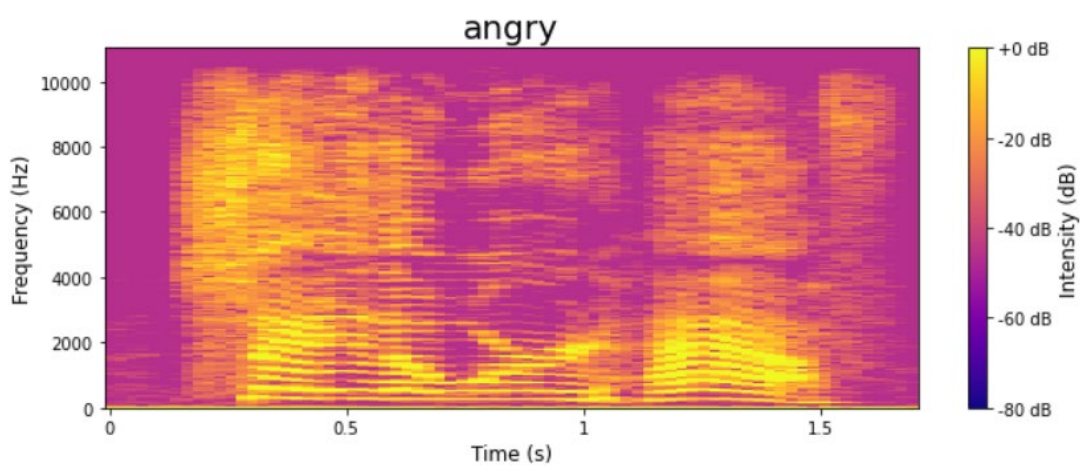


Figure 7.5: Angry speech data spectrogram

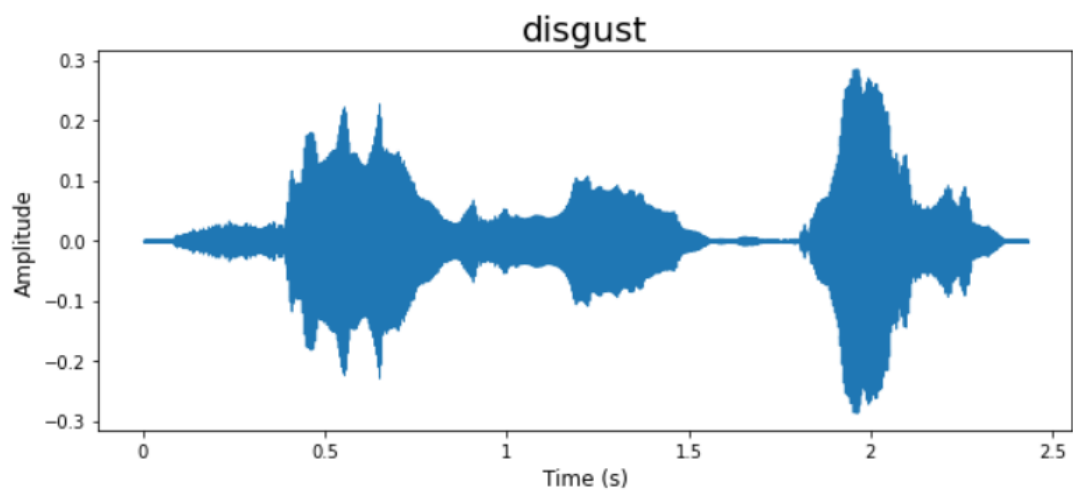


Figure 7.6: Disgust speech data waveform

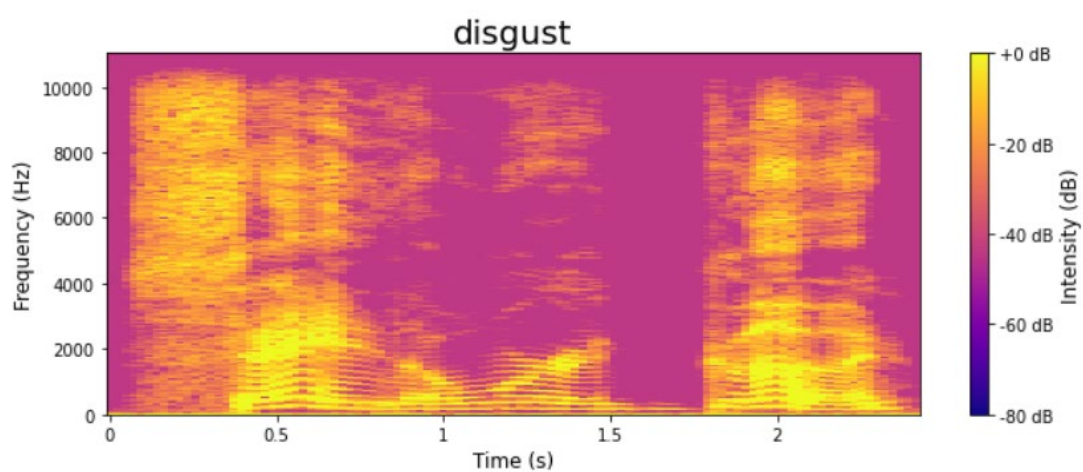


Figure 7.7: Disgust speech data spectrogram

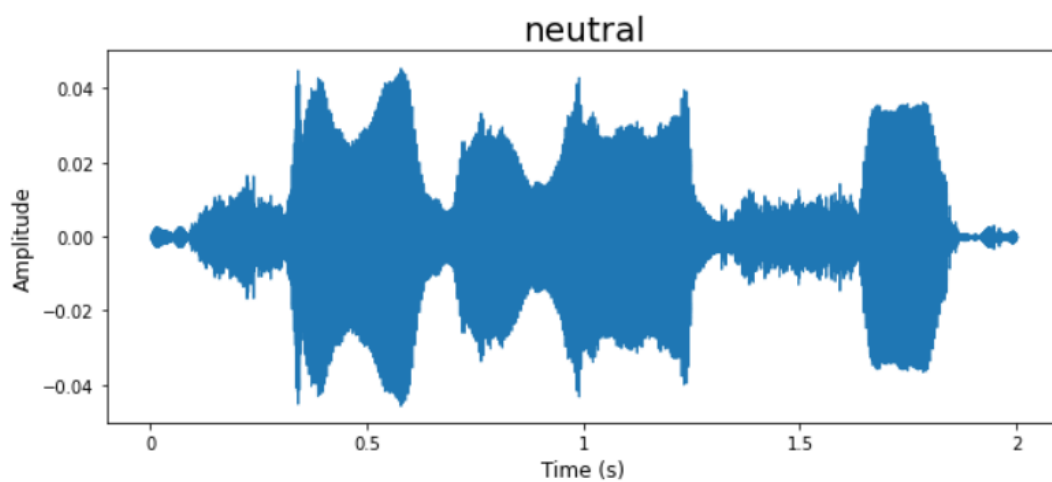


Figure 7.8: Neutral speech data waveform

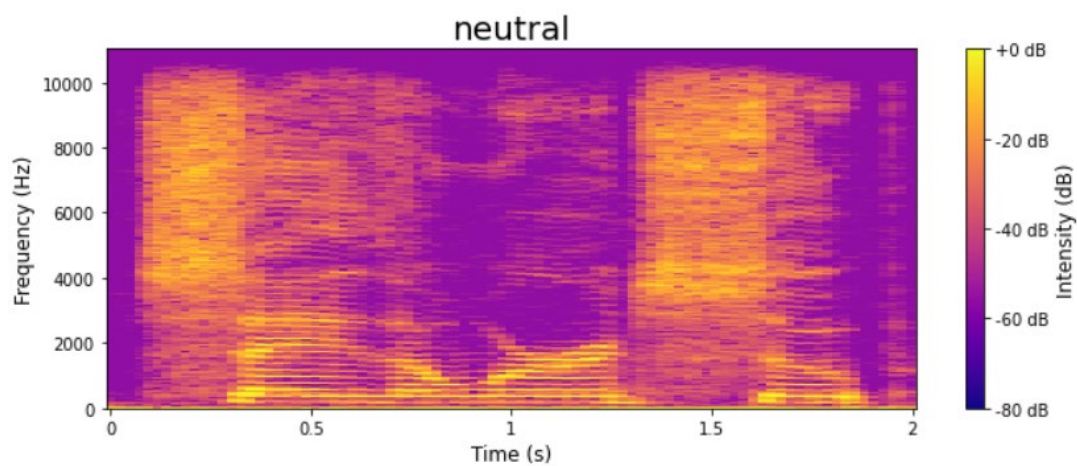


Figure 7.9: Neutral speech data spectrogram

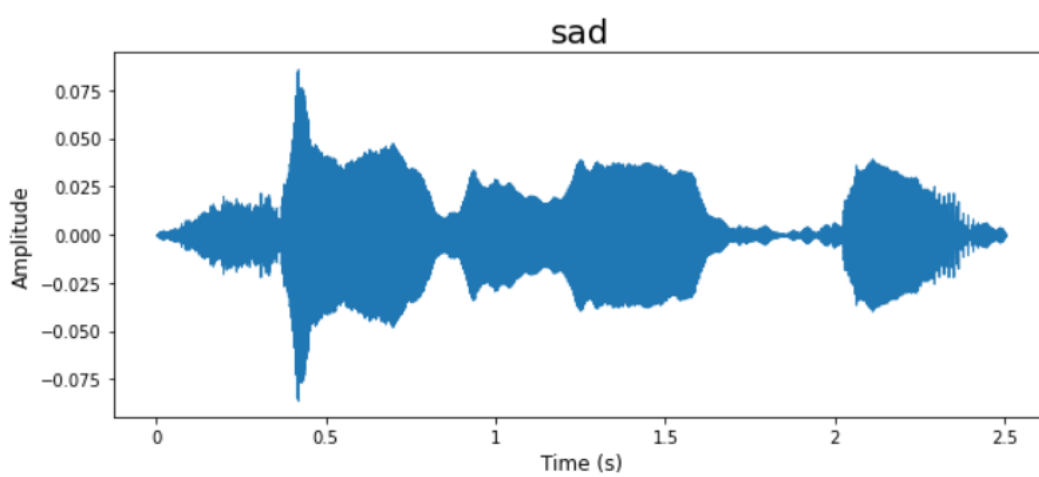


Figure 7.10: Sad speech data waveform

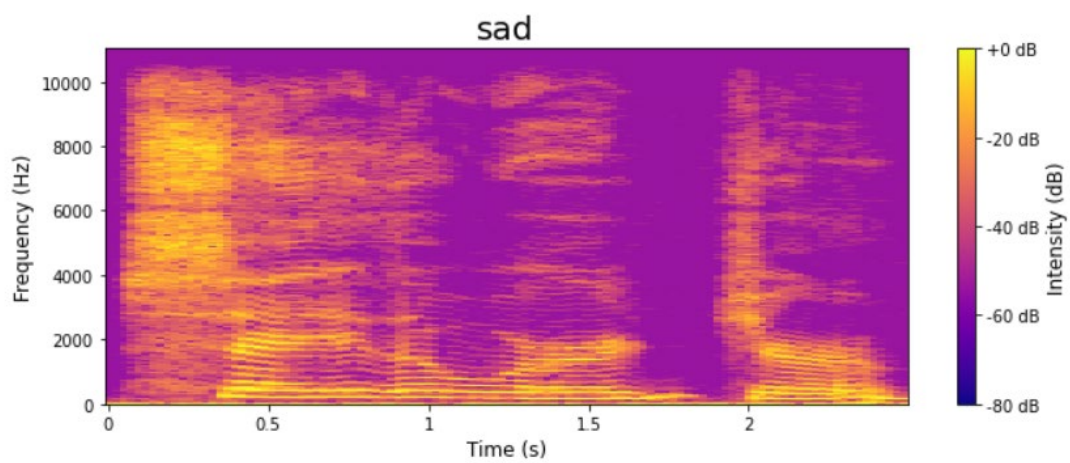


Figure 7.11: Sad speech data spectrogram

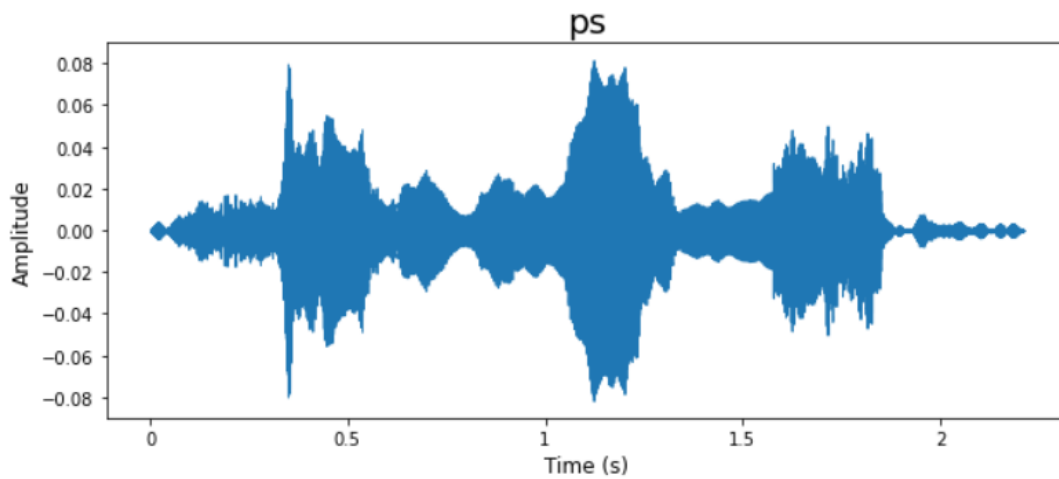


Figure 7.12: Pleasant surprise speech data waveform

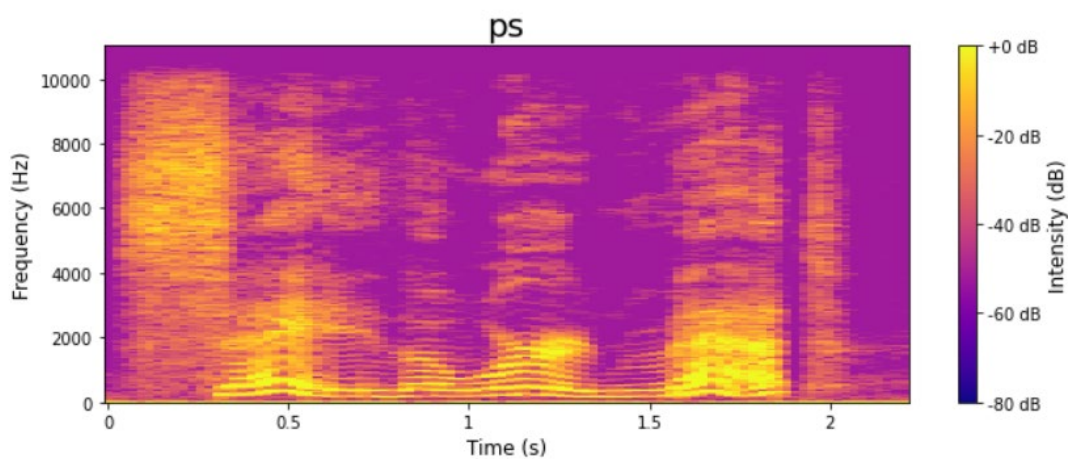


Figure 7.13: Pleasant surprise speech data spectrogram

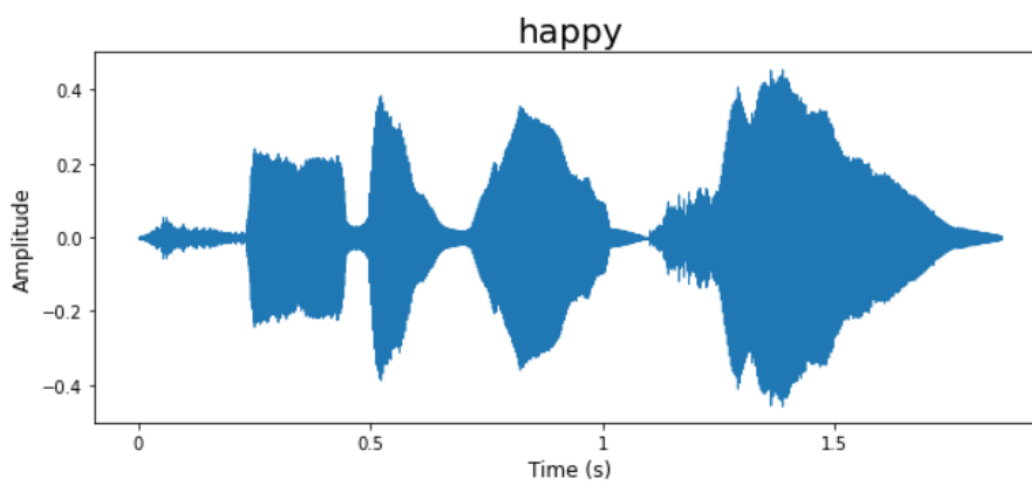


Figure 7.14: Happy speech data waveform

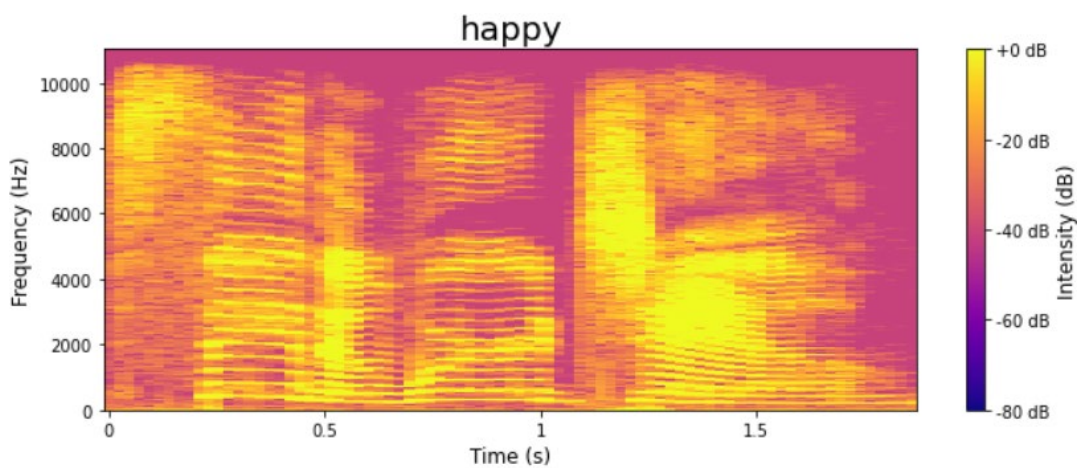


Figure 7.15: Happy speech data spectrogram

From the waveform diagram, we can observe the pitch, frequency, modulation, volume, speech rate and speech pattern of the speech spoken by the user. The variance of the volume and frequency can be seen when the wavelength height is longer. Usually, the higher the pitch, the louder the speech as well. Likewise, modulation, speech rate and speech pattern can be seen where their differences in the wavelength in the time spent talking.

To further analyse the speech data, a spectrogram is used to depict the frequency content of a voice signal across time, with various emotions exhibiting distinct characteristics in terms of pitch, intensity, and spectral shape. A higher pitch, corresponding to a higher fundamental frequency (F_0), is often connected with surprise, happiness, or fear emotions. In comparison, a lower pitch is associated with melancholy or tranquillity. The strength or energy of the speech signal may also convey emotion, with louder speech and greater intensity in the spectrogram frequently indicating emotions such as anger or surprise and softer speech and lower intensity indicating emotions such as sad or calm. Furthermore, emotions alter the spectral shape or distribution of energy across multiple frequencies, with anger or happiness possibly leading to a more spread-out energy distribution and sadness or calmness leading to a more concentrated energy distribution. Machine learning algorithms are typically used to effectively recognise emotions in speech signals by extracting features from spectrograms and identifying the intricate relationships between the spectral and temporal characteristics of the speech and the corresponding emotions.

7.4 Emotion Recognition models performance

Due to the short time, pre-trained models were used for the speech transcription module. In contrast, self-trained models were used for emotion recognition: LSTM, MLP Classifier and CNN-LSTM.

7.4.1 LSTM

As shown in Figure 7.16, the Long Short Term Memory Model is trained and tested to classify the emotion in the speech data set.

```

Model: "sequential"
-----
Layer (type)                Output Shape              Param #
-----
lstm (LSTM)                  (None, 256)              264192
-----
dropout (Dropout)           (None, 256)              0
-----
dense (Dense)                (None, 128)              32896
-----
dropout_1 (Dropout)         (None, 128)              0
-----
dense_1 (Dense)              (None, 64)               8256
-----
dropout_2 (Dropout)         (None, 64)               0
-----
dense_2 (Dense)              (None, 7)                455
-----
Total params: 305,799
Trainable params: 305,799
Non-trainable params: 0
-----

```

Figure 7.16: Summary of LSTM Model

7.4.1.1 Dataset used

The Toronto Emotional Speech Set (TESS), created by the University of Toronto, was used as the source of the speech data. To generate recordings of the set eliciting each of the seven emotions such as happy, anger, disgust, fear, sad, pleasant surprise, and neutral. Two actresses (aged 26 and 64) read a set of 200 target phrases in the carrier phrase "Say the word." There are 2800 data points in all (audio files). The information is organised so that each of the two female actresses and their emotions is included in a separate folder. That contains the audio file with 200 target words. WAV is the format used by the audio file. The code implementation and graph displaying the number of datasets for each mood are shown in Figure 7.17.

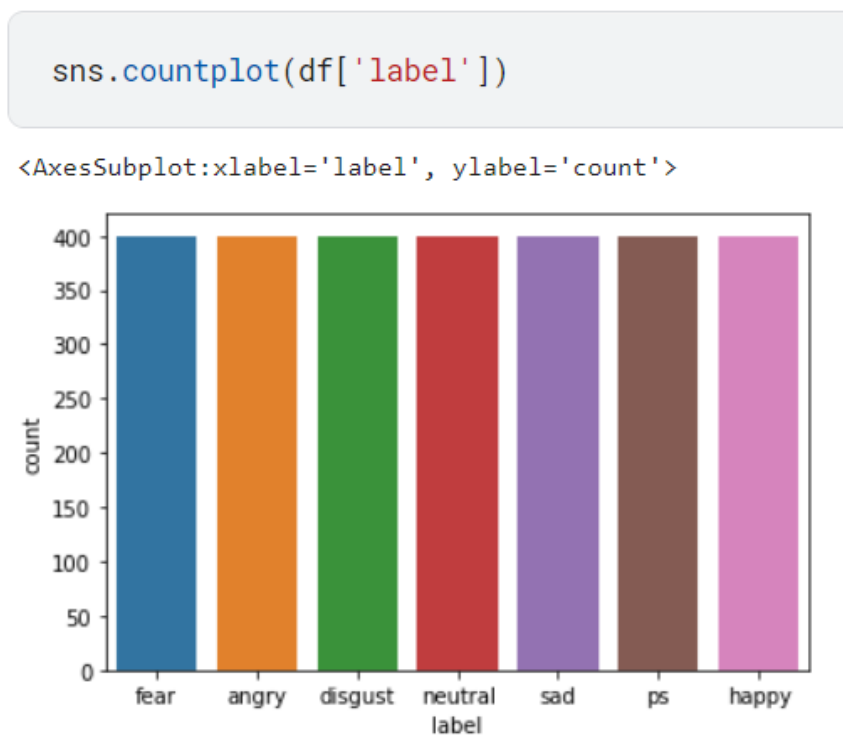


Figure 7.17: Number of speech dataset for each emotion

Next, below figures show each of the speech data wave form and spectrogram is printed out according to its emotion.

7.4.1.2 Results

Figures 7.19 and 7.20 show the training results from the LSTM Model. From the first iteration, the accuracy of the model increases as the number of trainings being done increases. However, the validation accuracy is low and inconsistent.

```
35/35 [=====] - 6s 109ms/step - loss: 1.0962 - accuracy: 0.6170 - val_loss: 3.0022 - val_accuracy: 0.0268
Epoch 2/50
35/35 [=====] - 3s 91ms/step - loss: 0.3840 - accuracy: 0.8629 - val_loss: 2.0082 - val_accuracy: 0.4446
Epoch 3/50
35/35 [=====] - 3s 91ms/step - loss: 0.2330 - accuracy: 0.9246 - val_loss: 0.9828 - val_accuracy: 0.6643
Epoch 4/50
35/35 [=====] - 3s 91ms/step - loss: 0.2002 - accuracy: 0.9339 - val_loss: 1.6514 - val_accuracy: 0.4196
Epoch 5/50
35/35 [=====] - 3s 93ms/step - loss: 0.1641 - accuracy: 0.9464 - val_loss: 2.3155 - val_accuracy: 0.3714
Epoch 6/50
35/35 [=====] - 3s 91ms/step - loss: 0.1127 - accuracy: 0.9661 - val_loss: 2.7867 - val_accuracy: 0.3464
Epoch 7/50
35/35 [=====] - 3s 90ms/step - loss: 0.1095 - accuracy: 0.9674 - val_loss: 2.5343 - val_accuracy: 0.4018
Epoch 8/50
35/35 [=====] - 3s 92ms/step - loss: 0.0866 - accuracy: 0.9719 - val_loss: 2.9856 - val_accuracy: 0.4250
Epoch 9/50
35/35 [=====] - 3s 90ms/step - loss: 0.0726 - accuracy: 0.9799 - val_loss: 2.6102 - val_accuracy: 0.5250
Epoch 10/50
35/35 [=====] - 3s 92ms/step - loss: 0.0750 - accuracy: 0.9781 - val_loss: 2.8661 - val_accuracy: 0.4000
Epoch 11/50
35/35 [=====] - 3s 91ms/step - loss: 0.0486 - accuracy: 0.9862 - val_loss: 1.6409 - val_accuracy: 0.6536
Epoch 12/50
35/35 [=====] - 4s 117ms/step - loss: 0.0477 - accuracy: 0.9871 - val_loss: 2.9480 - val_accuracy: 0.3893
Epoch 13/50
35/35 [=====] - 3s 91ms/step - loss: 0.0792 - accuracy: 0.9746 - val_loss: 3.3511 - val_accuracy: 0.3554
Epoch 14/50
35/35 [=====] - 3s 91ms/step - loss: 0.0719 - accuracy: 0.9777 - val_loss: 2.1031 - val_accuracy: 0.5375
Epoch 15/50
35/35 [=====] - 3s 94ms/step - loss: 0.0377 - accuracy: 0.9862 - val_loss: 2.1344 - val_accuracy: 0.4500
Epoch 16/50
```

Figure 7.18: Training results of LSTM Model part 1

```

Epoch 39/50
35/35 [=====] - 3s 91ms/step - loss: 0.0048 - accuracy: 0.9991 - val_loss: 4.3342 - val_accuracy: 0.4536
Epoch 40/50
35/35 [=====] - 3s 91ms/step - loss: 0.0039 - accuracy: 0.9996 - val_loss: 4.4968 - val_accuracy: 0.4643
Epoch 41/50
35/35 [=====] - 3s 92ms/step - loss: 0.0092 - accuracy: 0.9982 - val_loss: 5.3735 - val_accuracy: 0.4000
Epoch 42/50
35/35 [=====] - 3s 92ms/step - loss: 0.0125 - accuracy: 0.9969 - val_loss: 5.0857 - val_accuracy: 0.4304
Epoch 43/50
35/35 [=====] - 3s 98ms/step - loss: 0.0144 - accuracy: 0.9964 - val_loss: 5.7042 - val_accuracy: 0.3411
Epoch 44/50
35/35 [=====] - 4s 103ms/step - loss: 0.0550 - accuracy: 0.9857 - val_loss: 4.2149 - val_accuracy: 0.3143
Epoch 45/50
35/35 [=====] - 4s 103ms/step - loss: 0.0301 - accuracy: 0.9920 - val_loss: 3.0328 - val_accuracy: 0.5464
Epoch 46/50
35/35 [=====] - 4s 115ms/step - loss: 0.0311 - accuracy: 0.9897 - val_loss: 2.4058 - val_accuracy: 0.5125
Epoch 47/50
35/35 [=====] - 3s 91ms/step - loss: 0.0075 - accuracy: 0.9973 - val_loss: 3.3383 - val_accuracy: 0.5018
Epoch 48/50
35/35 [=====] - 3s 94ms/step - loss: 0.0370 - accuracy: 0.9920 - val_loss: 4.6324 - val_accuracy: 0.3482
Epoch 49/50
35/35 [=====] - 3s 91ms/step - loss: 0.0098 - accuracy: 0.9978 - val_loss: 2.9940 - val_accuracy: 0.4607
Epoch 50/50
35/35 [=====] - 3s 93ms/step - loss: 0.0027 - accuracy: 0.9991 - val_loss: 3.2176 - val_accuracy: 0.5179

```

Figure 7.19: Training results of LSTM Model part 2

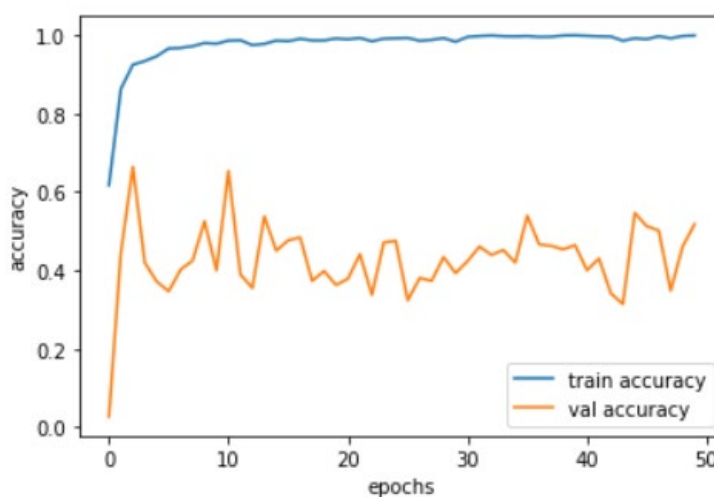


Figure 7.20: Relationship between train and validation accuracy of LSTM model

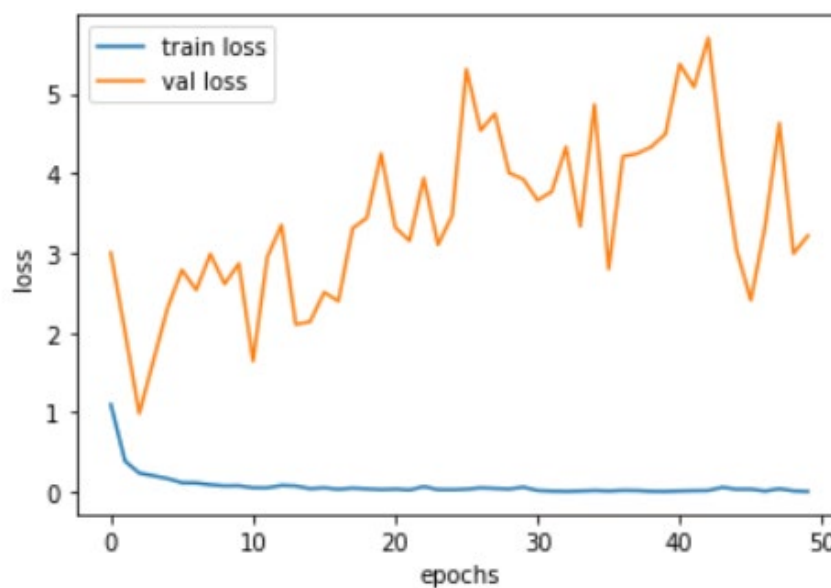


Figure 7.21: Relationship between the training loss and validation loss of LSTM model

According to Figure 7.21, the training accuracy increases as the epoch increases. In contrast, validation accuracy shows a low value of accuracy in an inconsistent rate as the epoch increases. According to Figure 7.22, the training loss decreases as the epoch increases. In contrast, the validation loss increases over the span of time at an inconsistent rate. This shows that the model is overfitting and not performing well. Hence, the model will not be able to classify or predict the data properly. Hence, the model needs more tuning and improvement to increase the validation accuracy to a more consistent rate.

7.4.2 MLP Classifier

As shown in Figure 7.23, the MLP Classifier Model is trained and tested to classify the emotion in the speech data set.

```
model=MLPClassifier  
(alpha=0.01, batch_size=256, epsilon=1e-08, hidden_layer_sizes=(300,),  
learning_rate='adaptive', max_iter=500)  
model.fit(x_train,y_train)
```

Figure 7.22: MLP Classifier architecture code implementation

7.4.2.1 Dataset used

The RAVDESS Emotional Speech audio dataset is used to train the MLP Classifier model. The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) is a collection developed to study emotion detection through speech. This section of the RAVDESS includes 1440 files: 60 attempts per character multiplied by 24 players equals 1440. The RAVDESS cast consists of 24 experienced performers (12 female, 12 male) who perform two lexically matched lines in a neutral North American dialect. Expressions of speech feelings include peaceful, joyful, sorrowful, furious, afraid, astonishment, and disgust. Each expression has two degrees of emotional strength (average and intense), as well as an indifferent expression.

Every one of the 1440 files has a distinct filename. The filename is a 7-part numerical identification (for example, 03-01-06-01-02-01-12.wav). These identifiers identify the qualities of the stimulus:

File naming conventions:

1. Media Type (01 = full-AV, 02 = video-only, 03 = audio-only).
2. Vocal Mode (01 = spoken, 02 = sung).
3. Emotion Category (01 = neutral, 02 = calm, 03 = happy, 04 = sad, 05 = angry, 06 = fearful, 07 = disgust, 08 = surprised).
4. Emotional Intensity (01 = regular, 02 = heightened). NOTE: 'Neutral' emotion has no heightened intensity.
5. Sentence (01 = "Kids are talking by the door", 02 = "Dogs are sitting by the door").
6. Iteration (01 = 1st attempt, 02 = 2nd attempt).
7. Actor (01 to 24. Odd-numbered actors are male, even-numbered actors are female).

File name example: 03-01-06-01-02-01-12.wav

1. Audio-only format (03)
2. Spoken mode (01)
3. Fearful emotion (06)
4. Regular intensity (01)
5. "Dogs" sentence (02)
6. 1st attempt (01)
7. 12th actor (12)
8. Female, since the actor ID is an even number.

7.4.2.2 Results

```

0.6966
Epoch 95/100
95/100 [=====] - 4s 102ms/step - loss: 0.0684 - accuracy: 0.7279 - val_loss: 0.1180 - val_accuracy:
0.7040
Epoch 96/100
96/100 [=====] - 4s 106ms/step - loss: 0.0143 - accuracy: 0.7229 - val_loss: 0.0336 - val_accuracy:
0.7054
Epoch 97/100
97/100 [=====] - 4s 102ms/step - loss: 0.0923 - accuracy: 0.7320 - val_loss: -0.0305 - val_accuracy:
0.6994
Epoch 98/100
98/100 [=====] - 4s 101ms/step - loss: 0.0611 - accuracy: 0.7322 - val_loss: 0.1211 - val_accuracy:
0.7049
Epoch 99/100
99/100 [=====] - 4s 101ms/step - loss: 0.0849 - accuracy: 0.7312 - val_loss: 0.0459 - val_accuracy:
0.7047
Epoch 100/100
100/100 [=====] - 4s 116ms/step - loss: 0.0743 - accuracy: 0.7303 - val_loss: 0.1118 - val_accuracy:
0.7130

```

Figure 7.23: Last 5 training/validation results epoch of MLP model

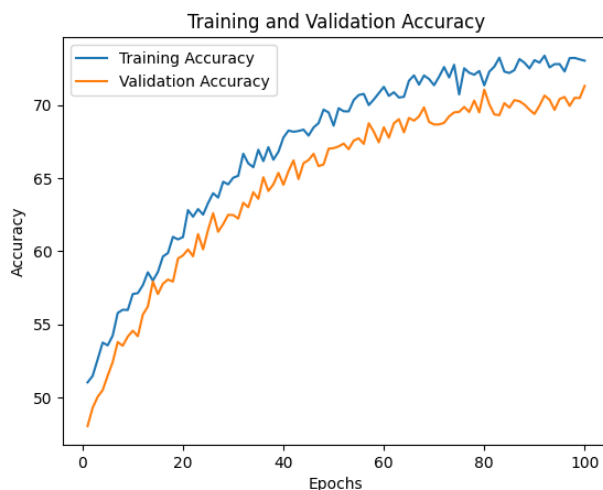


Figure 7.24: Relationship between train and validation accuracy of MLP model

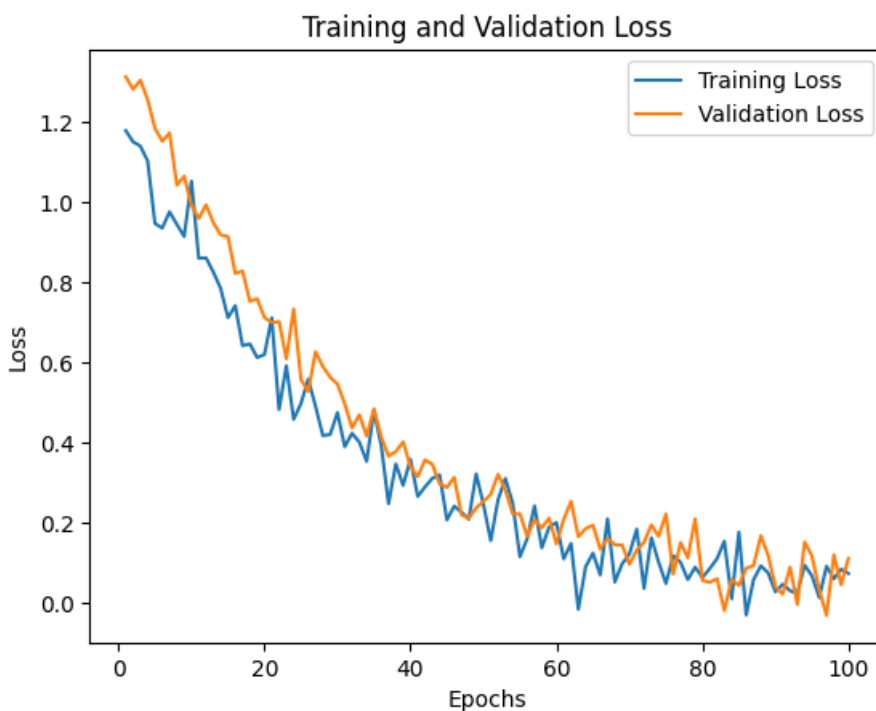


Figure 7.25: Relationship between the training loss and validation loss of MLP model

The MLP (Multilayer Perceptron) classification model adds to the 71.3% accuracy by capturing the complicated connections in the data using a feedforward artificial neural network with several layers. It can adapt to various classification issues because of its capacity to learn non-linear patterns and the appropriate activation functions in hidden layers. According to Figures 7.24, 7.25, and 7.26, the continuous

improvement in training and validation accuracies throughout 100 epochs and the tiny gap between them suggest that the model generalised well to the validation data. Due to its successful learning of the underlying patterns in the training data without overfitting or underfitting, the model attained a final validation accuracy of 71.3%. Despite that, the model was primarily confused about the three emotions: surprised, fearful and happy, as shown in Table 7.2. Given additional data, the model would recognise some emotions accurately but may risk failing when recognising other emotions. To enhance the model further, complexity can be increased as regularisation techniques, additional training data or data augmentation approaches, and fine-tuning hyperparameters using grid or random search techniques. Moreover, consideration in changing the model architecture to CNN-LSTM may be considered.

Table 7.2: Confusion matrix of the emotion recognised

Expected emotion	Actual emotion
Surprised	Sad
Fearful	Angry
Happy	Surprise

7.4.3 CNN-LSTM Model

As shown in Figure 7.27, the Convolutional Neural Networks - Long Short-Term Memory Model (CNN-LSTM) is trained and tested to classify the emotion in the speech data set.

Model: "sequential"		
Layer (type)	Output Shape	Param #
time_distributed (TimeDistributed)	(None, 22, 13, 128)	512
time_distributed_1 (TimeDistributed)	(None, 22, 13, 128)	512
time_distributed_5 (TimeDistributed)	(None, 22, 6, 256)	98560
time_distributed_6 (TimeDistributed)	(None, 22, 6, 256)	1024
time_distributed_10 (TimeDistributed)	(None, 22, 3, 512)	393728
time_distributed_11 (TimeDistributed)	(None, 22, 3, 512)	2048
bidirectional (Bidirectional)	(None, 22, 1024)	4198400
dropout_3 (Dropout)	(None, 22, 1024)	0
bidirectional_1 (Bidirectional)	(None, 22, 512)	2623488
dropout_4 (Dropout)	(None, 22, 512)	0
bidirectional_2 (Bidirectional)	(None, 256)	656384
dropout_5 (Dropout)	(None, 256)	0
dense (Dense)	(None, 6)	1542
Total params: 7,976,198		
Trainable params: 7,974,406		
Non-trainable params: 1,792		

Figure 7.26: CNN-LSTM model architecture summary

7.4.3.1 Dataset used.

The combination of three dataset were used two from the previous dataset: TESS and RAVDESS dataset and a new dataset CREMA-D dataset. The information about the TESS and RAVDESS dataset can be referred to on sub chapter, 7.4.1.1 and 7.4.2.1 respectively.

The CREMA-D dataset is a compilation of data collection of 7,442 original footage from 91 performers prepared by David Cooper Cheyney, an Assistant Professor at the West Chester University of Pennsylvania. These audios include 48 male and 43 female actors ranging in age from 20 to 74 and representing a variety of races and ethnicities (African American, Asian, Caucasian, Hispanic, and Unspecified). The actors read from a list of 12 phrases. The phrases were delivered with one of six possible emotions (Anger, Disgust, Fear, Happy, Neutral, or Sad) and four different emotion levels (Low, Medium, High, or Unspecified).

7.4.3.2 Results

```

cy: 0.7904
Epoch 95/100
100/100 [=====] - 11.8s 118ms/step - loss: 0.0402 - accuracy: 0.8413 - val_loss: 0.0858 - val_accuracy: 0.7978
Epoch 96/100
100/100 [=====] - 11.6s 116ms/step - loss: 0.0132 - accuracy: 0.8362 - val_loss: 0.0023 - val_accuracy: 0.7991
Epoch 97/100
100/100 [=====] - 10.4s 104ms/step - loss: 0.0656 - accuracy: 0.8453 - val_loss: 0.0609 - val_accuracy: 0.7929
Epoch 98/100
100/100 [=====] - 12.8s 128ms/step - loss: 0.0352 - accuracy: 0.8454 - val_loss: 0.0915 - val_accuracy: 0.7984
Epoch 99/100
100/100 [=====] - 10.3s 103ms/step - loss: 0.0597 - accuracy: 0.8443 - val_loss: 0.0171 - val_accuracy: 0.7981
Epoch 100/100
100/100 [=====] - 10.9s 109ms/step - loss: 0.0498 - accuracy: 0.8433 - val_loss: 0.0839 - val_accuracy: 0.8030

```

Figure 7.27: Last 5 training/validation results epoch of CNN-LSTM model

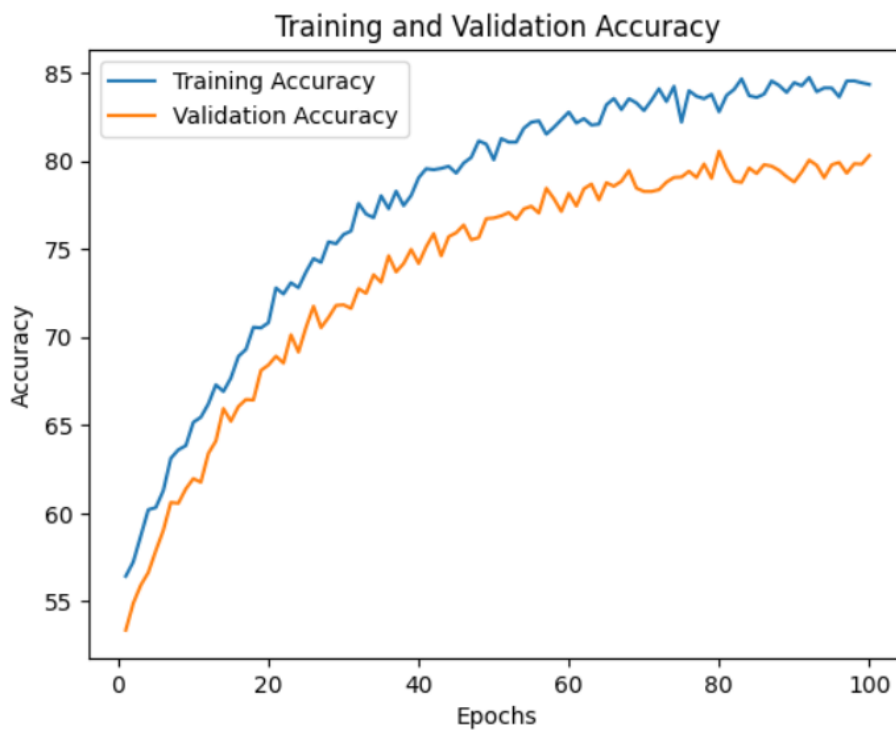


Figure 7.28: Relationship between train and validation accuracy of CNN-LSTM model



Figure 7.29: Relationship between the training loss and validation loss of CNN-LSTM model

The CNN-LSTM model outperforms the prior MLP classifier by 80.3%, most likely because it successfully integrates the capabilities of both convolutional and recurrent neural networks. CNN layers are good at finding local patterns and characteristics in input data, but LSTM layers are good at capturing long-term dependencies and temporal interactions. This combination enables the machine to learn more effectively complicated representations that generalise to previously unknown data. The connection between training and validation accuracy/loss shows that the model is converging and that the difference between the two is tolerable, suggesting a decent balance between underfitting and overfitting. This indicates that the model is learning patterns from the training data without overfitting to noise, increasing its chances of doing well on future data. When fresh data is supplied to the model, it is projected to generate accurate predictions, with an accuracy rate of roughly 80.3%. In the future, strategies like hyperparameter tweaking, data augmentation, extra features, testing other model architectures, or employing pre-trained embeddings might be investigated to enhance the model further.

7.4.4 Conclusion

In comparison with the results, the CNN-LSTM model surpasses the others with an accuracy of 80.3%. The CNN-LSTM model efficiently combines the capabilities of convolutional and recurrent neural networks, enabling it to learn complicated representations that generalise well to new input. The training and validation accuracies and losses show that the model effectively balanced between underfitting and overfitting, and it is likely to make accurate predictions when provided with fresh data. Although the MLP classifier performs well, it may struggle with some emotions and benefit from additional refinement. Overall, the results indicate that the CNN-LSTM model is a promising strategy for emotion identification tasks that might be improved further by experimenting with hyperparameter tweaking, data augmentation, and integrating new features or pre-trained embeddings.

7.5 Version Comparison and Selection

Table 7.3 below provides a comparison of the three app versions, summarizing the features, improvements, limitations, and drawbacks of each version. Hence, Version 3 would be chosen as the final version and will be used for user testing and deployment.

Table 7.3: Comparison on the different versions of the app

Version	Features & Improvements	Limitations & Drawbacks
1: Initial Prototype	<ul style="list-style-type: none"> • Voice-to-text and emotion-detection modules combined. • Text with emoji output from pre-recorded audio. • Wav2Vec model used for speech transcript. • LSTM model used for speech emotion recognition. 	<ul style="list-style-type: none"> • Limited to pre-recorded audio • No real-time speech-to-text • Slow Wav2Vec model for speech-to-text • Processing time 30s-50s • Speech transcript accuracy 99.5%

		<ul style="list-style-type: none"> • Emotion recognition accuracy 41%
2: Additional feature and enhanced Speech to Text and emotion recognition module	<ul style="list-style-type: none"> • Assembly AI API used for speech-to-text. • Chaquopy framework • Multithreading for better speed • Play/pause recording feature. • Real-time voice recording • MLP Classifier model used for speech emotion recognition. 	<ul style="list-style-type: none"> • Emotion prediction model not improved. • Chaquopy framework may still have limitations in performance. • Processing time 15s-20s • Speech transcript accuracy 99.5% • Emotion recognition accuracy 71.3%
3: Enhanced Emotion Prediction and new framework	<ul style="list-style-type: none"> • Improved emotion detection model • Pre-trained deep speech model for speech-to-text • TensorFlow Lite framework for AI models in Android Studio • Reduced app storage and processing time • Deep Speech pretrained model 	<ul style="list-style-type: none"> • No major limitations or drawbacks identified in this version, as it addresses most issues from previous versions. • Processing time 10s-15s • Speech transcript accuracy 99.5%

	<p>used for speech transcript.</p> <ul style="list-style-type: none"> • CNN LSTM model used for speech emotion recognition. 	<ul style="list-style-type: none"> • Emotion recognition accuracy 80.3%
--	--	--

7.6 Testing

Testing ensures that the application fulfils the criteria and that the users are satisfied. Unit tests, integration tests, as well as usability and user acceptance tests are all conducted.

7.6.1 Testing Objectives

The testing objectives are defined as below:

1. Ensure user can transcribe speech either by recording real-time or uploading pre-recorded speech.
2. Ensure user can view the accurate transcript text with emoji on screen.
3. Ensure user can play and pause the speech recorded or uploaded.
4. Ensure system transcribe speech accurately.
5. Ensure system recognise speech emotion accurately.

7.6.2 Unit Testing

Each module of the system was tested individually. The test data used was my voice conveying different emotions for each of the unit test case below.

Table 7.4: Unit Test Case for record audio.

Test case title	Record Audio	Test case ID	1
Design date	15 March 2023	Designed by	Tong Kah Pau
Execution date	20 March 2023	Executed by	Tong Kah Pau
Preconditions			

Description	Test steps	Test data	Expected result	Post-condition	Actual result	Status
Start voice recording	1. Hold on to microphone button		System starts recording	System starts recording	System starts recording	Pass
Stop voice recording	1. Releases microphone button		Stop recording and System saves the recorded audio to external storage	System analyse speech and convert to text with emoji and display it on screen	Stop recording and System saves the recorded audio to external storage	Pass
Speak into microphone	1. Hold on the microphone button	Audio input data	System listens to user speech	System records user speech	System listens to user speech	Pass
Permission for microphone is not allowed	1. Deny microphone access		An alert for permission denied is shown		An alert for permission denied is shown	Pass
Permission for external storage access is denied	1. Deny external storage access		An alert for permission denied is shown		An alert for permission denied is shown	Pass

Table 7.5: Unit Test Case for upload pre-recorded speech.

Test case title	Upload pre-recorded speech	Test case ID	2			
Design date	15 March 2023	Designed by	Tong Kah Pau			
Execution date	20 March 2023	Executed by	Tong Kah Pau			
Preconditions						
Description	Test steps	Test data	Expected result	Post-condition	Actual result	Status
Choose audio file to upload	1. Click the upload button		System prompts user to file explorer	User may choose audio file to upload	System prompts user to file explorer	Pass
Upload audio file to mobile app	1. Click on to audio file choice	Audio file data	System prompts back to app main screen and show loading screen	System analyse speech and convert to text with emoji and display it on screen	System prompts back to app main screen and show loading screen	Pass
Permission for external storage access is denied	1. Deny external storage access		An alert for permission denied is shown		An alert for permission denied is shown	Pass





Table 7.6: Unit Test Case for play and pause recording.





Test case title	Play and pause recording	Test case ID	3			
Design date	15 March 2023	Designed by	Tong Kah Pau			
Execution date	20 March 2023	Executed by	Tong Kah Pau			
Preconditions						
Description	Test steps	Test data	Expected result	Post-condition	Actual result	Status
Play recording	1. Click the play button		System plays the recording	Pause button is now enable	System plays the recording	Pass
Pause recording	1. Click the pause button		System pauses the recording	Play button is now enabled again	System pauses the recording	Pass
No speech recorded or uploaded	1. Click on play button		Nothing happens. Button is disabled		Nothing happens. Button is disabled	Pass

Table 7.7: Unit Test Case for transcript text



Test case title	Transcript text		Test case ID	4		
Design date	15 March 2023		Designed by	Tong Kah Pau		
Execution date	20 March 2023		Executed by	Tong Kah Pau		
Preconditions						
Description	Test steps	Test data	Expected result	Post-condition	Actual result	Status
Test the accuracy and efficiency of the speech to text transcription feature	<ol style="list-style-type: none"> 1. Press and hold the record button to start recording speech 2. Speak a predetermined test phrase into the microphone 3. Release the record button 4. System starts to transcribe 	“The quick brown fox jumps over the lazy dog”	The recorded speech is accurately transcribed as “The quick brown fox jumps over the lazy dog”	The transcribed text is ready for further processing such as integrating with the emoji so it can be displayed as text with emoji on screen	“The quick brown fox jumps over the lazy dog”	Pass

Table 7.8: Unit Test Case for Emoji recognition

Test case title	Emoji recognition		Test case ID	5		
Design date	15 March 2023		Designed by	Tong Kah Pau		
Execution date	20 March 2023		Executed by	Tong Kah Pau		
Preconditions						
Description	Test steps	Test data	Expected result	Post-condition	Actual result	Status
Test emotion recognition with happy tone speech	<ol style="list-style-type: none"> 1. Press and hold the record button to start recording speech 2. Speak a predetermined test phrase into the microphone 3. Release the record button 4. System recognise speech's emotion 	“The quick brown fox jumps over the lazy dog” in happy tone	Happy emoji 	The transcribed text is ready for further processing such as integrating with the emoji so it can be displayed as text with emoji on screen	Happy emoji 	Pass
Test emotion recognition with neutral	<ol style="list-style-type: none"> 1. Press and hold the record button to start recording speech 	“The quick brown fox jumps over the	Neutral emoji 	The transcribed text is ready for further processing	Neutral emoji 	Pass

tone speech	2. Speak a predetermined test phrase into the microphone 3. Release the record button 4. System recognise speech's emotion	lazy dog” in neutral tone		such as integrating with the emoji so it can be displayed as text with emoji on screen		
Test emotion recognition with surprised tone speech	1. Press and hold the record button to start recording speech 2. Speak a predetermined test phrase into the microphone 3. Release the record button 4. System recognise speech's emotion	“The quick brown fox jumps over the lazy dog” in surprised tone	Surprised emoji 	The transcribed text is ready for further processing such as integrating with the emoji so it can be displayed as text with emoji on screen	Surprised emoji 	Pass
Test emotion recognition with disgust	1. Press and hold the record button to start	“The quick brown fox jumps	Disgust emoji 	The transcribed text is ready for further	Disgust emoji 	Pass

tone speech	<p>recording speech</p> <p>2. Speak a predetermined test phrase into the microphone</p> <p>3. Release the record button</p> <p>4. System recognise speech's emotion</p>	<p>over the lazy dog” in disgust tone</p>		<p>processing such as integrating with the emoji so it can be displayed as text with emoji on screen</p>		
Test emotion recognition with scared tone speech	<p>1. Press and hold the record button to start recording speech</p> <p>2. Speak a predetermined test phrase into the microphone</p> <p>3. Release the record button</p> <p>4. System recognise speech's emotion</p>	<p>“The quick brown fox jumps over the lazy dog” in scared tone</p>	<p>Scared emoji</p> <p>😨</p>	<p>The transcribed text is ready for further processing such as integrating with the emoji so it can be displayed as text with emoji on screen</p>	<p>Scared emoji</p> <p>😨</p>	Pass
Test emotion recognition	<p>1. Press and hold the record button</p>	<p>“The quick brown</p>	<p>Angry emoji</p> <p>😡</p>	<p>The transcribed text is</p>	<p>Angry emoji</p> <p>😡</p>	Pass

n with angry tone speech	to start recording speech 2. Speak a predetermined test phrase into the microphone 3. Release the record button 4. System recognise speech's emotion	fox jumps over the lazy dog" in angry tone		ready for further processing such as integrating with the emoji so it can be displayed as text with emoji on screen		
Test emotion recognition with sad tone speech	1. Press and hold the record button to start recording speech 2. Speak a predetermined test phrase into the microphone 3. Release the record button 4. System recognise speech's emotion	"The quick brown fox jumps over the lazy dog" in sad tone	Sad emoji 	The transcribed text is ready for further processing such as integrating with the emoji so it can be displayed as text with emoji on screen	Sad emoji 	Pass

Test emotion recognition with calm tone speech	<ol style="list-style-type: none"> 1. Press and hold the record button to start recording speech 2. Speak a predetermined test phrase into the microphone 3. Release the record button 4. System recognise speech's emotion 	“The quick brown fox jumps over the lazy dog” in calm tone	Calm emoji 😌	The transcribed text is ready for further processing such as integrating with the emoji so it can be displayed as text with emoji on screen	Calm emoji 😌	Pass
--	---	--	--------------	---	--------------	------

The unit tests were conducted to assess the functionality of individual modules, including Record Audio, Upload Pre-recorded Speech, Play and Pause Recording, Transcript Text, and Emoji Recognition. The tests examined various scenarios, such as starting and stopping voice recordings, uploading audio files, playing, and pausing recordings, and accurately transcribing speech to text. Additionally, the tests evaluated the system's ability to recognize different emotions in speech and associate the appropriate emojis with them.

The results indicate that all the modules performed as expected, successfully handling various user scenarios, and achieving the desired outcomes. The system could record, upload, and play audio files and accurately transcribe speech to text. Furthermore, the emoji recognition module effectively identified emotions in speech and matched them with the corresponding emojis. The tests also demonstrated the system's ability to handle permission-related issues, displaying alerts when necessary. Overall, the unit tests confirmed that each module in the system functioned effectively, ensuring a reliable and efficient user experience.

7.6.3 Integration Testing

In this test, one or more modules from the system have been tested together. Lines from the movies were used as test input to observe the results from the system based on Table 7.9. The lines from the movies will be playing back from another device while the emulator microphone captures the audio for the recording function while the lines from the movies will be downloaded and stored in the emulator external storage for the upload function. The lines from the movies used are shown in Table 7.10.

Table 7.9: List of Integration Test Cases

Modules	Test Case Description	Test Steps	Expected results	Pass/Fail
1. Record audio 2. Transcribe text 3. Emoji recognition	User wants to transcribe their speech to text with emoji in real time	<ol style="list-style-type: none"> 1. User press and holds the microphone button. 2. User start speaking through the mic. 3. User releases the button and stop the recording. 4. Speech is processed, and user can view the text transcript with emoji on screen 	<ol style="list-style-type: none"> 1. Speech is recorded 2. Speech transcribed 3. Emotion recognised, and emoji is matched with it 4. Text with emoji is displayed on screen 	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>

<p>1. Upload audio</p> <p>2. Transcribe text</p> <p>3. Emoji recognition</p>	<p>User wants to transcribe a pre-recorded speech to text with emoji</p>	<p>1. user clicks the upload button</p> <p>2. user selects the audio file to be transcribe</p> <p>3. Speech is processed, and user can view the text transcript with emoji on screen</p>	<p>1. Speech is selected</p> <p>2. Speech transcribed</p> <p>3. Emotion recognised, and emoji is matched with it</p> <p>4. Text with emoji is displayed on screen</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>
<p>1. Play audio</p> <p>2. Pause audio</p>	<p>User wants to listen and pause to the speech they recorded or uploaded</p>	<p>1. After speech is processed into text with emoji on screen.</p> <p>2. user clicks the play button to listen</p> <p>3. user clicks the pause button to pause the audio.</p>	<p>1. Speech is playing</p> <p>2. Speech is paused</p>	<p>Pass</p>
<p>1. Record Audio</p> <p>2. Transcribe text</p>	<p>After getting their transcript text with emoji by recording their speech real-time, user wants to</p>	<p>1. User press and holds the microphone button.</p> <p>2. User start speaking</p>	<p>1. Speech is recorded</p> <p>2. Speech transcribed</p> <p>3. Emotion recognised,</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p>

<p>3. Recognise emotion</p> <p>4. upload audio</p>	<p>transcribe pre-recorded speech by uploading the audio to the application</p>	<p>through the mic.</p> <p>3. User releases the button and stop the recording.</p> <p>4. Speech is processed, and user can view the text transcript with emoji on screen</p> <p>5. User selects the upload button</p> <p>6. User selects audio</p> <p>7. The transcript text with emoji is updated with the new processed audio</p>	<p>and emoji is matched with it</p> <p>4. Text with emoji is displayed on screen</p> <p>5. Speech is selected</p> <p>6. Speech transcribed</p> <p>7. Emotion recognised, and emoji is matched with it</p> <p>8. Updated text with emoji is displayed on screen</p>	<p>Pass in terms of emotion recognised.</p>
--	---	---	--	--

Table 7.10: List of lines from different movies

No	Emotion	Lines	Movie name
1	Happy	"After all this time? Always."	Harry Potter and the Deathly Hallows: Part 2 (2011)

		"Adventure is out there!"	Up (2009)
		"I'm just a girl, standing in front of a boy, asking him to love her."	Notting Hill (1999)
2	Sad	"So, this is my life. And I want you to know that I am both happy and sad, and I'm still trying to figure out how that could be."	The Perks of Being a Wallflower (2012)
		"I wish I could freeze this moment, right here, right now, and live in it forever."	The Hunger Games (2012)
		"It's only after we've lost everything that we're free to do anything."	Fight Club (1999)
3	Angry	"You either die a hero, or you live long enough to see yourself become the villain."	The Dark Knight (2008)
		"I don't want to be a product of my environment. I want my environment to be a product of me."	The Departed (2006)
		"King Kong ain't got sh*t on me!"	Training Day (2001)
4	Disgust	"Why don't you make like a tree and get outta here?"	The Room (2003)
		"What is this? A center for ants?"	Zoolander (2001)
		"I drink your milkshake! I drink it up!"	There Will Be Blood (2007)
5	Surprise	"I'm not a smart man, but I know what love is."	Forrest Gump (1994)

		"I'm just a kid from Brooklyn."	Captain America: The First Avenger (2011)
		"I am your father."	Star Wars: Episode V - The Empire Strikes Back (1980)
6	Calm	"In case I don't see you, good afternoon, good evening, and good night!"	The Truman Show (1998)
		"The hardest choices require the strongest wills."	Avengers: Infinity War (2018)
		"You is kind. You is smart. You is important."	The Help (2011)
7	Neutral	"I am Iron Man."	Iron Man (2008)
		"Why so serious?"	The Dark Knight (2008)
		"I'll be back."	The Terminator (1984)
8	Fearful	"I see dead people."	The Sixth Sense (1999)
		"You better start believing in ghost stories, Miss Turner. You're in one!"	Pirates of the Caribbean: The Curse of the Black Pearl (2003)
		"We have to go back!"	Lost (TV Series, 2004-2010)

The test results showed successful outcomes for all test cases, indicating that the system could effectively transcribe speech to text with emojis in real-time and from pre-recorded audio, as well as allowing users to listen and pause the speech they recorded or uploaded. The test cases also demonstrated the successful recognition of emotions and the ability to update the transcript text with emojis when new audio was processed. Overall, the system passed in terms of functionalities, speech transcription, and emotion recognition, showcasing its ability to handle various user scenarios and perform the desired tasks effectively.

7.6.4 User Acceptance Test

A user acceptance test was performed. The final product will be tested by 10 selected individuals consist of 5 men and women from the age of 20-40. They will be given a checklist outlining what they need to do and the outcomes they can anticipate after completing those tasks. Their use of the mobile app is noticed along with their behaviour.

Users were provided a set of test cases to perform and attempt to finish all of them during the user approval test. The developer will not aid unless the customer is unable to finish the job.

7.6.4.1 Execution

Each user is expected to perform user testing by completing all provided scenarios. They will be tested under the guidance of the developer. If assistance is required, participants may seek it from the developer. The following procedures will be taken to conduct user testing:

1. Users read through all planned scenarios.
2. Users must follow the assignment scenarios that have been created and attempt to finish all the scenarios mentioned.
3. Monitor the users as they complete their tasks and offer help as needed.
4. After completing the test, participants complete the feedback questionnaire during the usability test.

7.6.4.2 List of Test Cases

Table 7.11 below shoes the test case listing which are linked to the task scenario given during usability test, and test cases description. Sample of user acceptance test cases are attached in Appendix B. Table 7.12 is used for users to convey their emotions by reading the sample speech prepared by me. User may also be creative and say their own speeches if they know what to say. For the upload speech feature, a list of speeches based on Table 7.12 was recorded by me for the user to test the feature. The results of the user acceptance test alongside with the informed consent agreement forms are attached in Appendix C.

Table 7.11: List of Task Scenario with Test Cases

No	Task Scenario Title	Task Scenario Description	Test Case Description
1	Express happiness in a recording	Assume you've just had a happy encounter and want to document your emotions with the programme. Make a brief statement conveying pleasure	<ol style="list-style-type: none"> 1. Able to start and stop recording by accessing mobile's microphone. 2. Able to view approximately accurate transcript text with emoji
2	Share a sad experience	Consider a scenario in which you were unhappy and wanted to express your emotions through the programme. Record a short statement detailing your sorrowful experience	<ol style="list-style-type: none"> 1. Able to start and stop recording by accessing mobile's microphone. 2. Able to view approximately accurate transcript text with emoji
3	Vent out anger	You recently encountered a stressful circumstance and want to express your frustration through the programme. Make a brief statement conveying your rage	<ol style="list-style-type: none"> 1. Able to start and stop recording by accessing mobile's microphone. 2. Able to view approximately accurate transcript text with emoji

			transcript text with emoji
4	Describe a disgusting incident	You recall an event that offended you and want to express your emotions through the app. Record a short description of the event	<ol style="list-style-type: none"> 1. Able to start and stop recording by accessing mobile's microphone. 2. Able to view approximately accurate transcript text with emoji
5	Upload and playback a speech expressing surprise	You have a pre-recorded speech in which you got surprising news, and you want to use the app to analyse what you said and your astonishment. Upload the statement, then play it back and pause it	<ol style="list-style-type: none"> 1. Able to upload audio file by choosing audio file from file explorer. 2. Able to view approximately accurate transcript text with emoji Able to play back and pause the audio
6	Upload and playback a speech about calming experience	You have a prepared speech in which you explain a peaceful and tranquil experience, and you want to use the app to analyse what you said and your mood. Upload the statement, then play it back and pause it	<ol style="list-style-type: none"> 1. Able to start and stop recording by accessing mobile's microphone. 2. Able to view approximately accurate

			transcript text with emoji
7	Upload and playback a neutral speech	You have a prepared speech in which you spoke without feeling, and you want to use the app to analyse what you said and your bland tone. Upload the voice and then hear it back after it has been processed.	<ol style="list-style-type: none"> 1. Able to start and stop recording by accessing mobile's microphone. 2. Able to view approximately accurate transcript text with emoji
8	Upload and playback a speech about a scary situation	You have a pre-recorded speech in which you explain a frightening experience, and you want to analyse what you said and your emotions using the app. Upload the voice and then hear it back after it has been processed.	<ol style="list-style-type: none"> 1. Able to start and stop recording by accessing mobile's microphone. 2. Able to view approximately accurate transcript text with emoji

Table 7.12: List of Test Speeches used for user acceptance test.

No	Emotions	Test Speeches
1	Happy	"Yesterday I discovered a new Nasi Lemak spot, it was so delicious"
2	Sad	"I did so bad in my exam today"
3	Angry	"Do you know how long I was stuck in a Traffic Jam today?"
4	Disgust	"Ewww can you clean your room"

5	Surprise	“I ran into our old schoolteacher at the mall today, and she still remembered us after all these years!”
6	Calm	“I spent my weekend lying in bed”
7	Neutral	“I just wanted to inform you about my report completion”
8	Scared	“I was driving home yesterday, and I saw a ghost!”

7.6.4.3 Results Analysis

All testers recorded documented their findings on the test case papers that had been made ahead of time and are appended as Appendix F.

All the test scenarios run by the 10 users were successful in terms of functionalities and the speech transcription accuracy. In contrast, there were some failures in terms of the emotion recognition accuracy. Not all users’ emotion was recognised. This is due to some users to that have the lack of ability to convey their emotions and this may confuse the AI model. However, 6/8 of the emotions were recognised.

In conclusion, this implies that the mobile application was able to execute all anticipated functions. On the other hand, it confirms that the mobile application provided features that met the user-defined criteria. This verifies that there has been no misunderstanding or misreading of criteria. When all test cases succeed, it is less likely that problems with the verified features will arise. After passing the user approval test, no flaws or errors were discovered. After passing the user approval test, the smartphone application is now available for distribution.

7.6.5 Usability Testing

After user acceptance test, usability test is conducted where each user will be given a brief survey to get their comments and opinions on the mobile app. This is to ensure users experience can be improved.

7.6.5.1 Results

An example of a user satisfaction survey form can be found in Appendix A. The full actual results and expected results from the task scenarios given is attached at Appendix C. Moreover, participants’ answered user satisfaction survey is attached in Appendix C. Upon completing the usability testing, all participants must complete a

user satisfaction survey. The results of the System Usability Scale (SUS) scores can be found in Table 7.13. According to results obtained, the participants are 91.75% satisfied with the mobile application. The score is 23.75% higher than the required 68%. The calculation for the SUS Scores is at Appendix C.

Table 7.13: System Usability Scale Scores

Participant	SUS Score										Total
	1	2	3	4	5	6	7	8	9	10	
1	5	1	5	1	4	1	5	1	4	1	95
2	5	1	5	1	5	1	5	1	5	1	100
3	5	1	5	1	3	1	4	1	4	1	90
4	5	1	4	1	4	1	4	1	4	1	90
5	5	1	4	2	4	1	4	1	5	1	90
6	4	1	5	1	4	1	4	1	4	1	90
7	5	1	4	2	5	1	5	1	4	1	92.5
8	4	1	4	2	4	1	3	1	5	1	85
9	5	1	5	1	4	1	5	1	4	1	95
10	5	1	4	2	5	2	5	1	4	1	90
Total Average SUS Score											91.75

In the user satisfaction survey, users' feedback on the app's least and most preferred features are collected. The following is a compilation of user feedback:

Positive feedback (what user like):

1. Speech-to-text translation that is simple and efficient: Users may enjoy the app's ability to convert spoken words swiftly and correctly into text, making it easier for them to interact or record their ideas.
2. Emoji integration: The app's feature that converts text sentences into pertinent emoticons can make conversations more engaging, entertaining, and emotive, which may appeal to users who like to communicate with emojis.
3. Emotion recognition: The ability to recognise and categorise the user's voice expression can add a fascinating and personalised element to the app experience, possibly increasing user enjoyment.

Areas for improvement (what users dislike):

1. Users may feel that the app is not as thorough or interesting if the app's character translation function does not encompass a broad variety of terms or sentences.
2. Emotion detection errors: If the app fails to correctly recognise or misinterprets the user's feelings, it may cause uncertainty or irritation.

In conclusion, due to its speech-to-text transcribing, emoticon incorporation, and expression detection features, the Speech to Text with emoji software might appeal to social media users, content producers, and people with disability requirements. However, due to its emphasis on emoticons and dynamic analysis, users who prefer conventional text communication, those who are worried about privacy, or workers wanting official conversation may find the app less pertinent or attractive. Moreover, users which have the lack of ability to convey emotions may experience a little less compared to the ones that know how to express emotions. Lastly, with the emotion recognition model being still yet to reach maximum accuracy, the model may be confused by other emotions as well.

CHAPTER 8

CONCLUSION

8.1 Introduction

Speech to text with emoji mobile app project is a project to develop an audio processor that can transcribe speech to text whilst recognising the speaker's emotions and combining them becoming text with emoji. This project started around June 2022 and ended in April 2023. From planning to completion, it took approximately six months. The project's targeted users are people who want to enhance their communication, deaf people, and students. With this mobile application, user can transcribe their speech to text with emojis anytime.

8.2 Fulfilment of Objectives

Upon completion of development of the project, the objectives in Chapter 1.3 were fulfilled by including certain features and backend development techniques in the mobile application. All objectives are considered achieved with these functions included.

The first objective to investigate the feasibility of the voice recognition system for emotion detection was implemented by undertaking comprehensive study on voice recognition, auditory processing, and mood identification approaches already in use. The viability of the system is assured by comparing various mood recognition algorithms and selecting the one that best suits the project specifications. These can be seen that different emotion recognition have been tried out during the evaluation and testing phase.

The second project objective to develop an emotional voice recognition system which can measure the highest accuracy of the number of decibels, pitch/octave, speech rate, timbre, speech pattern and common phrases from a user's speech to display the correct emojis on screen after the right model has been chosen. The speech-to-text engine and emotion recognition model are combined in a smooth manner to produce text with the appropriate emoticons based on the identified feelings. The system is continuously tested and adjusted to increase its effectiveness and precision in identifying feelings. This can be seen as Version 3 has the improved version of the MLP Classifier model where the parameters are tune, increasing the model accuracy.

The third project objective to develop a speech to text with emoji solutions which the transcript text with emojis rate is as similar as the speech rate and the emotions portrait by the user. This is implemented by adopting the multithreading technique. Both the modules are executed smoothly and the rate of processing their inputs are dependent on one another. Hence, the transcript text with emoji will be display on screen at a timely manner.

The project effectively develops an AI-driven smartphone application capable of understanding user voice and translating it into text with matching emoticons based on the identified feelings through research, development, and continuous enhancement.

8.3 Suggestions and Recommendations

1. History and arrangements

Add a function that enables users to browse, arrange, and control their transcript past using emoticons. Users could soon locate talks or spot patterns in emotional expression by sorting the record past by date, subject, or emotion. The ability to quickly access and examine previous transcriptions would allow users to monitor and meditate on their feelings and communication styles over time.

2. Options for cooperation and sharing.

Integrating with well-known work and communication tools would be essential to guarantee smooth sharing and cooperation. Include teamwork and sharing tools that enable users to post their emoji-filled transcriptions to social media, email, or messaging services. Multiple individuals could simultaneously participate in a discussion or paper by working together on transcriptions in real-time. These elements increase the programme's usefulness beyond personal use and promote a more engaging social user experience.

3. Automatic language recognition and multilingual assistance

By introducing multilingual support, you can increase the app's functionality and reach a wider audience. Use automated language identification to recognise and correctly translate words in the user's preferred language while keeping accurate context and mood recognition. Integrating language-specific models for speech-to-text and

emotion detection into this feature would necessitate further study and development to ensure precise writing and affective depiction across various languages and societies.

4. Real-time input and guidance for emotions

Introduce a tool that gives users immediate feedback on their emotional expression. At the same time, speech is being recorded, providing gentle guidance or cues to assist users in adjusting their speech and emotional expression as needed. The software could give a user a little notice or recommendation to change their tone, for instance, if they speak furiously but want to sound tranquil. This function, which promotes self-awareness and mental control, may benefit users who wish to practise public speaking or improve their communication skills.

5. Voice characteristics and customised mood recognition

Create a tool that lets users customise their voice profiles so that the app can adjust mood recognition to match their speaking habits and vocal traits. The software would be able to offer more accurate mood detection and emoticon recommendations by learning from each user's unique speech style. Users could also actively modify the app's susceptibility to various feelings or give the app performance input, which would help the app's emotion recognition become more accurate over time.

6. Client and Server-side system architecture

Deploy the speech transcription and emotion recognition AI models integration to cloud. This shall further decrease the app storage size and processing time since the task handled by the mobile application decreased. However, the application may be Internet dependant.

REFERENCES

Álvarez, A., Arzelus, H., Torre, I.G. and González-Docasal, A. (2022). Evaluating Novel Speech Transcription Architectures on the Spanish RTVE2020 Database. *Applied Sciences*, 12(4), p.1889. doi:10.3390/app12041889.

Android Developers. (2019). *SpeechRecognizer* | Android Developers. [online] Available at: <https://developer.android.com/reference/android/speech/SpeechRecognizer>.

AssemblyAI. (n.d.). *AssemblyAI* | AI models to transcribe and understand speech. [online] Available at: https://www.assemblyai.com/?utm_source=google&utm_medium=cpc&utm_campaign=Brand&utm_term=assemblyai%20api

Castanyer, R.C., Martínez-Fernández, S. and Franch, X. (2021). Integration of Convolutional Neural Networks in Mobile Applications. [online] *IEEE Xplore*. doi:10.1109/WAIN52551.2021.00010.

Chavhan, Y., Dhore, M.L. and Yesaware, P. (2010). Speech Emotion Recognition using Support Vector Machine. *International Journal of Computer Applications*, 1(20), pp.8–11. doi:10.5120/431-636.

Codecademy. (n.d.). *Deep Learning Workflow*. [online] Available at: <https://www.codecademy.com/article/deep-learning-workflow>

Fiscus, J., Ajot, J., Radde, N. and Laprun, C. (2006). Multiple Dimension Levenshtein Edit Distance Calculations for Evaluating Automatic Speech Recognition Systems During Simultaneous Speech.

Fox, C.B., Israelsen-Augenstein, M., Jones, S. and Gillam, S.L. (2021). An Evaluation of Expedited Transcription Methods for School-Age Children’s Narrative Language: Automatic Speech Recognition and Real-Time Transcription. *Journal of Speech, Language, and Hearing Research*, 64(9), pp.3533–3548. doi:10.1044/2021_jslhr-21-00096.

Full Emoji List, v12.0. (2019). Unicode.org. <https://unicode.org/emoji/charts/full-emoji-list.html>

Glenn, M., Strassel, S., Lee, H., Maeda, K., Zakhary, R. and Li, X. (n.d.). Transcription methods for consistency, volume and efficiency

Hashemnia, S., Grasse, L., Soni, S. and Tata, M.S. (2021). Human EEG and Recurrent Neural Networks Exhibit Common Temporal Dynamics During Speech Recognition. *Frontiers in Systems Neuroscience*, 15. doi:<https://doi.org/10.3389/fnsys.2021.617605>.

huggingface.co. (n.d.). *facebook/wav2vec2-base-960h* · Hugging Face. [online] Available at: <https://huggingface.co/facebook/wav2vec2-base-960h> [Accessed 2 Apr. 2023].

Intellipaat Blog. (2020). What is LSTM - Introduction to Long Short Term Memory. [online] Available at: <https://intellipaat.com/blog/what-is-lstm/>

Javed, A.R., Sarwar, M.U., Khan, S., Iwendi, C., Mittal, M. and Kumar, N. (2020). Analyzing the Effectiveness and Contribution of Each Axis of Tri-Axial Accelerometer Sensor for Accurate Activity Recognition. *Sensors*, 20(8), p.2216. doi:<https://doi.org/10.3390/s20082216>.

Joshi, A. and Kaur, R. (2013). A Study of Speech Emotion Recognition Methods. *International Journal of Computer Science and Mobile Computing*, 2(4), pp.28–31.

Kerkeni, L., Serrestou, Y., Mbarki, M., Raouf, K. and Mahjoub, M.A. (2018). Speech Emotion Recognition: Methods and Cases Study. *Proceedings of the 10th International Conference on Agents and Artificial Intelligence*. [online] doi:[10.5220/0006611601750182](https://doi.org/10.5220/0006611601750182).

Kissell, K.D. (2007). What is multithreading and multi-core? *ACM SIGDA Newsletter*, 37(21), pp.1–1. doi:<https://doi.org/10.1145/1859872.1859873>.

Lim, W., Jang, D. and Lee, T. (2016). Speech emotion recognition using convolutional and Recurrent Neural Networks. [online] *IEEE Xplore*. doi:[10.1109/APSIPA.2016.7820699](https://doi.org/10.1109/APSIPA.2016.7820699).

Mamorsky, J., individual, M.-L. at myTherapyNYCOffers, depression, couples counseling S. specializes in, anxiety, Issues, R. and trauma. (2019). The Importance of Understanding Your Emotions. [online] myTherapyNYC. Available at: <https://mytherapynyc.com/understanding-emotions/>.

Miraz, M., Ali, M. and Excell, P.S. (2022). Cross-cultural usability evaluation of AI-based adaptive user interface for mobile applications. *Acta Scientiarum. Technology*, 44, p.e61112. doi:[10.4025/actascitechnol.v44i1.61112](https://doi.org/10.4025/actascitechnol.v44i1.61112).

mozilla (2019). mozilla/DeepSpeech. [online] *GitHub*. Available at: <https://github.com/mozilla/DeepSpeech>.

Naseem, I., Togneri, R. and Bennamoun, M. (2010). Linear Regression for Face Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, [online] 32(11), pp.2106–2112. doi:[10.1109/TPAMI.2010.128](https://doi.org/10.1109/TPAMI.2010.128).

News, I.B.L. (2020). A Survey Shows that Many College Students Struggle to Maintain Focus and Discipline in Distance Learning | IBL News. [online] IBL News. Available at: <https://iblnews.org/a-survey-shows-that-many-college-students-struggle-to-maintain-focus-and-discipline-in-distance-learning/>.

Passricha, V. and Aggarwal, R.K. (2019). A Hybrid of Deep CNN and Bidirectional LSTM for Automatic Speech Recognition. *Journal of Intelligent Systems*, 29(1), pp.1261–1274. doi:<https://doi.org/10.1515/jisys-2018-0372>.

Sarker, I.H., Hoque, M.M., Uddin, Md.K. and Alsanoosy, T. (2020). Mobile Data Science and Intelligent Apps: Concepts, AI-Based Modeling and Research Directions. Mobile Networks and Applications. doi:10.1007/s11036-020-01650-z.

Sudhkar, R. and Anil, M. (2015). Analysis of Speech Features for Emotion Detection : A review.

V. Srinivasan, V. Ramalingam, and P. Arulmozhi (2014). Artificial Neural Network Based Pathological Voice Classification Using Mfcc Features.

Wu, S., Falk, T.H. and Chan, W.-Y. (2011). Automatic speech emotion recognition using modulation spectral features. *Speech Communication*, 53(5), pp.768–785. doi:10.1016/j.specom.2010.08.013.

www.businesswire.com. (2014). Survey Says: Voicemail-to-Text Service Brings the Speed and Convenience to Voice Messaging that Consumers Want. [online] Available at: <https://www.businesswire.com/news/home/20140211005345/en/Survey-Says-Voicemail-to-Text-Service-Brings-the-Speed-and-Convenience-to-Voice-Messaging-that-Consumers-Want>

APPENDICES

APPENDIX A: User Satisfaction Survey Sample

Participant #

User Satisfaction survey

Please mark 'x' to rate the following statements.

	Strongly Agree 5	Agree 4	Neutral 3	Disagree 2	Strongly Disagree 1
1. I prefer using this system for enhancing my communication experience.					
2. I found the system unnecessarily complex.					
3. I thought the system was easy to use.					
4. I found the system very awkward to use					
5. I am satisfied with the speed and responsiveness of the Speech to Text with Emoji app.					

6. I felt frustrated when trying to accomplish tasks using the system					
7. I think that the app may integrate with my preferred messaging or communication apps					
8. I felt this system had too many differences or changes in it.					
9. I think that the Speech to Text with Emoji app is important for your daily communication needs					
10. I had to gain ample knowledge before using the system effectively.					

Please answer all the following questions.

1. Which part of the system you like the most?

Ans:

2. Which part of the system you like the least?

Ans:

3. Any final comments or questions?

Ans:

APPENDIX B: User Acceptance Test Sample

Test Date			
Starting Time			
Ending Time			
Participant Name			
Test Module	Test Scenario	Pass/Fail	Feedbacks
Transcribe happy speech in real time	<p>A user is happy, and they want to transcribe their speech into text with emojis in real time and plays it back.</p> <ol style="list-style-type: none"> 1. Start speech recording. 2. Stop speech recording. 3. View transcript text with emoji. <ol style="list-style-type: none"> 1. Plays back recording. 		
Transcribe sad speech in real time	<p>A user is sad, and they want to transcribe their speech into text with emojis in real time.</p> <ol style="list-style-type: none"> 1. Start speech recording. 2. Stop speech recording. 3. View transcript text with emoji. <ol style="list-style-type: none"> 1. Plays back recording. 		
Transcribe pre- recorded angry speech	<p>A user is angry, and they want to transcribe their speech into text with emojis by uploading.</p> <ol style="list-style-type: none"> 1. Choose audio file to upload. 		

	<ol style="list-style-type: none"> 2. Upload audio file 3. Views transcript text with emoji 1. Plays back recording 		
Transcribe pre-recorded disgust speech in real time	<p>A user is disgusted, and they want to transcribe their speech into text with emojis in real time.</p> <ol style="list-style-type: none"> 1. Choose audio file to upload. 2. Upload audio file 3. Views transcript text with emoji 1. Plays back recording 		
Transcribe pre-recorded surprise speech in real time	<p>A user is disgusted, and they want to transcribe their speech into text with emojis in real time.</p> <ol style="list-style-type: none"> 1. Choose audio file to upload. 2. Upload audio file 3. Views transcript text with emoji <p>Plays back recording</p>		
Transcribe pre-recorded calm speech in real time	<p>A user is calm, and they want to transcribe their speech into text with emojis in real time.</p> <ol style="list-style-type: none"> 1. Choose audio file to upload. 2. Upload audio file 3. Views transcript text with emoji 1. Plays back recording 		

<p>Transcribe neutral speech in real time</p>	<p>A user is neutral, and they want to transcribe their speech into text with emojis in real time and plays it back.</p> <ol style="list-style-type: none"> 1. Start speech recording. 2. Stop speech recording. 3. View transcript text with emoji. <ol style="list-style-type: none"> 1. Plays back recording. 		
<p>Transcribe scared speech in real time</p>	<p>A user is scared, and they want to transcribe their speech into text with emojis in real time and plays it back.</p> <ol style="list-style-type: none"> 1. Start speech recording. 2. Stop speech recording. 3. View transcript text with emoji. <ol style="list-style-type: none"> 1. Plays back recording. 		

APPENDIX C: User Acceptance Test and System Usability Test Results

Informed Consent Agreement

The purpose of this research project, led by Tong Kah Pau, a Software Engineering student at Universiti Tunku Abdul Rahman (UTAR), is to examine the practicality and user-friendliness of a system designed under the project named "Speech to Text with Emojis."

We cordially invite you to take part in this research study, which will consist of the following elements:

1. User Acceptance Test (UAT)
2. System Usability Test (SUT)

By providing your signature on this document, you confirm that:

- Your participation in this study is entirely voluntary, and you have agreed to do so without coercion.
- You are aware that involvement in this usability research is optional, and you have the right to raise any concerns or discomfort throughout the research and to withdraw at any moment.
- You acknowledge that the User Acceptance Test will necessitate your engagement with the developed system according to the instructions provided.

Name : How Jia Le
Signature : *How Jia Le*
Date : 26/3/2023

We are grateful for your involvement in this study.

Rest assured that any information you provide will be used solely for the purpose of evaluating the system's effectiveness and will not be shared or used for any other reasons.

User Acceptance Test

Test Date	26/3/2023		
Starting Time	1.00pm		
Ending Time	1.30pm		
Participant Name	How Jia Le		
Test Module	Test Scenario	Pass/Fail	Feedbacks
Transcribe happy speech in real time	<p>A user is happy, and they want to transcribe their speech into text with emojis in real time and plays it back.</p> <ol style="list-style-type: none"> 1. Start speech recording. 2. Stop speech recording. 3. View transcript text with emoji. 4. Plays back recording. 	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
Transcribe sad speech in real time	<p>A user is sad, and they want to transcribe their speech into text with emojis in real time.</p> <ol style="list-style-type: none"> 1. Start speech recording. 2. Stop speech recording. 3. View transcript text with emoji. 4. Plays back recording. 	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	

Transcribe pre-recorded speech angry	A user is angry, and they want to transcribe their speech into text with emojis by uploading. <ol style="list-style-type: none"> 1. Choose audio file to upload. 2. Upload audio file 3. Views transcript text with emoji 4. Plays back recording 	Pass in terms of functionalities. Pass in terms of speech transcribed. Pass in terms of emotion recognised.	
Transcribe pre-recorded speech in real time disgust	A user is disgusted, and they want to transcribe their speech into text with emojis in real time. <ol style="list-style-type: none"> 1. Choose audio file to upload. 2. Upload audio file 3. Views transcript text with emoji 4. Plays back recording 	Pass in terms of functionalities. Pass in terms of speech transcribed. Fail in terms of emotion recognised	
Transcribe pre-recorded speech in real time surprise	A user is disgusted, and they want to transcribe their speech into text with emojis in real time. <ol style="list-style-type: none"> 1. Choose audio file to upload. 2. Upload audio file 3. Views transcript text with emoji Plays back recording	Pass in terms of functionalities. Pass in terms of speech transcribed. Pass in terms of emotion recognised.	
Transcribe pre-recorded speech in real time calm	A user is calm, and they want to transcribe their	Pass in terms of functionalities.	

	<p>speech into text with emojis in real time.</p> <ol style="list-style-type: none"> 1. Choose audio file to upload. 2. Upload audio file 3. Views transcript text with emoji 4. Plays back recording 	<p>Pass in terms of speech transcribed.</p> <p>Fail in terms of emotion recognised.</p>	
Transcribe neutral speech in real time	<p>A user is neutral, and they want to transcribe their speech into text with emojis in real time and plays it back.</p> <ol style="list-style-type: none"> 1. Start speech recording. 2. Stop speech recording. 3. View transcript text with emoji. 4. Plays back recording. 	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
Transcribe scared speech in real time	<p>A user is scared, and they want to transcribe their speech into text with emojis in real time and plays it back.</p> <ol style="list-style-type: none"> 1. Start speech recording. 2. Stop speech recording. 3. View transcript text with emoji. 4. Plays back recording. 	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	

System Usability Test

Participant 1 How Jia Le

User Satisfaction survey

Please mark 'x' to rate the following statements.

	Strongly Agree 5	Agree 4	Neutral 3	Disagree 2	Strongly Disagree 1
1. I prefer using this system for enhancing my communication experience.	x				
2. I found the system unnecessarily complex.					x
3. I thought the system was easy to use.	x				
4. I found the system very awkward to use					x
5. I am satisfied with the speed and responsiveness of the Speech to Text with Emoji app.		x			
6. I felt frustrated when trying to					x

accomplish tasks using the system					
7. I think that the app may integrate with my preferred messaging or communication apps	x				
8. I felt this system had too many differences or changes in it.					x
9. I think that the Speech to Text with Emoji app is important for your daily communication needs		x			
10. I had to gain ample knowledge before using the system effectively.					x

Please answer all the following questions.

1. Which part of the system you like the most?

Ans: I really like how speech can be converted to text with emoji without saying the emoji.

2. Which part of the system you like the least?

Ans: The system should recognise different slangs more.

3. Any final comments or questions?

Ans: Overall, it's a great app that makes communication easier and more fun.

Informed Consent Agreement

The purpose of this research project, led by Tong Kah Pau, a Software Engineering student at Universiti Tunku Abdul Rahman (UTAR), is to examine the practicality and user-friendliness of a system designed under the project named "Speech to Text with Emojis."

We cordially invite you to take part in this research study, which will consist of the following elements:

1. User Acceptance Test (UAT)
2. System Usability Test (SUT)

By providing your signature on this document, you confirm that:

- Your participation in this study is entirely voluntary, and you have agreed to do so without coercion.
- You are aware that involvement in this usability research is optional, and you have the right to raise any concerns or discomfort throughout the research and to withdraw at any moment.
- You acknowledge that the User Acceptance Test will necessitate your engagement with the developed system according to the instructions provided.

Name : Tan Yu Zhe
Signature : *Yu Zhe*
Date : 26/3/2023

We are grateful for your involvement in this study.

Rest assured that any information you provide will be used solely for the purpose of evaluating the system's effectiveness and will not be shared or used for any other reasons.

User Acceptance Test

Test Date	26/3/2023		
Starting Time	1.30pm		
Ending Time	2.00pm		
Participant Name	Tan Yu Zhe		
Test Module	Test Scenario	Pass/Fail	Feedbacks
Transcribe happy speech in real time	<p>A user is happy, and they want to transcribe their speech into text with emojis in real time and plays it back.</p> <ol style="list-style-type: none"> 5. Start speech recording. 6. Stop speech recording. 7. View transcript text with emoji. 8. Plays back recording. 	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
Transcribe sad speech in real time	<p>A user is sad, and they want to transcribe their speech into text with emojis in real time.</p> <ol style="list-style-type: none"> 5. Start speech recording. 6. Stop speech recording. 7. View transcript text with emoji. 8. Plays back recording. 	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
Transcribe pre-recorded angry speech	<p>A user is angry, and they want to transcribe their</p>	<p>Pass in terms of functionalities.</p>	

	<p>speech into text with emojis by uploading.</p> <ol style="list-style-type: none"> 5. Choose audio file to upload. 6. Upload audio file 7. Views transcript text with emoji 8. Plays back recording 	<p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
Transcribe pre-recorded disgust speech in real time	<p>A user is disgusted, and they want to transcribe their speech into text with emojis in real time.</p> <ol style="list-style-type: none"> 5. Choose audio file to upload. 6. Upload audio file 7. Views transcript text with emoji 8. Plays back recording 	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
Transcribe pre-recorded surprise speech in real time	<p>A user is disgusted, and they want to transcribe their speech into text with emojis in real time.</p> <ol style="list-style-type: none"> 4. Choose audio file to upload. 5. Upload audio file 6. Views transcript text with emoji <p>Plays back recording</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
Transcribe pre-recorded calm speech in real time	<p>A user is calm, and they want to transcribe their speech into text with emojis in real time.</p>	<p>Pass in terms of functionalities.</p>	

	<ol style="list-style-type: none"> 5. Choose audio file to upload. 6. Upload audio file 7. Views transcript text with emoji 8. Plays back recording 	<p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
Transcribe neutral speech in real time	<p>A user is neutral, and they want to transcribe their speech into text with emojis in real time and plays it back.</p> <ol style="list-style-type: none"> 5. Start speech recording. 6. Stop speech recording. 7. View transcript text with emoji. 8. Plays back recording. 	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
Transcribe scared speech in real time	<p>A user is scared, and they want to transcribe their speech into text with emojis in real time and plays it back.</p> <ol style="list-style-type: none"> 5. Start speech recording. 6. Stop speech recording. 7. View transcript text with emoji. 8. Plays back recording. 	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	

System Usability Test

Participant 2 Tan Yu Zhe

User Satisfaction survey

Please mark 'x' to rate the following statements.

	Strongly Agree 5	Agree 4	Neutral 3	Disagree 2	Strongly Disagree 1
1. I prefer using this system for enhancing my communication experience.	x				
2. I found the system unnecessarily complex.					x
3. I thought the system was easy to use.	x				
4. I found the system very awkward to use					x
5. I am satisfied with the speed and responsiveness of the Speech to Text with Emoji app.	x				
6. I felt frustrated when					x

trying to accomplish tasks using the system					
7. I think that the app may integrate with my preferred messaging or communication apps	x				
8. I felt this system had too many differences or changes in it.					x
9. I think that the Speech to Text with Emoji app is important for your daily communication needs	x				
10. I had to gain ample knowledge before using the system effectively.					x

Please answer all the following questions.

1. Which part of the system you like the most?

Ans: The ability to accurately detect emotions and add appropriate emojis is impressive.

2. Which part of the system you like the least?

Ans: The user interface could be more intuitive.

3. Any final comments or questions?

Ans: Would love to see more language support in the future.

Informed Consent Agreement

The purpose of this research project, led by Tong Kah Pau, a Software Engineering student at Universiti Tunku Abdul Rahman (UTAR), is to examine the practicality and user-friendliness of a system designed under the project named "Speech to Text with Emojis."

We cordially invite you to take part in this research study, which will consist of the following elements:

1. User Acceptance Test (UAT)
2. System Usability Test (SUT)

By providing your signature on this document, you confirm that:

- Your participation in this study is entirely voluntary, and you have agreed to do so without coercion.
- You are aware that involvement in this usability research is optional, and you have the right to raise any concerns or discomfort throughout the research and to withdraw at any moment.
- You acknowledge that the User Acceptance Test will necessitate your engagement with the developed system according to the instructions provided.

Name : Kang Shen Ni
Signature : *Shenni*
Date : 26/3/2023

We are grateful for your involvement in this study.

Rest assured that any information you provide will be used solely for the purpose of evaluating the system's effectiveness and will not be shared or used for any other reasons.

User Acceptance Test

Test Date	26/3/2023		
Starting Time	2.00pm		
Ending Time	2.30pm		
Participant Name	Kang Shen Ni		
Test Module	Test Scenario	Pass/Fail	Feedbacks
Transcribe happy speech in real time	<p>A user is happy, and they want to transcribe their speech into text with emojis in real time and plays it back.</p> <p>9. Start speech recording.</p> <p>10. Stop speech recording.</p> <p>11. View transcript text with emoji.</p> <p>12. Plays back recording.</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
Transcribe sad speech in real time	<p>A user is sad, and they want to transcribe their speech into text with emojis in real time.</p> <p>9. Start speech recording.</p> <p>10. Stop speech recording.</p> <p>11. View transcript text with emoji.</p> <p>12. Plays back recording.</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	

Transcribe pre-recorded speech angry	A user is angry, and they want to transcribe their speech into text with emojis by uploading. 9. Choose audio file to upload. 10. Upload audio file 11. Views transcript text with emoji 12. Plays back recording	Pass in terms of functionalities. Pass in terms of speech transcribed. Fail in terms of emotion recognised	
Transcribe pre-recorded speech in real time disgust	A user is disgusted, and they want to transcribe their speech into text with emojis in real time. 9. Choose audio file to upload. 10. Upload audio file 11. Views transcript text with emoji 12. Plays back recording	Pass in terms of functionalities. Pass in terms of speech transcribed. Fail in terms of emotion recognised	
Transcribe pre-recorded speech in real time surprise	A user is disgusted, and they want to transcribe their speech into text with emojis in real time. 7. Choose audio file to upload. 8. Upload audio file 9. Views transcript text with emoji Plays back recording	Pass in terms of functionalities. Pass in terms of speech transcribed. Fail in terms of emotion recognised.	
Transcribe pre-recorded speech in real time calm	A user is calm, and they want to transcribe their	Pass in terms of functionalities.	

	<p>speech into text with emojis in real time.</p> <p>9. Choose audio file to upload.</p> <p>10. Upload audio file</p> <p>11. Views transcript text with emoji</p> <p>12. Plays back recording</p>	<p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
Transcribe neutral speech in real time	<p>A user is neutral, and they want to transcribe their speech into text with emojis in real time and plays it back.</p> <p>9. Start speech recording.</p> <p>10. Stop speech recording.</p> <p>11. View transcript text with emoji.</p> <p>12. Plays back recording.</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
Transcribe scared speech in real time	<p>A user is scared, and they want to transcribe their speech into text with emojis in real time and plays it back.</p> <p>9. Start speech recording.</p> <p>10. Stop speech recording.</p> <p>11. View transcript text with emoji.</p> <p>12. Plays back recording.</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	

System Usability Test

Participant 3 Kang Shen Ni

User Satisfaction survey

Please mark 'x' to rate the following statements.

	Strongly Agree 5	Agree 4	Neutral 3	Disagree 2	Strongly Disagree 1
1. I prefer using this system for enhancing my communication experience.	x				
2. I found the system unnecessarily complex.					x
3. I thought the system was easy to use.	x				
4. I found the system very awkward to use					x
5. I am satisfied with the speed and responsiveness of the Speech to Text with Emoji app.			x		
6. I felt frustrated when trying to					x

accomplish tasks using the system					
7. I think that the app may integrate with my preferred messaging or communication apps		x			
8. I felt this system had too many differences or changes in it.					x
9. I think that the Speech to Text with Emoji app is important for your daily communication needs		x			
10. I had to gain ample knowledge before using the system effectively.					x

Please answer all the following questions.

1. Which part of the system you like the most?

Ans: The ease of use and the variety of emojis available make the app enjoyable.

2. Which part of the system you like the least?

Ans: -

3. Any final comments or questions?

Ans: Great work! This app has made texting a more interactive experience.

Informed Consent Agreement

The purpose of this research project, led by Tong Kah Pau, a Software Engineering student at Universiti Tunku Abdul Rahman (UTAR), is to examine the practicality and user-friendliness of a system designed under the project named "Speech to Text with Emojis."

We cordially invite you to take part in this research study, which will consist of the following elements:

1. User Acceptance Test (UAT)
2. System Usability Test (SUT)

By providing your signature on this document, you confirm that:

- Your participation in this study is entirely voluntary, and you have agreed to do so without coercion.
- You are aware that involvement in this usability research is optional, and you have the right to raise any concerns or discomfort throughout the research and to withdraw at any moment.
- You acknowledge that the User Acceptance Test will necessitate your engagement with the developed system according to the instructions provided.

Name : Caleb Kumar
Signature : *Caleb*
Date : 26/3/2023

We are grateful for your involvement in this study.

Rest assured that any information you provide will be used solely for the purpose of evaluating the system's effectiveness and will not be shared or used for any other reasons.

User Acceptance Test

Test Date	26/3/2023		
Starting Time	2.30pm		
Ending Time	3.00pm		
Participant Name	Caleb Kumar		
Test Module	Test Scenario	Pass/Fail	Feedbacks
Transcribe happy speech in real time	<p>A user is happy, and they want to transcribe their speech into text with emojis in real time and plays it back.</p> <p>13. Start speech recording.</p> <p>14. Stop speech recording.</p> <p>15. View transcript text with emoji.</p> <p>16. Plays back recording.</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
Transcribe sad speech in real time	<p>A user is sad, and they want to transcribe their speech into text with emojis in real time.</p> <p>13. Start speech recording.</p> <p>14. Stop speech recording.</p> <p>15. View transcript text with emoji.</p> <p>16. Plays back recording.</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	

Transcribe pre-recorded speech angry	A user is angry, and they want to transcribe their speech into text with emojis by uploading. 13. Choose audio file to upload. 14. Upload audio file 15. Views transcript text with emoji 16. Plays back recording	Pass in terms of functionalities. Pass in terms of speech transcribed. Pass in terms of emotion recognised.	
Transcribe pre-recorded speech in real time disgust	A user is disgusted, and they want to transcribe their speech into text with emojis in real time. 13. Choose audio file to upload. 14. Upload audio file 15. Views transcript text with emoji 16. Plays back recording	Pass in terms of functionalities. Pass in terms of speech transcribed. Fail in terms of emotion recognised	
Transcribe pre-recorded speech in real time surprise	A user is disgusted, and they want to transcribe their speech into text with emojis in real time. 10. Choose audio file to upload. 11. Upload audio file 12. Views transcript text with emoji Plays back recording	Pass in terms of functionalities. Pass in terms of speech transcribed. Pass in terms of emotion recognised.	
Transcribe pre-recorded speech in real time calm	A user is calm, and they want to transcribe their	Pass in terms of functionalities.	

	<p>speech into text with emojis in real time.</p> <p>13. Choose audio file to upload.</p> <p>14. Upload audio file</p> <p>15. Views transcript text with emoji</p> <p>16. Plays back recording</p>	<p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
Transcribe neutral speech in real time	<p>A user is neutral, and they want to transcribe their speech into text with emojis in real time and plays it back.</p> <p>13. Start speech recording.</p> <p>14. Stop speech recording.</p> <p>15. View transcript text with emoji.</p> <p>16. Plays back recording.</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
Transcribe scared speech in real time	<p>A user is scared, and they want to transcribe their speech into text with emojis in real time and plays it back.</p> <p>13. Start speech recording.</p> <p>14. Stop speech recording.</p> <p>15. View transcript text with emoji.</p> <p>16. Plays back recording.</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	

System Usability Test

Participant 4 Caleb Kumar

User Satisfaction survey

Please mark 'x' to rate the following statements.

	Strongly Agree 5	Agree 4	Neutral 3	Disagree 2	Strongly Disagree 1
1. I prefer using this system for enhancing my communication experience.	x				
2. I found the system unnecessarily complex.					x
3. I thought the system was easy to use.		x			
4. I found the system very awkward to use					x
5. I am satisfied with the speed and responsiveness of the Speech to Text with Emoji app.		x			
6. I felt frustrated when trying to					x

accomplish tasks using the system					
7. I think that the app may integrate with my preferred messaging or communication apps		x			
8. I felt this system had too many differences or changes in it.					x
9. I think that the Speech to Text with Emoji app is important for your daily communication needs		x			
10. I had to gain ample knowledge before using the system effectively.					x

Please answer all the following questions.

1. Which part of the system you like the most?

Ans: Technology to transcribe speech and recognise emotion at the same time.

2. Which part of the system you like the least?

Ans: The inconsistency in recognizing certain phrases could be improved.

3. Any final comments or questions?

Ans: Keep up the good work! This app has great potential.

Informed Consent Agreement

The purpose of this research project, led by Tong Kah Pau, a Software Engineering student at Universiti Tunku Abdul Rahman (UTAR), is to examine the practicality and user-friendliness of a system designed under the project named "Speech to Text with Emojis."

We cordially invite you to take part in this research study, which will consist of the following elements:

1. User Acceptance Test (UAT)
2. System Usability Test (SUT)

By providing your signature on this document, you confirm that:

- Your participation in this study is entirely voluntary, and you have agreed to do so without coercion.
- You are aware that involvement in this usability research is optional, and you have the right to raise any concerns or discomfort throughout the research and to withdraw at any moment.
- You acknowledge that the User Acceptance Test will necessitate your engagement with the developed system according to the instructions provided.

Name : Liang Shuat Teng
Signature : *Liang*
Date : 26/3/2023

We are grateful for your involvement in this study.

Rest assured that any information you provide will be used solely for the purpose of evaluating the system's effectiveness and will not be shared or used for any other reasons.

User Acceptance Test

Test Date	26/3/2023		
Starting Time	3.00pm		
Ending Time	3.30pm		
Participant Name	Liang Shuat Teng		
Test Module	Test Scenario	Pass/Fail	Feedbacks
Transcribe happy speech in real time	<p>A user is happy, and they want to transcribe their speech into text with emojis in real time and plays it back.</p> <p>17. Start speech recording.</p> <p>18. Stop speech recording.</p> <p>19. View transcript text with emoji.</p> <p>20. Plays back recording.</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
Transcribe sad speech in real time	<p>A user is sad, and they want to transcribe their speech into text with emojis in real time.</p> <p>17. Start speech recording.</p> <p>18. Stop speech recording.</p> <p>19. View transcript text with emoji.</p> <p>20. Plays back recording.</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
Transcribe pre-recorded angry speech	<p>A user is angry, and they want to transcribe their</p>	<p>Pass in terms of functionalities.</p>	

	<p>speech into text with emojis by uploading.</p> <p>17. Choose audio file to upload.</p> <p>18. Upload audio file</p> <p>19. Views transcript text with emoji</p> <p>20. Plays back recording</p>	<p>Pass in terms of speech transcribed.</p> <p>Fail in terms of emotion recognised.</p>	
Transcribe pre-recorded disgust speech in real time	<p>A user is disgusted, and they want to transcribe their speech into text with emojis in real time.</p> <p>17. Choose audio file to upload.</p> <p>18. Upload audio file</p> <p>19. Views transcript text with emoji</p> <p>20. Plays back recording</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Fail in terms of emotion recognised</p>	
Transcribe pre-recorded surprise speech in real time	<p>A user is disgusted, and they want to transcribe their speech into text with emojis in real time.</p> <p>13. Choose audio file to upload.</p> <p>14. Upload audio file</p> <p>15. Views transcript text with emoji</p> <p>Plays back recording</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Fail in terms of emotion recognised.</p>	
Transcribe pre-recorded calm speech in real time	<p>A user is calm, and they want to transcribe their speech into text with emojis in real time.</p>	<p>Pass in terms of functionalities.</p>	

	<p>17. Choose audio file to upload.</p> <p>18. Upload audio file</p> <p>19. Views transcript text with emoji</p> <p>20. Plays back recording</p>	<p>Pass in terms of speech transcribed.</p> <p>Fail in terms of emotion recognised.</p>	
Transcribe neutral speech in real time	<p>A user is neutral, and they want to transcribe their speech into text with emojis in real time and plays it back.</p> <p>17. Start speech recording.</p> <p>18. Stop speech recording.</p> <p>19. View transcript text with emoji.</p> <p>20. Plays back recording.</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
Transcribe scared speech in real time	<p>A user is scared, and they want to transcribe their speech into text with emojis in real time and plays it back.</p> <p>17. Start speech recording.</p> <p>18. Stop speech recording.</p> <p>19. View transcript text with emoji.</p> <p>20. Plays back recording.</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	

System Usability Test

Participant 5 Liang Shuat Teng

User Satisfaction survey

Please mark 'x' to rate the following statements.

	Strongly Agree 5	Agree 4	Neutral 3	Disagree 2	Strongly Disagree 1
1. I prefer using this system for enhancing my communication experience.	x				
2. I found the system unnecessarily complex.					x
3. I thought the system was easy to use.		x			
4. I found the system very awkward to use				x	
5. I am satisfied with the speed and responsiveness of the Speech to Text with Emoji app.		x			
6. I felt frustrated when trying to accomplish					x

tasks using the system					
7. I think that the app may integrate with my preferred messaging or communication apps		x			
8. I felt this system had too many differences or changes in it.					x
9. I think that the Speech to Text with Emoji app is important for your daily communication needs	x				
10. I had to gain ample knowledge before using the system effectively.					x

Please answer all the following questions.

1. Which part of the system you like the most?

Ans: I love how this could help me enhance my texting experience.

2. Which part of the system you like the least?

Ans: -

3. Any final comments or questions?

Ans: Overall, a very useful app for daily communication.

Informed Consent Agreement

The purpose of this research project, led by Tong Kah Pau, a Software Engineering student at Universiti Tunku Abdul Rahman (UTAR), is to examine the practicality and user-friendliness of a system designed under the project named "Speech to Text with Emojis."

We cordially invite you to take part in this research study, which will consist of the following elements:

1. User Acceptance Test (UAT)
2. System Usability Test (SUT)

By providing your signature on this document, you confirm that:

- Your participation in this study is entirely voluntary, and you have agreed to do so without coercion.
- You are aware that involvement in this usability research is optional, and you have the right to raise any concerns or discomfort throughout the research and to withdraw at any moment.
- You acknowledge that the User Acceptance Test will necessitate your engagement with the developed system according to the instructions provided.

Name : Ian Cheong
Signature : *Cheong*
Date : 26/3/2023

We are grateful for your involvement in this study.

Rest assured that any information you provide will be used solely for the purpose of evaluating the system's effectiveness and will not be shared or used for any other reasons.

User Acceptance Test

Test Date	26/3/2023		
Starting Time	3.30pm		
Ending Time	4.00pm		
Participant Name	Ian Cheong		
Test Module	Test Scenario	Pass/Fail	Feedbacks
Transcribe happy speech in real time	<p>A user is happy, and they want to transcribe their speech into text with emojis in real time and plays it back.</p> <p>21. Start speech recording.</p> <p>22. Stop speech recording.</p> <p>23. View transcript text with emoji.</p> <p>24. Plays back recording.</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
Transcribe sad speech in real time	<p>A user is sad, and they want to transcribe their speech into text with emojis in real time.</p> <p>21. Start speech recording.</p> <p>22. Stop speech recording.</p> <p>23. View transcript text with emoji.</p> <p>24. Plays back recording.</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	

Transcribe pre-recorded speech angry	A user is angry, and they want to transcribe their speech into text with emojis by uploading. 21. Choose audio file to upload. 22. Upload audio file 23. Views transcript text with emoji 24. Plays back recording	Pass in terms of functionalities. Pass in terms of speech transcribed. Pass in terms of emotion recognised.	
Transcribe pre-recorded speech in real time disgust	A user is disgusted, and they want to transcribe their speech into text with emojis in real time. 21. Choose audio file to upload. 22. Upload audio file 23. Views transcript text with emoji 24. Plays back recording	Pass in terms of functionalities. Pass in terms of speech transcribed. Pass in terms of emotion recognised	
Transcribe pre-recorded speech in real time surprise	A user is disgusted, and they want to transcribe their speech into text with emojis in real time. 16. Choose audio file to upload. 17. Upload audio file 18. Views transcript text with emoji Plays back recording	Pass in terms of functionalities. Pass in terms of speech transcribed. Pass in terms of emotion recognised.	
Transcribe pre-recorded speech in real time calm	A user is calm, and they want to transcribe their	Pass in terms of functionalities.	

	<p>speech into text with emojis in real time.</p> <ol style="list-style-type: none"> 21. Choose audio file to upload. 22. Upload audio file 23. Views transcript text with emoji 24. Plays back recording 	<p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
Transcribe neutral speech in real time	<p>A user is neutral, and they want to transcribe their speech into text with emojis in real time and plays it back.</p> <ol style="list-style-type: none"> 21. Start speech recording. 22. Stop speech recording. 23. View transcript text with emoji. 24. Plays back recording. 	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
Transcribe scared speech in real time	<p>A user is scared, and they want to transcribe their speech into text with emojis in real time and plays it back.</p> <ol style="list-style-type: none"> 21. Start speech recording. 22. Stop speech recording. 23. View transcript text with emoji. 24. Plays back recording. 	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	

System Usability Test

Participant 6 Ian Cheong

User Satisfaction survey

Please mark 'x' to rate the following statements.

	Strongly Agree 5	Agree 4	Neutral 3	Disagree 2	Strongly Disagree 1
1. I prefer using this system for enhancing my communication experience.		x			
2. I found the system unnecessarily complex.					x
3. I thought the system was easy to use.	x				
4. I found the system very awkward to use					x
5. I am satisfied with the speed and responsiveness of the Speech to Text with Emoji app.		x			

6. I felt frustrated when trying to accomplish tasks using the system					x
7. I think that the app may integrate with my preferred messaging or communication apps		x			
8. I felt this system had too many differences or changes in it.					x
9. I think that the Speech to Text with Emoji app is important for your daily communication needs		x			
10. I needed to learn a lot of things before I could get going with the system					x

Please answer all the following questions.

1. Which part of the system you like the most?

Ans: The app's ability to understand speech and emotions is amazing.

2. Which part of the system you like the least?

Ans: The user interface feels a bit cluttered at times.

3. Any final comments or questions?

Ans: I'm excited to see how this app will evolve in the future!

Informed Consent Agreement

The purpose of this research project, led by Tong Kah Pau, a Software Engineering student at Universiti Tunku Abdul Rahman (UTAR), is to examine the practicality and user-friendliness of a system designed under the project named "Speech to Text with Emojis."

We cordially invite you to take part in this research study, which will consist of the following elements:

1. User Acceptance Test (UAT)
2. System Usability Test (SUT)

By providing your signature on this document, you confirm that:

- Your participation in this study is entirely voluntary, and you have agreed to do so without coercion.
- You are aware that involvement in this usability research is optional, and you have the right to raise any concerns or discomfort throughout the research and to withdraw at any moment.
- You acknowledge that the User Acceptance Test will necessitate your engagement with the developed system according to the instructions provided.

Name : Hau Han Chuan
Signature : *Hau*
Date : 26/3/2023

We are grateful for your involvement in this study.

Rest assured that any information you provide will be used solely for the purpose of evaluating the system's effectiveness and will not be shared or used for any other reasons.

User Acceptance Test

Test Date	26/3/2023		
Starting Time	4.00pm		
Ending Time	4.30pm		
Participant Name	Hau Han Chuan		
Test Module	Test Scenario	Pass/Fail	Feedbacks
Transcribe happy speech in real time	<p>A user is happy, and they want to transcribe their speech into text with emojis in real time and plays it back.</p> <p>25. Start speech recording.</p> <p>26. Stop speech recording.</p> <p>27. View transcript text with emoji.</p> <p>28. Plays back recording.</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
Transcribe sad speech in real time	<p>A user is sad, and they want to transcribe their speech into text with emojis in real time.</p> <p>25. Start speech recording.</p> <p>26. Stop speech recording.</p> <p>27. View transcript text with emoji.</p> <p>28. Plays back recording.</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	

<p>Transcribe pre-recorded speech</p> <p>pre-angry</p>	<p>A user is angry, and they want to transcribe their speech into text with emojis by uploading.</p> <p>25. Choose audio file to upload.</p> <p>26. Upload audio file</p> <p>27. Views transcript text with emoji</p> <p>28. Plays back recording</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
<p>Transcribe pre-recorded speech in real time</p> <p>pre-disgust</p>	<p>A user is disgusted, and they want to transcribe their speech into text with emojis in real time.</p> <p>25. Choose audio file to upload.</p> <p>26. Upload audio file</p> <p>27. Views transcript text with emoji</p> <p>28. Plays back recording</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
<p>Transcribe pre-recorded speech in real time</p> <p>pre-surprise</p>	<p>A user is disgusted, and they want to transcribe their speech into text with emojis in real time.</p> <p>19. Choose audio file to upload.</p> <p>20. Upload audio file</p> <p>21. Views transcript text with emoji</p> <p>Plays back recording</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
<p>Transcribe pre-recorded speech in real time</p> <p>pre-calm</p>	<p>A user is calm, and they want to transcribe their</p>	<p>Pass in terms of functionalities.</p>	

	<p>speech into text with emojis in real time.</p> <p>25. Choose audio file to upload.</p> <p>26. Upload audio file</p> <p>27. Views transcript text with emoji</p> <p>28. Plays back recording</p>	<p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
Transcribe neutral speech in real time	<p>A user is neutral, and they want to transcribe their speech into text with emojis in real time and plays it back.</p> <p>25. Start speech recording.</p> <p>26. Stop speech recording.</p> <p>27. View transcript text with emoji.</p> <p>28. Plays back recording.</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
Transcribe scared speech in real time	<p>A user is scared, and they want to transcribe their speech into text with emojis in real time and plays it back.</p> <p>25. Start speech recording.</p> <p>26. Stop speech recording.</p> <p>27. View transcript text with emoji.</p> <p>28. Plays back recording.</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	

System Usability Test

Participant 7 Hau Han Chuan

User Satisfaction survey

Please mark 'x' to rate the following statements.

	Strongly Agree 5	Agree 4	Neutral 3	Disagree 2	Strongly Disagree 1
1. I prefer using this system for enhancing my communication experience.	x				
2. I found the system unnecessarily complex.					x
3. I thought the system was easy to use.		x			
4. I found the system very awkward to use				x	
5. I am satisfied with the speed and responsiveness of the Speech to Text with Emoji app.	x				
6. I felt frustrated when trying to					x

accomplish tasks using the system					
7. I think that the app may integrate with my preferred messaging or communication apps	x				
8. I felt this system had too many differences or changes in it.					x
9. I think that the Speech to Text with Emoji app is important for your daily communication needs		x			
10. I had to gain ample knowledge before using the system effectively.					x

Please answer all the following questions.

1. Which part of the system you like the most?

Ans: The responsiveness of the app is great and it keeps up with my speech without any issues.

2. Which part of the system you like the least?

Ans: -

3. Any final comments or questions?

Ans: I would recommend this app to friends who want to enhance their communication experience.

Informed Consent Agreement

The purpose of this research project, led by Tong Kah Pau, a Software Engineering student at Universiti Tunku Abdul Rahman (UTAR), is to examine the practicality and user-friendliness of a system designed under the project named "Speech to Text with Emojis."

We cordially invite you to take part in this research study, which will consist of the following elements:

1. User Acceptance Test (UAT)
2. System Usability Test (SUT)

By providing your signature on this document, you confirm that:

- Your participation in this study is entirely voluntary, and you have agreed to do so without coercion.
- You are aware that involvement in this usability research is optional, and you have the right to raise any concerns or discomfort throughout the research and to withdraw at any moment.
- You acknowledge that the User Acceptance Test will necessitate your engagement with the developed system according to the instructions provided.

Name : Wayne Boo
Signature : *Wayne*
Date : 26/3/2023

We are grateful for your involvement in this study.

Rest assured that any information you provide will be used solely for the purpose of evaluating the system's effectiveness and will not be shared or used for any other reasons.

User Acceptance Test

Test Date	26/3/2023		
Starting Time	4.30pm		
Ending Time	5.00pm		
Participant Name	Wayne Boo		
Test Module	Test Scenario	Pass/Fail	Feedbacks
Transcribe happy speech in real time	<p>A user is happy, and they want to transcribe their speech into text with emojis in real time and plays it back.</p> <p>29. Start speech recording.</p> <p>30. Stop speech recording.</p> <p>31. View transcript text with emoji.</p> <p>32. Plays back recording.</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
Transcribe sad speech in real time	<p>A user is sad, and they want to transcribe their speech into text with emojis in real time.</p> <p>29. Start speech recording.</p> <p>30. Stop speech recording.</p> <p>31. View transcript text with emoji.</p> <p>32. Plays back recording.</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	

Transcribe pre-recorded speech angry	A user is angry, and they want to transcribe their speech into text with emojis by uploading. 29. Choose audio file to upload. 30. Upload audio file 31. Views transcript text with emoji 32. Plays back recording	Pass in terms of functionalities. Pass in terms of speech transcribed. Pass in terms of emotion recognised.	
Transcribe pre-recorded speech in real time disgust	A user is disgusted, and they want to transcribe their speech into text with emojis in real time. 29. Choose audio file to upload. 30. Upload audio file 31. Views transcript text with emoji 32. Plays back recording	Pass in terms of functionalities. Pass in terms of speech transcribed. Pass in terms of emotion recognised.	
Transcribe pre-recorded speech in real time surprise	A user is disgusted, and they want to transcribe their speech into text with emojis in real time. 22. Choose audio file to upload. 23. Upload audio file 24. Views transcript text with emoji Plays back recording	Pass in terms of functionalities. Pass in terms of speech transcribed. Pass in terms of emotion recognised.	
Transcribe pre-recorded speech in real time calm	A user is calm, and they want to transcribe their	Pass in terms of functionalities.	

	<p>speech into text with emojis in real time.</p> <p>29. Choose audio file to upload.</p> <p>30. Upload audio file</p> <p>31. Views transcript text with emoji</p> <p>32. Plays back recording</p>	<p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
Transcribe neutral speech in real time	<p>A user is neutral, and they want to transcribe their speech into text with emojis in real time and plays it back.</p> <p>29. Start speech recording.</p> <p>30. Stop speech recording.</p> <p>31. View transcript text with emoji.</p> <p>32. Plays back recording.</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
Transcribe scared speech in real time	<p>A user is scared, and they want to transcribe their speech into text with emojis in real time and plays it back.</p> <p>29. Start speech recording.</p> <p>30. Stop speech recording.</p> <p>31. View transcript text with emoji.</p> <p>32. Plays back recording.</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	

System Usability Test

Participant 8 Wayne Boo

User Satisfaction survey

Please mark 'x' to rate the following statements.

	Strongly Agree 5	Agree 4	Neutral 3	Disagree 2	Strongly Disagree 1
1. I prefer using this system for enhancing my communication experience.		x			
2. I found the system unnecessarily complex.					x
3. I thought the system was easy to use.		x			
4. I found the system very awkward to use				x	
5. I am satisfied with the speed and responsiveness of the Speech to Text with Emoji app.		x			

6. I felt frustrated when trying to accomplish tasks using the system					X
7. I think that the app may integrate with my preferred messaging or communication apps			x		
8. I felt this system had too many differences or changes in it.					x
9. I think that the Speech to Text with Emoji app is important for your daily communication needs	x				
10. I had to gain ample knowledge before using the system effectively.					x

Please answer all the following questions.

1. Which part of the system you like the most?

Ans: The wide range of emotions and emojis that the system can recognise is fantastic.

2. Which part of the system you like the least?

Ans: -

3. Any final comments or questions?

Ans: Overall, a fun and engaging app for expressing emotions in text conversations.

Informed Consent Agreement

The purpose of this research project, led by Tong Kah Pau, a Software Engineering student at Universiti Tunku Abdul Rahman (UTAR), is to examine the practicality and user-friendliness of a system designed under the project named "Speech to Text with Emojis."

We cordially invite you to take part in this research study, which will consist of the following elements:

1. User Acceptance Test (UAT)
2. System Usability Test (SUT)

By providing your signature on this document, you confirm that:

- Your participation in this study is entirely voluntary, and you have agreed to do so without coercion.
- You are aware that involvement in this usability research is optional, and you have the right to raise any concerns or discomfort throughout the research and to withdraw at any moment.
- You acknowledge that the User Acceptance Test will necessitate your engagement with the developed system according to the instructions provided.

Name : Wong Zi Ying
Signature : *Wong*
Date : 26/3/2023

We are grateful for your involvement in this study.

Rest assured that any information you provide will be used solely for the purpose of evaluating the system's effectiveness and will not be shared or used for any other reasons.

User Acceptance Test

Test Date	26/3/2023		
Starting Time	5.00pm		
Ending Time	5.30pm		
Participant Name	Wong Zi Ying		
Test Module	Test Scenario	Pass/Fail	Feedbacks
Transcribe happy speech in real time	<p>A user is happy, and they want to transcribe their speech into text with emojis in real time and plays it back.</p> <p>33. Start speech recording.</p> <p>34. Stop speech recording.</p> <p>35. View transcript text with emoji.</p> <p>36. Plays back recording.</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
Transcribe sad speech in real time	<p>A user is sad, and they want to transcribe their speech into text with emojis in real time.</p> <p>33. Start speech recording.</p> <p>34. Stop speech recording.</p> <p>35. View transcript text with emoji.</p> <p>36. Plays back recording.</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	

<p>Transcribe pre-recorded speech</p> <p>pre-angry</p>	<p>A user is angry, and they want to transcribe their speech into text with emojis by uploading.</p> <p>33. Choose audio file to upload.</p> <p>34. Upload audio file</p> <p>35. Views transcript text with emoji</p> <p>36. Plays back recording</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Fail in terms of emotion recognised</p>	
<p>Transcribe pre-recorded speech in real time</p> <p>pre-disgust</p>	<p>A user is disgusted, and they want to transcribe their speech into text with emojis in real time.</p> <p>33. Choose audio file to upload.</p> <p>34. Upload audio file</p> <p>35. Views transcript text with emoji</p> <p>36. Plays back recording</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Fail in terms of emotion recognised</p>	
<p>Transcribe pre-recorded speech in real time</p> <p>pre-surprise</p>	<p>A user is disgusted, and they want to transcribe their speech into text with emojis in real time.</p> <p>25. Choose audio file to upload.</p> <p>26. Upload audio file</p> <p>27. Views transcript text with emoji</p> <p>Plays back recording</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Fail in terms of emotion recognised</p>	
<p>Transcribe pre-recorded speech in real time</p> <p>pre-calm</p>	<p>A user is calm, and they want to transcribe their</p>	<p>Pass in terms of functionalities.</p>	

	<p>speech into text with emojis in real time.</p> <p>33. Choose audio file to upload.</p> <p>34. Upload audio file</p> <p>35. Views transcript text with emoji</p> <p>36. Plays back recording</p>	<p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
Transcribe neutral speech in real time	<p>A user is neutral, and they want to transcribe their speech into text with emojis in real time and plays it back.</p> <p>33. Start speech recording.</p> <p>34. Stop speech recording.</p> <p>35. View transcript text with emoji.</p> <p>36. Plays back recording.</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
Transcribe scared speech in real time	<p>A user is scared, and they want to transcribe their speech into text with emojis in real time and plays it back.</p> <p>33. Start speech recording.</p> <p>34. Stop speech recording.</p> <p>35. View transcript text with emoji.</p> <p>36. Plays back recording.</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	

System Usability Test

Participant 9 Wong Zi Ying

User Satisfaction survey

Please mark 'x' to rate the following statements.

	Strongly Agree 5	Agree 4	Neutral 3	Agree 2	Strongly Disagree 1
1. I prefer using this system for enhancing my communication experience.	x				
2. I found the system unnecessarily complex.					x
3. I thought the system was easy to use.	x				
4. I found the system very awkward to use					x
5. I am satisfied with the speed and responsiveness of the Speech to Text with Emoji app.		x			
6. I felt frustrated when					x

trying to accomplish tasks using the system					
7. I think that the app may integrate with my preferred messaging or communication apps	x				
8. I felt this system had too many differences or changes in it.				x	
9. I think that the Speech to Text with Emoji app is important for your daily communication needs		x			
10. I had to gain ample knowledge before using the system effectively.					x

Please answer all the following questions.

1. Which part of the system you like the most?

Ans: I appreciate how easy it is to use and how quickly it transcribes my speech.

2. Which part of the system you like the least?

Ans: I wish the app could recognise emotions for those people that couldn't convey their emotions well.

3. Any final comments or questions?

Ans: The app has been a game-changer for my daily communication needs.

Informed Consent Agreement

The purpose of this research project, led by Tong Kah Pau, a Software Engineering student at Universiti Tunku Abdul Rahman (UTAR), is to examine the practicality and user-friendliness of a system designed under the project named "Speech to Text with Emojis."

We cordially invite you to take part in this research study, which will consist of the following elements:

1. User Acceptance Test (UAT)
2. System Usability Test (SUT)

By providing your signature on this document, you confirm that:

- Your participation in this study is entirely voluntary, and you have agreed to do so without coercion.
- You are aware that involvement in this usability research is optional, and you have the right to raise any concerns or discomfort throughout the research and to withdraw at any moment.
- You acknowledge that the User Acceptance Test will necessitate your engagement with the developed system according to the instructions provided.

Name : Farha Sofia binti Muzafah
Signature : *Farha*
Date : 26/3/2023

We are grateful for your involvement in this study.

Rest assured that any information you provide will be used solely for the purpose of evaluating the system's effectiveness and will not be shared or used for any other reasons.

User Acceptance Test

Test Date	26/3/2023		
Starting Time	5.30pm		
Ending Time	6.00pm		
Participant Name	Farha Sofia binti Muzafah		
Test Module	Test Scenario	Pass/Fail	Feedbacks
Transcribe happy speech in real time	<p>A user is happy, and they want to transcribe their speech into text with emojis in real time and plays it back.</p> <p>37. Start speech recording.</p> <p>38. Stop speech recording.</p> <p>39. View transcript text with emoji.</p> <p>40. Plays back recording.</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
Transcribe sad speech in real time	<p>A user is sad, and they want to transcribe their speech into text with emojis in real time.</p> <p>37. Start speech recording.</p> <p>38. Stop speech recording.</p> <p>39. View transcript text with emoji.</p> <p>40. Plays back recording.</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	

<p>Transcribe pre-recorded speech</p> <p>pre-angry</p>	<p>A user is angry, and they want to transcribe their speech into text with emojis by uploading.</p> <p>37. Choose audio file to upload.</p> <p>38. Upload audio file</p> <p>39. Views transcript text with emoji</p> <p>40. Plays back recording</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
<p>Transcribe pre-recorded speech in real time</p> <p>pre-disgust</p>	<p>A user is disgusted, and they want to transcribe their speech into text with emojis in real time.</p> <p>37. Choose audio file to upload.</p> <p>38. Upload audio file</p> <p>39. Views transcript text with emoji</p> <p>40. Plays back recording</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Fail in terms of emotion recognised</p>	
<p>Transcribe pre-recorded speech in real time</p> <p>pre-surprise</p>	<p>A user is disgusted, and they want to transcribe their speech into text with emojis in real time.</p> <p>28. Choose audio file to upload.</p> <p>29. Upload audio file</p> <p>30. Views transcript text with emoji</p> <p>Plays back recording</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
<p>Transcribe pre-recorded speech in real time</p> <p>pre-calm</p>	<p>A user is calm, and they want to transcribe their</p>	<p>Pass in terms of functionalities.</p>	

	<p>speech into text with emojis in real time.</p> <p>37. Choose audio file to upload.</p> <p>38. Upload audio file</p> <p>39. Views transcript text with emoji</p> <p>40. Plays back recording</p>	<p>Pass in terms of speech transcribed.</p> <p>Fail in terms of emotion recognised.</p>	
Transcribe neutral speech in real time	<p>A user is neutral, and they want to transcribe their speech into text with emojis in real time and plays it back.</p> <p>37. Start speech recording.</p> <p>38. Stop speech recording.</p> <p>39. View transcript text with emoji.</p> <p>40. Plays back recording.</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	
Transcribe scared speech in real time	<p>A user is scared, and they want to transcribe their speech into text with emojis in real time and plays it back.</p> <p>37. Start speech recording.</p> <p>38. Stop speech recording.</p> <p>39. View transcript text with emoji.</p> <p>40. Plays back recording.</p>	<p>Pass in terms of functionalities.</p> <p>Pass in terms of speech transcribed.</p> <p>Pass in terms of emotion recognised.</p>	

System Usability Test

Participant 10 Farha Sofia binti Muzafah

User Satisfaction survey

Please mark 'x' to rate the following statements.

	Strongly Agree 5	Agree 4	Neutral 3	Disagree 2	Strongly Disagree 1
1. I prefer using this system for enhancing my communication experience.	x				
2. I found the system unnecessarily complex.					x
3. I thought the system was easy to use.		x			
4. I found the system very awkward to use				x	
5. I am satisfied with the speed and responsiveness of the Speech to Text with Emoji app.	x				
6. I felt frustrated when trying to				x	

accomplish tasks using the system					
7. I think that the app may integrate with my preferred messaging or communication apps	x				
8. I felt this system had too many differences or changes in it.					x
9. I think that the Speech to Text with Emoji app is important for your daily communication needs		x			
10. I had to gain ample knowledge before using the system effectively.					x

Please answer all the following questions.

1. Which part of the system you like the most?

Ans: The accuracy of the speech recognition and emoji selection is impressive.

2. Which part of the system you like the least?

Ans: The app's design could be more visually appealing.

3. Any final comments or questions?

Ans: I'm looking forward to seeing more features and improvements in future updates.

Calculation for SUS Scores

Participant 1:

Odd-numbered questions: $(5 - 1) + (5 - 1) + (4 - 1) + (5 - 1) + (4 - 1) = 4 + 4 + 3 + 4 + 3 = 18$

Even-numbered questions: $(5 - 1) + (5 - 1) + (5 - 1) + (5 - 1) + (5 - 1) = 4 + 4 + 4 + 4 + 4 = 20$

SUS Score: $(18 + 20) * 2.5 = 95$

Participant 2:

Odd-numbered questions: $(5 - 1) + (5 - 1) + (4 - 1) + (5 - 1) + (5 - 1) = 4 + 4 + 3 + 4 + 3 = 20$

Even-numbered questions: $(5 - 1) + (5 - 1) + (5 - 1) + (5 - 1) + (5 - 1) = 4 + 4 + 4 + 4 + 4 = 20$

SUS Score: $(20 + 20) * 2.5 = 100$

Participant 3:

Odd-numbered questions: $(5 - 1) + (5 - 1) + (3 - 1) + (4 - 1) + (4 - 1) = 4 + 4 + 2 + 3 + 3 = 16$

Even-numbered questions: $(5 - 1) + (5 - 1) + (5 - 1) + (5 - 1) + (5 - 1) = 4 + 4 + 4 + 4 + 4 = 20$

SUS Score: $(16 + 20) * 2.5 = 90$

Participant 4:

Odd-numbered questions: $(5 - 1) + (4 - 1) + (4 - 1) + (4 - 1) + (4 - 1) = 4 + 3 + 3 + 3 + 3 = 16$

Even-numbered questions: $(5 - 1) + (5 - 1) + (5 - 1) + (5 - 1) + (5 - 1) = 4 + 4 + 4 + 4 + 4 = 20$

SUS Score: $(16 + 20) * 2.5 = 90$

Participant 5:

Odd-numbered questions: $(5 - 1) + (4 - 1) + (4 - 1) + (4 - 1) + (5 - 1) = 4 + 3 + 3 + 3 + 4 = 17$

Even-numbered questions: $(5 - 1) + (5 - 2) + (5 - 1) + (5 - 1) + (5 - 1) = 4 + 3 + 4 + 4 + 4 = 19$
 SUS Score: $(17 + 19) * 2.5 = 90$

Participant 6:

Odd-numbered questions: $(4 - 1) + (5 - 1) + (4 - 1) + (4 - 1) + (4 - 1) = 3 + 4 + 3 + 3 + 3 = 16$
 Even-numbered questions: $(5 - 1) + (5 - 1) + (5 - 1) + (5 - 1) + (5 - 1) = 4 + 4 + 4 + 4 + 4 = 20$
 SUS Score: $(16 + 20) * 2.5 = 90$

Participant 7:

Odd-numbered questions: $(5 - 1) + (4 - 1) + (5 - 1) + (5 - 1) + (4 - 1) = 4 + 3 + 4 + 4 + 3 = 18$
 Even-numbered questions: $(5 - 1) + (5 - 2) + (5 - 1) + (5 - 1) + (5 - 1) = 4 + 3 + 4 + 4 + 4 = 19$
 SUS Score: $(18 + 19) * 2.5 = 92.5$

Participant 8:

Odd-numbered questions: $(4 - 1) + (4 - 1) + (4 - 1) + (3 - 1) + (5 - 1) = 3 + 3 + 3 + 2 + 4 = 15$
 Even-numbered questions: $(5 - 1) + (5 - 2) + (5 - 1) + (5 - 1) + (5 - 1) = 4 + 3 + 4 + 4 + 4 = 19$
 SUS Score: $(15 + 19) * 2.5 = 85$

Participant 9:

Odd-numbered questions: $(5 - 1) + (5 - 1) + (4 - 1) + (5 - 1) + (4 - 1) = 4 + 4 + 3 + 4 + 3 = 18$
 Even-numbered questions: $(5 - 1) + (5 - 1) + (5 - 1) + (5 - 1) + (5 - 1) = 4 + 4 + 4 + 4 + 4 = 20$
 SUS Score: $(18 + 20) * 2.5 = 95$

Participant 10:

Odd-numbered questions: $(5 - 1) + (4 - 1) + (5 - 1) + (5 - 1) + (4 - 1) = 4 + 3 + 4 + 4 + 3 = 18$
 Even-numbered questions: $(5 - 1) + (5 - 2) + (5 - 2) + (5 - 1) + (5 - 1) = 4 + 3 + 3 + 4 + 4 = 18$
 SUS Score: $(18 + 18) * 2.5 = 90$

Total Average SUS Score = $(95 + 100 + 90 + 90 + 90 + 90 + 92.5 + 85 + 95 + 90) / 10 = 91.75$