Jimma University

School of Post Graduate Studies

Jimma Institute of Technology

Faculty of Electrical and Computer Engineering

Control and Instrumentation Engineering

MSc. Thesis on

Artificial Neural Network Based Afan Oromo Speech Recognition System

BY

MUHIDIN MOHAMMED DEBISO

A Thesis submitted to School of Post Graduate Studies of Jimma Institute of Technology in Partial Fulfillment for the Requirement of Masters of Science in Electrical and Computer Engineering (Control and Instrumentation Engineering)

Jimma University

School of Post Graduate Studies,

Jimma Institute of Technology

Faculty of Electrical and Computer Engineering

Control and Instrumentation Engineering

TITLE: Artificial Neural Network Based Afan Oromo Speech Recognition System

BY

MUHIDIN MOHAMMED DEBISO

A Thesis submitted to School of Post Graduate Studies of Jimma Institute of Technology in Partial Fulfillment for the Requirement of Masters of Science in Electrical and Computer Engineering (Control and Instrumentation Engineering)

Advisor: Dr. A. Prashanth

Co-advisor: Mr. Fetulhak Abdurrahman

DATE: APRIL 2018

JIMMA, ETHIOPIA

Jimma University

School of Post Graduate Studies

Jimma Institute of Technology

Faculty of Electrical and Computer Engineering

(Control and Instrumentation Engineering Stream)

As member of the Examining Board of the Final MSc., Open defense, we certify that we have read and evaluated the thesis prepared by Mr. Muhidin Mohammed, Entitled**: Artificial Neural Network Based Afan Oromo Speech Recognition System**. And recommended that it be accepted as fulfilling the thesis requirement for the degree of Master of Science in Electrical and Computer Engineering under Control and Instrumentation Engineering Stream.

Name	Signature	Date
1. Dr. Henok Mulugeta External Examiner	----------	--------/--------/------
2. Dr. Amruth Ramesh Thelkar Internal Examiner	----------	--------/--------/------
3. Mr. Mengstu Fentaw Chairman	----------	--------/--------/------
4. Dr. A. Prashanth Alluvada Main advisor	----------	--------/--------/------
5. Mr. Fetulhak Abdurrahman Co-advisor	----------	--------/--------/------

Declaration

I hereby declare that, this thesis is my original work and has not been submitted as a partial requirement for a degree by any university. The related references that have been used were properly cited through the thesis.

Name of the student	Date	Signature
Muhidin Mohammed Debiso	April 2018	……

We (his advisors) hereby approved that he has done the thesis from the scratch. And we put our names and signatures for confirmation.

Name of Advisor	Date	Signature.
Dr. A. Prashanth	April 2018	……

Name of Co-Advisor	Date	Signature
Mr. Fetulhak Abdurrahman	April 2018	……

Acknowledgement

My first and foremost heartfelt gratitude goes to Almighty Allah for his keeping me in sustaining any hindrances and reaching my destination.

Secondly, I need to thank my both advisor Dr. A. Prashanth Alluvada and Co-advisor Mr. Fetulhak Abdurrahman for their restless support with no hesitation through my all duties.

Thirdly, I would like to thank my all families chiefly my lovely wife Mrs. Mebruka Shukuri for her unforgettable support and inspiration through my all activities.

Finally, the dissemination of my thanks recognize all my mates and peoples who support me in any case.

Abstract

The speech recognition system sometimes mistakenly taken as voice or speaker recognition system. However, they are different technologies. Because the speech recognition aims at understanding and comprehending what was spoken. It is used in hand-free computing, map, or menu navigation. Whereas the objective of voice or speaker recognition is to recognize who is speaking. It is used to identify a person by analyzing its tone, voice pitch, and accent. The former system has been done for different foreign languages. Especially for English language, a number of papers were produced.

On the other hand, for local languages like Afan Oromo it is still at infant stage. Though Afan Oromo may benefit from researches conducted on other languages, it also needs its own specific research since there are many grammatical and syntactical differences between languages. The thesis explored speech recognition for Afan Oromo and the possibility of its applicability.

In order to ease the way for the thesis, the 29 Afan Oromo and 5 loan phonemes were collected. Then the phonemes were grouped in 9 sentences which inturn either uttered to computer through microphone and stored in it or used in creating the sound by praat software and again stored in a computer. The system has different algorithms like receiving the Afan Oromo speech signal, preprocessing it, feature extraction, speech classification and recognizing the speech. In accomplishing these all algorithms, artificial neural network toolboxes and some scripts of MATLAB software were used.

*For developing the system, 21144 * 45 input datasets and 9*45 target datasets were made. 70% of input datasets were used for training whereas 30% of input datasets shared between validation and testing algorithms. Then confusion matrix was resulted. It shown the correctly and incorrectly classified samples.*

Out of total samples, 91.1% were perfectly classified to their corresponding classes whereas the rest 8.9% were misclassified. That is, they were classified to other classes.

Finally, the recognition ability of the system was tested by one sample of MFCC traindataset at a time. Consequently, the corresponding text form of the recognized sample was displayed.

Key words: Speech recognition, Afan Oromo, Phoneme, Artificial neural network, Speech classification, pattern Recognition.

Table contents

Contents Page

Declaration I
Acknowledgement II
Abstract III
Table contents IV
List of Tables VI
List of Figures VII
Acronyms IX
Chapter 1 1
Introduction 1
1.1. Background 1
1.2. Statement of Problem 3
1.3. Research Questions 3
1.4. Scope of the Thesis 3
1.5. Significance of the Thesis 3
1.6. Objectives 4
1.7. Methodology 4
1.8. Organizing of the Thesis 5
Chapter 2 6
Literature Review 6
2.1. Speech Recognition 6
2.2. Fundamentals of Speech Recognition 6
Chapter 3 9
Overview of Artificial Neural Network and Afan Oromo language 9
3.1. Biological neural network and its artificial models 9
3.1.1. Biological neural network 9
3.1.2. Neuron Models 10
3.2. Artificial Neural Network 12
3.2.1. Historical Development 12
3.3. Afan Oromo language 13
3.3.1. Afan Oromo Phonemes 14
3.3.2. Lengthened and shortened sounds 15
3.3.3. Geminated or ungeminated 16

3.3.4. Rules in Afan Oromo phoneme Distribution 17
Chapter 4 18
Developing Afan Oromo Speech Recognition system by Artificial Neural Network 18
4.1. Introduction 18
4.2. Hardware, Software and Analysis methods 19
4.2.1. Hardware and Software used 19
4.2.2. Analysis Methods 19
4.3. Developing Overall system 20
4.3.1. Afan Oromo speech signal preparation 20
4.3.2. Speech-acquiring stage 24
4.3.3. Speech pre- processing stage 25
4.3.4. Feature Extraction 27
4.3.5. Speech Classification 29
4.3.6. Speech Recognition 35
Chapter 5 36
Result and Discussion 36
5.1. Preprocessing Results 36
5.2. Feature Extraction Results 37
5.3. Classification Results 38
5.4. Recognition Results 39
Chapter 6 41
Conclusion and Future Work 41
6.1. Conclusion 41
6.2. Future Work 42
References 43
Appendices 45
Appendix A: Matlab source Code for developing of the system 45
Appendix B: The First 50*10 Input Dataset 53
Appendix C: Afan Oromo phonemes and their formation place 55
Appendix D: The International Phonetic Alphabet 57

List of Tables

Table 3. 1. Phonemes and their corresponding graphemes of native Qubes 15
Table 3. 2. Phonemes and their corresponding graphemes of loan qubes 15
Table 3. 3. Examples of shortening and lengthening vowels in Afan Oromo 16
Table 3. 4. Examples of geminated and ungeminated consonants of Afan Oromo 16
Table 4. 1. Confusion Matrix of the trained system 32
Table 5. 1. Classification and misclassification of samples from confusion matrix 38
Table 5. 2. Correct and wrong responses from confusion matrix 39
Table B. 1. The first 50*10 Input dataset 53
Table C. 1. The consonant phoneme lists of Afan Oromo and their formation place 55
Table C. 2. The vowel phoneme lists of Afan Oromo and their formation place 56

List of Figures

Figure 3. 1. Schematic diagram of a neuron and a sample of pulse train 10
Figure 3. 2. McCulloch-Pitts model neuron and elementary logic networks 11
Figure 4. 1. Block Diagram of the system 18
Figure 4. 2. Transcription of audio of the waliigalan alaa galan into its corresponding phonemes 20
Figure 4. 3. Transcription of audio of the barumsi hundee qaroomaati into its corresponding phonemes 21
Figure 4. 4. Transcription of audio of the dachaasaan barataa kutaa keenyaati into its corresponding phonemes 21
Figure 4. 5. Transcription of audio of the lakkoofsi kopheesaa afurtamadha into its corresponding phonemes 22
Figure 4. 6. Transcription of audio of the caalaan hiriyaa isheeti into its corresponding phonemes 22
Figure 4. 7. Transcription of audio of the xalayaan ergama qaba into its corresponding phonemes 23
Figure 4. 8. Transcription of audio of the har'a guyyaan meeqa into its corresponding phonemes 23
Figure 4. 9. Transcription of audio of the Poostaan televizyiiniirra jira into its corresponding phonemes 24
Figure 4. 10. Transcription of audio of the Zeeroo fi Tsaggaan walinbeekan into its corresponding phonemes 24
Figure 4. 11. Digitizing audio signal 24
Figure 4. 12. Speech Pre-processing illustration 26
Figure 4. 13. Audio and MFCC of waliigalan alaa galan 27
Figure 4. 14. Audio and MFCC of Barumsi hundee qaroomaati 28
Figure 4. 15. Audio and MFCC of Dachaasaan barataa kutaa keenyaati 28
Figure 4. 16. Audio and MFCC of Lakkoofsi kopheesaa meeqa 29
Figure 4. 17. Sigmoid function 30
Figure 4. 18. Neural Network of the system 30
Figure 4. 19. Training the neural network 31

Figure 4. 20. ROC of the trained system 33
Figure 4. 21. Flowchart of Speech Recognition Stage..... 35
Figure 5. 1. Raw audios and their corresponding pre-processed signals…….....36
Figure 5. 2. Feature Extracted from the first two audio files……...37
Figure 5. 3. Feature Extracted from the second two audio files……...37

Acronyms

A/D:	Analog to Digital
ANN:	Artificial Neural Network
ASR:	Automatic Speech Recognition
CD:	Compact Disk
DTW:	Dynamic Time Warping
FECE:	Faculty of Electrical and Computer Engineering
FFT:	Fast Fourier Transform
HMM:	Hidden Markov Model
JiT:	Jimma Institute of Technology
JU:	Jimma University
LPC:	Linear Predictive Coding
MATLAB:	Mathematics Laboratory
MFCC:	Mel Frequency Cepstral Coefficient
NN:	Neural Network
SR:	Speech Recognition
STT:	Speech to Text
Vd:	Voiced
Vl:	Voiceless
VQ:	Vector Quantization

Chapter 1

Introduction

1.1. Background

The speech recognition system sometimes wrongly taken as voice or speaker recognition system. However, they are different technologies. Because the speech recognition aims at understanding and comprehending what was spoken. It is used in hand-free computing, map, or menu navigation. Whereas the objective of voice or speaker recognition is to recognize who is speaking. It is used to identify a person by analyzing its tone, voice pitch, and accent.

Speech recognition can be a complicated problem. It begins with lungs producing airflow and air pressure. This pressure then vibrates the vocal chords, which separate the airflow into audible pulses. The muscles of the larynx then adjust the length and tension of the vocal cords to change the pitch and tone of the sound produced. The vocal tract, consisting of the tongue, palate, cheek and lips then articulate and filter the sound. Each instance of speech that occurs after this process is unique. Voices can vary in volume, speed, pitch, roughness, tone and other different aspects. Due to different cultural backgrounds, voices can also differ in terms of accent, articulation, and pronunciation. All of these differences make the implementation of speech recognition a very challenging problem [1].

However, in today's society when technology is consistently striving for a hands-free or voice driven implementation, speech recognition can be a very useful tool. With speech being such a fascinating phenomenon of the human body, many different properties of speech end up being unique for each person. This leads to diverse options when considering implementing a speech recognition system.

The speech has many characteristics. They include vocabulary size and confusability, speaker dependence vs. independence, isolated, discontinuous, or continuous speech and read vs spontaneous speech. Speech recognition is more efficient with a small vocabulary size, a single speaker, an isolated word and read speech (reference word). With large number of words, multiple speakers, continuous and spontaneous speech; the amount of confusion and percentage of error increase [1].

The speech recognition system has been done for different foreign languages. Especially for English language, a number of papers were produced.

On the other hand, for local languages like Afan Oromo it is still at infant stage. Though Afan Oromo may benefit from researches conducted on other languages, it also needs its own specific research since there are many grammatical and syntactical differences between languages. This research explored speech recognition for Afan Oromo and the possibility of its applicability.

An artificial neural network (or simply a neural network) is a biologically inspired computational model that consists of processing elements (neurons) and connections between them, as well as of training and recall algorithms [2]. The structure of an artificial neuron is defined by inputs, having weights bound to them; an input function, which calculates the aggregated net input signal to a neuron coming from all its inputs; an activation (signal) function, which calculates the activation level of a neuron as a function of its aggregated input signal and (possibly) of its previous state. An output signal equal to the activation value is emitted through the output (the axon) of the neuron. Neural networks are also called connectionist models owing to the main role of the connections. The weights bound to them are a result of the training process and represent the long-term memory of the model [2]. The main characteristics of a neural network are:

a) Learning: a network can start with no knowledge and can be trained using a given set of data examples, that is, input-output pairs (a supervised training), or only input data (unsupervised training); through learning, the connection weights change in such a way that the network learns to produce desired outputs for known inputs; learning may require repetition.
b) Generalization: if a new input vector that differs from the known examples is supplied to the network, it produces the best output according to the examples used.
c) Massive potential parallelism: during the processing of data, many neurons fire simultaneously.
d) Robustness: if some neurons go wrong, the whole system may still perform well.
e) Partial match: it is what is required in many cases as the already known data do not coincide exactly with the new facts

These main characteristics of neural networks make them useful for knowledge engineering. They can be used for building expert systems and have been applied to almost every application area, where a data set is available and a good solution is sought. They can survive with noisy data, missing data, imprecise or corrupted data, and still produce a good solution. It is the connectionist

model for natural language processing, speech recognition, pattern recognition, image processing, and so forth [2].

1.2. Statement of Problem

There are a number of studies regarding to speech recognition system for different languages of the world. Among them English has got more chances of studying by many scholars. Consequently, different English language speech recognition applications are developed. Nevertheless, many local languages of our country have no such sophisticated speech recognition systems. For instance, Afan Oromo language. Still, the learning and developing different applications of foreign languages are ongoing. In the future, this may reverses the learning direction. I.e. the machine will learn the human languages and produce something accordingly. So a great attention is needed in preparing the local languages for coming technologies. As a beginning, this thesis selected the Afan Oromo and paved the way for developing its speech recognition system. That means the purpose of this research in general is to study how computer can process Afan Oromo like it do for English language.

1.3. Research Questions

How to apply speech-processing system to afan Oromo like for English? Is there any suitable software for developing a system by artificial neural network? Are there a fixed number of phonemes for afan Oromo? What are they? How can we process them for speech recognition?

1.4. Scope of the Thesis

The thesis was mainly confined to finding phonemes of Afan Oromo language, making the sentences to group those all phonemes, taking audio records of sentences and processing them by artificial neural network toolboxes. By this fashion, the study shown how the speech recognition system can be developed for Afan Oromo. The study duration was from May 2017 G.C to April 1 2018G.C.

1.5. Significance of the Thesis

The thesis can contribute many inputs for Afan Oromo language and the researchers for farther aligning of the language to the new applications. It enables the language in communicating with different languages. That is, once it developed the speech recognition system, it will encourage researchers to go for studying the translating system of Afan Oromo to other languages. It also will initiate other researchers like phonologists since it raised phonemes of the languages.

This implies that the thesis will ease the process of changing the auditory explanation to text record. It will also increase the speed of recording (no need of spending the time for touching the key board). This will also decrease costs (i.e.it will save costs for typing, energy needed to record, typewriter and facility needed for typist). From point of view of dramatically growing of current technologies, future machines will learn speech to communicate with human beings instead of human beings learn computer languages or operations as usual. Hence, the thesis will give a great chance for Afan Oromo language in preparing itself for coming technologies.

The other importance of this thesis is that it will enable the language to be internationally known. For instance taking this thesis as the base, the google speech recognition for Afan Oromo can be developed by other study.

Generally, the thesis has many significances in developing the language with sciences and technologies.

1.6. Objectives

1.6.1. General objective

The general objective of the thesis is to develop Artificial Neural Network Based Afan Oromo Speech Recognition System.

1.6.2. Specific objectives

The specific objectives of the thesis are the following.

1. Collecting Afan Oromo phonemes
2. Grouping those phonemes in some sentences and taking their audios
3. Developing the MATLAB source code to analyze those audios data
4. Creating the neural network
5. Training, validating and testing the neural network by the audio data
6. Testing the neural network for recognition of new audio input data

1.7. Methodology

Early of the thesis started by reading the related literatures. After a while, parallel to continuation of reading, the practicing of MATLAB script was also began. Then in order to ease the way for the thesis, the 29 Afan Oromo and 5 loan phonemes were collected. Then the phonemes were grouped within 9 sentences which inturn either uttered to computer through microphone and stored in it or used in creating the sound by praat software and again stored in a computer. The system has different algorithms like receiving the Afan Oromo speech signal, preprocessing it, feature

extraction, speech classification and recognizing the speech. In accomplishing these all algorithms, artificial neural network toolboxes and some scripts of MATLAB software were used.

1.8. Organizing of the Thesis

The rest of the document parts are as follows:

Chapter 2: discusses the literature review.

Chapter 3: overviews the ANN and Afan Oromo language.

In Chapter 4, the main part of the thesis, which is the developing of Afan Oromo Speech recognition system by Artificial Neural Network, is going to be raised.

Chapter 5: discusses the result and discussion whereas

Chapter 6: completes the thesis by conclusion and recommending the future work.

Then the document is finalized by appending the references and appendices to it.

Chapter 2

Literature Review

2.1. Speech Recognition

Speech Recognition is the inter-disciplinary sub-field of computational linguistics that develops methodologies and technologies that enables the recognition and translation of spoken language into text by computers. It is also known as ASR, computer speech recognition, or just STT. It incorporates knowledge and research in the linguistics, computer science, and electrical engineering fields [3].

Some Speech Recognition systems use training where an individual speaker reads text or isolated vocabulary into the system. The system analyzes the person's specific voice and uses it to fine-tune the recognition of that person's speech, resulting in increased accuracy. SR systems can be speaker dependent or independent depending on whether it is using training algorithms or not. Systems, which use training, are known as speaker dependent whereas which do not are speaker independent systems [3].

Speech recognition applications include voice user interfaces such as voice dialing, call routing, domestic appliance control, simple data entry, preparation of structured documents, speech-to-text processing and aircraft [3]. The term voice recognition or speaker identification [4]refers to identifying the speaker, rather than what they are saying. Recognizing the speaker can simplify the task of translating speech in system that has been trained on a specific person's voice or be used to authenticate or verify the identity of a speaker as part of a security process.

2.2. Fundamentals of Speech Recognition

Speech recognition is multi-leveled pattern recognition task. It examines acoustical signals and structured into a hierarchy of phonemes, words, phrases, and sentences. Each level may provide additional temporal constraints such as known pronunciations or legal word sequences [5].

There are different basics of SR system. Some of them are raw speech, signal analysis, speech frame etc.

The speech is sampled at a standard frequency of 16 KHz over a microphone. This sampling produces a sequence of amplitude values over time. To simplify the recognition procedure, sampled raw speech must be converted and compacted. This can be done by different signal

analysis techniques, which extract features from raw speech and compress the data without loss of it. For instance, Fourier analysis, Perceptual Linear Prediction, Linear Predictive Coding and Cepstral analysis. They properly process the raw speech into a more usable state. Then the audio is broken up into frames, which is typically 10ms intervals of the processed audio. This provides unique information relative to the speech recognition process [5].

Before this study, some papers had been done regarding to English speech recognition systems. However, as to our knowledge there were no such tangible published studies for local languages. This implies that, there are many burdens waiting the researchers who need to study speech recognition system of local languages particularly for Afan Oromo.

The studies of some authors regarding to the English Speech Recognition systems are summarized as follows:

B.C. Kamble [1] wrote a review on Speech Recognition Using ANN. The study discussed speech recognition processes that include speech, speech pre-processing, feature extraction, speech classification and the techniques such as MFCC, LPC, HMM, DTW and VQ. The study also overviewed the ANN with its types, merits and demerits. Finally, the study was concluded by suggesting as it inspires the research group working on Automatic Speech Recognition.

However, the study has no any experiments or simulations that justified what had been discussed. It also has no the speech acquiring unit that is used in conversion of analog input speech to digital signal. Hence, it needs justification through experiments.

A. Murphy [5] conducted the study on Implementing Speech Recognition with ANN. The study discussed NN, SR, design of NN and its implementation using MATLAB.

However, the implemented system has no mechanisms for removing unwanted (noisy) signals from the speech signal. Therefore, this gap must be filled by adding speech-preprocessing algorithm.

Shikha Gupta et al [6] studied the classification of speech, its recognition and modelling techniques. As conclusion, it was found that MFCC is widely used for feature Extraction and VQ is better over DTW. Finally, it recommended that as comprehensive investigation, both experimental and theoretical of the problem should be done for better results and for making the system more robust. Therefore, there is the gaps that needs the further justification both interms of experimental and theoretical activities; like adding the simulation of the system.

Takialddin Al Smadi et al [7] discussed the different approaches for pattern recognition, types of NN, recognition processes and some equations for calculating of NN outputs. Finally, concluded as model of speech recognition was based on artificial neural networks and was investigated to develop a learning neural network using genetic algorithm. Again, this also has no any simulation to find the results that might show the reality of the discussion. This implies that, as the study has to be filled.

As a summary of the literature reviewed, it is obvious that there are many gaps those need to be filled through successive research. The aim of this thesis is therefore to fill those gaps.

Chapter 3

Overview of Artificial Neural Network and Afan Oromo language

This chapter is going to review the two main parts of the thesis; which are artificial neural network and Afan Oromo language. It starts with overviewing of biological neural network. Aligned with a biological neural network their artificial models will also be discussed here.

3.1. Biological neural network and its artificial models

3.1.1. Biological neural network

A human brain comprises of approximately 10^{11} neurons. For each neuron, there are approximately 10^4 synapses. They communicate by means of electrical impulses through a connection network of these synapses and axons. They operate in a chemical environment that is even more important in terms of actual brain behavior. We thus can consider the brain to be a densely connected electrical switching network conditioned largely by the biochemical processes [8].

The enormous NN has an elaborate structure with very complex interconnections. The input to the network is provided by sensory receptors. Receptors deliver stimuli both from within the body as well as from sense organs when the stimuli originate in the external world. The stimuli are in the form of electrical impulses that convey the information into the network of neurons. Due to information processing in the central nervous systems, the effectors are controlled and give human responses in the form of various actions. This process has three organs. Such as receptors (sensory organs), NN (central nervous system) and effectors (motor organs). They control the organism and its actions [8].

The fundamental building block of the NN is known as a neuron. It is the elementary nerve cell. Schematically it is shown in Figure 3.1. It has three major regions. They are the cell body (soma), axon, and dendrites. The collection of dendrites create dendritic tree. It is a very fine bush of thin fibers around the neuron's body. They receive information from neurons through axons long fibers. These fibers serve as transmission lines. The axon is a long hollow connection, which carries impulses from the neuron. The closure part of an axon splits into a fine tree-like structure. Each branch of it ends in a small endbulb almost touching the dendrites of neighboring neurons by a connection organ known as a synapse. The synapse is the place where the neuron announces its signal to the adjacent neuron. The electrical impulses are the signals reaching it and received by

dendrites. Even though interneuron transmission is electrical but is usually caused by the release of chemical transmitters at the synapse [8].

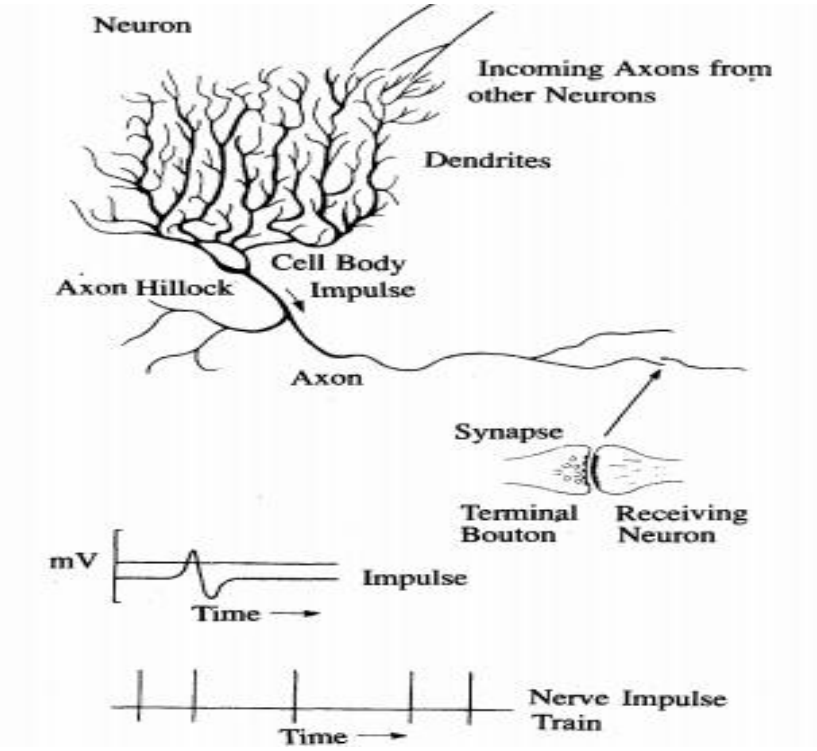


Figure 3. 1. Schematic diagram of a neuron and a sample of pulse train [8]

Here, there are two terminals such as terminal boutons and receiving neuron. The terminal boutons produce the chemical that affects the receiving neuron. The receiving neuron either produces an impulse to its axon, or harvests no response. The neuron's response is generated if the total potential of its membrane reaches a certain level. Incoming impulses can be excitatory or inhibitory depending on whether they cause or hinder the firing of the response respectively. In firing the response, the excitation exceeded the inhibition by the amount called the threshold of the neuron typically a value of about 40mV [8].

3.1.2. Neuron Models

Based on the highly simplified considerations of the biological model, the first formal definition of a synthetic neuron model was formulated by McCulloch and Pitts. Its model is shown in Figure 3.2a. The inputs x_i, for i = 1, 2. . . n, are 0 or 1, depending on the absence or presence of the input impulse at instant k. The neuron's output signal is denoted as *o*. The firing rule for this model is defined as follows:

$$o^{k+1} = \begin{cases} 1, & if \ \sum_{i=1}^{n} w_i {x_i}^k \geq T \\ 0, & if \ \sum_{i=1}^{n} w_i {x_i}^k < T \end{cases} \quad (3.1)\ [8]$$

where k = 0, 1, 2… denotes the discrete-time instant, w_i is the multiplicative weight connecting the i[th] input with the neuron's membrane and *T* is the neuron's threshold value, which needs to be exceeded by the weighted sum of signals for the neuron to fire [8].

Examples of three-input NOR and NAND gates using the McCulloch-Pitts neuron model are shown in Figure 3.2(b) and (c). We can easily inspect the implemented functions by compiling a truth table for each of the logic gates shown in the figure [8].

A single neuron with a single input *x* and with the weight and threshold values both of unity, computes $o^{k+1} = x^k$. Such a simple network thus behaves as a single register cell able to retain the input for one period elapsing between two instants. Consequently, once a feedback loop is closed around the neuron as shown in Figure 3.2(d), we obtain a memory cell. An excitatory input of 1 initializes the firing in this memory cell, and an inhibitory input of 1 initializes a non-firing state. The output value, at the absence of inputs, is then continued indefinitely. This is because the output of 0 fed back to the input does not cause firing at the next instant, while the output of 1 does [8].

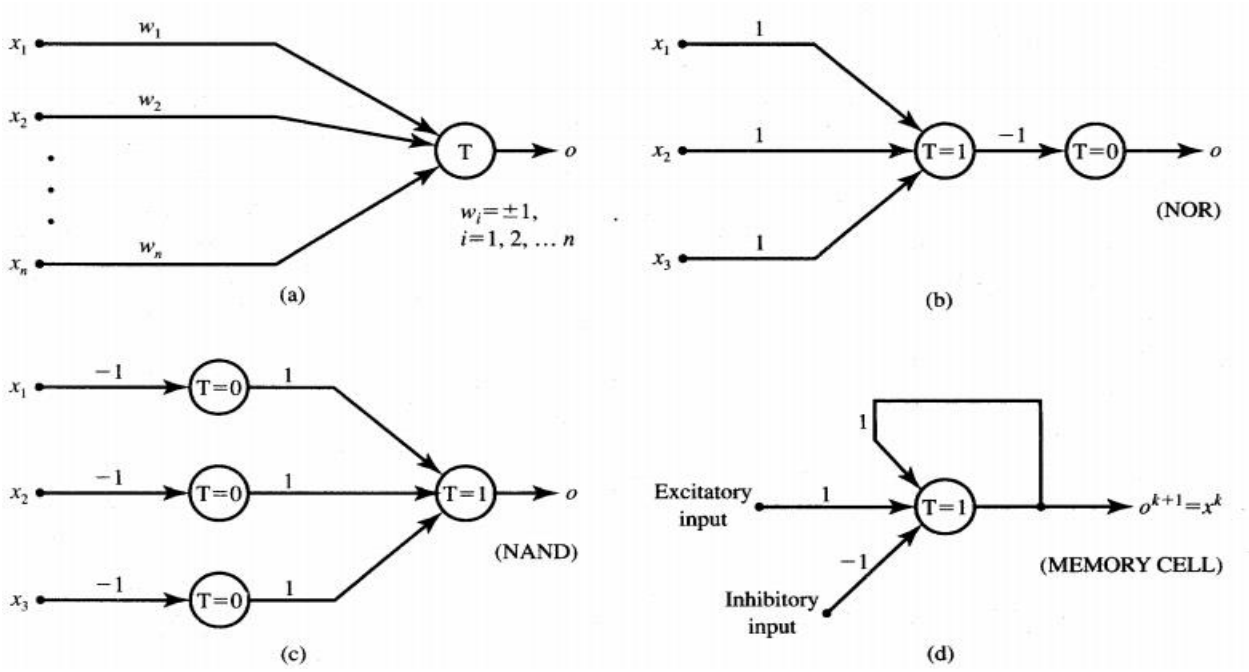


Figure 3. 2. McCulloch-Pitts model neuron and elementary logic networks

(a) Model diagram, (b) NOR gate, (c) NAND gate, and (d) memory cell [8].

3.2. Artificial Neural Network

Neural networks are composed of simple elements operating in parallel. Biological nervous systems inspire these elements. As in nature, the connections between elements largely determine the network function. You can train a neural network to perform a particular function by adjusting the values of the connections (weights) between elements [8].

Typically, neural networks are adjusted or trained. So that a particular input leads to a specific target output. Here based on a comparison of the output and the target the network is adjusted until its output matches the target. Typically, many such input/target pairs are needed to train a network. Neural networks have been trained to perform complex functions in various fields including pattern recognition, identification, classification, speech, vision and control systems. They can also be trained to solve problems that are difficult for conventional computers or human beings. The toolbox emphasizes the use of neural network paradigms that build up to or are themselves used in engineering, financial, and other practical applications [8].

3.2.1. Historical Development

The modern study of neural networks actually began in the 19th century, when neurobiologists first began extensive studies of the human nervous system. Cajal determined that the nervous system is comprised of discrete neurons, which communicate with each other by sending electrical signals down their long axons, which ultimately branch out and touch the dendrites (receptive areas) of thousands of other neurons, transmitting the electrical signals through synapses (points of contact, with variable resistance). This basic picture was elaborated on in the following decades, as different kinds of neurons were identified, their electrical responses were analyzed, and their patterns of connectivity and the brain's gross functional areas were mapped out. While neurobiologists found it relatively easy to study the functionality of individual neurons (and to map out the brain's gross functional areas), it was extremely difficult to determine how neurons worked together to achieve high-level functionality, such as perception and cognition. With the advent of high-speed computers, however, it finally became possible to build working models of neural systems, allowing researchers to freely experiment with such systems and better understand their properties [9].

McCulloch and Pitts proposed the first computational model of a neuron, namely the binary threshold unit, whose output was either 0 or 1 depending on whether its net input exceeded a given threshold. This model caused a great deal of excitement, for it was shown that a system of such

neurons, assembled into a finite state automaton, could compute any arbitrary function, given suitable values of weights between the neurons [9].

Researchers soon began searching for learning procedures that would automatically find the values of weights enabling such a network to compute any specific function. Rosenblatt discovered an iterative learning procedure for a particular type of network, the single- layer perceptron, and he proved that this learning procedure always converged to a set of weights that produced the desired function, as long as the desired function was potentially computable by the network. This discovery caused another great wave of excitement, as many AI researchers imagined that the goal of machine intelligence was within reach [9].

However, in a rigorous analysis, Minsky and Papert showed that the set of functions potentially computable by a single-layer perceptron is actually quite limited, and they expressed pessimism about the potential of multi-layer perceptrons as well; as a direct result, funding for connectionist research suddenly dried up, and the field lay dormant for 15 years [9].

Interest in neural networks was gradually revived when Hopfield suggested that a network can be analyzed in terms of an energy function, triggering the development of the Boltzmann Machine which is a stochastic network that could be trained to produce any kind of desired behavior from arbitrary pattern mapping to pattern completion. Soon thereafter, Rumelhart et al popularized a much faster learning procedure called backpropagation, which could train a multi-layer perceptron to compute any desired function, showing that Minsky and Papert's earlier pessimism was unfounded. With the advent of backpropagation, neural networks have enjoyed a third wave of popularity and have now found many useful applications [9].

3.3. Afan Oromo language

Afan Oromo is a highly spoken language in Ethiopia as per its speakers. It is found at the top of the list of the distinct and separate languages used in Africa. It is classified as one of the Cushitic languages spoken in the Ethiopian Empire, Somalia, Sudan, Tanzania, and Kenya. Its native speakers Oromo are the largest ethnic group. Oromia is their fertile region in Ethiopia. The region is located between 2 and 12 N and 34 and 44 East [10].

Afan Oromo language is written with a Latin alphabet called Qube. It was formally adopted in 1991 [10]. There are 34 Qubes (letters) including the loaned ones (p, ts, v, z, zh). Such as a, b, c, ch, d, dh, e, f, g, h, i, j, k, l, m, n, ny, o, p, ph, q, r, s, sh, t, ts, u, v, w, x, y, z, zy and ʔ. Among these 34 qubes, five of them are vowels namely a, e, i, o and u whereas twenty-nine are consonants.

They include b, c, ch, d, dh, f, g, h, j, k, l, m, n, ny, p, ph, q, r, s, sh, t, ts, v, w, x, y, z, zh and ʔ. All qubes have its corresponding upper case letters like English language except the last letter (ʔ). In Afan Oromo it is called "Hudha (')" which is glottal stop in English.

In addition to these 34 symbols, a learner of Oromo writing system will have to be taught the principles that:

1. Two vowels in succession indicate that the vowel is long, e.g. Gaarii (Good);
2. Gemination (a doubling of a same consonant) is phonemic in Oromo, e.g. damee (branch), dammee (sweet potato);
3. h is not geminated;
4. The same word can have two or more forms depending on its context, e.g. nama kadhu (ask people) namaa kadhu (ask for people);
5. Understandably, instead of accent signs, the combined Latin letters ch, dh, ny, ph, sh, ts and zh (zy) are used so as to align them with typewriter characters.

The Latin alphabet was adapted to many languages such as Germanic languages (English, German, Swedish, Danish, Norwegian and Dutch), Romance languages (Italian, French, Spanish, Portuguese and Rumanian), Slavonic languages (Polish, Czech, Croatian and Sloven), Finno-Ugrian languages (Finnish and Hungarian), Baltic languages (Lithuanian and Lettish), Somali, Swahili and etc. [10].

Qube Afan Oromo also aligned itself with so many countries that use the Latin script. One obvious advantage of this is that an oromo child who has learned his own alphabet can learn English script in a relatively short period. Another is the adaptability to computer technology, which gives alphabetic writing an edge over even the simplest of syllabic writing [10].

3.3.1. Afan Oromo Phonemes

Phoneme is an indivisible unit of sound (phone) in a given language. Alternatively, it is an abstraction of a physical speech sounds.

As discussed above out of 34 Afan Oromo's qubes; five of them are the loaned qubes. This implies that there are 29 native qubes and the corresponding phonemes are there in Afan Oromo language. Afan Oromo has 29 segmental phonemes. Among these, five of them are vowels ('dubbachiiftuu') and the remaining 24 are consonants ('dubbifamaa'). The 29 phoneme lists of the afan Oromo language are given as table 3.1.

Table 3. 1. Phonemes and their corresponding graphemes of native Qubes [11]

S. Nº	Graphemes (qubes)	Phonemes	S. Nº	Graphemes (letters)	Phonemes	S. Nº	Graphemes (letters)	Phonemes
1	a	/a/	11	i	/i/	21	r	/r/
2	B	/b/	12	j	/dʒ/	22	s	/s/
3	c	/tʃ'/	13	k	/k/	23	sh	/ʃ/
4	ch	/tʃ/	14	l	/l/	24	t	/t/
5	d	/d/	15	m	/m/	25	u	/u/
6	dh	/ɗ/	16	n	/n/	26	w	/w/
7	e	/e/	17	ny	/ɲ/	27	x	/t'/
8	f	/f/	18	o	/o/	28	y	/j/
9	g	/g/	19	ph	/ Φ/	29	'	/ʔ/
10	h	/h/	20	q	/k'/			

In afan Oromo, five sounds are used only in loan words, but are not parts of the phoneme lists of the language. Table 3.2 gives these phonemes and their corresponding graphemes.

Table 3. 2. Phonemes and their corresponding graphemes of loan qubes [11]

Serial Nº	Graphemes (qubes)	Phonemes
1	p	/p/
2	v	/v/
3	z	/z/
4	zy	/ʒ/
5	ts	/s'/

As can be seen from the tables, both consonants and vowels are regular in their grapheme representations [11].

In afan Oromo sounds can be

a) Lengthened or shortened

b) Geminated or ungeminated

3.3.2. Lengthened and shortened sounds

The lengthened sounds have the same two successive vowels in a word. Whereas in shortened sound there is no more than one vowels of the same type, appeared next to each other in a word. In Afan Oromo language, the lengthened sound is known as “dhera (long)” whereas the shortened one is called “gababa (short)”. The following table 3.3 shows the examples of shortening and lengthening vowels:

Table 3. 3. Examples of shortening and lengthening vowels in Afan Oromo

Vowel shortening			Lengthening		
Vowel(s)(phonemes)	Word(s)	meaning	Vowel(s)(phonemes)	Word(s)	Meaning
/a/	Hara	Lake	/aa/	haaraa	New
/a/	Ana	I	/aa/	aanaa	District
/i/, /a/	Bira	Near	/ii/, /aa/	biiraa	Beer
/a/	Laga	River	/aa/	laagaa	Palate

3.3.3. **Geminated** or **ungeminated**

The afan Oromo sound geminated when the same two consonants appear next to each other in a word. The geminated sound is called “jabaa (strong)” and whereas ungeminated one is known as “lafa (weak)”.

Gemination is not obligatorily marked for digraphs like ch, dh, ny, ph, sh. However, some writers indicate it by doubling the first element like in the word 'qopphaa'uu' meaning be prepared.

Examples of geminated and ungeminated Afan Oromo’s sounds are given as table 3.4.

Table 3. 4. Examples of geminated and ungeminated consonants of Afan Oromo

Ungeminated Consonants			Geminated consonants		
Consonant(s)	Word(s)	meaning	Vowel(s)(phonemes)	Word(s)	Meaning
/t/	Bitaa	left	/tt/	Bittaa	to buy
/l/	Balaa	accident	/ll/	Ballaa	Blind
/r/	Bira	near	/rr/	Birraa	autumn

Vowel lengthening and consonant gemination are phonemic in Afan Oromo language.

3.3.4. Rules in Afan Oromo phoneme Distribution

1. More than two similar or different consecutive consonant phonemes are not allowed.
2. In afan Oromo, consonant clusters and gemination at the beginning and ending of words are forbidden.
3. Diphthongs or more different vowel phonemes are not allowed in a word.
4. More than two different or similar vowels are not allowed in a word [11].

Chapter 4

Developing Afan Oromo Speech Recognition system by Artificial Neural Network

4.1. Introduction

In order to develop this system, the different algorithms are required. They include Afan Oromo speech signal inputting, speech acquiring, speech pre-processing, feature extraction, speech classification and recognition algorithms. By putting them one after the other, the block diagram of the system can be made as figure 4.1.

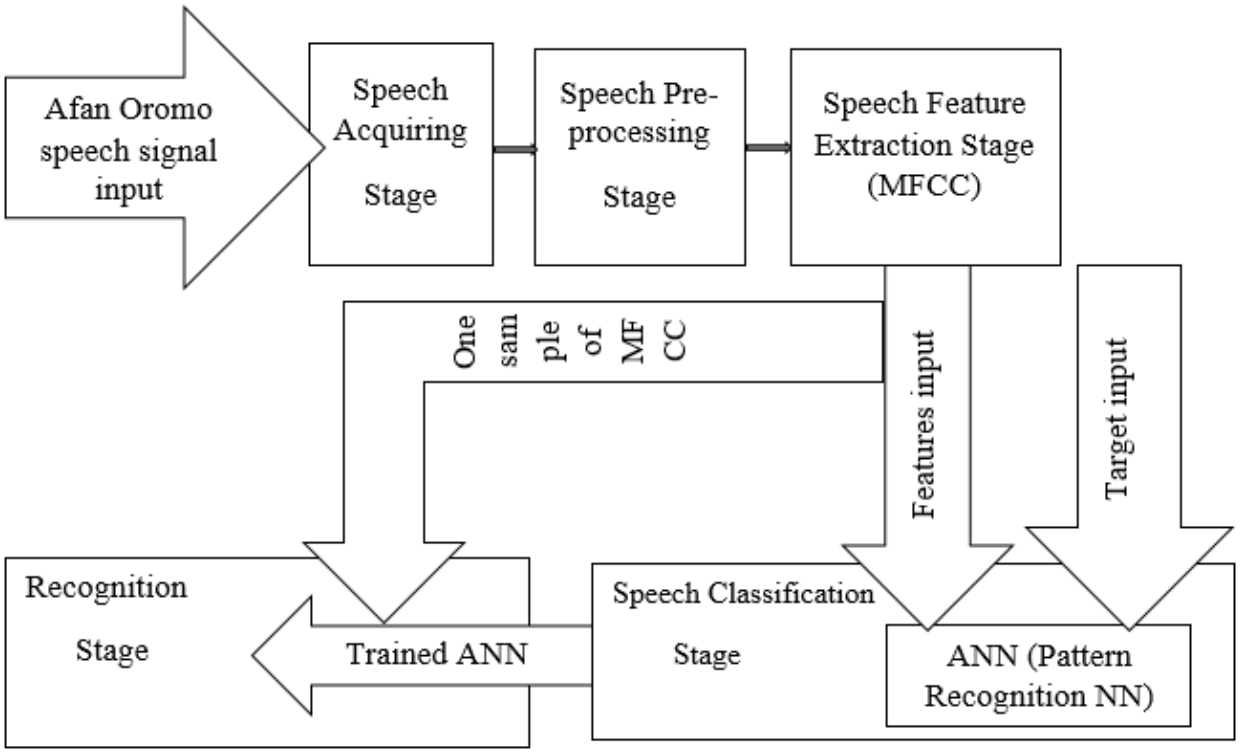


Figure 4. 1. Block Diagram of the system

As shown in fig.4.1, the speech-acquiring unit receives the speech signal(s) and converts to digital signals. Secondly, the speech pre-processing stage removes unwanted signals like noise, echoes etc. Feature extraction stage is another stage, which extracts the feature(s) of the previous stage's outputs. Then speech signals are easily classified and finally recognized.

4.2. Hardware, Software and Analysis methods

In developing this system, different hardware, software and analysis have been used.

4.2.1. Hardware and Software used

The hardware parts of this system include microphone (headphone), sound card and computer whereas the software were Praat and MATLAB. The Praat software was used in recording or creation and transcripting audios in to their corresponding phonemes. Again, it can also analysis the audio interms of its pitchs, spectra, intensities and son on. The MATLAB was mainly used in recording audios and writing the scripts that was used for reading, training, testing and validating the speech audios dataset.

4.2.2. Analysis Methods

The analysis methods of the thesis to achieve its outputs are collecting the phonemes of Afan Oromo language, grouping the phonemes within some sentences, uttering those sentences and recording their audios using speech-acquiring unit, pre-processing the speech, transcripting those audios into their corresponding phonemes by feature extraction algorithm, speech classification, and recognition.

As we have discussed in previous chapter there are 29 qubes with their 29 corresponding phonemes in afan Oromo language. In order to get those phonemes from audios, we grouped the qubes within the following sentences:

1) Waliigalan alaa galan (being agree together is being save oneself).
2) Barumsi hundee qaroomaati (Education is the base of civilization).
3) Dachaasaan barataa kutaa keenyaati (Dachasa is our class's student).
4) Lakkoofsi kopheesaa afurtamadha (his shoes number is forty).
5) Caalaan hiriyaa isheeti (chala is her friend).
6) Xalayaan ergama qaba (a letter has a mission).
7) har'a guyyaan meeqa (what is the date today)

There are also loan qubes (letters) in Afan Oromo language. They are five in numbers. Such as p, v, z, ts and zh. They can also mapped to their own phonemes. To see their phonemes separately, we made the following sentences that have grouped them together.

8) Poostaan televizyiiniirra jira (Post is on the television)
9) Zeeroo fi Tsaggaan walinbeekan (zero and Tsega do not know each other)

4.3. Developing Overall system

The overall system contained different algorithms as sub-blocks. All sub blocks are connected in cascade fashion. This implies that the output of the first sub-block is used as the input of the next one. Therefore, we preferred to develop each sub-block separately as discussed in the following sub-sections.

4.3.1. Afan Oromo speech signal preparation

There are two options in preparation of speech signals. They are (a) recording audios by uttering the already made sentences in section 4.2.2 or (b) synthesizing speech audios by inputting those sentences into Praat software and storing in computer.

In recording them, we need to have microphone, soundcard and different speakers for each sentences. Here we can use Praat or MATLAB software with appropriate script. However, in synthesizing audios the Praat software is appropriate for Afan Oromo language. Because it has different speakers options with Afan Oromo phoneme pronunciations. Again, Praat software has the ability of transcripting the synthesized audios to their appropriate phonemes. Beyond its long duration, manually splitting recorded audios to their phonemes is also possible.

Consecutively, in this thesis the Praat software was used in showing the audios signals with their corresponding phonemes as the following figures.

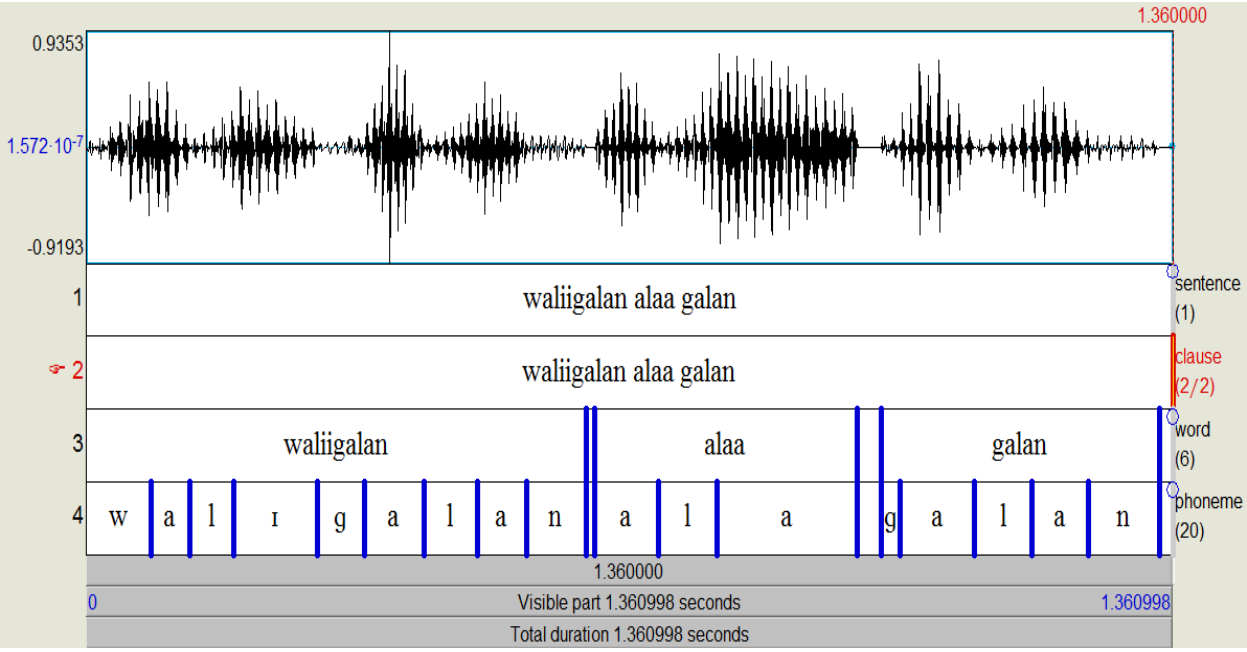


Figure 4. 2. Transcription of audio of the waliigalan alaa galan into its corresponding phonemes

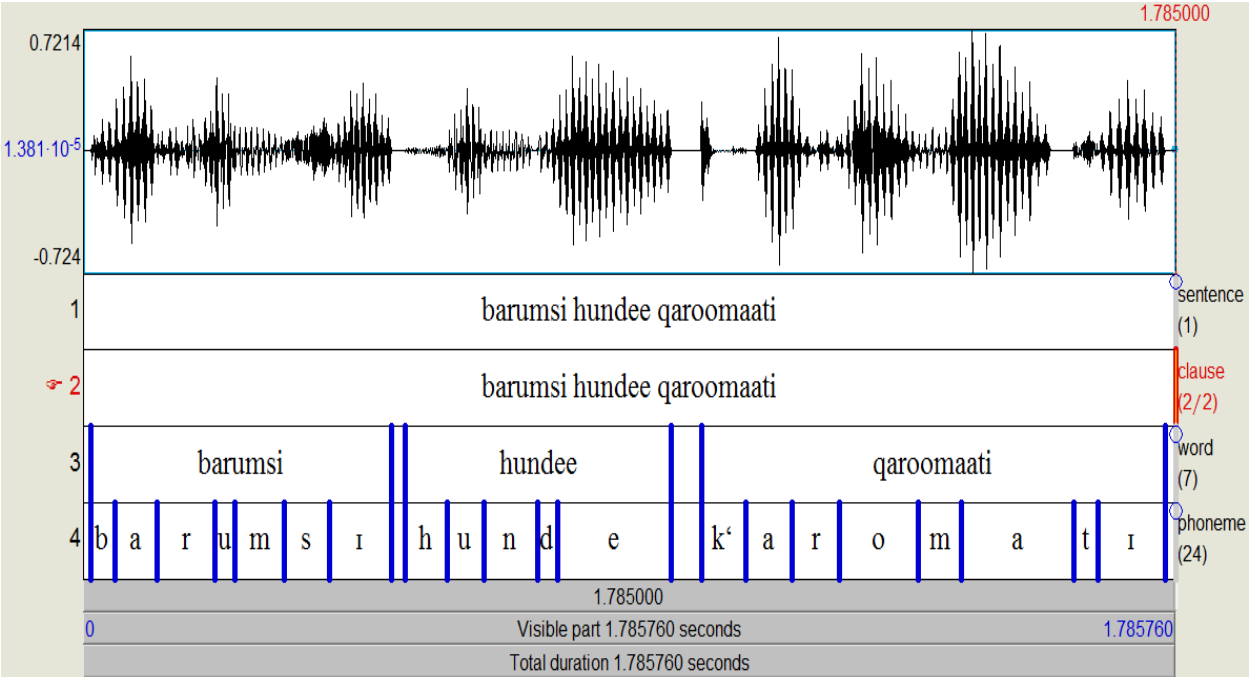


Figure 4. 3. Transcription of audio of the barumsi hundee qaroomaati into its corresponding phonemes

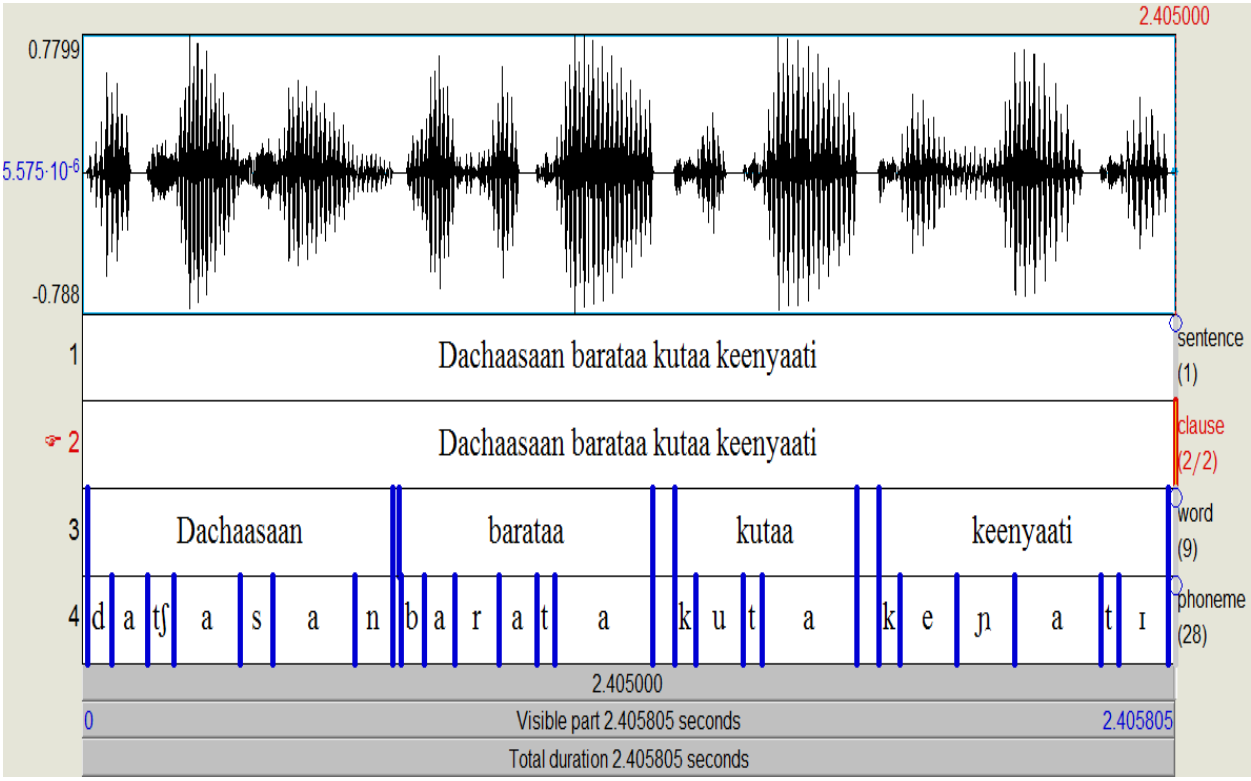


Figure 4. 4. Transcription of audio of the Dachaasaan barataa kutaa keenyaati into its corresponding phonemes

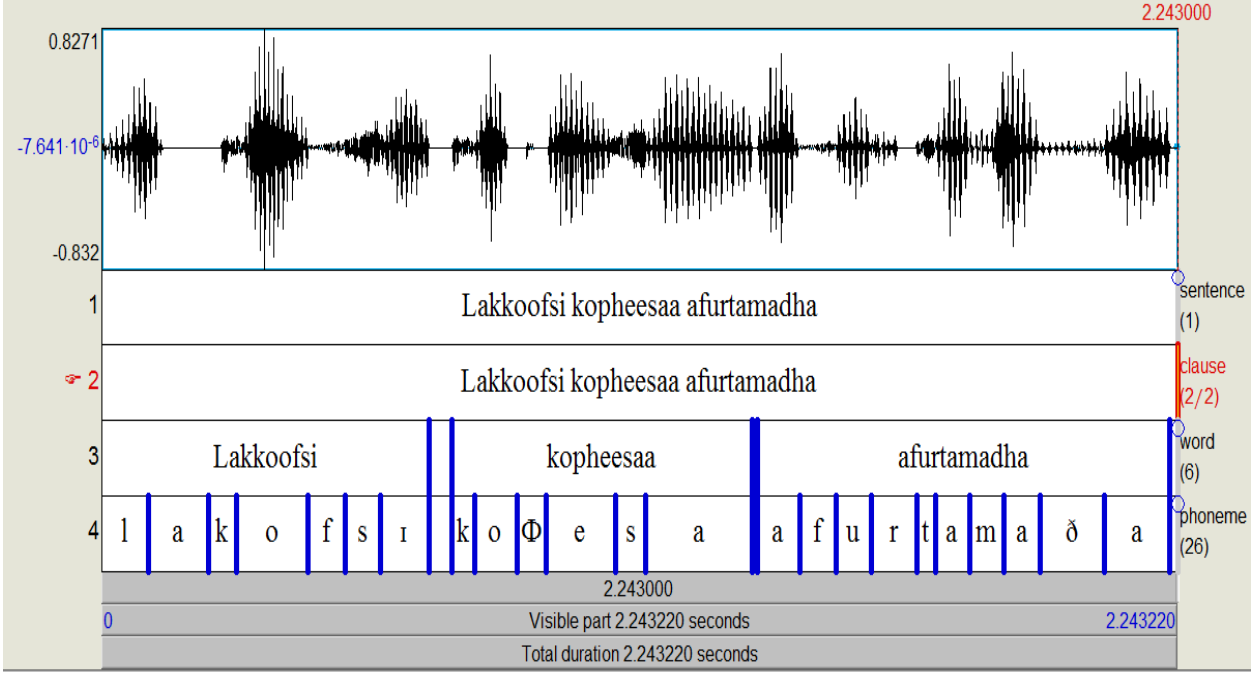


Figure 4. 5. Transcription of audio of the lakkoofsi kopheesaa afurtamadha into its corresponding phonemes

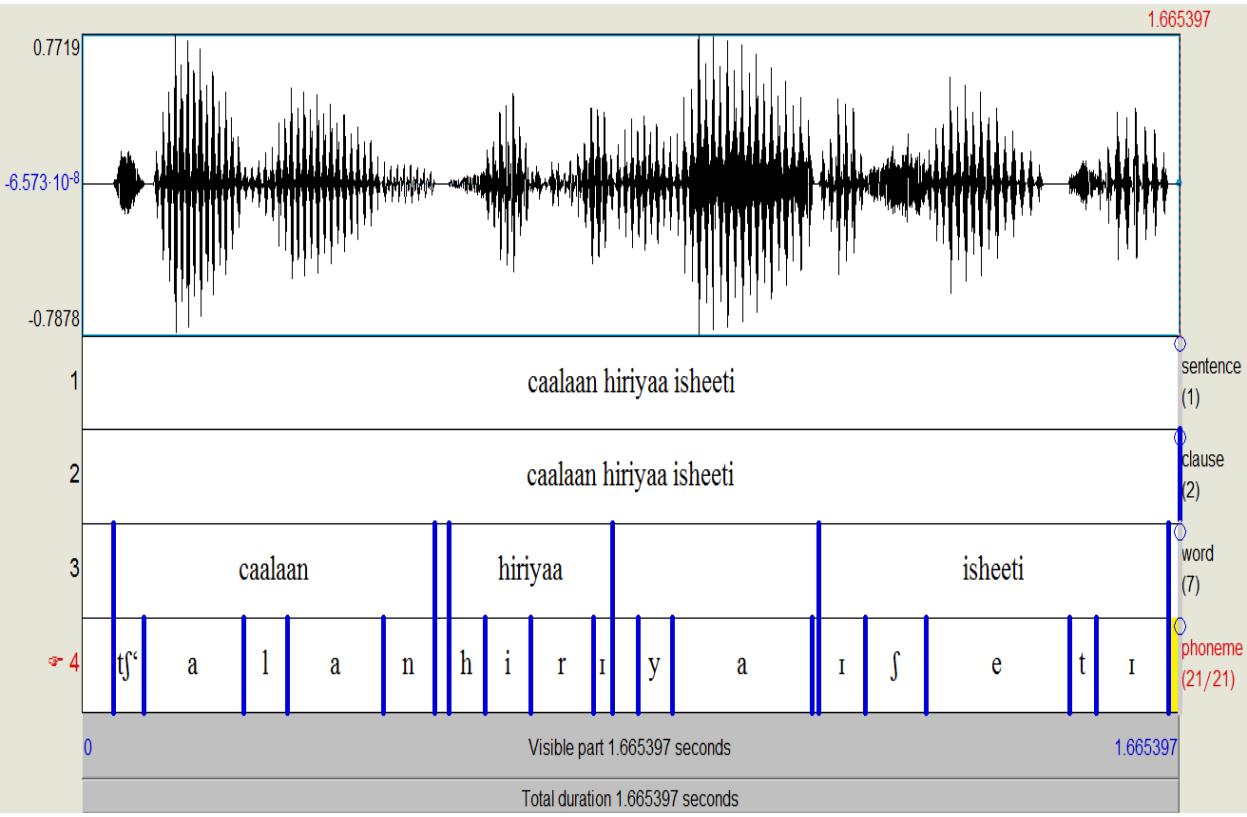


Figure 4. 6. Transcription of audio of the caalaan hiriyaa isheeti into its corresponding phonemes

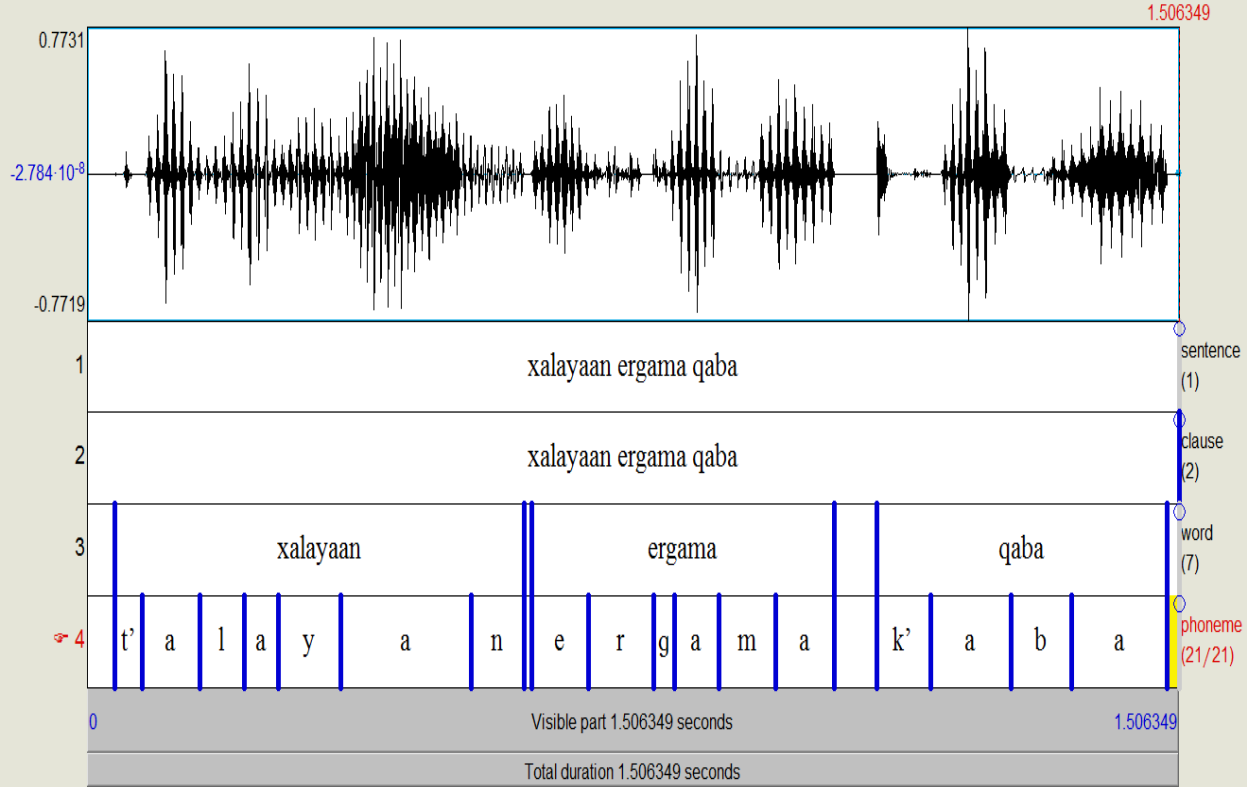


Figure 4. 7. Transcription of audio of the xalayaan ergama qaba into its corresponding phonemes

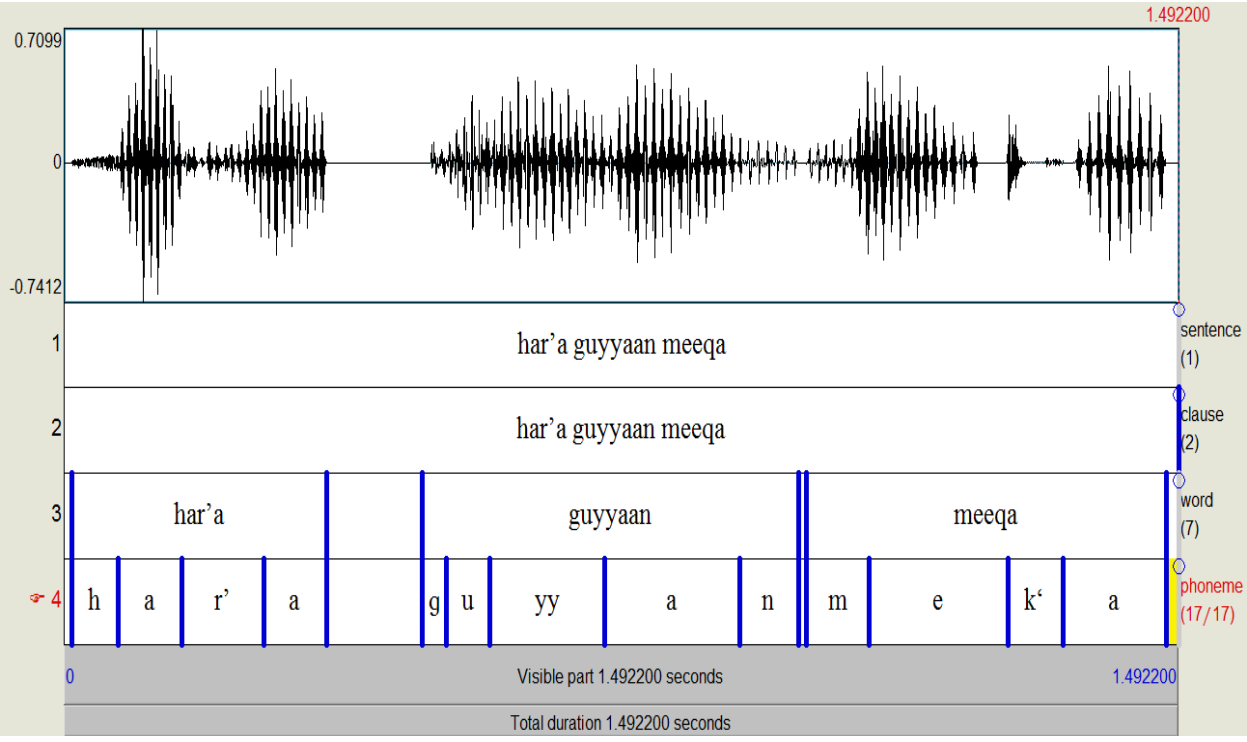


Figure 4. 8. Transcription of audio of the har’a guyyaan meeqa into its corresponding phonemes

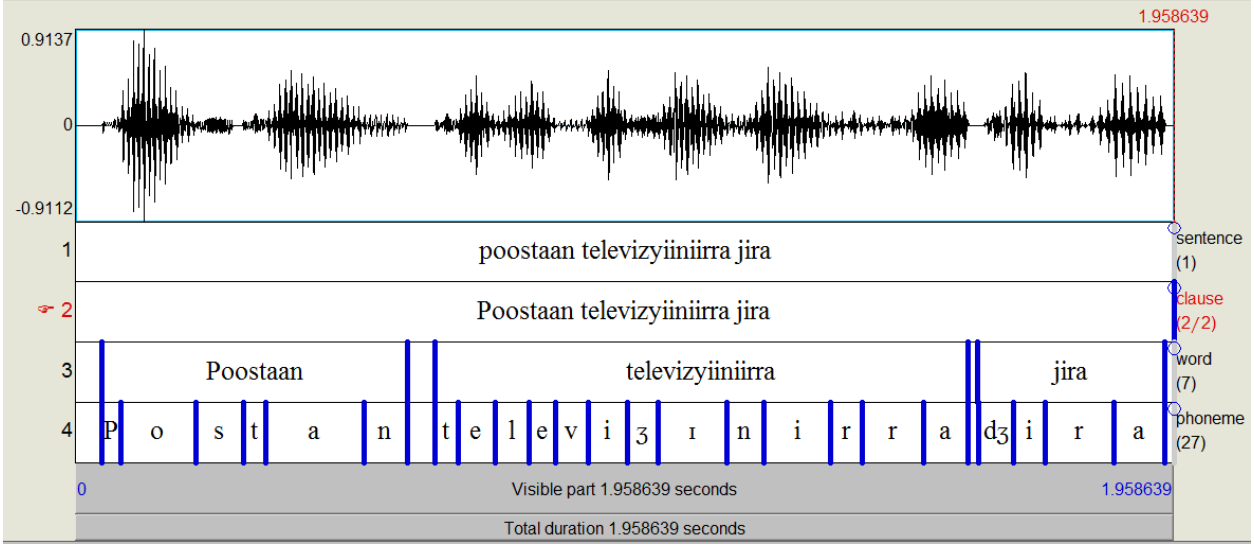


Figure 4. 9. Transcription of audio of the Poostaan televizyiiniirra jira into its corresponding phonemes

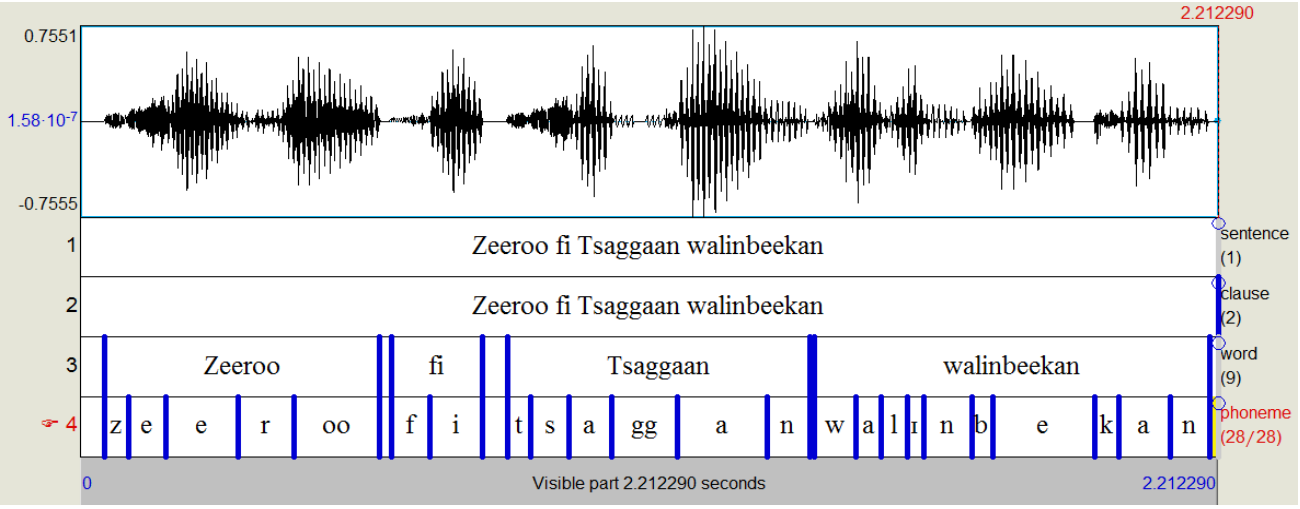


Figure 4. 10. Transcription of audio of the Zeeroo fi Tsaggaan walinbeekan into its corresponding phonemes

4.3.2. Speech-acquiring stage

The speech-acquiring stage is used in receiving and storing audio signals. It contains A/D converter and storage system. This is shown by figure 4.11.

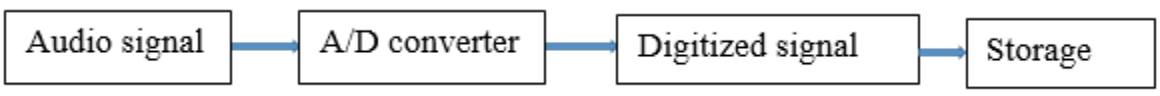


Figure 4. 11. Digitizing audio signal

Sound waves are converted to electrical signals by a microphone. If the sounds are already stored in a CD or computer, its output is available in an output jack of all players. Then this signal is fed to an appropriate A/D converter. By appropriate, its means if we want to digitize a telephone conversation, we use A/D converter which gives 8-bits digital output and samples the input signal 6700 times/second. If the signal to be digitized is high fidelity audio, we use an A/D converter that uses 16-bit signal output for each value sampled by it. The number of samples per second per audio channel will be 16000 per second. The digital output is stored in storage unit [12].

In our case, the electronic circuit called A/D converter used in digitizing audio signals is built into an electronic circuit called soundcard. Sound card is an add-on card on the motherboard of a PC. It has the audio input jacks that need appropriate connector. Then personal computer hard disk is used as the storage unit for digitized audio signal. The file has .wav extension and is called as a wave file.

In order to directly record audio signal to MATLAB software, the following scripts were used.

```
audio1=audiorecorder (16000, 16, 1); % creating audio object
% with sampling freq=16000Hz, 16bits and channel 1
record (audio1, 5); % recording audio1 for 5 seconds
get1=getaudiodata (audio1); %converting audio recorder data to double datatype
```

4.3.3. Speech pre- processing stage

Pre-processing of speech signals segregates the voiced region from the silence/unvoiced portion of the captured signal. This is usually encouraged as a crucial step in the development of a reliable speech or speaker recognition system. This is because most of the speech or speaker specific attributes are present in the voiced part of the speech signals; moreover, extraction of the voiced part of the speech signal by marking and/or removing the silence and unvoiced region leads to substantial reduction in computational complexity at later stages [13].

The preprocessing stage in speech recognition systems is used in order to increase the efficiency of subsequent feature extraction and classification stages and therefore to improve the overall recognition performance. Commonly the preprocessing includes the sampling step, a windowing and a denoising (filtering) step. At the end of the preprocessing, the compressed and filtered speech frames are forwarded to the feature extraction stage.

The audio data were already sampled in the previous section. Hence, the two common preprocessing steps that are windowing and pre-emphasis (high-pass) filtering are applied to

speech waveforms. The following is the MATLAB script for pre-processing of the first two audio datasets.

```
% Reading the audiorecords
m1=audioread ('Oromo_Male1.wav');
m2=audioread ('Oromo_Male2.wav');
% Apply a pre-emphasis filter. The pre-emphasis filter is a high-pass all-pole filter.
preemph = [1 0.9];
% filters the data in vector m with the filter described by numerator
% coefficient vector 1 and denominator coefficient vector preemph
m1f=filter (1, preemph, m1);
m2f=filter (1, preemph, m2);
% Window the speech segment using a Hamming window
m1w = m1f.*hamming (length (m1f));
m2w = m2f.*hamming (length (m2f));
```

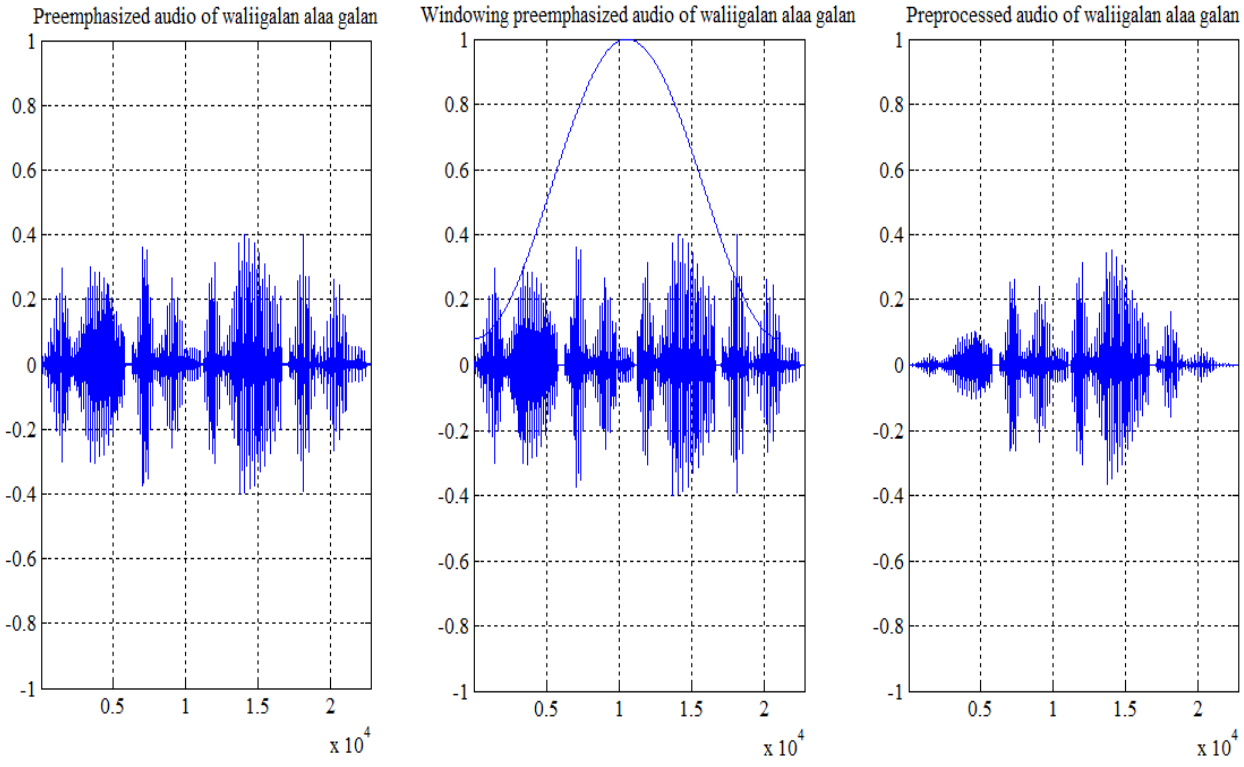


Figure 4. 12. Speech Pre-processing illustration

4.3.4. Feature Extraction

The digitized signal consists of a stream of periodic signals sampled at 16000 times per second and is not suitable to carry out actual speech recognition process, as the pattern cannot be easily located. To extract the actual information, the signal in time domain is converted to signal in frequency domain. This is done by MFCCs.

MFCC is the Mel-frequency Cepstral Coefficients are a feature widely used in automatic speech and speaker recognition. The Mel scale relates perceived frequency, or pitch, of a pure tone to its actual measured frequency. The formula for converting from frequency to Mel scale is:

$$M(f) = 1125 * \ln\left(1 + \frac{f}{700}\right) = 2591 * \log\left(1 + \frac{f}{700}\right) \quad (4.1)\ [14]$$

The reverse conversion is

$$M^{-1}(m) = f = 700\left(e^{\left(\frac{m}{1125}\right)} - 1\right) \quad (4.2)\ [14]$$

The MATLAB script of MFCC is the following:

```
% Extract features of filtered dataset by MFCC

m1c=2591.*log(1.+m1w(1:sl)/700);
m2c=2591.*log(1.+m2w(1:sl)/700);
```

The script was repeated for all filtered audio dataset of all samples. Then the features extracted and their corresponding original audios of first four sentences could be shown by the following figures.

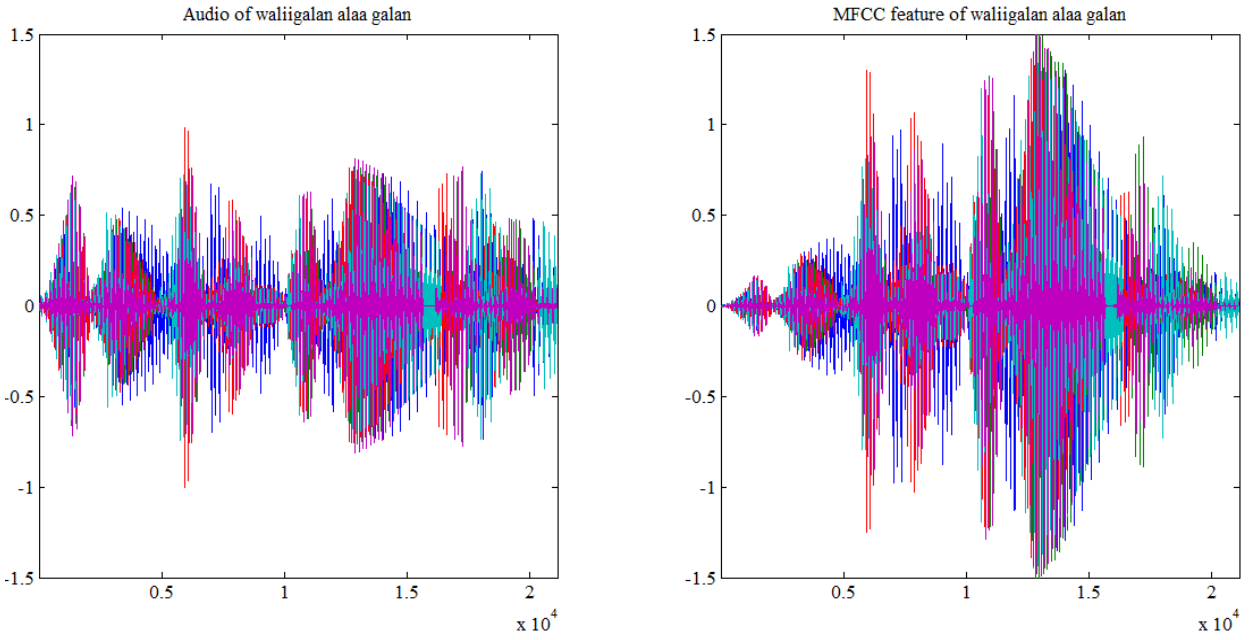


Figure 4. 13. Audio and MFCC of waliigalan alaa galan

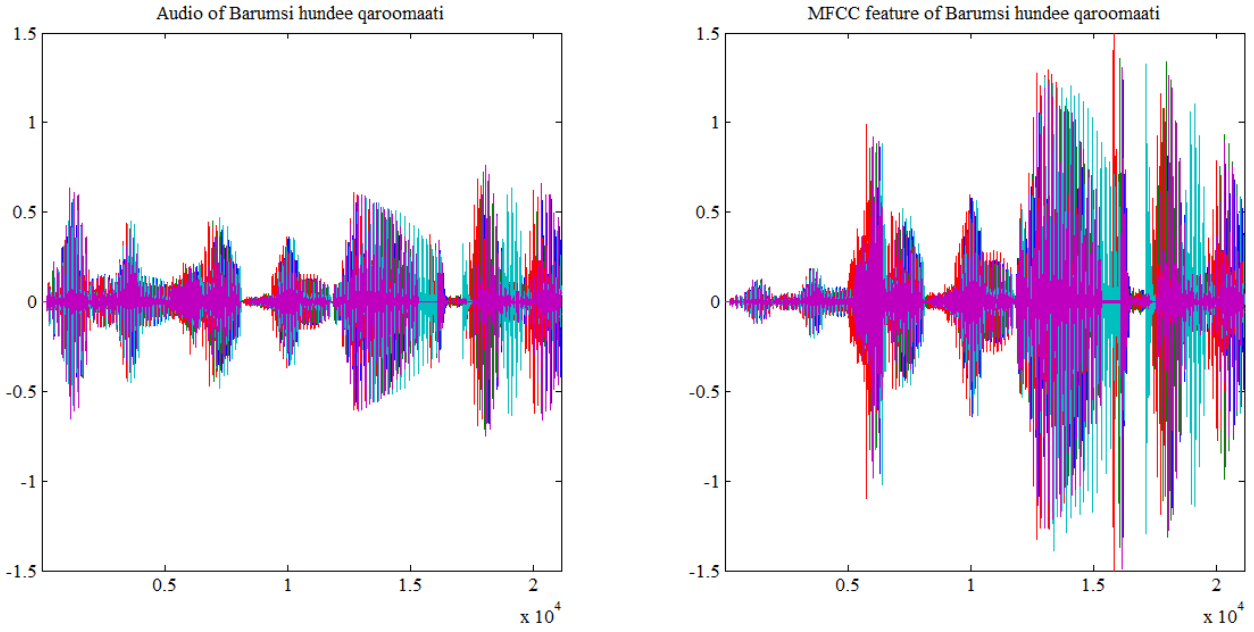


Figure 4. 14. Audio and MFCC of Barumsi hundee qaroomaati

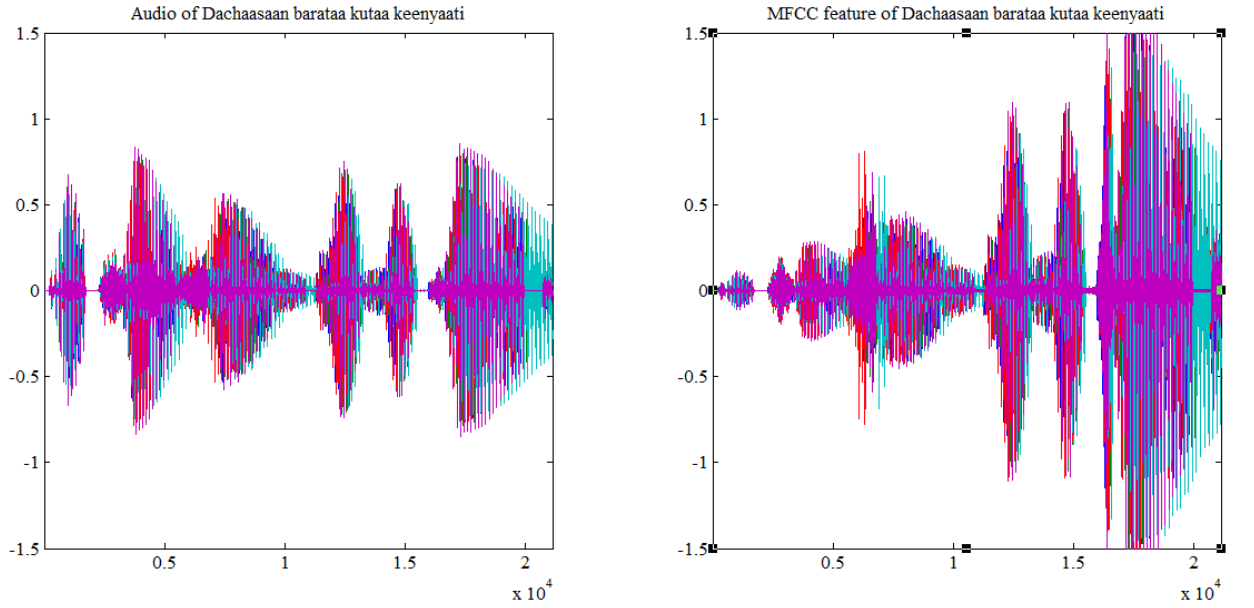


Figure 4. 15. Audio and MFCC of Dachaasaan barataa kutaa keenyaati

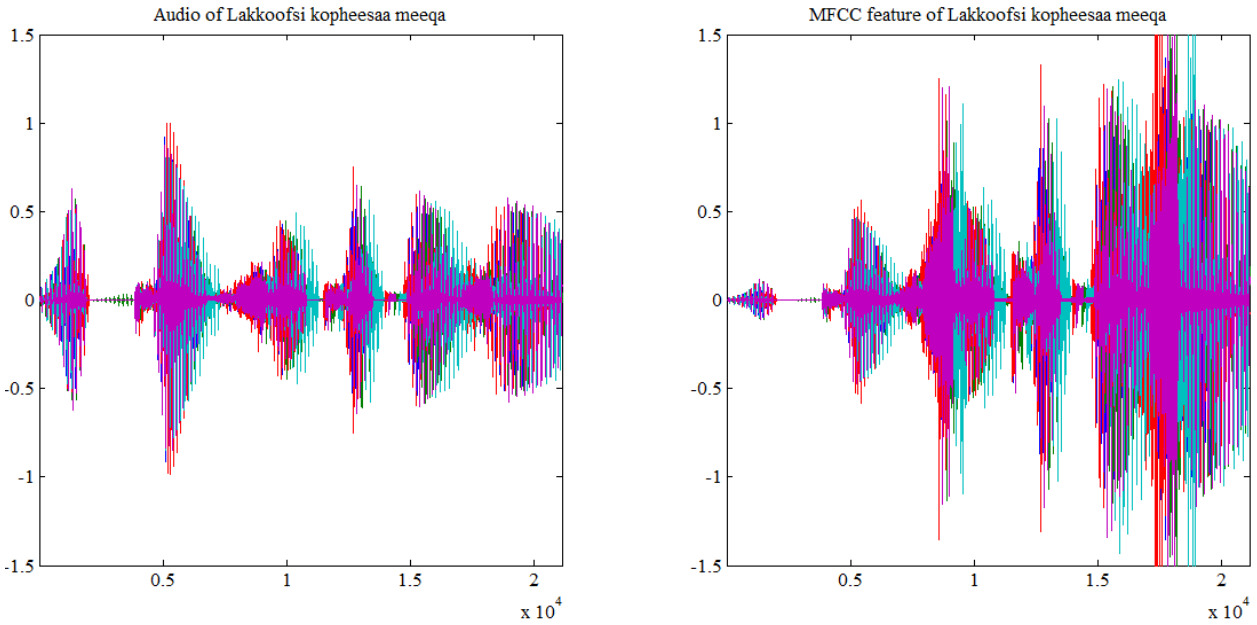


Figure 4. 16. Audio and MFCC of Lakkoofsi kopheesaa meeqa

4.3.5. Speech Classification

This is a stage where the speech is classified under its corresponding classes or labels. Alternatively, it is known as pattern matching block because it matches or classifies the pattern from feature extraction block to its most related classes according to the target inputs.

In this thesis, the neural network pattern recognition tool of MATLAB was used. The following steps were followed to classify input datasets.

1. Write nprtool on MATLAB command window; it will open GUI of NN a two-layer network with sigmoid hidden and softmax output neurons.

a) Sigmoid function

The sigmoid function is characterized by S-shaped curve (sigmoid curve). It is given by the following formula.

$$S(x) = \frac{1}{1+e^{-x}} = \frac{e^{x}}{e^{x+1}} \quad (4.3)\ [15]$$

It is used in backpropagation neural network. The created system network replaces the variable x.

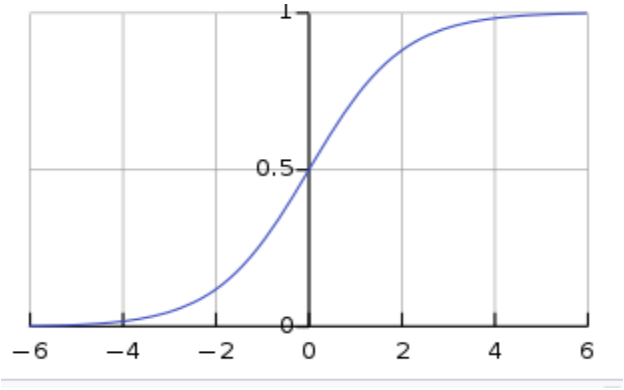


Figure 4. 17. Sigmoid function [15]

b) Softmax function

This function is used in the final layer of a neural network-based classifier. Such networks are trained under a cross-entropy rule, giving a non-linear variant of multinomial logistic regression. It is given by

$$softmax(z_j) = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad for\ j = 1, \ldots, K \qquad (4.4)\ [15]$$

where K is number of classes, z_j is the input to neuron j.

Then Click Next.

2. Get input and target data with the same number of samples from Workspace and click Next
3. Divide your samples into training, validation and testing data. The default divisions are 70%, 15% and 15% for training, validation and testing respectively. Here we used the default divisions as standard. Then click next.
4. Define a pattern recognition NN by adding number of hidden neurons and click next.

In this thesis, 10 hidden neurons were selected and then as a result the following NN was created.

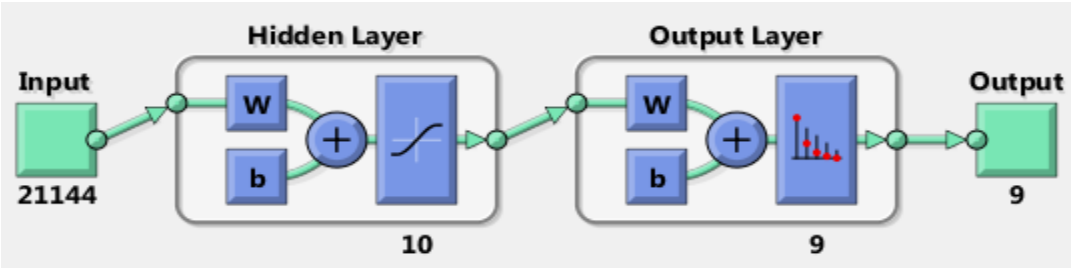


Figure 4. 18. Neural Network of the system

5. Train the created network. Here different outputs can be seen. The network was trained as figure 4.19.

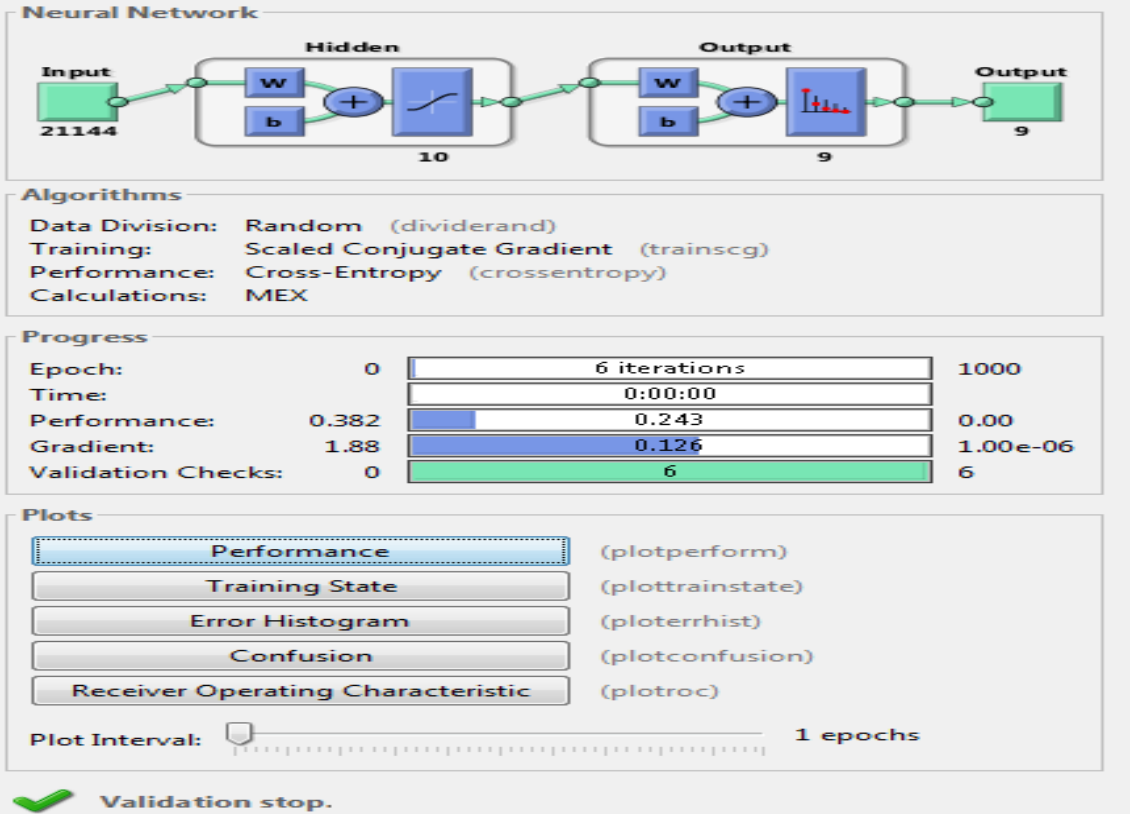


Figure 4. 19. Training the neural network

6. Click Next and test by re inputting the input datasets on step 2. Here, the total confusion matrix, cross-entropy, percentage-error and ROC will be gained.
7. Deploy application, Simulink diagram and generate code and graphical diagram of NN.
8. Finally save the results.

It has different algorithms like Logistic regression (Stochastic Gradient Descent), Naïve Bayes Classification, HMM etc.

We usually read regression along with classification. Actually, there is a difference between them. A classification involves a categorical target variable while a regression involves a numeric target variable. Classification predicts whether something will happen and regression predicts how much something will happen [16].

Naïve Bayes classification is a very popular algorithm for text classification. It uses the concept of probability to classify new items. It based on Bayes theorem. In HMM, we observe a sequence of emission but do not have a sequence of states, which a model uses to generate the emission [16].

There is no a single evaluation technique in fitting all the classifier models. However, here some common issues like confusion matrix, ROC graph, AUC and Entropy matrix can be used.

a) The confusion matrix

The confusion matrix shows the number of correct and incorrect predictions made by the model compared with the actual outcomes (target values) in the data. It is an N*N matrix, where N is the number of labels (classes). Each column is an instance in the predicted class whereas each row is an instance in the actual class. Using this it is possible to find out how one class is confused with another.

In this thesis, the 9*9-confusion matrix was made from nine classes of 21144*45 input datasets of the system compared with its 9*45 target values. This was after training the system network by giving the two inputs and repeating the train algorithm until errors in training, validation and testing become 0.0, 0.14 and 0.42 respectively. Then confusion matrix shown in table 4.1 was resulted.

Table 4. 1. Confusion Matrix of the trained system

Output Class \ Target Class	1	2	3	4	5	6	7	8	9	
1	**4** 8.9%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	100% 0.0%
2	**0** 0.0%	**4** 8.9%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	100% 0.0%
3	**0** 0.0%	**0** 0.0%	**5** 11.1%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	100% 0.0%
4	**0** 0.0%	**1** 2.2%	**0** 0.0%	**5** 11.1%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**2** 4.4%	**0** 0.0%	62.5% 37.5%
5	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**5** 11.1%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	100% 0.0%
6	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**5** 11.1%	**0** 0.0%	**0** 0.0%	**0** 0.0%	100% 0.0%
7	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**5** 11.1%	**0** 0.0%	**0** 0.0%	100% 0.0%
8	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**3** 6.7%	**0** 0.0%	100% 0.0%
9	**1** 2.2%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**5** 11.1%	83.3% 16.7%
	80.0% 20.0%	80.0% 20.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	60.0% 40.0%	100% 0.0%	**91.1%** **8.9%**

b) ROC graph

It is the receiver operating characteristics graph, which is a two-dimensional plot of a classifier with false positive rate on the x-axis and true positive rate on the y-axis. The lower point (0, 0) in the figure represents never issuing a positive classification. Point (0, 1) represents a prefect classification. The diagonal from (0, 0) to (1, 1) divides the ROC space into two. Points above the diagonal represent good classification results whereas the points below it represent poor results [16] [17] [18].

The ROC of the above confusion matrix is shown in figure 4.20.

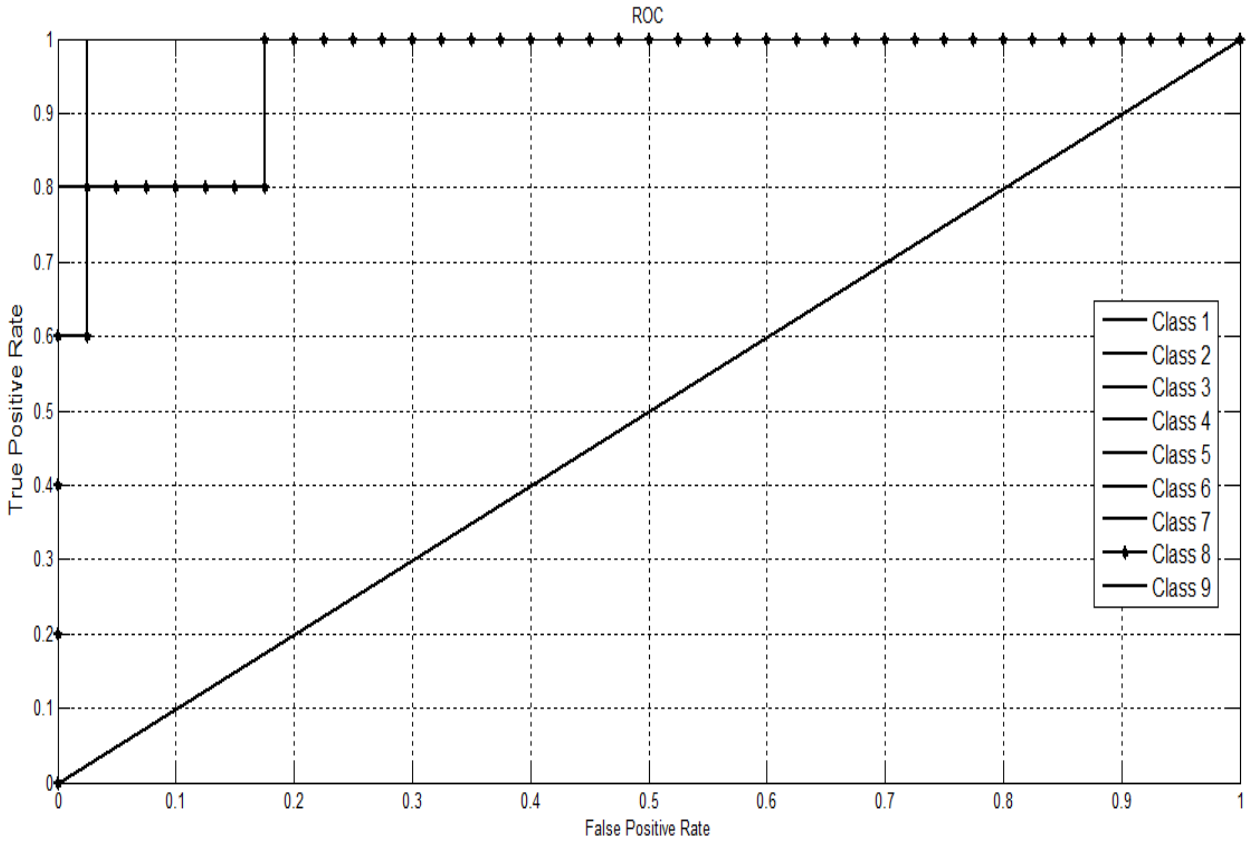


Figure 4. 20. ROC of the trained system

In a ROC curve the true positive rate (Sensitivity) is plotted in function of the false positive rate (100-Specificity) for different cut-off points of a parameter. Each point on the ROC curve represents a sensitivity/specificity pair corresponding to a particular decision threshold.

In general, the ROC can be analyzed interms of the following points [16] [17] [18].

i. True Positive: The test result that detects the sample when the sample is present.

ii. True Negative: The test result does not detect the sample when the sample is absent.

iii. False Positive: The test result that detects the sample when the sample is absent.

iv. False Negative: The test result that does not detect the sample when the sample is present

v. Sensitivity: probability that a test result will be positive when the sample is present (true positive rate, expressed as a percentage). It measures the ability of a test to detect the sample when the sample is present. Given by

$$\text{Sensitivity} = \frac{\text{True Positive}}{\text{True positive} + \text{False Negative}} \quad (4.3)\ [17]$$

vi. Specificity: probability that a test result will be negative when the sample is not present (true negative rate, expressed as a percentage). It measures the ability of a test to correctly exclude the condition (not detect the condition) when the condition is absent. Given by

$$\text{Specificity} = \frac{\text{True Negative}}{\text{False Positive} + \text{True Negative}} \quad (4.4)\ [17]$$

c) AUC

This is the area under the ROC curve and is known as AUC. It is used to measure the quality of the classification model. In practice, most of the classification models have an AUC between 0.5 and 1. The closer the value is to 1, the better is our classifier [16] [17] [18].

d) Entropy Matrix

Entropy is a measure of disorder that can be applied to a set. It is defined as:

$$\text{Entropy} = -P_1 \log_2(P_1) - P_2 \log_2(P_2) - \cdots \quad (4.9)\ [16]$$

Each P is the probability of a particular property (class) within the set.

Entropy is useful in acquiring knowledge of information gain. Information gain measures the change in entropy due to any new information being added in model creation. Therefore, if entropy decreases from new information, it indicates that the model is performing well at that moment. Information gain is calculated as

$$\text{IG}(\text{classes}, \text{subclasses}) = \text{enropy}(\text{class}) - [\text{P}(\text{subclass}_1) * \text{entropy}(\text{subclass}_1) + \text{P}(\text{subclass}_2) * \text{entropy}(\text{subclass}_2) + \cdots] \quad (4.10)\ [16]$$

Entropy matrix is the same as the confusion matrix defined earlier except that the elements in its matrix are the average of the log of the probability score for each true or estimated category

combination. A good model will have small negative entropy values along the diagonal and will have large negative entropy values in the off-diagonal position [16].

4.3.6. Speech Recognition

This is the last stage of system design where the speech signal is recognized. Once the system is trained, validated and tested with dataset, the new input speech signal is recognized as per its already stored reference signal. The flow chart of speech recognition process is shown as figure 4.21.

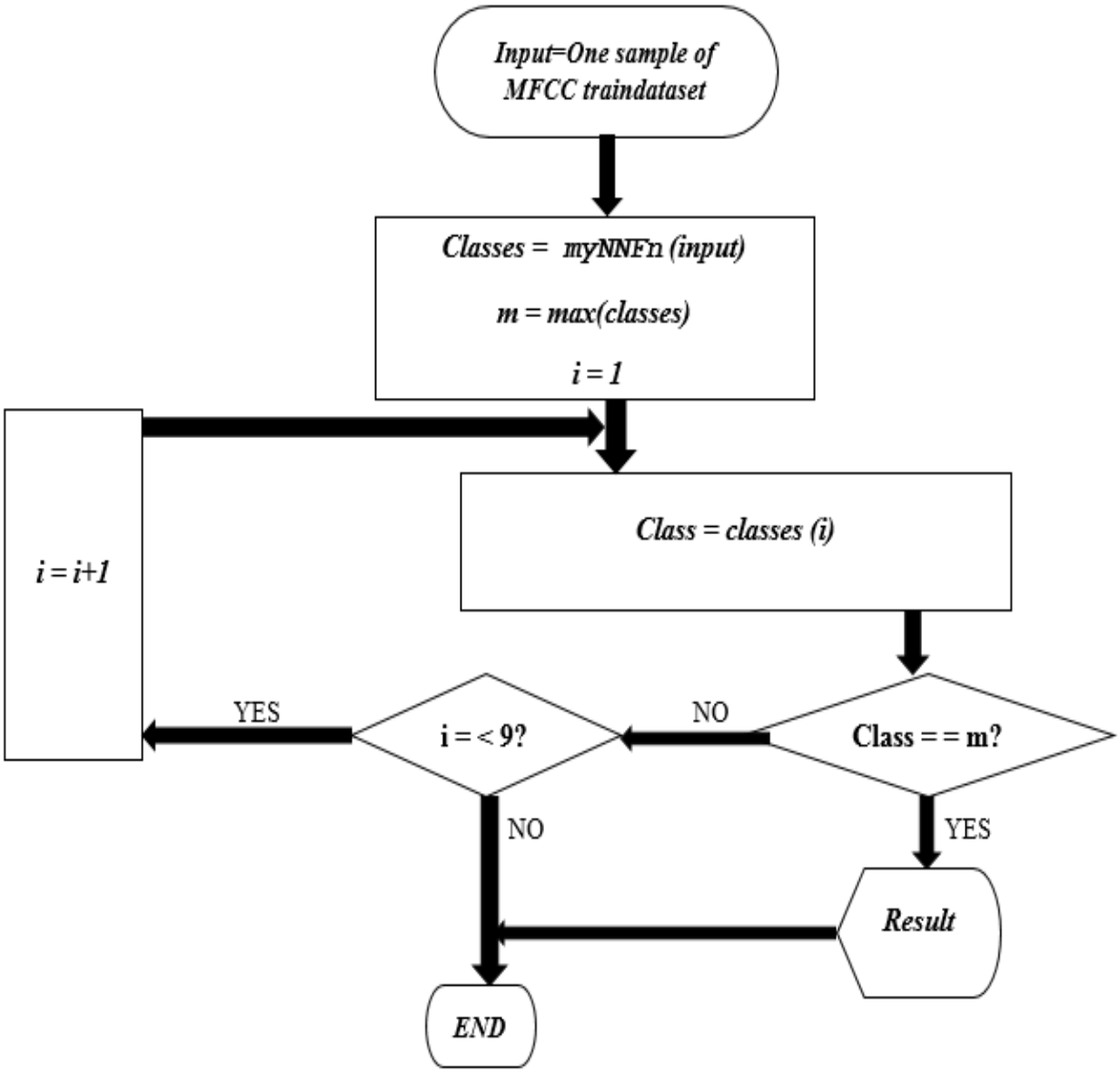


Figure 4. 21. Flowchart of Speech Recognition Stage

Chapter 5

Result and Discussion

5.1. Preprocessing Results

The raw audios and their corresponding pre-processed signals of the first two of 45 samples are shown in the following figures.

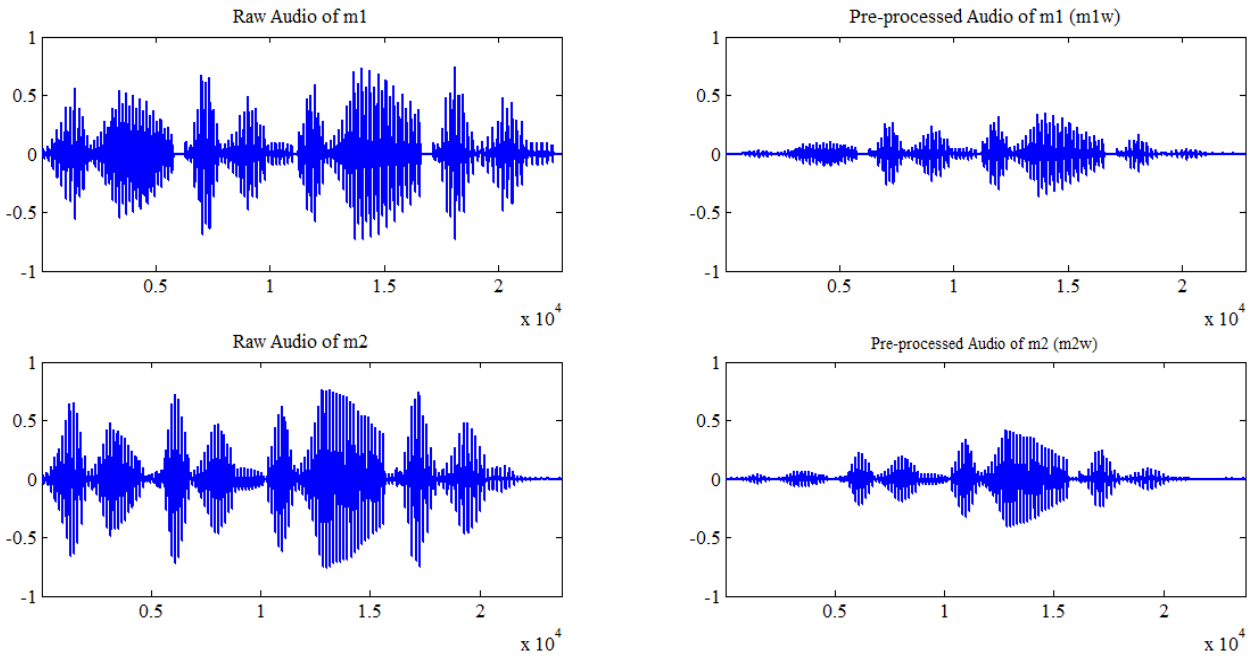


Figure 5. 1. The first two raw audios and their corresponding pre-processed signals

The left side of figure 5.1 shows waveform of raw audios data whereas its right side is showing the corresponding preprocessed version of waveforms. The preprocessed of waves are compressed in their amplitudes. They are the results of the preemphasis (high pass) filter by which lower frequency noises were suppressed. Afterwards, they can be used as inputs for feature extraction stage.

5.2. Feature Extraction Results

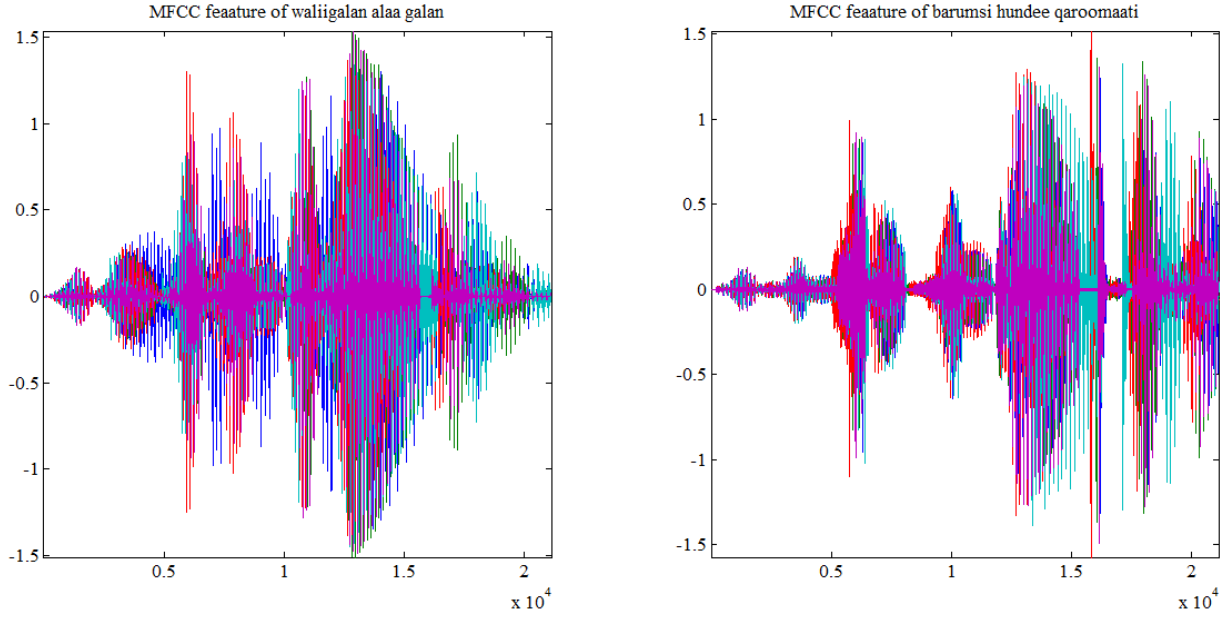


Figure 5. 2. Feature Extracted from the first two audio files

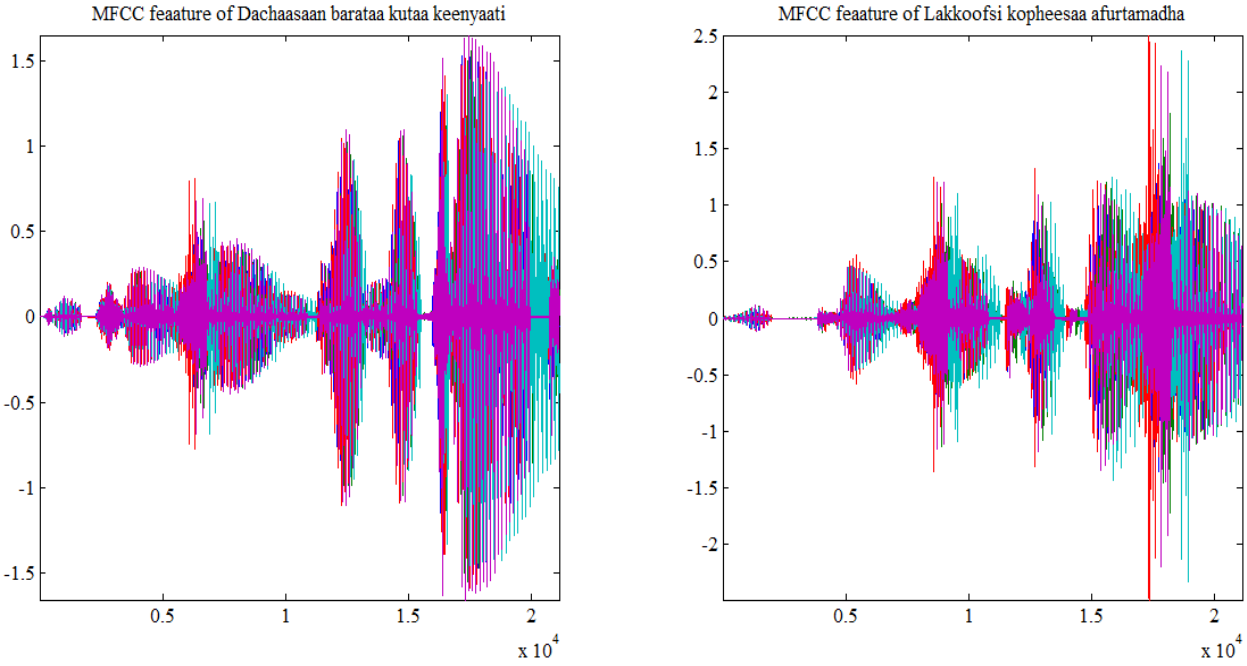


Figure 5. 3. Feature Extracted from the second two audio files

As can be seen from the figures 5.2 and 5.3, the extracted features are unique for each audio. Hence, they can be used in differentiating one audio from the other. That means they are identity for speech recognition.

5.3. Classification Results

The responses were structured as a confusion matrix, in which each row corresponds to a signal (input) and each column to a response. The entries in the confusion matrix are the frequencies with which particular responses were made. From there, the conditional proportions of the responses given each stimulus are calculated. Entries along the diagonal of the confusion matrix describe correct responses, and off-diagonal entries describe the various types of incorrect responses.
The following table summarized this.

Table 5. 1. Classification and misclassification of samples from confusion matrix

Classes	Nº of Samples		Percentage	
	Fitted class	Missed class	%Fit class	% Missed class
1	4	0	100	0
2	4	0	100	0
3	5	0	100	0
4	5	3	62.5	37.5
5	5	0	100	0
6	5	0	100	0
7	5	0	100	0
8	3	0	100	0
9	5	1	83.3	16.7
Total	41	4	91.1	8.9

Table 5.1 shown that the samples of classes 1 to 3 and 5 to 8 are perfectly (100%) classified to their corresponding classes. Whereas classes 4 and 9 got the misclassified samples. The overall percentage was 91.1% for perfect classification and 8.9% for misclassification of samples. It is located within the last row of the table.

Table 5.2 analyzed that the correct and wrong responses due to the exact classification and misclassification of samples. The overall percentage was 91.1% for correct and 8.9% for wrong responses.

Table 5. 2. Correct and wrong responses from confusion matrix

Target outputs	Nº of		Percentage	
	Correct responses	Wrong responses	%Correct response	% Wrong response
1	4	1	80	20
2	4	1	80	20
3	5	0	100	0
4	5	0	100	0
5	5	0	100	0
6	5	0	100	0
7	5	0	100	0
8	3	2	60	40
9	5	0	100	0
Total	41	4	91.1	8.9

5.4. Recognition Results

```
>> Recognition                    >> Recognition

recognizedmatrix =                recognizedmatrix =

    0.6054                            0.5280
    0.5207                            0.5257
    0.6033                            0.6721
    0.5293                            0.5539
    0.5598                            0.6228
    0.5417                            0.5463
    0.5315                            0.5232
    0.5147                            0.5051
    0.5936                            0.5229

Waliigalan alaa galan             Dachaasaan barataa kutaa keenyaati
```

```
>> Recognition

recognizedmatrix =

    0.5296
    0.5372
    0.5620
    0.7217
    0.5850
    0.5265
    0.5263
    0.5012
    0.5106

Lakkoofsi kopheesaa afurtamadha
```

```
>> Recognition

recognizedmatrix =

    0.5123
    0.5079
    0.5619
    0.5015
    0.5101
    0.5356
    0.5042
    0.5115
    0.8550

Zeeroo fi Tsaggaan walinbeekan
```

```
>> Recognition

recognizedmatrix =

    0.5299
    0.5512
    0.6101
    0.5511
    0.7074
    0.5198
    0.5233
    0.5011
    0.5061

Caalaan hiriyaa isheeti
```

In testing the system for recognition, the new input was given to the trained NN and classified under nine classes. Then the class with maximum value was displayed its corresponding text.

Chapter 6

Conclusion and Future Work

6.1. Conclusion

In this thesis, a number of concepts and methods had been raised for developing afan Oromo speech recognition system. Among those concepts afan Oromo phonemes, their corresponding qubes (letters) and rules in spelling them in Oromo words like doubling the same consonants at the beginning and ending of the word is forbidden. The Artificial Neural Network MATLAB toolboxes and how they can be used in speech recognition system was another concept that has been discussed. Methods like uttering the sentences made from collected phonemes and preparing the audios for system development, analyzing them by algorithms, which include acquiring audio, preprocessing, feature extraction by MFCC, classification and finally recognition had been used. In order to develop the system, 21144 * 45 input datasets and 9*45 target datasets were made. 70% of input datasets were used for training whereas 30% of input datasets shared between validation and testing algorithms. Then confusion matrix was resulted. It shown the correctly and incorrectly classified samples.

Out of total samples, 91.1% were perfectly classified to their corresponding classes whereas the rest 8.9% were misclassified. That is, they were classified to other classes.

Finally, the recognition ability of the system was tested by one sample of MFCC traindataset at a time. Consequently, the corresponding text form of the recognized sample was displayed.

6.2. Future Work

In the future, an acoustic model could be developed for afan Oromo speech recognition system. The model will develop the audio-phoneme transcription system. It will use Hidden Markov Model. This model again develops a non-deterministic probability model for the speech recognition. It consists of two variables namely the hidden states of the phonemes stored in the computer memory and the visible frequency segment of the digital signal. Each phoneme has its own probability to which the segment will compared with accordingly. Then the matched phonemes will be collected together to form the correct words according to the stored speech corpus and grammar rules of the language.

References

[1] B. C. Kamble, "Speech Recognition Using Artificial Neural Network -A Review," *Int'l Journal of Computing, Communications & Instrumentation Engg. (IJCCIE),* vol. 3, no. 1, pp. ISSN 2349-1469 EISSN 2349-1477, 2016.

[2] N. K. Kasabov, Foundations of neural networks, fuzzy systems, and knowledge Engineerimg, London, England: MIT Press, 1998, pp. 17-19.

[3] FifthGen Computer Corporation, "Global tree system software," Fifth Generation Software, 15 june 2013. [Online]. Available: http://www.fifthgen.com/speaker-independent-connected-s-r.htm. [Accessed 28 April 2017].

[4] D. Reynolds and R. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 1, 21 February 2014.

[5] A. Murphy, "Implementing Speech Recognition with Artificial Neural Networks," Algoma University, 2015.

[6] Shikha Gupta, Mr.Amit Pathak, Mr.Achal Saraf, "A Study On Speech Recognition System: A Literature Review," *International Journal of Science, Engineering and Technology Research (IJSETR),* vol. 3, no. 8, August 2014.

[7] Takialddin Al Smadi,Huthaifa A. Al Issa,Esam Trad,Khalid A. Al Smadi, "Artificial Intelligence for Speech Recognition Based on Neural Networks," *Journal of Signal and Information Processing*, no. 6, pp. 66-72, May 2015.

[8] J. M. Zurada, Introduction to Artificial Neural Systems, New York: West Publishing Company , 1992.

[9] J. Tebelskis, "Speech Recognition using Neural Networks," May 1995. [Online]. Available: www.Google.com. [Accessed 12 January 2017].

[10] P. T. Gamta, "wikipedia," 26 6 1994. [Online]. Available: www.africa.upenn.edu/Hornet/Afaan_ Oromo_19777.html. [Accessed 7 4 2018].

[11] F. B. Kebede, "Dissimilation in Afan oromo phonology," *international Journal of Innovative Research and Development* , vol. 3, no. 13, pp. 187-196, December 2014.

[12] V.Rajaraman, Introduction to Information Technology, 3 ed., vol. IV, Bangalore: Asoke K.Ghosh., 2018, pp. 73-75.

[13] Keerio, Ayaz, Mitra, Bhargav Kumar, Birch, Philip, Young, Rupert and Chatwin, Chris, "On Pre-processing of speech signals," *International Journal ofSignal Processing,* vol. 5, no. 3, pp. 216-222, 2009.

[14] P. M. Steven B. Davis, "Comparison of Parametric Representations For Monosyllabic Word Recognition In Continuously Spoken Sentences," *In IEEE Transactions on Acoustics, Speech, and Signal Processing,* Vol. 28, No. 4, Pp. 357-366, 2001.

[15] Han Jun, Moraga Claudo, "The influence of the sigmoid function parameters on the speed of backpropagation learning," in *From Natural to Artificial Neural Computation*, vol. 930, In Mira, Jose, Sandoval, Francisco, Springer, Berlin, Heidelberg, IWANN 1995, pp. 195-201.

[16] A. Gupta, Learning Apache Mahout classifiction, BIRMINGHAM: Packt, 2015.

[17] D. G.Altman, Practical Statistics For Medical Research, Boca Raton London New York Washington, D.C.: CHAPMAN & HALL/CRC, 2014.

[18] "Amazon," january 2017. [Online]. Available: https://www.amazon.com/dp/1520321570. [Accessed 9 April 2018].

Appendices

Appendix A: Matlab source Code for developing of the system

```
% Reading the audiorecords
m1=audioread('Oromo_Male1.wav');
m2=audioread('Oromo_Male2.wav');
m3=audioread('Oromo_Male3.wav');
m4=audioread('Oromo_Male4.wav');
m5=audioread('Oromo_Male5.wav');
m6=audioread('Oromo_Male6.wav');
m7=audioread('Oromo_Male7.wav');
m8=audioread('Oromo_Male8.wav');
m9=audioread('Oromo_Male9.wav');
m10=audioread('Oromo_Male10.wav');
m11=audioread('Oromo_Male11.wav');
m12=audioread('Oromo_Male12.wav');
m13=audioread('Oromo_Male13.wav');
m14=audioread('Oromo_Male14.wav');
m15=audioread('Oromo_Male15.wav');
m16=audioread('Oromo_Male16.wav');
m17=audioread('Oromo_Male17.wav');
m18=audioread('Oromo_Male18.wav');
m19=audioread('Oromo_Male19.wav');
m20=audioread('Oromo_Male20.wav');
m21=audioread('Oromo_Male21.wav');
m22=audioread('Oromo_Male22.wav');
m23=audioread('Oromo_Male23.wav');
m24=audioread('Oromo_Male24.wav');
m25=audioread('Oromo_Male25.wav');
m26=audioread('Oromo_Male26.wav');
m27=audioread('Oromo_Male27.wav');
m28=audioread('Oromo_Male28.wav');
m29=audioread('Oromo_Male29.wav');
m30=audioread('Oromo_Male30.wav');
m31=audioread('Oromo_Male31.wav');
m32=audioread('Oromo_Male32.wav');
m33=audioread('Oromo_Male33.wav');
m34=audioread('Oromo_Male34.wav');
m35=audioread('Oromo_Male35.wav');
m36=audioread('Oromo_Male36.wav');
m37=audioread('Oromo_Male37.wav');
m38=audioread('Oromo_Male38.wav');
m39=audioread('Oromo_Male39.wav');
m40=audioread('Oromo_Male40.wav');
m41=audioread('Oromo_Male41.wav');
m42=audioread('Oromo_Male42.wav');
```

```
m43=audioread('Oromo_Male43.wav');
m44=audioread('Oromo_Male44.wav');
m45=audioread('Oromo_Male45.wav');
% Preprocessing Stage (windowing and preemphasis)
% Apply a pre-emphasis filter. The pre-emphasis filter is
% a highpass all-pole filter.
preemph=[1 0.9];
% filters the data in vector m1 with the filter described by numerator
% coefficient vector 1 and denominator coefficient vector preemph
m1f=filter(1,preemph,m1);
m2f=filter(1,preemph,m2);
m3f=filter(1,preemph,m3);
m4f=filter(1,preemph,m4);
m5f=filter(1,preemph,m5);
m6f=filter(1,preemph,m6);
m7f=filter(1,preemph,m7);
m8f=filter(1,preemph,m8);
m9f=filter(1,preemph,m9);
m10f=filter(1,preemph,m10);
m11f=filter(1,preemph,m11);
m12f=filter(1,preemph,m12);
m13f=filter(1,preemph,m13);
m14f=filter(1,preemph,m14);
m15f=filter(1,preemph,m15);
m16f=filter(1,preemph,m16);
m17f=filter(1,preemph,m17);
m18f=filter(1,preemph,m18);
m19f=filter(1,preemph,m19);
m20f=filter(1,preemph,m20);
m21f=filter(1,preemph,m21);
m22f=filter(1,preemph,m22);
m23f=filter(1,preemph,m23);
m24f=filter(1,preemph,m24);
m25f=filter(1,preemph,m25);
m26f=filter(1,preemph,m26);
m27f=filter(1,preemph,m27);
m28f=filter(1,preemph,m28);
m29f=filter(1,preemph,m29);
m30f=filter(1,preemph,m30);
m31f=filter(1,preemph,m31);
m32f=filter(1,preemph,m32);
m33f=filter(1,preemph,m33);
m34f=filter(1,preemph,m34);
m35f=filter(1,preemph,m35);
m36f=filter(1,preemph,m36);
m37f=filter(1,preemph,m37);
```

```
m38f=filter(1,preemph,m38);
m39f=filter(1,preemph,m39);
m40f=filter(1,preemph,m40);
m41f=filter(1,preemph,m41);
m42f=filter(1,preemph,m42);
m43f=filter(1,preemph,m43);
m44f=filter(1,preemph,m44);
m45f=filter(1,preemph,m45);
% Window the speech segment using a Hamming window
m1w = m1f.*hamming(length(m1f));% w represented windowed data
m2w = m2f.*hamming(length(m2f));
m3w = m3f.*hamming(length(m3f));
m4w = m4f.*hamming(length(m4f));
m5w = m5f.*hamming(length(m5f));
m6w = m6f.*hamming(length(m6f));
m7w = m7f.*hamming(length(m7f));
m8w = m8f.*hamming(length(m8f));
m9w = m9f.*hamming(length(m9f));
m10w = m10f.*hamming(length(m10f));
m11w = m11f.*hamming(length(m11f));
m12w = m12f.*hamming(length(m12f));
m13w = m13f.*hamming(length(m13f));
m14w = m14f.*hamming(length(m14f));
m15w = m15f.*hamming(length(m15f));
m16w = m16f.*hamming(length(m16f));
m17w = m17f.*hamming(length(m17f));
m18w = m18f.*hamming(length(m18f));
m19w = m19f.*hamming(length(m19f));
m20w = m20f.*hamming(length(m20f));
m21w = m21f.*hamming(length(m21f));
m22w = m22f.*hamming(length(m22f));
m23w = m23f.*hamming(length(m23f));
m24w = m24f.*hamming(length(m24f));
m25w = m25f.*hamming(length(m25f));
m26w = m26f.*hamming(length(m26f));
m27w = m27f.*hamming(length(m27f));
m28w = m28f.*hamming(length(m28f));
m29w = m29f.*hamming(length(m29f));
m30w = m30f.*hamming(length(m30f));
m31w = m31f.*hamming(length(m31f));
m32w = m32f.*hamming(length(m32f));
m33w = m33f.*hamming(length(m33f));
m34w = m34f.*hamming(length(m34f));
m35w = m35f.*hamming(length(m35f));
m36w = m36f.*hamming(length(m36f));
m37w = m37f.*hamming(length(m37f));
```

```
m38w = m38f.*hamming(length(m38f));
m39w = m39f.*hamming(length(m39f));
m40w = m40f.*hamming(length(m40f));
m41w = m41f.*hamming(length(m41f));
m42w = m42f.*hamming(length(m42f));
m43w = m43f.*hamming(length(m43f));
m44w = m44f.*hamming(length(m44f));
m45w =m45f.*hamming(length(m45f));
% sample length (sl)
sl=min([length(m1f),length(m2f),length(m3f),length(m4f),length(m5f),length(m6f),length(m7f),le
ngth(m8f),length(m9f),length(m10f),length(m11f),length(m12f),length(m13f),length(m14f),length
(m15f),length(m16f),length(m17f),length(m18f),length(m19f),length(m20f),length(m21f),length(m
22f),length(m23f),length(m24f),length(m25f),length(m26f),length(m27f),length(m28f),length(m29
f),length(m30f),length(m31f),length(m32f),length(m33f),length(m34f),length(m35f),length(m36f),
length(m37f),length(m38f),length(m39f),length(m40f),length(m41f),length(m42f),length(m43f),le
ngth(m44f),length(m45f)]);
% extracting real mel frequency cepstrals (vectors) of read audio
m1c=2591.*log(1.+m1w(1:sl)/700);
m2c=2591.*log(1.+m2w(1:sl)/700);
m3c=2591.*log(1.+m3w(1:sl)/700);
m4c=2591.*log(1.+m4w(1:sl)/700);
m5c=2591.*log(1.+m5w(1:sl)/700);
m6c=2591.*log(1.+m6w(1:sl)/700);
m7c=2591.*log(1.+m7w(1:sl)/700);
m8c=2591.*log(1.+m8w(1:sl)/700);
m9c=2591.*log(1.+m9w(1:sl)/700);
m10c=2591.*log(1.+m10w(1:sl)/700);
m11c=2591.*log(1.+m11w(1:sl)/700);
m12c=2591.*log(1.+m12w(1:sl)/700);
m13c=2591.*log(1.+m13w(1:sl)/700);
m14c=2591.*log(1.+m14w(1:sl)/700);
m15c=2591.*log(1.+m15w(1:sl)/700);
m16c=2591.*log(1.+m16w(1:sl)/700);
m17c=2591.*log(1.+m17w(1:sl)/700);
m18c=2591.*log(1.+m18w(1:sl)/700);
m19c=2591.*log(1.+m19w(1:sl)/700);
m20c=2591.*log(1.+m20w(1:sl)/700);
m21c=2591.*log(1.+m21w(1:sl)/700);
m22c=2591.*log(1.+m22w(1:sl)/700);
m23c=2591.*log(1.+m23w(1:sl)/700);
m24c=2591.*log(1.+m24w(1:sl)/700);
m25c=2591.*log(1.+m25w(1:sl)/700);
m26c=2591.*log(1.+m26w(1:sl)/700);
m27c=2591.*log(1.+m27w(1:sl)/700);
m28c=2591.*log(1.+m28w(1:sl)/700);
m29c=2591.*log(1.+m29w(1:sl)/700);
```

```
m30c=2591.*log(1.+m30w(1:sl)/700);
m31c=2591.*log(1.+m31w(1:sl)/700);
m32c=2591.*log(1.+m32w(1:sl)/700);
m33c=2591.*log(1.+m33w(1:sl)/700);
m34c=2591.*log(1.+m34w(1:sl)/700);
m35c=2591.*log(1.+m35w(1:sl)/700);
m36c=2591.*log(1.+m36w(1:sl)/700);
m37c=2591.*log(1.+m37w(1:sl)/700);
m38c=2591.*log(1.+m38w(1:sl)/700);
m39c=2591.*log(1.+m39w(1:sl)/700);
m40c=2591.*log(1.+m40w(1:sl)/700);
m41c=2591.*log(1.+m41w(1:sl)/700);
m42c=2591.*log(1.+m42w(1:sl)/700);
m43c=2591.*log(1.+m43w(1:sl)/700);
m44c=2591.*log(1.+m44w(1:sl)/700);
m45c=2591.*log(1.+m45w(1:sl)/700);
% constructing the matrix from the created vectors
mfcctraindataset(:,1)=m1c;%mfcc is mel frequency cepstral coefficient
mfcctraindataset(:,2)=m2c;
mfcctraindataset(:,3)=m3c;
mfcctraindataset(:,4)=m4c;
mfcctraindataset(:,5)=m5c;
mfcctraindataset(:,6)=m6c;
mfcctraindataset(:,7)=m7c;
mfcctraindataset(:,8)=m8c;
mfcctraindataset(:,9)=m9c;
mfcctraindataset(:,10)=m10c;
mfcctraindataset(:,11)=m11c;
mfcctraindataset(:,12)=m12c;
mfcctraindataset(:,13)=m13c;
mfcctraindataset(:,14)=m14c;
mfcctraindataset(:,15)=m15c;
mfcctraindataset(:,16)=m16c;
mfcctraindataset(:,17)=m17c;
mfcctraindataset(:,18)=m18c;
mfcctraindataset(:,19)=m19c;
mfcctraindataset(:,20)=m20c;
mfcctraindataset(:,21)=m21c;
mfcctraindataset(:,22)=m22c;
mfcctraindataset(:,23)=m23c;
mfcctraindataset(:,24)=m24c;
mfcctraindataset(:,25)=m25c;
mfcctraindataset(:,26)=m26c;
mfcctraindataset(:,27)=m27c;
mfcctraindataset(:,28)=m28c;
mfcctraindataset(:,29)=m29c;
```

```
mfcctraindataset(:,30)=m30c;
mfcctraindataset(:,31)=m31c;
mfcctraindataset(:,32)=m32c;
mfcctraindataset(:,33)=m33c;
mfcctraindataset(:,34)=m34c;
mfcctraindataset(:,35)=m35c;
mfcctraindataset(:,36)=m36c;
mfcctraindataset(:,37)=m37c;
mfcctraindataset(:,38)=m38c;
mfcctraindataset(:,39)=m39c;
mfcctraindataset(:,40)=m40c;
mfcctraindataset(:,41)=m41c;
mfcctraindataset(:,42)=m42c;
mfcctraindataset(:,43)=m43c;
mfcctraindataset(:,44)=m44c;
mfcctraindataset(:,45)=m45c;
%packing our created matrix in order to use small space
% pack('trainmatrix')%PACK can only be used from the MATLAB command line.
%forming the target matrix with the same columns as train matrix
targetdataset=[1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
               0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
               0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
               0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
               0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
               0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
               0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0;
               0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0;
               0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1;];

% Solve a Pattern Recognition Problem with a Neural Network
% Script generated by Neural Pattern Recognition app
% Created Sun Apr 22 22:40:03 PDT 2018
% This script assumes these variables are defined:
%mfcctraindataset - input data.
%targetdataset - target data.
x = mfcctraindataset;
t = targetdataset;
% Create a Pattern Recognition Network
hiddenLayerSize = 10;
net = patternnet(hiddenLayerSize);
% Choose Input and Output Pre/Post-Processing Functions
% For a list of all processing functions type: help nnprocess
net.input.processFcns = {'removeconstantrows','mapminmax'};
net.output.processFcns = {'removeconstantrows','mapminmax'};
% Setup Division of Data for Training, Validation, Testing
% For a list of all data division functions type: help nndivide
net.divideFcn = 'dividerand';  % Divide data randomly
```

```
net.divideMode = 'sample';  % Divide up every sample
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;
% For help on training function 'trainscg' type: help trainscg
% For a list of all training functions type: help nntrain
net.trainFcn = 'trainscg';  % Scaled conjugate gradient
% Choose a Performance Function
% For a list of all performance functions type: help nnperformance
net.performFcn = 'crossentropy'; % Cross-entropy
% Choose Plot Functions
% For a list of all plot functions type: help nnplot
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
  'plotregression', 'plotfit'};
% Train the Network
[net,tr] = train(net,x,t);
% Test the Network
y = net(x);
e = gsubtract(t,y);
tind = vec2ind(t);
yind = vec2ind(y);
percentErrors = sum(tind ~= yind)/numel(tind);
performance = perform(net,t,y)
% Recalculate Training, Validation and Test Performance
trainTargets = t .* tr.trainMask{1};
valTargets = t  .* tr.valMask{1};
testTargets = t  .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,y)
valPerformance = perform(net,valTargets,y)
testPerformance = perform(net,testTargets,y)
% View the Network
view(net)
% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, plotconfusion(t,y)
%figure, plotroc(t,y)
%figure, ploterrhist(e)
% Deployment
% Change the (false) values to (true) to enable the following code blocks.
if (false)
  % Generate MATLAB function for neural network for application deployment
  % in MATLAB scripts or with MATLAB Compiler and Builder tools, or simply
  % to examine the calculations of your trained neural network performs.
  genFunction(net,'myNeuralNetworkFunction');
```

```
  y = myNeuralNetworkFunction(x);
end
if (false)
  % Generate a matrix-only MATLAB function for neural network code
  % generation with MATLAB Coder tools.
  genFunction(net,'myNeuralNetworkFunction','MatrixOnly','yes');
  y = myNeuralNetworkFunction(x);
end
if (false)
  % Generate a Simulink diagram for simulation or deployment with.
  % Simulink Coder tools.
  gensim(net);
end
% Code for Testing system's recognition for newinput audio
% new audio need to have same row length with trained datasamples
% Enter one of melcepstrals input in place of x
recognizedmatrix=myNeuralNetworkFunction(x);
class1=recognizedmatrix(1);
class2=recognizedmatrix(2);
class3=recognizedmatrix(3);
class4=recognizedmatrix(4);
class5=recognizedmatrix(5);
class6=recognizedmatrix(6);
class7=recognizedmatrix(7);
class8=recognizedmatrix(8);
class9=recognizedmatrix(9);
m=max([class1,class2,class3,class4,class5,class6,class7,class8,class9]);
if class1==m
   disp('Waliigalan alaa galan');
elseif class2==m
   disp('Barumsi hundee qaroomaati');
elseif class3==m
   disp('Dachaasaan barataa kutaa keenyaati');
elseif class4==m
   disp('Lakkoofsi kopheesaa afurtamadha');
elseif class5==m
   disp('Caalaan hiriyaa isheeti');
elseif class6==m
   disp('Xalayaan ergama qaba ');
elseif class7==m
   disp('har'a guyyaan meeqa');
elseif class8==m
   disp('Poostaan televizyiiniirra jira ');
elseif class9==m
   disp('Zeeroo fi Tsaggaan walinbeekan ');
end
```

Appendix B: The First 50*10 Input Dataset

Table B. 1. The first 50*10 Input dataset

-9.04E-06	0.00028917	0.00030725	-0.0003795	-0.0002259	-9.04E-06	-9.04E-06	1.81E-05	1.81E-05	3.61E-05
2.62E-05	0.00030905	0.00034701	-0.0003994	-0.0002666	1.72E-05	8.13E-06	-5.24E-05	-2.53E-05	-4.16E-05
3.06E-05	0.00055323	0.00056425	-0.0006978	-0.0003926	-3.35E-05	-1.64E-05	8.33E-05	4.08E-05	6.45E-05
8.09E-05	0.0005865	0.00062176	-0.0007727	-0.0003334	4.82E-05	2.38E-05	-0.0001111	-5.48E-05	-8.52E-05
5.37E-05	0.00079151	0.00079592	-0.0009945	-0.0004229	-6.15E-05	-3.04E-05	0.0001361 7	6.74E-05	0.0001037 7
0.00011430 3	0.00083292	0.00091028	-0.0010569	-0.0002701	7.34E-05	3.64E-05	-0.0001587	-7.88E-05	-0.0001205
7.79E-05	0.00099446	0.00105136	-0.0012086	-0.0003082	-8.42E-05	-4.18E-05	0.0001789 7	8.90E-05	0.0001355 7
0.00015584	0.00102982	0.00115032	-0.0012709	-8.41E-05	9.38E-05	4.67E-05	-0.0001972	-9.81E-05	-0.0001491
0.00010373 5	0.00110644	0.00122392	-0.0013323	-8.70E-05	-0.0001025	-5.10E-05	0.0002136 5	0.0001063 9	0.0001613 2
0.00020485 5	0.00111882	0.0012842	-0.0013764	0.00016864	0.00011033	5.50E-05	-0.0002284	-0.0001138	-0.0001723
0.00014095 8	0.00112576	0.00131129	-0.0013548	0.00015548	-0.0001174	-5.85E-05	0.0002417 4	0.0001205 2	0.0001821 8
0.0002798	0.00111952	0.00135922	-0.0013472	0.0003571	0.0001237	6.17E-05	-0.0002537	-0.0001265	-0.0001911
0.00019099 2	0.00109804	0.00137936	-0.0012275	0.00025697	-0.0001294	-6.46E-05	0.0002644 9	0.0001319 6	0.0001990 8
0.00031610 7	0.00106316	0.00144258	-0.0011816	0.00038324	0.00013454	6.71E-05	-0.0002742	-0.0001368	-0.0002063
0.00019446 9	0.00096804	0.00143992	-0.001006	0.00025153	-0.0001392	-6.95E-05	0.0002829 2	0.0001412 3	0.0002127 6
0.00034009 6	0.00088195	0.00152367	-0.0009381	0.00036104	0.00014332	7.16E-05	-0.0002908	-0.0001452	-0.0002186
0.00023614 6	0.00075159	0.00145735	-0.0007191	0.00019019	-0.0001471	-7.34E-05	0.0002978 5	0.0001487 4	0.0002238 5
0.00042008	0.00067915	0.00146284	-0.000618	0.00029877	0.00015043	7.51E-05	-0.0003042	-0.0001519	-0.0002286
0.00030876 8	0.00056361	0.00129525	-0.0003385	0.00014682	-0.0001535	-7.67E-05	0.0003099 4	0.0001548 2	0.0002328 3
0.00047221 8	0.00051397	0.00131958	-0.0002376	0.00031069	0.00015619	7.80E-05	-0.0003151	-0.0001574	-0.0002367
0.00034319 4	0.00036887	0.00124348	3.31E-05	0.00021744	-0.0001586	-7.93E-05	0.0003197 4	0.0001597 5	0.0002401 1
0.00049547 5	0.00027353	0.00136622	0.00012385	0.00047309	0.00016086	8.04E-05	-0.0003239	-0.0001618	-0.0002432
0.00036746 6	0.00014244	0.00121963	0.00035849	0.00045088	-0.0001628	-8.14E-05	0.0003276 7	0.0001637 4	0.000246
0.00057306 2	9.77E-05	0.00113468	0.0003823	0.00076009	0.00016464	8.23E-05	-0.0003311	-0.0001654	-0.0002485
0.00047841 4	1.14E-05	0.00078638	0.0006049	0.00077102	-0.0001663	-8.31E-05	0.0003341	0.0001669 7	0.0002507 8
0.00070821 9	-1.93E-05	0.00065701	0.00066666	0.00103234	0.0001677	8.38E-05	-0.0003368	-0.0001683	-0.0002528
0.00061890 2	-7.30E-05	0.00041193	0.00090934	0.00099601	-0.000169	-8.45E-05	0.0003393 1	0.0001695 9	0.0002546 5
0.00078968 8	-0.0001332	0.0003704	0.00090786	0.00114622	0.00017018	8.51E-05	-0.0003415	-0.0001707	-0.0002563
0.00065407 3	-0.0002236	9.14E-05	0.00105382	0.00104721	-0.0001712	-8.56E-05	0.0003435 4	0.0001717 1	0.0002577 8
0.00079422 4	-0.0003501	-8.23E-05	0.00094055	0.00110924	0.00017219	8.61E-05	-0.0003453	-0.0001726	-0.0002591
0.00072233 8	-0.000417	-0.0005496	0.00112386	0.00084555	-0.0001731	-8.65E-05	0.0003469 6	0.0001734 3	0.0002603 2

0.000949756	-0.0004652	-0.0009064	0.00113967	0.00072135	0.00017382	8.69E-05	-0.0003484	-0.0001742	-0.0002614
0.000871646	-0.0004851	-0.0013807	0.00140567	0.00025466	-0.0001745	-8.72E-05	0.00034973	0.0001748 3	0.0002623 8
0.001023325	-0.0005757	-0.0016679	0.00133804	0.00010524	0.00017514	8.76E-05	-0.0003509	-0.0001754	-0.0002633
0.000796449	-0.0007744	-0.0020694	0.00148029	-0.0003297	-0.0001757	-8.78E-05	0.00035198	0.0001759 6	0.0002640 5
0.000756606	-0.0009933	-0.0022595	0.0013523	-0.0004083	0.00017621	8.81E-05	-0.0003529	-0.0001764	-0.0002648
0.000494185	-0.0012121	-0.0026128	0.00152177	-0.0008167	-0.0001767	-8.83E-05	0.0003538	0.0001768 7	0.0002654
0.000549594	-0.0012502	-0.0028012	0.00146872	-0.0008469	0.00017708	8.85E-05	-0.0003546	-0.0001773	-0.000266
0.000337025	-0.0012612	-0.0032555	0.00171538	-0.0011994	-0.0001775	-8.87E-05	0.00035528	0.0001776 1	0.0002665
0.000329475	-0.0013237	-0.003579	0.00159287	-0.0010901	0.00017779	8.89E-05	-0.0003559	-0.0001779	-0.000267
-0.000115729	-0.0016019	-0.003966	0.00175742	-0.0012337	-0.0001781	-8.90E-05	0.00035648	0.0001782 2	0.0002673 9
-0.000311701	-0.0018487	-0.0038438	0.00152801	-0.0009056	0.00017836	8.92E-05	-0.000357	-0.0001785	-0.0002678
-0.0008857	-0.0019973	-0.0037098	0.00150853	-0.0009478	-0.0001786	-8.93E-05	0.00035745	0.0001787	0.0002681 1
-0.001083335	-0.0018546	-0.0033514	0.00111931	-0.0006025	0.00017882	8.94E-05	-0.0003579	-0.0001789	-0.0002684
-0.001583572	-0.001739	-0.00343	0.00111709	-0.0006873	-0.000179	-8.95E-05	0.00035824	0.0001791	0.0002687
-0.001657792	-0.0015538	-0.0033775	0.00090215	-0.0005025	0.0001792	8.96E-05	-0.0003586	-0.0001793	-0.000269
-0.002151626	-0.0015397	-0.0034701	0.0008877	-0.0007773	-0.0001794	-8.97E-05	0.00035888	0.0001794 2	0.0002691 8
-0.002276867	-0.0014078	-0.0029891	0.00034021	-0.0006566	0.0001795	8.97E-05	-0.0003592	-0.0001796	-0.0002694
-0.002860462	-0.0012915	-0.0024999	-8.02E-05	-0.0008376	-0.0001796	-8.98E-05	0.0003594	0.0001796 9	0.0002695 6
-0.003022522	-0.0008086	-0.0016111	-0.000841	-0.000738	0.00017975	8.99E-05	-0.0003596	-0.0001798	-0.0002697

NB: Columns are samples whereas are rows are elements

Appendix C: Afan Oromo phonemes and their formation place

The following Tables represent the phoneme list of the language with their formation place, which is accepted and used by many researchers like Ishetu (1981), Waqo (1981), Fikadu (2010) and Dejene (2010) [11].

Table C. 1. The consonant phoneme lists of Afan Oromo and their formation place *[11]*

		Bilabial	Labio-dental	Alveolar/ Alveo-dental	Palatal	Velar	Glottal
Ejectives stop, Vl		*p'*		*t'*	*c'*	*k'*	
Plosives	Vl			*t*		*k*	*ʔ*
	Vd	*b*		*d*		*g*	
Implosive, Vd				*ɗ*			
Fricatives, Vl			*f*	*s*	*ʃ*		*h*
Affricate	Vl				*c*		
	Vd				*ɟ*		
Nasals, Vd		*m*		*n*	*ɲ*		
Lateral, Vd				*l*			
Flap, Vd				*r*			
Glides, Vd		*w*			*j*		

Ejective phoneme: non-pulmonic consonants formed by squeezing air trapped between the glottis and articulator further forward and releasing it suddenly.

Plosive phonemes: produced from opening a previously closed oral passage.

Implosive phonemes: formed by implosion.

Fricative phonemes: any of several sounds produced by air flowing through a constriction in the oral cavity and typically producing a sibilant, hissing or buzzing quality.

Affricative phonemes: a sound produced by a combination of a plosive and a fricative.

Nasal phonemes: the sounds that have a quality of imparted by means of a nose and specifically made by lowering the soft palate in some cases with closure oral passage, the voice thus issuing wholly or partially through the nose.

Lateral phonemes: generated by partially blocking the egress of the airstream with the tip of the tongue touching the alveolar ridge leaving space on one or both sides of the occlusion for the air passage.

Glide phonemes: semivowels.

Table C. 2. The vowel phoneme lists of Afan Oromo and their formation place [11]

	Front	central	Back
Close	i /ɪ/, ii /iː/		u /ʊ/, uu /uː/
Mid	e /ɛ/, ee /eː/		o /ɔ/, oo /oː/
Open		a /ɐ/	aa /ɑː/

In table C.1 the cʹ, c and pʹ are used as phonemes of c, ch and ph respectively. However, in international phonetic alphabet symbols the tʃʹ, tʃ and Φ are used instead. Therefore, in this thesis the international version of Afan Oromo phonemes were used.

Appendix D: The International Phonetic Alphabet (revised to 2015)

CONSONANTS (PULMONIC) © 2015 IPA

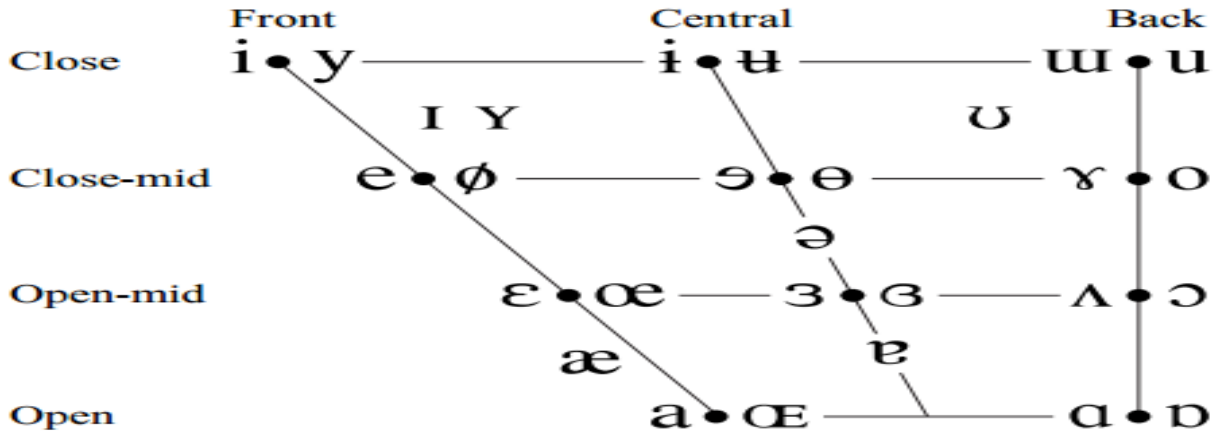
	Bilabial	Labiodental	Dental	Alveolar	Postalveolar	Retroflex	Palatal	Velar	Uvular	Pharyngeal	Glottal
Plosive	p b			t d		ʈ ɖ	c ɟ	k g	q ɢ		ʔ
Nasal	m	ɱ		n		ɳ	ɲ	ŋ	ɴ		
Trill	ʙ			r					ʀ		
Tap or Flap		ⱱ		ɾ		ɽ					
Fricative	ɸ β	f v	θ ð	s z	ʃ ʒ	ʂ ʐ	ç ʝ	x ɣ	χ ʁ	ħ ʕ	h ɦ
Lateral fricative				ɬ ɮ							
Approximant		ʋ		ɹ		ɻ	j	ɰ			
Lateral approximant				l		ɭ	ʎ	ʟ			

Symbols to the right in a cell are voiced, to the left are voiceless. Shaded areas denote articulations judged impossible.

CONSONANTS (NON-PULMONIC)

Clicks	Voiced implosives	Ejectives
ʘ Bilabial	ɓ Bilabial	’ Examples:
ǀ Dental	ɗ Dental/alveolar	p’ Bilabial
ǃ (Post)alveolar	ʄ Palatal	t’ Dental/alveolar
ǂ Palatoalveolar	ɠ Velar	k’ Velar
ǁ Alveolar lateral	ʛ Uvular	s’ Alveolar fricative

VOWELS



Where symbols appear in pairs, the one to the right represents a rounded vowel.