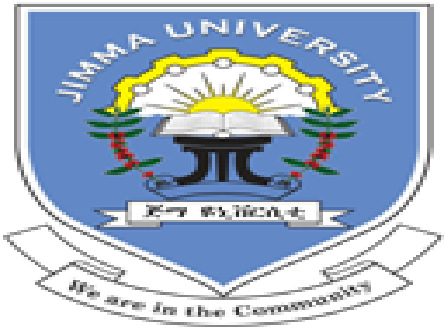


JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

FACULTY OF COMPUTING AND INFORMATICS

Afaan Oromo Continuous Speech Recognition

Using Deep Learning

Sifen Dadi Degefa

A THESIS SUBMITTED TO FACULTY OF COMPUTING AND INFORMATICS OF JIMMA INSTITUTE OF TECHNOLOGY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN INFORMATION TECHNOLOGY

April, 2021

Jimma, Ethiopia

JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

FACULTY OF COMPUTING AND INFORMATICS

GRADUATE PROGRAM IN INFORMATION TECHNOLOGY

Afaan Oromo Continuous Speech Recognition Using Deep Learning

Sifen Dadi Degefa

Advisor: Dr. Million Meshesha (PhD)

Co-Advisor: Mr. Teferi Kebebew (MSc)

A THESIS SUBMITTED TO FACULTY OF COMPUTING AND INFORMATICS OF JIMMA INSTITUTE OF TECHNOLOGY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN

INFORMATION TECHNOLOGY

April , 2021

Jimma, Ethiopia

Declaration

This thesis is my original work and has not been presented for a degree in any other universities.

Research Submitted By

Sifen Dadi Degefa _____ _____

Signature Date

Approved by advisors:

_____ _____ _____

Advisor Signature Date

_____ _____ _____

Co- Advisor Signature Date

Approved by faculty of computing and informatics Thesis Examination members

1. _____ _____ _____

Signature Date

2. _____ _____ _____

Signature Date

3. _____ _____ _____

Signature Date

Acknowledgement

First of all, I would like to thank the son of God Jesus Christ for he given me his life to save me from hell and not only that the almighty is also helping me in every activities of my life. Thank you Jesus for all the miracles in my life and family.

My heartfelt thanks should go to my advisor Dr. Million Meshesha for his proper guidance, constructive suggestions, comments and technical supports. Despite his busy schedule of work, starting from title selection up to end of the thesis work Dr. Million was supporting and guiding me a lot without any hesitation and boredom.

I am also thankful to Mr. Teferi Kebebew because his guidance technical supports and open comments supported me for the completion of this research. He was also very busy by work but starting from this thesis up to end Mr. Teferi was supporting me. Without my advisors it was very difficult to me to finalize it.

I also want to thank Jimma University Afaan Oromo department staffs and Mr. Endale Teshome for supported me a lot during corpus preparation for the study.

Table of Contents

Acronyms

AI:	Artificial Intelligence
ASR:	Automatic Speech Recognition
CNN:	Convolutional Neural Networks
CTC:	Connectionist Temporal Classification
DL:	Deep Learning
EBC:	Ethiopian Broadcasting Corporation
FBC:	Fana Broadcasting Corporate
FFT:	Fast Fourier Transform
HMM:	Hidden Markov Model
LVCSR:	Large Vocabulary Continuous Speech Recognition
NLP:	Natural Language Processing
OBN:	Oromia Broadcasting Network
OMN:	Oromia Media Network
RNN:	Recurrent Neural Networks
TV:	Television
VOA:	Voice of America

List of Tables

List of Figures

Abstract

Automatic Speech Recognition (ASR) works by taking an audio speech as an input and convert it to text as an output. In this study an attempt is made to design an automatic Afaan Oromo speech to text recognition using the state-of-the-art deep learning algorithm. Accordingly, the study explored the possibilities of developing a continuous speech recognition system for Afaan Oromo.

Previous related works on local languages and also for Afaan Oromo was reviewed but there was no any work on Afaan Oromo using deep learning algorithms; all the previous Afaan Oromo ASRs were based on traditional machine learning models. For this thesis, deep bidirectional RNN and CNN/RNN hybrid models have been proposed to show the possibility of developing ASR for local languages and Afaan Oromo using deep learning and to improve the performance of Afaan Oromo continuous speech recognition systems. For the purpose of conducting the experiment towards training, validating, and visualizing the model, Tensor flow, Keras, Jupyter Notebook, PyDub, Matplotlib and Pydot are tools used.

The speech corpus was prepared by collecting broadcast news audios from Ethiopian Broadcasting Corporation (EBC), Oromia Broadcasting Network (OBN), Oromia Media Network (OMN), Voice of America (VOA), Fana Broadcasting Corporation (FBC), and BBC Afaan Oromo program. Totally about 8000 utterances from 101 speakers (80 males and 21 females), which have 10:01:38 hours long data set was collected and transcribed. The dataset was used for both training, validation and testing.

We trained and evaluated both RNN and CNN/RNN models with connectionist temporal classification CTC to tackle sequence problems. We also tried to adjust learning rate, optimizers, number of neurons and number of layers of the recognizer model according to the available resources so as to increase the performance of the recognizer. Accordingly, multiple experiments were done and CNN/RNN hybrid model was chosen as the best model for our case.

Experimental results shown that, the best performance achieved was 69% WER and 16.3 loss by CNN/RNN hybrid model. Even if we get a promising result, from all experiments we understand that an increase in data and use of high performing GPUs for constructing large models could improve the performance of Afaan Oromo deep ASR. So we recommend further study needs to

be conducted with large vocabulary and better GPU to enhance the accuracy of ASR for Afaan Oromo language.

Key words: Afaan Oromo Continuous Speech Recognition; Automatic Speech Recognition; Broadcasting News Speech; Deep Learning; CNN, RNN

Chapter One

Introduction

1.1. Background

Speech is the most common way of human beings' communication and speech processing is one of the most exciting investigation areas of the signal processing [1]. Speech processing is learning of language signals and the processing techniques of these signals. The signals are usually processed in a digital format, so speech processing can be viewed as a unique case of digital signal processing which is applied to speech signal.

The task of speech recognition is to convert speech into a sequence of words by a computer program [2]. Computer-based processing and identification of human voices is known as speech recognition [2]. It can be used to authenticate users in certain systems, as well as provide instructions to smart devices like the Google Assistant, Siri or Cortana. Essentially, it works by storing a human voice and training an automatic speech recognition system to recognize vocabulary and speech patterns in that voice [3].

Automatic Speech Recognition (ASR) works by taking an audio speech as input and then giving the string of words as an output. For the development of ASR system, the audio format is needed that used as an input parameter, and a large text data is also required. The necessary audio could be found from read speech or from real world speech like spontaneous speech, telephone conversational speech and Radio/Television (TV) broadcasts [4].

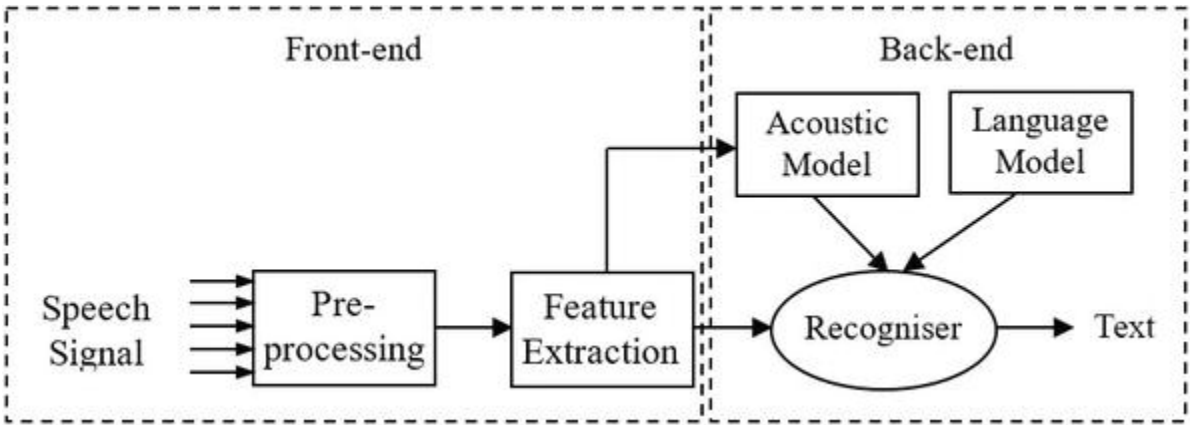


Figure 1.1 General overview of speech recognition

Many literatures reveal that an ASR system can be classified to different categories based on the nature of utterance, speaker type, and vocabulary size [5]. Based on the nature of utterances ASR system can be categorized as: isolated words speech recognition, connected words speech recognition, continuous speech recognition, and spontaneous speech recognition. Based on the type of speaker, speech recognition system can be categorized as speaker dependent and speaker independent. The last parameter to categorize speech recognition system is vocabulary size. Based on the size of vocabulary speech recognition system can be classified as small vocabulary, medium vocabulary, large vocabulary, and very large vocabulary [5] see their differences under chapter two.

One of the most critical issues in speech and language research is speech recognition, i.e. how machine recognizes human speech [6]. If the machine (computer) can perfectly understand what human says, it can be very useful for many other applications such as voice command, voice assistant, smart home system and robot audition [7]. This is why further research is needed which has high performance of recognition for local languages. If a speech recognition system results high performance in evaluation, it will improve the performance on other applications that are listed above.

Nowadays the speech recognition technique itself is rapidly developed from Hidden Markov Model (HMM) to Deep Neural Network (DNN), because using deep learning has achieved performance beyond HMM [8]. Non-Linearity is at the heart of Deep Learning and it is what makes the networks more expressive and more adaptive at learning from real world data. Linear models

lag behind by a large margin in terms of the results. Hence it is inevitable that DL models like RNNs will take over linear models like HMMs [9]. The trend of deep learning is to remove a lot of the complexity and human knowledge that was necessary in the past to build good ASR systems (e.g. speaker adaptation, phonetic context modeling, discriminative feature processing) and to replace them with a powerful neural network architecture that can be trained agnostically on a lot of data [10].

1.2. Statement of the problem

The need for automatic speech recognition is for facilitating interaction of machine with human with easier and in a way which is preferable by human. Because, natural speech is considered to be easier than other mechanisms of interaction, such as typing, pressing, rolling and sliding [11]. One of the reasons for such ease of use is that it is trivial to observe human beings learning the speaking skill of one's mother tongue language before any of the aforementioned skills used as human beings and machine interaction mechanism.

Speech recognition nowadays is unsolved for (very) under-resourced languages like Ethiopian languages, even if it can be considered as solved for well-resourced language like English, Spanish, Mandarin and Japanese. Therefore, further researches are needed for solving these problems.

Afaan Oromo is an Afro-Asiatic language, and the most widely spoken language of the Cushitic family. In addition to this, in Africa, it is the language with the fourth (4th) most speakers, after Arabic, Hausa and Swahili [12]. The speech technologies (speech synthesis, speech recognition) are at a very infant stage for Ethiopian languages and particularly for Afaan Oromo [13]. So, developing ASR for Afaan Oromo solves many human machine interaction problems related to this language.

There is no previous work which is conducted on Afaan Oromo speech recognition system using deep learning techniques. Previous researches for Afaan Oromo speech recognition were focused on HMM [14] [15] [16] [4] and hybrid of HMM/ANN [17], and the data set they use for experimental purpose was also not more than 4 hours.

This above mentioned machine learning techniques which used for Afaan Oromo speech recognition had exploited shallow-structured architectures [18]. That means these architectures typically contain a single layer of nonlinear feature transformations and they lack multiple layers of adaptive non-linear features.

A common property of these shallow learning models is the relatively simple architecture that consists of only one layer responsible for transforming the raw input signals or features into a problem-specific feature space, which may be unobservable [18].

Actually shallow architectures have been shown effective in solving many simple or well-constrained problems, but their limited modeling and representational power can cause difficulties when dealing with more complicated real-world applications involving natural signals such as human speech, natural sound and language, and natural image and visual scenes [19].

According to the review of related literatures, there are few works which developed for different Ethiopian languages and most of them were based on machine learning systems. Specifically, by using deep learning approaches, the work we found on local language is the study done by Tilaye [20]. In this study, the researcher applied Deep Recurrent Network to train a speech recognizer for Amharic. The researcher used a speech corpus which was prepared by Solomon et al [21] which was read speech. The researcher used acoustic model to train the recognizer. However, he did not put the accuracy level of his work and conclusion. However, for Afaan Oromo as discussed on related works under chapter two the researchers used HMM, hybrid of ANN/HMM and the like and none of Afaan Oromo ASR were done by DL.

So as the gap we have seen that Afaan Oromo ASR is still in it's infant stage and the techniques used are also traditional machine learnings. Therefore using deep learning for Afaan Oromo ASR solves or fills this gap because we can get better model in terms of their performance and the type of data they use to train and test. That means the possibility of having a system which works in real world environment would be widened.

Technological spectrum in the underdeveloped countries may be widened if the solution is available in regional spoken languages. It will open the gateway to a human being to talk to the computer with one's own mother tongue without bothering to learn internationally recognized

languages. Human want to speak to computer without losing naturalness then all the challenges and difficulties must be addressed and to address that we have to start from using state of the art techniques deep learning for Afaan Oromo ASR [22].

The aim of this study was therefore to investigate the possibility of developing a continuous Afaan Oromo broadcast news speech recognition systems using deep learning.

1.3. Research Questions

After conducting this study, the researcher must be able to answer the following research questions.

1. What are the challenges of collecting datasets for deep learning speech recognitions?
2. Which type of speech feature extraction technique is better for Afaan Oromo continuous speech features?
3. What are the challenges in developing speech recognition system for Afaan Oromo using deep learning approach?
4. To what extent the performance of Afaan Oromo speech recognition can be improved using deep learning technique?

1.4. Objective of the Study

1.4.1. General objective

The general objective of the research is exploring and designing better model for Afaan Oromo continuous speech recognition using deep learning so as to increase performance of the recognizer.

1.4.2. Specific objectives

To achieve the general objective of the study, this research attempted the following specific objectives.

- To review literature to understand the different approaches, techniques and tools that are used in speech recognition
- To prepare corresponding speech and text corpus of Afaan Oromo
- To prepare the transcriptions for Afaan Oromo speech corpus.
- To build acoustic model from Afaan Oromo speech corpus

- To build a prototype of continuous speech recognizer using deep neural network
- To evaluate the performance of the model

1.5. Scope and limitations of the study

The main aim of this study is to investigate the possibility of developing ASR System for large vocabulary continues speaker independent Afaan Oromo speech using deep learning. Accordingly, about 10 hours long speech data was collected from different sources of Afaan Oromo news speech.

However, on this research, problems like dialects, speaker gender, age, health and emotional state were not addressed because of time and budget. However, the researchers tried to address performance problem of speech recognizer of previous works and because of previous works used very small datasets we also tried to prepare a good quality data which can be used for this and further researches.

The experiment is conducted for only recognizing speech and convert to text, without identifying the varieties of dialects, gender and age. The varieties of gender are explained only to identify the number of speakers from collected speech corpus.

1.6. Significance of the study

In the advancement of speech technology human beings reach to a desire to communicate with machine such as Computer by using natural languages specifically using voice.

Since this research was conducted on continuous, speaker-independent Afaan Oromo speech using deep learning approaches, it showed that the possibilities of developing speech recognizer system for Afaan Oromo and other local languages by using deep learning techniques.

In addition, using this language in speech technology is one from technologically significant and good for the development of using the language in technology, because Afaan Oromo has wider number of speakers and it is also official regional government's working language of Oromia.

A broadcast news corpus was prepared for this study. Therefore, this corpus can be an input to other researchers who will conduct a research in speech recognition, speech translation and other similar areas. If we have a good Afaan Oromo ASR system, we can also use it for other applications, which are listed below.

The speech recognition has a various advantage as discussed by several researchers [23]. Among these advantages with the help of speech technology, users can easily control devices and create documents by speaking. Speech recognition allows documents to be created faster because the software generally produces words as quickly as they are uttered, which is usually much faster than a person can type. Dictation solutions are not only used by individuals but also by organizations that require massive transcription tasks such as healthcare and legal [24].

Speech recognition technology also makes invaluable contributions to organizations. Businesses that provide customer services benefit from speech technology to improve self-service in a way that enriches the customer experience and reduces organizational costs. With the help of the voice recognition technology, callers can input information such as name, account number, the reason for their call, etc. without interacting with a live agent. Instead of having callers remain idly on hold while agents are busy, organizations can engage their callers without live customer representatives. That is why speech recognition technology contributes to cost savings by minimizing or even eliminating the need for live agents while improving customer experience [24].

There are estimated to be about 15 million people in the United States alone who are disabled to some degree, and it has been hypothesized that at least 10% of the world's population experience some sort of physical impairment [25]. One of the most promising application areas for Automatic Speech Recognition (ASR) is then in helping people with disabilities. For disabled users, the range of possible applications for voice commands is wide, covering microcomputer and wheelchair control, to operating appliances in the domestic environment and many others [25].

ASR can also be used to analyze social media data. The social data contains not only textual data, but also a large number of audios, video, and image data. ASR for Afaan Oromo can be used to transcribe various audio recordings and videos on social media, and then developing ASR system can help other applications like hate speech detection and the like also.

The remaining of the document contained chapter two literature review and related works, chapter three methods and techniques, chapter four experiment and result discussion and last chapter which is chapter Five is about conclusions and future works.

Chapter Two

Literature Review

2.1. Overview of speech recognition

The goal of speech recognition is to make machines understand human speech and do accordingly. Automatic Speech Recognition (ASR) is one of the important tasks in the Artificial intelligence field and it is the process of converting speech data into text as shown in figure 2.1.

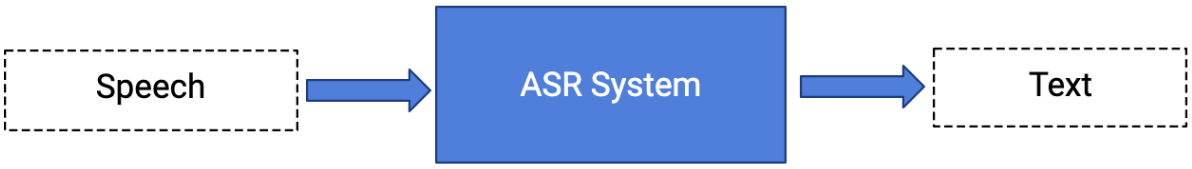


Figure 2.1. A generic ASR architecture

Speech recognition process is composed of components, which are responsible for the recognition process starting from analyzing the input sound signal to the decoding step which produces the corresponding text to the input signal. For every component of ASR system, there are different approaches. Modern and successful methods or approaches have to be used at each of the components for effective functioning and improved overall recognition performance. The other very important thing is unit of recognition such as phoneme, syllable, word, etc.
Therefore selecting an appropriate unit is another challenging task [32].

2.2. Types of speech recognition

There are different categories of automatic speech recognition systems based on the nature of utterance, speaker type, and vocabulary size [5]. Based on the nature of utterances ASR system can be categorized as isolated words vs. connected words speech recognition, as well as continuous vs. spontaneous speech recognition [5].

Isolated word recognizers usually require each utterance to have silence on both side of sample windows [33]. It accepts single word or single utterances at a time. It is mostly used for conditions that user is required to give only a single word. This type of ASR system is simple and easiest because of the word boundaries are simply identified [4]. Whereas, connected word is similar with the isolated words. However, the difference is allowing separate utterances to be run-together by having a minimal pause among words, which run together. Pause is used to show the boundary of words.

Continuous speech recognizer system allows user to speak almost naturally, while the computer will examine the content. Continuous speech recognizer system is developed by removing silences among connected words and word boundary is not simply identified like isolated words speech recognition system. Therefore, it is more difficult to develop speech recognizer from continuous speech [33]. Spontaneous speech is on the other hand a variety of natural speech feature such as words being run together. As discussed by [4] a conversation between two or more peoples can be taken as a good example for spontaneous speech. A System with spontaneous speech ability should be able to handle a variety of speech features like words being run together and even mispronunciation.

Based on the type of speaker, speech recognition system can be categorized as Speaker dependent vs. Speaker independent speech recognizer [5].

Speaker dependent speech recognizer system is developed for particular speaker. Speaker dependent speech recognition system is more accurate since developed for individual speaker and easier to develop and cheaper when compared with speaker independent. But, because of it works for particular speakers it is not flexible as speaker independent recognizer [4]. Unlike speaker dependent, speaker independent speech recognition system can recognize different speakers. However, implementation of such system is not easy and cheap like speaker dependent, and the accuracy of this system is lower than the speaker dependent [4].

The last parameter to categorize speech recognition system is vocabulary size. Based on the size of vocabulary speech recognition system can be classified as: small vocabulary, medium vocabulary, large vocabulary and very large vocabulary [5].

Small vocabulary speech recognizer contains tens of words or 1 to 100 words and mostly suitable for developing command-control. Medium vocabulary speech recognition system, on the other hands includes hundreds of words or 101 to 1000. Also, large vocabulary speech recognizer

system includes thousands of words from 1001 to 10000. Finally, very large size vocabulary includes tens of thousands of words or more than 10000 words [5].

2.3. Steps in speech recognition

According to [34] in speech recognition system an unknown speech signal is transformed into sequence of feature vectors by different speech processing techniques. It converts feature vector to phoneme lattice by applying an algorithms [7]. A recognition module transforms the phoneme lattice into a word lattice by lexicon and then grammar is applied to word lattice to recognize the specific words or text. The figure below shows the information for general steps in speech recognition system (SRS). The speech recognition process is divided into several steps (See figure 2.2) [35].

Step 1: In this step, speech signal is divided into equally spaced blocks to get signal characteristics such as, total energy, zero crossing strength across various frequency ranges etc. By using these characteristics feature vectors combine each block with the phoneme to produce a string of phonemes.

Step 2: In this step spectrum analysis is applied on each block by using linear predictive coding technique, fast Fourier transform(FFT) and bank of frequency filters.

Step 3: In this step decision process is performed on each block. Each phoneme has distinguished features which narrow the field.

Step 4: This step is used to enhance the performance of decision process to get high degree of success using different algorithms. For each word of vocabulary an algorithm is constructed and then string of phonemes is compared against each algorithm.

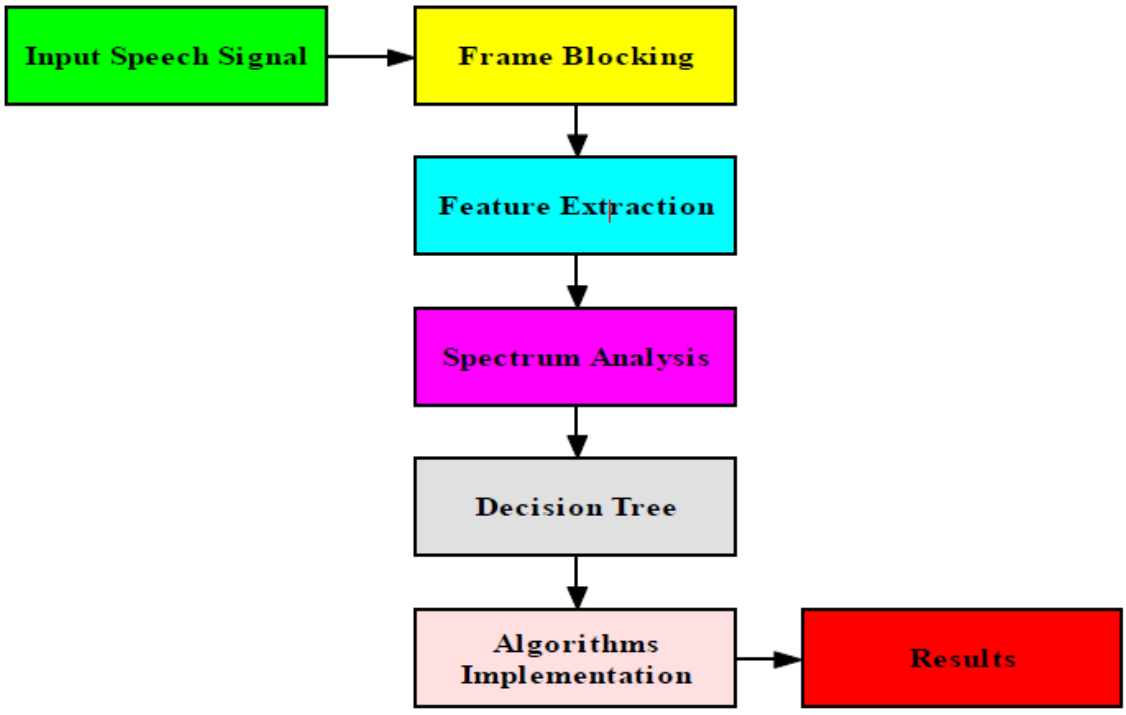


Figure 2.2 General steps for speech recognition system from [35]

2.4. Speech Feature extraction

Feature extraction is accomplished by changing the speech waveform to a form of parametric representation at a relatively minimized data rate for subsequent processing and analysis. Therefore, acceptable classification is derived from excellent and quality features. Mel Frequency Cepstral Coefficients (MFCC), Spectrograms, Linear Prediction Coefficients (LPC), Linear Prediction Cepstral Coefficients (LPCC), Line Spectral Frequencies (LSF), Discrete Wavelet Transform (DWT) and Perceptual Linear Prediction (PLP) are the speech feature extraction techniques.

MFCCs are avery common speech feature extraction methods in which it tries to mimics the human auditory system. It's measure advantage is MFCC captures main characteristics of phones in

speech and also has low Complexity [36]. But in background noise MFCC does not give accurate results so they perform better in clean speech than speech with background noise [36].

The spectrogram of sound signal is derived from the discrete-time Fourier series DTFS of shifted and windowed copies of the signal [37]. Spectrograms are good for noisy speech feature extraction.

Linear Prediction Coefficients (LPC) analyzes the speech signal by estimating the formants, removing speech signal, and estimating the intensity and frequency of the remaining buzz [36]. It provides autoregression based speech features. This technique's advantage is computation speed of LPC is good and provides with accurate parameters of speech. But it's weakness is LPC generates residual error as output that means some amount of important speech gets left in the residue resulting in poor speech quality [36]. . Figure 2.3 shows general steps of Linear Prediction Coefficients.

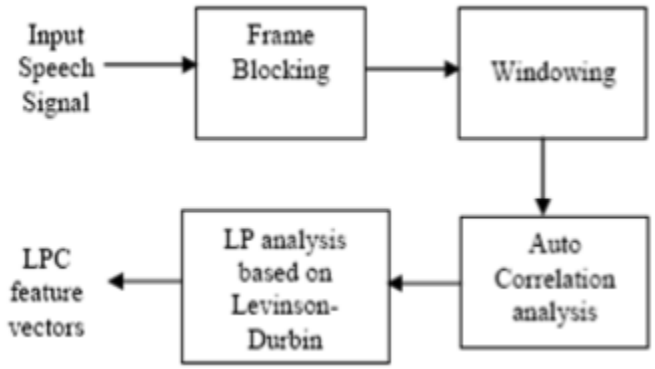


Figure 2.3 steps of LPC from [36]

Linear Prediction Cepstral Coefficients (LPCC) is also another feature extraction method in speech recognition derived from LPC calculated spectral envelope. One advantage of LPCC is it have low vulnerability to noise. It's disadvantage is cepstral analysis on high-pitch speech signal gives small source-filter separability in the quefrency domain [38].

.

Discrete Wavelet Transform (DWT) : Wavelet transform decomposes a signal into a group of basic functions called wavelets. It's advantage is it's speed of computation is high. But it has no good noise resistance ability [39].

2.5. Recognition and classification techniques

Until recently many ASR systems recognizing and classifying methods was built using one of the following methods which are Hidden Markov Model (HMM), Dynamic Time wrapping (DTW), Dynamic Bayesian Networks (DBN), Support Vector Machine (SVM). In order to cover some ASR sub tasks such as acoustic modeling and language modeling Artificial Neural Network (ANN) and N-grams were widely used [40].

The hidden Markov models are statistical models used in many real-world applications and communities. The strengths of the HMM method is its mathematical framework which provides straightforward solution to related problems and its implementational structure which provides flexibility in dealing with various speech recognition tasks and the ease of implementation [18]. One of it's main drawback is the Markov assumption itself. Hidden Markov Modeling is based on the Markov property, which states that the probability of being in a given state at time t only depends on the state at time t-1.This is not always the case for speech sounds where dependencies sometimes extend through several states [18].

Dynamic Time wrapping (DTW): The technique of dynamic time warping (DTW) is relied on heavily in isolated word recognition systems [41]. The advantage of using DTW is that reliable time alignment between reference and test patterns is obtained. The disadvantage of using DTW is the heavy computational burden required to find the optimal time alignment path. Several alternative procedures have been proposed for reducing the computation of DTW algorithms. However these alternative methods generally suffer from a loss of optimality or precision in defining points along the alignment path [41].

Support vector machine or SVM works relatively well when there is a clear margin of separation between classes. But it's drawback is SVM algorithm is not suitable for large data sets and does not perform very well when the data set has more noise i.e. target classes are overlapping [42].

However deep learning finally made speech recognition and classification accurate enough to be useful in real and uncontrolled environments [30]. Convolutional neural networks, recurrent neural networks and connectionist temporal classifications are among the best performing deep learning speech classification and recognition architectures [43] [30].

2.6. Related works

Research on Automatic Speech Recognition (ASR) has been the focus of many researchers. As mentioned by Juang et al. [44], the speech recognition was started by building a system for isolated word recognition system for a single speaker in Bell Laboratories in the year of 1952 by Davis, Biddulph, and Balashek for digits using the formant frequencies measured (or estimated) during vowel regions of each digit. This was done by designing to develop the ASR system for the ten digits (one to nine and zero). Automatic Speech Recognition requires still more investigation because of the variety of languages spoken in the world.

Saon et al. [10] used a collection of acoustic and language modeling techniques that lowered the word error rate of their English conversational telephone LVCSR system to a switchboard dataset. The researchers use 2000 hours of English conversational telephone speech, On the acoustic side, the researchers use a score fusion of three strong models: recurrent nets with maxout activations, very deep convolutional nets with 3x3 kernels, and bidirectional long short-term memory nets which operate on FMLLR and i-vector features. On the language modeling side, they use an updated model "M" and hierarchical neural network LMs. As the researcher discussed the performance is less than 3% away from achieving human level accuracy on the Switchboard data. Finally the researcher concludes as, it looks like future improvements on this task will be considerably harder to get and will probably require a breakthrough in direct sequence-to-sequence modeling and a significant increase in training data.

For local languages there are a few works which are done on ASR. Among them:

Wubshet [23] Developed Hidden Markov Model Based Large Vocabulary, Speaker Independent, Continuous Amharic Speech Recognition and HTK tools was used. At the end of his experiment,

he achieved the accuracy of 79% word level correctness, 76.18% word accuracy, and 30.01% sentence level correctness. Finally, he concluded that as the result of his experiment is a proof of the fact that it is possible to construct an Amharic large vocabulary, speaker independent continuous speech recognizer using the HTK toolkit and the HMM modeling technique.

Solomon [11] tried to develop a Large Vocabulary Speaker Independent Continuous Speech Recognition System for Amharic by developing an Amharic speech corpus that can be used for several kinds of investigations into the nature of spoken Amharic. Then researcher used HMM topology for Amharic CV syllables and also used speaker adaptation. Then they obtained word recognition accuracy of 90.43% on evaluation test set. The researcher concluded that Amharic CV syllables as represented by orthographic symbols is better alternative to prevailing modeling units of elementary sounds like phones.

Adugna [45] Develop a spontaneous, speaker independent Amharic speech recognizer by using speeches such as conversation between two or more speakers in the form of conversation. The researcher use the speech data with 44100 Hz sampling rate obtained from web that are recorded in local Medias. The speech data used for both training and testing are conversational speeches which are made between two or more speakers. Hidden Markov Model (HMM) was used and HTK toolkit was also employed. The performance result that the researcher got was 41.60% and 23.25% of word accuracy for test data from speakers those are involved in training and speakers those do not involved in training, respectively.

Abera [34] tried to design speaker independent continuous Tigrigna recognition system. The researcher used the HMM as modeling techniques and the work is done by HTK toolkit. Therefore the performance result obtained are 60.32%,58.38%, and 20 % for word level correctness, word accuracy, and sentence level correctness, respectively.

Specifically by using deep learning approaches, the work we found on local language is the study done by Tilaye [20]. In this study, the researcher applied Deep Recurrent Network to train a speech recognizer for Amharic. The researcher used a speech corpus which was prepared by Solomon et al [21]. He also used MFCC to extract features from audio. The researcher used acoustic model to train the recognizer. However, he does not put the accuracy level of his work and conclusion.

In addition to the above-mentioned works, the researchers try to discuss speech recognition, which conducted particularly for Afaan Oromo.

Ashenafi [14] This thesis work was on Speech Recognition System for Afaan Oromo isolated words and used the HMM model and an open source speech recognition toolkit Sphinx4. A researcher prepared 50 Afaan Oromo words as corpus by consulting the domain experts. Then 20 persons read these words and 1000 utterances of Afaan Oromo isolated words were obtained. When the 66.67% of the data was used for training, the remaining (33.33% of the data) was used for testing purpose. At the end, the word level accuracy achieved by the researcher's work was 82.83% and 81.081% for context dependent phoneme based model and context independent word based, respectively.

Kasahun [15] Tried to develop a Continuous, Speaker Independent Speech Recognizer for Afaan Oromo. The researcher try to explore the possibility of developing continuous Afaan Oromo speech recognition system using HMM model and sphinx system (Sphinx train for training and Sphinx4 for decoding). In this study 70 Afaan Oromo long words, phrases, and simple sentences were selected that were read by 30 people who are different by their age and gender. Thus, generally a corpus consisting of 2100 utterances was prepared. He used the 66.67% of the data for training; and the remaining (33.33% of the data) for testing purpose. The performance level which achieved by the researcher was 68.514% with sentence accuracy of 28% and 89.459% with the sentence accuracy of 42% for word level and tri-phone based recognition system, respectively.

Teferi [17] Also try to develop a speech recognition for Afan Oromo and the possibility of its applicability. The researcher used a hybrid HMM/ANN model to see its effectiveness compared to the more common HMM. For developing and testing the hybrid ASR system, CSLU hybrid toolkit was used along with other tools used for recording and labeling the speech corpus. The experiment was conducted on recognition of limited vocabulary. A total of hundred Afan Oromo word are selected and all the 29 phonemes of the language are considered during the selection. The words are organized to form sentences to make the recording easy. For his research, speech was recorded and then labeled manually for the experimental process. The system was trained and tested with the labeled speech and the final result achieved was 98.11%.

Yadeta [4] tried to explore the possibilities of developing a large vocabulary, speaker-independent, continuous speech recognition system for Afaan Oromo using broadcast news speech corpus. In the study statistical (stochastic) approach and Hidden Markov Model (HMM) modeling techniques were used and tools like HTK, Audacity, and SRILM were also used. The speech corpus was collected from different sources like: Oromia Radio and Television Organization (ORTO), Voice of America Afaan Oromo program (VOA), and Fana Broadcasting Corporate (FBC). Totally, 2953 utterances (about 6 hours speech) were prepared from 57 speakers (42 males and 15 females), a text corpus that is required for language modeling was collected from Bariisaa Afaan Oromo newspaper and bigram language model was developed using the SRILM language modeling tool. Out of 2953 utterances, 2653 were used for training and the remaining 300 utterances prepared from 12 speakers (9 males and 3 females) which are about 40 minutes long were used for testing the developed speech recognizer. Speakers who are involved in testing were not involved in training. The researcher developed context independent (mono-phones based) and context dependent (tri-phones based) acoustic models and the best performance he obtained in terms of word error rate was 91.46% WER and 89.84% WER, for context-independent and context-dependent, respectively. Based on the findings he concluded that increasing the Gaussian number to 12 and tuning parameters for word insertion penalty 1.0 and grammar scale factors to 15.0 can improve the performance of the system.

Duressa [16] tried to explore the possibility of developing large vocabulary continuous read speech recognizer prototype for Afaan Oromo using Hidden Markov Model. The researcher use one hour read speech data, which was segmented and labeled into sentences with freely available toolkits. Prototype of large vocabulary continuous speech recognition system for Afaan Oromo was modeled using CMU sphinx open source speech recognition toolkit. Then he tested the prototype using two types of test data set. The first is speaker dependent test data set in which speaker that participated in training were participated in testing. The second is speaker independent test data set in which speaker participated in testing were do not participated in training. The experimentation was done in two distinct phases. The first phase experimentation was done using with skip transition topology, context dependent tri-phone and using 8 Gaussian mixture. The second phase experimentation was done using without skip transition topology, context dependent tri-phone and using 8 Gaussian mixture. The language model the researcher used for this study

were both bigram and tri-gram. From the experimentation result bigram language model performed the highest word accuracy 93% for speaker dependent test data set and 43.6% word accuracy for speaker independent test data set. From the experimentation result the researcher concluded that increasing the training data and language model do not a guarantee to increase the performance of recognizer. In addition to corpus size training acoustic model by using different acoustic model parameters is important. In this study the researcher trained his acoustic model by Gaussian mixture, tied state, acoustic model transition topology and acoustic model types.

Research gap

Therefore, none of earlier Afaan Oromo language speech recognition studies conducted using deep learning technique and the speech corpus they use were not more than 4 hours.

This above mentioned machine learning and signal processing techniques which used for Afaan Oromo speech recognition had exploited shallow-structured architectures. That means these architectures typically contain a single layer of nonlinear feature transformations and they lack multiple layers of adaptive non-linear features.

Examples of the shallow architectures including above mentioned ones are, commonly used Gaussian mixture models (GMMs) and hidden Markov models (HMMs), linear or nonlinear dynamical systems, conditional random fields (CRFs), maximum entropy (MaxEnt) models, support vector machines (SVMs), logistic regression, kernel regression, and multi-layer perceptron (MLP) neural network with a single hidden layer including extreme learning machine [19].

A common property of these shallow learning models is the relatively simple architecture that consists of only one layer responsible for transforming the raw input signals or features into a problem-specific feature space, which may be unobservable. We can take the example of a SVM and other conventional kernel methods. These algorithms use a shallow linear pattern separation model with one or zero feature transformation layer when kernel trick is used or otherwise.

Actually shallow architectures have been shown effective in solving many simple or well-constrained problems, but their limited modeling and representational power can cause difficulties

when dealing with more complicated real-world applications involving natural signals such as human speech, natural sound and language, and natural image and visual scenes [19].

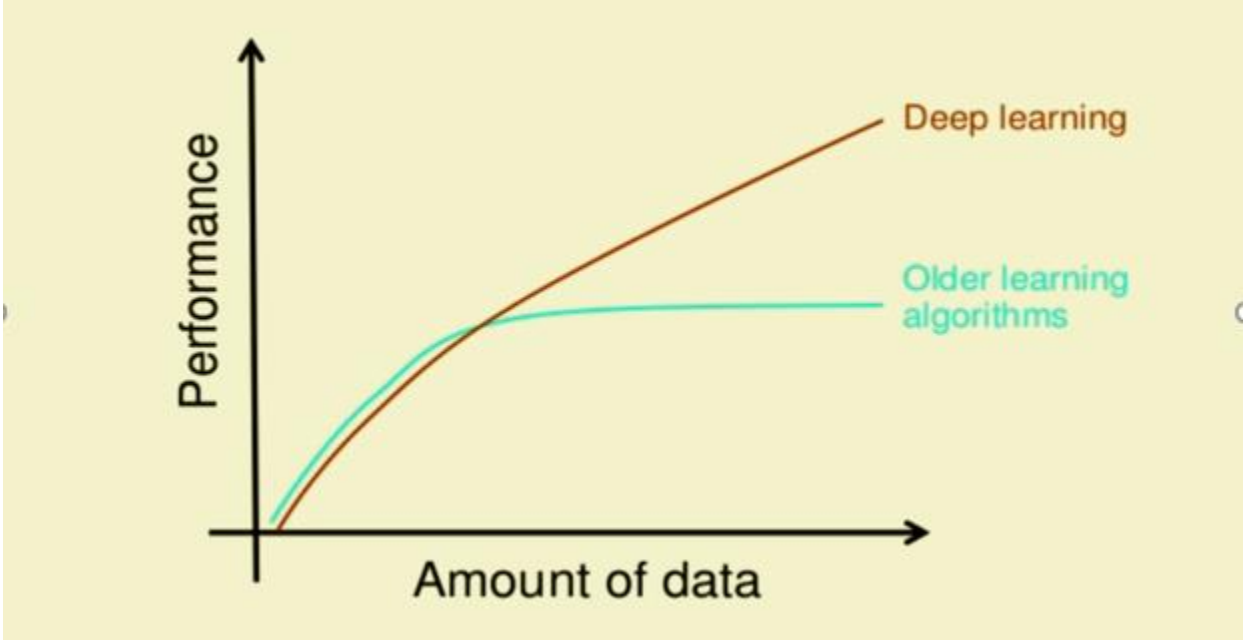


Figure 2.2 Why deep learning [[46]]

So the gap or the thing which motivates us to do this research is the above mentioned problems of already used methods. In addition to that, because of machine learning techniques need the applied features to be identified by domain expert in order to reduce the complexity of the data and make patterns more visible to learning algorithms to work that means it needs more cleaned and structured data.

To use real world data and system that is more real especially when it comes to complex problems such as image classification, natural language processing, and speech recognition deep learning really shines.

The biggest advantage of Deep Learning algorithms as discussed before are that they try to learn high-level features from data in an incremental manner. This eliminates the need of domain expertise and hard-core feature extraction. In addition to that, machine-learning can only handles small amount of data and they are only good at them.

Therefore, applying deep learning techniques for Afaan Oromo speech recognition solves these above mentioned problems and can have more applicable systems for real world problem.

Overview of Afaan Oromo language and deep learning

2.7. Afaan Oromo language

The great Oromo people has their own culture, tradition, customs and language and they are the largest ethnic group in Ethiopia and account for more than 40% of the population [47] [16]. This ethnic group also lives in some other parts of Africa like parts of Kenya and Somalia. Afaan Oromo is an Afro-Asiatic language, and the most widely spoken language of the Cushitic family. In addition to this, in Africa, it is the language with the fourth (4th) most speakers, after Arabic, Hausa and Swahili [12]. As discussed on [16] in 1991 G.C, the Latin alphabet started being used for Afaan Oromo writing and adopted as official alphabet of Afaan Oromo. This writing system is called Qubee in Afaan Oromo. Now it is language of public media, education, social issues, religion, political affairs, and technology.

2.8. Afaan Oromo alphabets (Qubee Afaan Oromoo)

Afaan Oromo Alphabets 'Qubee' contain 32 letters, twenty-six Latin letters and additional 6 double letters. The Afaan Oromo alphabets are classified into two main categories namely vowels ***'Qubee Dubbachiiftuu'*** (a, e, i, o, u) and consonants ***'Qubee Dubbifamaa'*** (b, c, d, f, g, h, j, k, l, m, n, p, q, r, s, t, v, w, x, y, z, ch, dh, ny, ph, sh, ts) . Vowels in Afaan Oromo are 5 in number. However, the number of consonants is 27 [4].

Afaan Oromo vowels (Qubee Dubbachiiftuu) are categorized as long vowels and short vowels [12]. Short vowels use single vowel letter (like 'Dhuga' which means drink) and have short sound and long vowels use double vowel letters ('Dhugaa' meaning truth) and have long sound.

Afaan Oromo consonants (Qubee dubbifamaa) are also two types geminated consonants and non-geminated consonants. Geminated consonants are consonants that have geminated sound by doubling same consonants and non-geminated ones have single consonant followed by vowel. For instance let us see two Oromo words 'Gadaa' which have non-geminated sound, which means Oromo peoples' traditional democratic governmental system (Gada system) and Gadda has geminated sound which means sad. All Afaan Oromo consonants have geminate forms except 'h' and compound symbols [16].

2.9. Afaan Oromo phonetics

Phonetics is the study of sounds (phonemes) of a language that produce a word. Afaan Oromo alphabets or letters have their International Phonetics Alphabet (IPA) representation [12].

Alphabets/Qubee	Aa	Bb	Cc	CH ch	Dd	DH dh	Ee	Ff	Gg	Hh	Ii
IPA for Qubee	[a]	[b]	[ʧ']	[ʧ]	[d]	[ɗ]	[e]	[f]	[g]	[h]	[i]
Alphabets/Qubee	**Jj**	**Kk**	**Ll**	**Mm**	**Nn**	**NY ny**	**Oo**	**Pp**	**PH ph**	**Qq**	**Rr**
IPA for Qubee	[ʤ]	[k]	[l]	[m]	[n]	[ɲ]	[o]	[p]	[p']	[k']	[r]
Alphabets/Qubee	**Ss**	**SH sh**	**Tt**	**TS ts**	**Uu**	**Vv**	**Ww**	**Xx**	**Yy**	**Zz**	**'**
IPA for Qubee	[s]	[ʃ]	[t]	[ʦ]	[u]	[v]	[w]	[t']	[j]	[z]	[ʔ]

Table 3.1 Afaan Oromo alphabets with their IPA representations

2.10. Overview of Deep learning

Deep learning is machine learning algorithm as a new name for an approach to artificial intelligence called neural networks with multiple hidden layers, which have been going in and out of fashion for more than 70 years [48].

The history of deep learning dates back to 1943 when Neural networks were first proposed by Warren McCullough and Walter Pitts. A neurophysiologist Warren McCulloch and a mathematician Walter Pitts created a computer model based on the neural networks of the human brain [49]. Walter Pitts and Warren McCulloch used a combination of mathematics and algorithms, and they called threshold logic to mimic the thought process. Their model was known as MCP neural model [49] [50].

Despite the resemblance between MCP Neural Model and modern perceptron, they are still different distinctly in many different aspects like MCP Neural Model is initially built as electrical circuits and the weights of MCP Neural Model are fixed, in contrast to the adjustable weights in modern perceptron [49].

Deep Learning (DL) uses multiple layers of algorithms to process data, to understand human speech, and visually recognize objects. Information is passed through each layer, with the output of the previous layer providing input for the next layer [50].

Deep learning is not a single approach but rather a class of algorithms and topologies that you can apply to a broad spectrum of problems. Deep learning is not certainly new, but it is experiencing explosive growth because of the intersection of deeply layered neural networks and the use of GPUs (graphical processing units) to accelerate their execution [51]. Big data has also fed this growth. Because deep learning relies on supervised learning algorithms (those that train neural networks with example data and reward them based on their success), the more data, the better to build these deep learning structure.

There are different types of deep learning architectures like recurrent neural networks (RNNs), long short-term memory (LSTM)/gated recurrent unit (GRU), convolutional neural networks (CNNs), deep belief networks (DBN), and deep stacking networks (DSNs) [51].

Convolutional neural networks

Kunihiko Fukushima who designed the neural networks with multiple pooling and convolutional layers first used convolutional neural networks (CNNs). Neocognitron Kunihiko was an artificial neural network developed by Fukushima in 1979, which used a multi-layered and hierarchical design. The multi-layered and hierarchical design allowed the computer to learn to recognize visual patterns [52].

CNNs have a special architecture, which is particularly well adapted to classify images [53]. This architecture makes convolutional networks fast to train. This, in turn, helps us train deep, multi-layer networks, which are very good at classifying images. These days, deep convolutional networks are widely used in most neural networks for image recognition. This architecture uses three basic ideas: local receptive fields, shared weights, and pooling [53]. In local receptive fields, each neuron in the first (or any) hidden layer will be connected to a small region of the input(or previous layer's) neurons. Shared weights mean that the same weights and bias are used for each of the local receptive field. This shows that all the neurons in the hidden layer detect exactly the same feature, just at different locations in the input image. There are also pooling layers that are contained in convolutional neural networks in addition to the convolutional layers just described. These Pooling layers are usually used immediately

after convolutional layers. The purpose of pooling layers is to simplify the information in the output from the convolutional layer. Recent and currently widely used convolutional network architectures have 10 to 20 hidden layers and billions of connections between units.

Convolutional neural networks can be classified as 1D, 2D and 3D convolutions, which stands for one-dimensional CNN, two-dimensional CNN and three-dimensional CNN respectively [54].

One-dimensional or Conv1D CNNs are CNNs with one dimension or kernel slides along one dimension. This property of Conv1D CNNs make them suitable for time series data like speech or sound. **Two-dimensional CNN** is generally used on Image data. Because of the **kernel** slides along two dimensions on the data, it is called two-dimensional CNN. In three-dimensional or **Conv3D CNNs,** the **kernel** slides in three dimensions. We use Conv3D CNNs mostly with 3D image data, such as **Magnetic Resonance Imaging** (MRI) data.

Recurrent neural networks (RNNs)

The other deep learning architecture is recurrent neural networks (RNNs). RNN is one of the fundamental network architectures from which other deep learning architectures are built [43]. Reccurent neural networks includes a rich set of deep learning architectures. RNNs can use their internal state memory to process variable length sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition [55]. Unlike feedforward neural networks, RNNs can use their internal memory to process arbitrary sequences of inputs [53]. Every information which is processed, is captured, stored, and utilized to calculate the final outcome. Furthermore, the recurrent network might have connections that feedback into prior layers (or even into the same layer). This feedback allows them to maintain the memory of past inputs and solve problems in time [43].

RNNs are very useful when it comes to fields where the sequence of presented information is mandatory.

Previous early RNN models turned out to be very difficult to train, harder even than deep feedforward networks. This was because of the unstable gradient problem such as vanishing gradient and exploding gradient. Gradient can get smaller and smaller as it is propagated back through layers. This makes learning in early layers extremely slow. The problem actually gets worse in RNNs, since gradients are not just propagated backward through layers; they are propagated backward through time. When the network runs for a long time, that can make the gradient extremely unstable and hard to learn from. To

solve unstable gradient problem it is possible to incorporate an idea known as long short-term memory units (LSTMs) into RNNs. LSTMs make it much easier to get good results when training RNNs and many recent papers make use of LSTMs or related ideas [53].

There are two types of RNN [43]:

1. Bidirectional RNN: In this architecture, the output layer can get information from past and future states simultaneously.
2. Deep RNN: because of multiple layers are present, the DL model can extract more hierarchical information.

Long short-term memory (LSTM) and gated recurrent unit (GRU)

LSTM was created by Hochreiter and Schimdhube in 1997 [51]. Long short-term memory (LSTM) is an RNN architecture used in the field of deep learning and has feedback connections. It can process sequential data such as speech or video so it is applicable to tasks such as connected handwriting recognition and speech recognition [56].

LSTM is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell [55].

Long short term memory LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. LSTMs were developed to deal with the vanishing gradient problem that can be encountered when training RNNs [43]. The **most Control-ability and thus, better Results,** is an advantage of LSTM over RNNs [57].

The main advantage of an LSTM cell when compared to a common recurrent unit is its cell memory unit. The cell vector has the ability to encapsulate the notion of forgetting part of its previously stored memory, as well as to add part of the new information [43].

The other deep learning architecture is Gated recurrent units (GRUs). GRUs are a gating mechanism in recurrent neural networks and are like LSTMs with forget gates and it lacks an output gate [58]. However, GRU is simpler than the LSTM and can be trained more quickly because it has fewer parameters than LSTM [43]. GRU's performance on certain tasks like speech

signal modeling and natural language processing was found to be similar to that of LSTM and can be more efficient in its execution. However, the LSTM can be more expressive and with more data, can lead to better results [43].

Deep belief networks (DBN)

DBN is a multilayer network with many hidden layers in which each pair of connected layers is a Restricted Boltzmann Machine (RBM). DBN is a stack of RBMs. DBN is composed of multiple layers of latent variables ("hidden units"), with connections between the layers but not between units within each layer [59].

Deep stacking networks

DSN is deep learning network that contains a deep set of individual networks, each with its own hidden layers [43]. Each layer in a deep learning architecture exponentially increases the complexity of training, so the DSN views training not as a single problem but as a set of individual training problems. Because of DSNs can perform better than typical DBNs they become popular and efficient network architectures [43].

Deep learning methods have many application areas, and really outperforms in complex and real world problems. Particularly in the image processing, speech recognitions and self-driving cars [60].

For instance, a deep learning architecture called Convolutional Neural Networks (CNNs) are designed to imitate the behavior of human visual cortex. So it performs very well on any visual recognition tasks. The CNN architecture consists of special layers called convolutional layers and pooling layers. These layers allow the network to encode certain images properties. Auto encoders is another class of a deep learning architecture. Auto encoders like stacked denoising auto encoders are used to reduce the input data by decreasing the dimensionality of the feature space and cleaned a partially corrupted output.

Another area where deep learning is successfully applied is automatic speech recognition. In automatic speech recognition, good acoustic and language models are combined [60]. The speech recognition problem involves time series data. Recurrent Neural Network (RNN) is the best architecture for time series sequencial data that contains loops in the hidden layer to retain the

information at the previous time step to predict the value of the current time step. This mechanism helps RNNs to handle different speaking rates[6].

While analyzing speech recognition tasks temporal dependencies could also be an issue. LSTMs which is one type of RNN can solve temporal dependencies that may present in the short term or long term depending on the speech recognition problem. RNNs can be applied to a variety of problems such as machine translation, image captioning, and speech recognition [60].

The Connectionist Temporal Classification (CTC) is used to convert continuous and unsegmented data into labeled sequences. That means for labeling sequence data in training with RNN, CTCs solves difficulty of facing with more observations than actual labels which associated with observations or the CTC method has been proven to be helpful where alignment between input and output labels is unknown [61].

Training deep learning networks has taken weeks several years ago, but thanks to progress in GPU and algorithm enhancement, training time has reduced to several hours [53].

Chapter Three

Methods and Techniques

In this chapter general architecture of the proposed model, feature extraction algorithms, recognition algorithms, decoders, evaluation techniques and speech recognition tools are discussed.

3.1. The architecture

Speech recognition is the ability of a machine or program to identify words and phrases in spoken language and convert them to a machine readable format. Usually, simple implementations of these algorithms which is as traditional machine learning base ones has a limited vocabulary and small amount of data, and it may only identify words/phrases if they are spoken very clearly [62].

However, over time, the computer can learn to understand speech from experience, thanks to incredible recent advances in deep learning. Deep learning has dramatically improved the state-of-the-art in many different artificial intelligent tasks including speech recognition [49]. Its deep architecture nature grants deep learning the possibility of solving many more complicated AI tasks [49].

As a result, in this thesis work researchers try to apply deep learning to a continuous speech recognition for under resourced local language Afaan Oromo automatic speech recognition system.

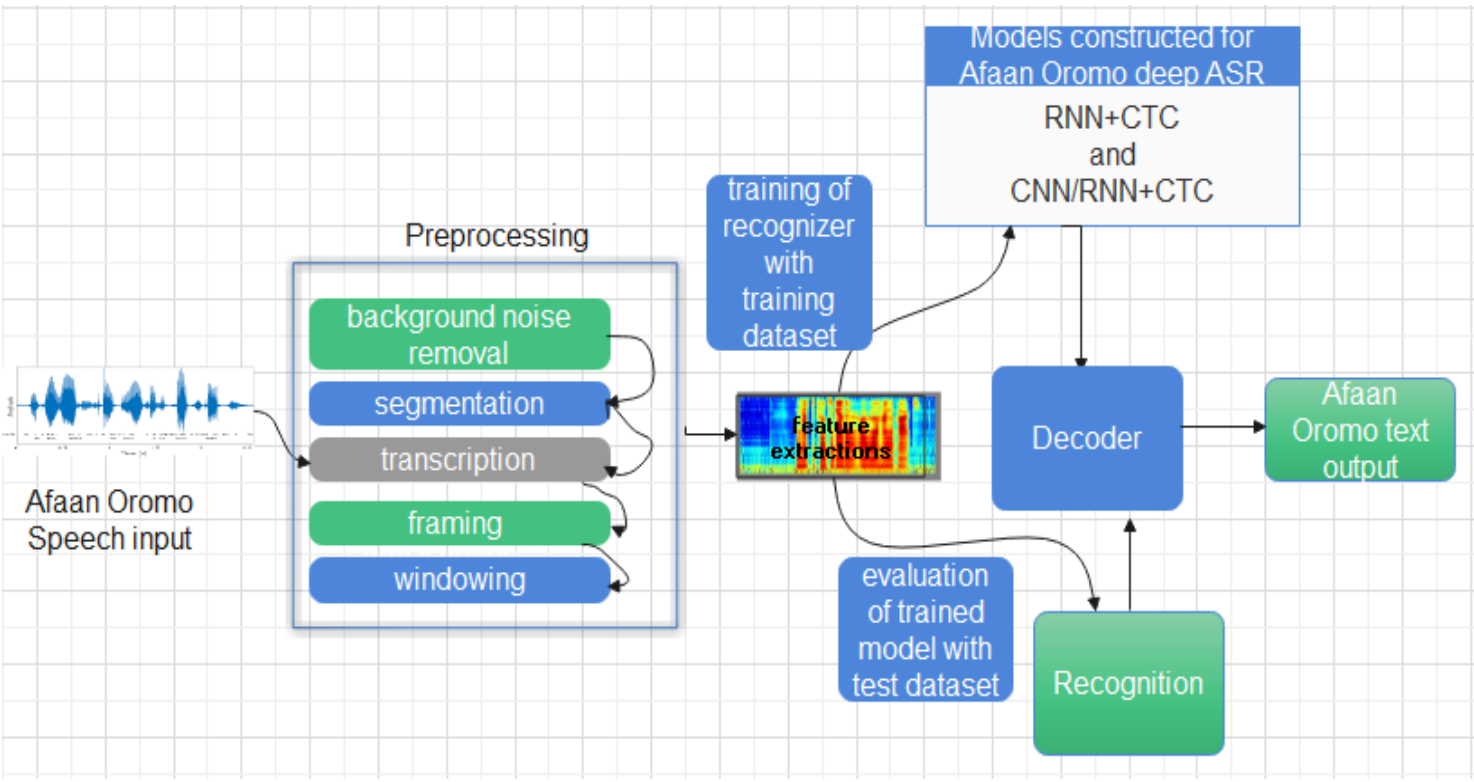


Figure 4.1 Architecture of how the proposed prototype works

3.1.1. Overview of the architecture

As shown from figure 4.1 collecting speech data is the first step and required input speech will be collected from Afaan Oromo broadcast news speech from OBN, OMN, VOA, FANA, EBC and BBC for preparing the required corpus. Therefore after collecting required corpus preprocessing was the next step. Preprocessing will be done on the collected speech data and as indicated on the architecture it has five steps.

i. Background noise removal: this is the first step of preprocessing speech data for the corpus. Using speech or non speech discriminator we separate speech with non speech.

ii. Segmentation: as known broadcast news speech is long and not suitable to directly use for transcription and also training purpose. So we segmented the long audio

data in to sentence level. That means every broadcast news are segmented and saved separately in single sentence form.

iii. Transcription: it is the process of labeling sound data with corresponding text. So because of there is no any available automatic tool to transcribe Afaan Oromo speech we manually transcribed it.

iv. Framing: here split the signal into short-time frames. The rationale behind this step is that frequencies in a signal change over time, so in most cases it doesn't make sense to do the Fourier transform across the entire signal in that there may be loss of the frequency contours of the signal over time. To avoid that, assume the frequencies in a signal are stationary over a very short period of time. Therefore, by doing a Fourier transform over this short-time frame, we can obtain a good approximation of the frequency contours of the signal by concatenating adjacent frames. Typical frame sizes in speech processing range from 20 ms to 40 ms [26].

v. Windowing: After segmenting the signal into frames, a window function such as the Hamming window will be applied to each frame. There are several reasons why we need to apply a window function to the frames, notably to counteract the assumption made by the FFT that the data is infinite and to reduce spectral leakage.

Then features were extracted from the preprocessed speech and after these the extracted features were used for training deep learning models that are chosen as a recognizer for this thesis work. The researchers also propose to use connectionist temporal classification (CTC) with combination of deep learning algorithms for solving alignment problems during training. Finally Afaan Oromo text will be the Output and using test dataset the system was tested for it's performance. Details of the rest components, which included in the architecture, are described below.

3.2. Feature extraction algorithms

Feature extraction means a process that identifies important features or attributes of the data. It is accomplished by changing the speech waveform to a form of parametric representation at a relatively minimized data rate for subsequent processing and analysis [38]. First, we have to convert sound data in format we can input in our model. One of the most common way is to split the sound in a sequence of frames and then extract features from them.

The use of Mel Frequency Cepstral Coefficients is considered as one of the standard method for feature extraction in speech recognition [63]. It is popularly used because it approximates the human system response more closely than any other system as the frequency bands are positioned logarithmically [64].

Mel Frequency Cepstral Coefficients (MFCC) is most widely used algorithm used to extract spectral features. MFCCs used in speech recognition are based on frequency domain using the Mel scale and they are one of the most accepted feature extraction techniques [65]. Therefore, we choose MFCC feature extraction techniques for this thesis work.

MFCC computation is a replication of the human hearing system which the Mel scale is based on the human ear scale intending to artificially implement the ear's working principle with the assumption that the human ear is a reliable speaker recognizer [65].

The MFCCs are calculated using the equation [9, 19]:

$$\hat{C}\, n = \sum kn = 1\big(log\hat{S}\, k\big)\boldsymbol{cos}[n(k-12)\pi k]\hat{C}n = \sum n = 1klog\hat{S}kcosnk - 12\pi k$$

Where k is the number of mel cepstrum coefficients, $\hat{S}k$ is the output of filterbank and $\hat{C}n$ is the final mfcc coefficients.

MFCCs which are well thought-out to be frequency domain features are to a great extent more precise than time domain features. Human Speech as a function of the frequencies is not linear in nature; therefore the pitch of an acoustic speech signal of single frequency is mapped into a "Mel" scale. In Mel scale, the frequencies spacing below 1 kHz is linear and the frequencies spacing above 1 kHz is logarithmic [1].

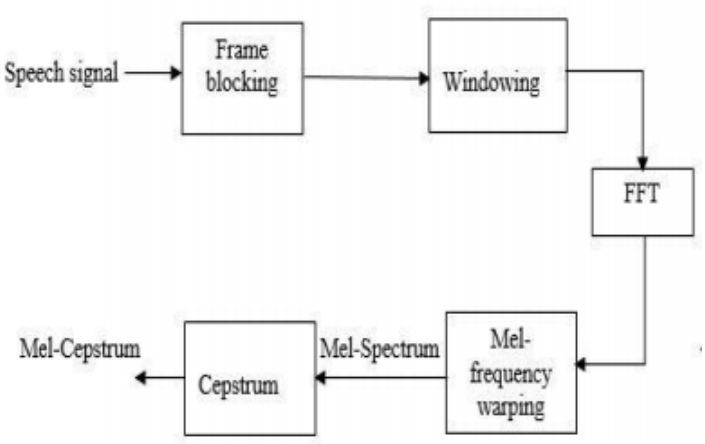


Figure 4.2 MFCC feature extraction process [66]

Frame blocking: or we can simply call it framing or segmentation: here split the signal into short-time frames. The rationale behind this step is that frequencies in a signal change over time, so in most cases it does not make sense to do the Fourier transform across the entire signal in that there may be loss of the frequency contours of the signal over time. To avoid that, assume the frequencies in a signal are stationary over a very short period of time. Therefore, we can obtain a good approximation of the frequency contours of the signal by concatenating adjacent frames by doing a Fourier transform over this short-time frame. Typical frame sizes in speech processing range from 20ms to 40ms [46].

An audio signal can be divided into a numbers of frames and some parts of each frame is overlap to each other. So each frame can be analyzed and synthesized individually without loss of information and denoted by a single vector. In this method a continuous audio signal is divided into N numbers of samples in a frame, where each neighboring frames are disjointed by M (M < N). Here the first frame contains N numbers of samples and the next frame starts M numbers of the samples so the first frame and second frame overlap by (N – M) numbers of samples. Correspondingly, the next frame contains 2M number of samples. Therefore, the first frame and the third frame are overlap by (N – 2M) numbers of samples. By using this technique speech signal is framed and no discontinuity occurs in audio samples. Frame blocking of the audio data is important for the angle of computational complexity and analyzing sufficiently over a short duration of time. For computation, the value of N is equal to 149 as a resolution between frequency

and time resolution. When the frequency and time coefficients are observed then the corresponding power spectrum of the audio data will be appeared in the outcome segment.

Hamming window: After segmenting the signal into frames, a window function such as the Hamming window will be applied to each frame. There are several reasons why we need to apply a window function to the frames, notably to counteract the assumption made by the FFT that the data is infinite and to reduce spectral leakage. Window is multiplied with each frames so the continuity will be achieved in between the first point of the first frame and last point of the last frame.

FFT: Fast Fourier transform is used for calculating of the discrete fourier transform (DFT) of signal. This step is performed to transform the signal into frequency domain. To calculate FFT

$$x[k] = \sum_{n=0}^{N-1} x[n] e^{-2\frac{\pi}{N}kn}$$ Where, N is the size of FFT.

Mel frequency warping: The next step is transformation from Hertz to Mel Scale, the spectrums power is transformed into a Mel scale. The audio sound-related framework is scaled by frequency called mel scale frequency.

Cepstrum: In this step DCT is applied to have melscale Cepstral coefficients.

Spectrograms: we can define a spectrogram as a bunch of FFTs stacked on top of each other [67]. Alternatively, a spectrogram is a visual representation of discretized frequencies (spectrum) at each time interval. The spectrogram we use for extracting speech features has 161 dimensions.

3.3. RECOGNITION ALGORITHMS

Speech recognition works using algorithms through recognizer modeling. The recognizer which is acoustic modeling represents the relationship between linguistic units of speech and audio signals.

3.3.1. Building the Recognizer

The recognizer model took MFCC and Spectrogram represented features as input. Then recognizer model is trained with CTC loss criterion.

On this thesis work experimentation on deep GRU RNN and CNN/RNN, combination of Deep convulusional neural networks (CNNs), Batch normalization and bidirectional recurrent neural

networks (RNNs) are done to see which one will perform best for our recognizer modeling purpose.

As many literatures reveals that RNNs are the most utilized and powerful deep learning architectures among all of the deep learning architectures in terms of error rates and speech recognition performance [68]. So GRU RNN with three hidden layers are used to build the first recognizer model.

In CNN/RNN hybrid model we use bidirectional reccurent neural networks with 1D convolutions. This Bidirectional Recurrent Neural Networks (BRNN) connect two hidden layers of opposite directions to the same output. In BRNN with this form of generative deep learning, the output layer can get information from past (backwards) and future (forward) states simultaneously. To train BRNN models we use two time directions, input information from the past and future of the current time frame can be used unlike standard RNN, for forward pass, forward states and backward states are passed first, then output neurons are passed. For backward pass also, output neurons are passed first, then forward states and backward states are passed next. Therefore, after forward and backward passes are done, the weights are updated.

The reason we want to experiment with this architecture is that using CNN allows us to extract common and useful patterns from features; this means that our RNN can then use this common features to extract information more efficiently.

Therefore we experiment with combination of Deep convulusional neural networks (CNNs), Batch normalization and bidirectional recurrent neural networks (RNNs).

It seems that the acoustic model can gets advantages of hybrid of both models like using CNN and RNNs. Therefore we try to combine CNN with Batch normalization.

That means, we try to combine a 1D convolutional layer as we try to describe under chapter three with Bidirectional GRU layers. We also used Batch Normalizations, which is used to reduce overfitting by adding some noise, but alone it was not enough to avoid the overfitting so Dropout is used right after Bidirectional GRU layers also.

Bidirectional layer seems little slower but produces a better result according to our experiments and at the end softmax is used to calculate probabilities.

CNNs find spatial relationship between the features, while RNNs find temporal relationships that will allow us to decide what letter to associate with a sound. The RNNs used as hybrid of CNN/RNN are also RNNs that are bidirectional.

Therefore, it is obviously known that deep models works better than a shallower model. CNNs find spatial relationship between the features, so having multiple CNNs allows the model to find more complex spatial features, while RNNs find temporal relationships that will allow us to decide what letter to associate with a sound. The RNNs used as hybrid of CNN/RNN are also RNNs with three hidden layers. The researchers also propose to use connectionist temporal classification (CTC) with combination of acoustic model for converting similar speeches which are spoken by different time durations or which are stretched and spoken rapidly to the same word.

Deeper and bigger models obviously improves the model accuracy but because of computational resources like GPU cost and time constraints we cannot make our model larger and deeper.

3.3.2. Decoders

Decoder is the most important component on the ASR systems and for each audio frame there is a process of pattern matching. Hereafter, the decoder evaluates the received feature against all other patterns. The best match can be achieved when more frames are processed [40].

Decoding is done using the trained recognizer model which is trained by speech data sets with corresponding transcription. However, it is not made directly on the components described. A decoding graph, which is a combination of the listed components, is first created and then the decoding is performed based on this graph. Both the creation of the graph and decoding are done on tensor flow and the result is printed in a human understandable form.

3.3.3. Speech recognition tools

Tensor flow:

Tensor flow is a software library or framework, designed by the Google team to implement machine learning and deep learning concepts in the easiest manner. It combines the computational algebra of optimization techniques for easy calculation of many mathematical expressions. It includes a variety of machine learning and deep learning algorithms, and can train and run deep neural networks for handwritten digit classification, image recognition, word embedding speech

recognition and creation of various sequence models. Because of tensor flow is easy to use that means it can be easily installed on personal computers and cloud and it is also open source meaning a freely available deep learning speech recognition tool we used it for this particular research work.

Keras:

Keras is compact, easy to learn, high-level Python library run on top of Tensor Flow framework. It is made with focus of understanding deep learning techniques, such as creating layers for neural networks maintaining the concepts of shapes and mathematical details. It is a deep learning framework or a library providing high-level building blocks for developing deep learning models.

Jupyter Notebook: Jupyter notebooks are great way to run deep-learning experiments. It allows you to break up a long experiment into smaller pieces that can be executed independently which makes the development interactive. All the experiments in this research were run in Jupyter.

NumPy: It is a multi-dimensional array (tensor) manipulation library. When doing deep learning every data must be represented in a tensor of different size and for storing and manipulating the arrays NumPy was used.

Librosa: It is a library for music and audio analysis and it provides the building blocks necessary to create audio information retrieval systems. we use this library to extract features from the audio.

PyDub :It is a library to manipulate audio data with a simple high-level interface.

Matplotlib: It is a python 2D plotting library. We use it to plot our constructed neural network model.

Seaborn: It is a data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphs.

Pydot :It is an interface to Graphviz and we use it to draw the graphical representation of the neural network architectures.

Programing language: The programming language that we use was python. Python is a high-level programming language based on object, translator, and dynamic words. Python programming is used for preprocessing of the data and building the models using the Jupiter notebook editor.

3.4. Evaluation techniques

In speech recognition, there are two different types of performance or evaluation measures, which are accuracy, and speed. Evaluation measures based on accuracy include WER, loss, and mean edit distance. WER is the most commonly used error measurement in ASR. The interpretation of loss and WER is that the lower the loss and WER, the better the speech recognition is [60].

Then the system was evaluated by analyzes and calculation of WER. Training loss, validation loss and WER are used for this work. The calculation of the error is done based on the following elements [32].

Loss: the loss that we calculated is the CTC loss. For calculating the loss, all the possible alignments' scores of the ground truth (that means the probability of input sentences)are summed up. After summing up the score of the individual path we get the probability of the ground truth occurring, then the loss is the negative logarithm of calculated probability [61]. Here is the formula [61]

$$Loss = \sum_{(y,x)\varepsilon D} (-\log p(Y \mid X))$$

Where x and y are variables and D is datasets

Training loss: Training loss is the error on the training set of data.

Validation loss: Validation loss is the error after running the validation set of data through the trained network. Train/valid is the ratio between the two.

WER: The word error rate is based on how much the word string returned by the recognizer differs from a correct or reference transcription.

Substitution: At the position of a unit (word or phoneme) which in our case is word, a different unit has been recognized.

Deletion: In the recognition result a unit is omitted.

Insertion: The result contains new units than were not spoken

The measures of WER is therefore given as follows:

$$WER = ((D + S + I)/ N) * 100\% = WER\%$$

Where D is deletion, S is substitution, I is insertion and N is the total number of words.

Based on this, the results (in terms of loss and WER) for the system are shown and discussed in chapter six.

3.5. Corpus preparation

Corpus preparation is the first step in speech recognition tasks and is designed according to best practice guidelines established for other languages. Standard speech corpora consist of a training set and evaluation test sets [21]. The training set is intended to collect speech data for training the recognizer and the evaluation test set is for the purpose of final evaluation of the recognizer. So the corpus preparation includes text corpus and speech or audio corpus. We used 80% of the dataset for training and 20% for testing based on [64] that used 10% and 20% for testing and remaining for training in different papers and [4] also uses 10% for testing and 90% for training, [65] and [66] uses Libri speech on github uses 80% training dataset and remaining for evaluation purpose.

3.5.1. Speech corpus preparation

Afaan Oromo does not has easily available electronic audio and text sources like other developed languages like Chinese and English because of Afaan Oromo is categorized as under resourced language.

The researcher did not record the audio datasets from individuals but instead collected the required speech corpus from audio and video broadcast news from different media which have Afaan Oromo program. And the video was converted to audio form before we proceed to next steps.

The speech corpus is primary input for the recognizer system. The speech corpus is prepared by collecting news audios from Ethiopian broadcasting corporation (EBC), Oromia broadcasting network (OBN), Oromia media network (OMN), voice of America (VOA), Fana broadcasting corporation (FBC), and BBC Afaan Oromo program.

The researcher also prepared therefore corresponding text transcription for the news data already collected by consulting with linguistic experts.

3.5.2. Preprocessing of speech data

The collected speech from different Medias have to be preprocessed because it cannot be used directly. This means broadcast news speeches are not clean and noise free as read speech. News speech is also not slowly spoken with some pauses between words like which is done in read speech.

So these properties of news speeches makes them more difficult to the recognizer than read speech corpuses. By considering previous other language broadcast news corpuses like Swahili and English language we also tried to prepare our corpus by using them as benchmark.

Therefore, from the news speech we get we only need continuous speech, which are not spontaneous like interviews, and additional reports like telephone reports, which are not clearly audible, are removed. Speech with music background and Afaan Oromo mixed with other languages like Amharic and English is also removed. Praat software was used for removing unnecessary sounds which mentioned above and long silences from collected news speech like in figure 5.1 and 5.2.

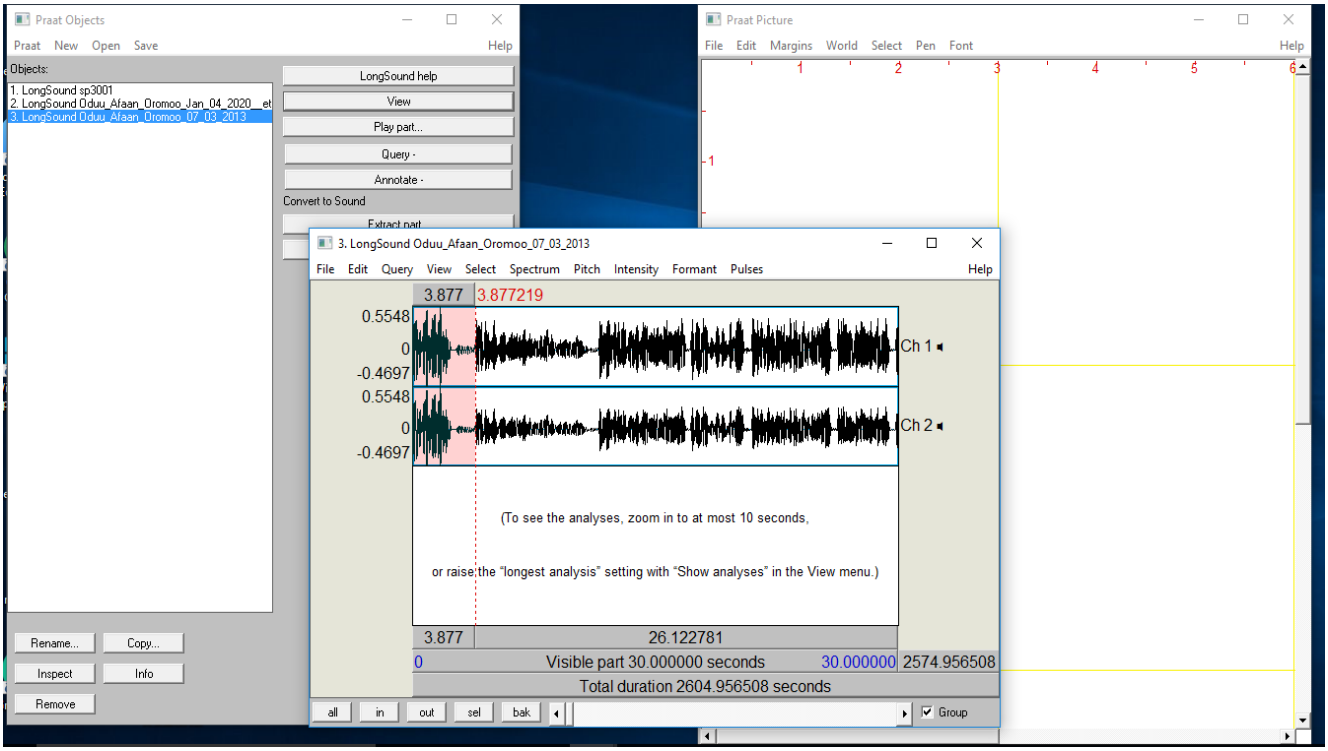


Figure 5.1 Removing non Afaan Oromo speech sounds from collected speech corpus

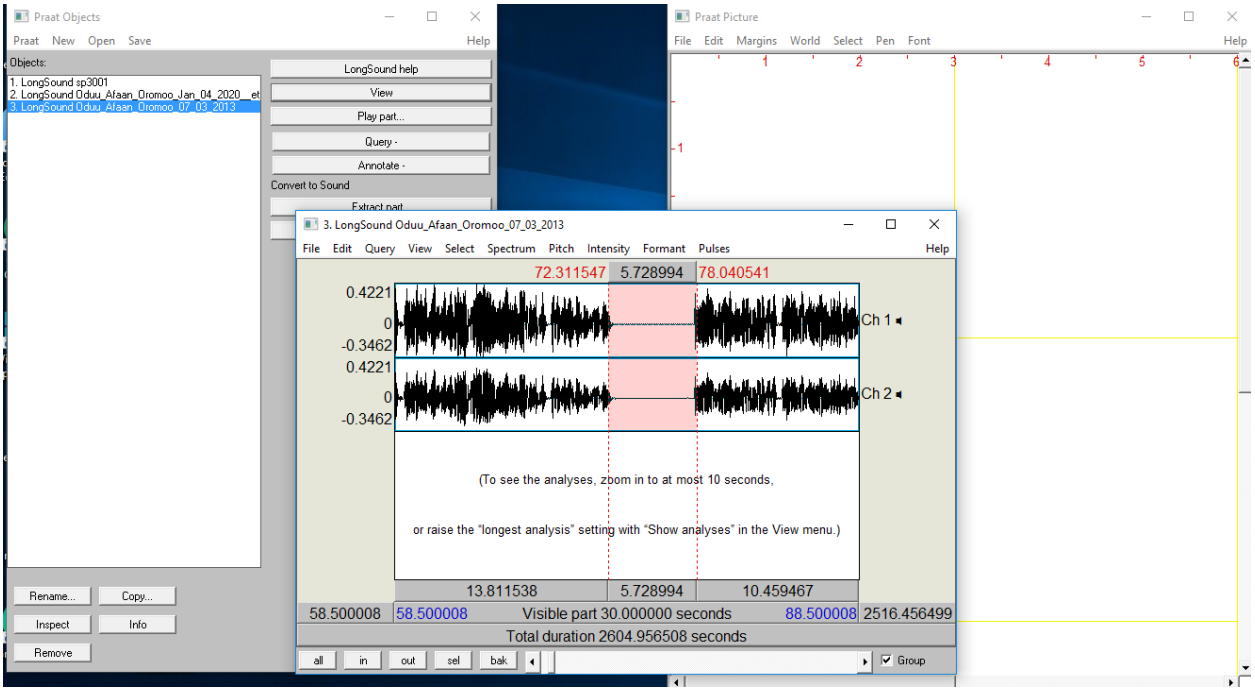


Figure 5.2 Removing long silence from collected speech corpus

Speech Segmentation: After removing and cleaning, the speech data have to be segmented to sentence level. Up on the knowledge of the researcher, there is no automatic method or tool to segment Afaan Oromo sentences automatically. So the segmentation is on sentence level or we used sentences to train our model because the recognition was also in sentence level not word not phoneme.

Here is praat segmentation of sentences.

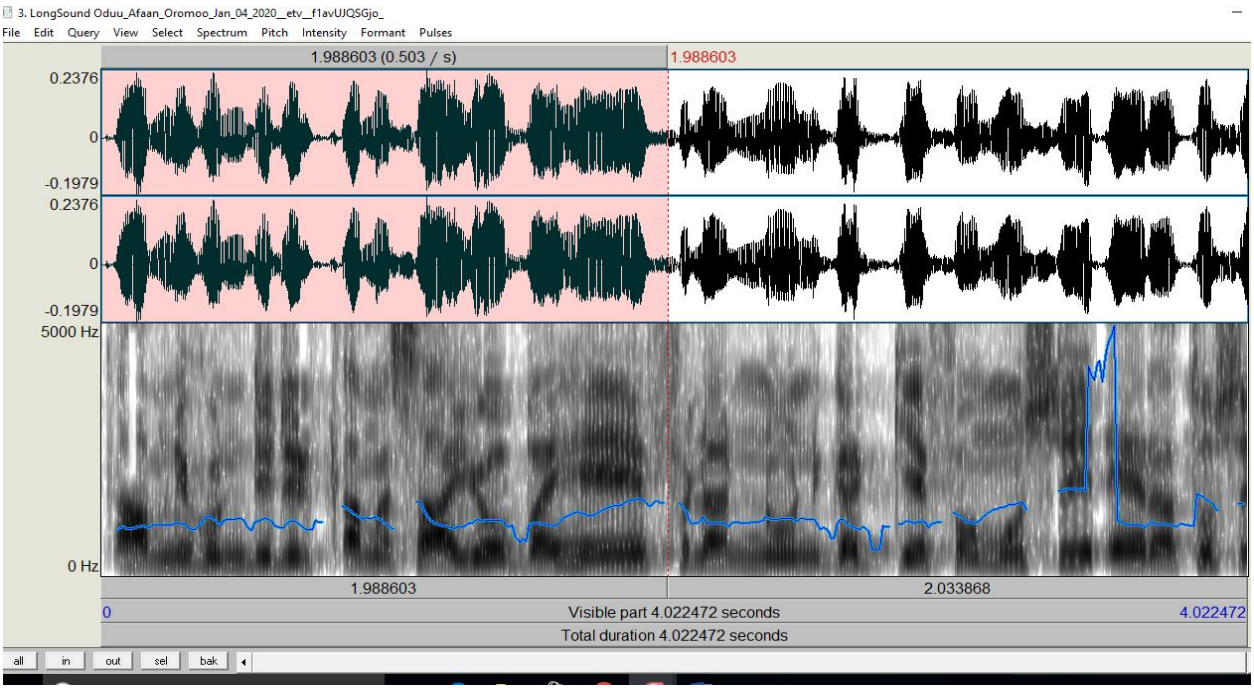


Figure 5.3 segmenting long speeches of broadcast news into short sentences

Therefore, sentence segmentation is done manually by listening audio files. Accordingly, 8,000 of total speech utterances that means about 10:01:38 hours speech from 101 different speakers were collected. The speakers were 80 males and 21 females. But to make easier our segmentation we use praat to visualize, listen and segment our datasets.

We consider to balance male and female speakers but because of most of the speakers from broadcast news including the journalists, peoples who participated in interviews and reporters were unfortunately males we could not. Therefore because of this we could not balance their number. Similarly many other literatures also did not use equal number of female and male speakers even if they do not put their reason. Among them to mention some Solomon [21] records 96 males and 84 females even if the corpus was read speech and Yadeta [4] collect his corpus which was broadcast news from 57 speakers (42 males and 15 females).

An open source tool known as praat was used to convert the segmented audio file in .WAV form because .WAV files are more suitable for machine learning purpose.

3.5.3. Transcription of segmented speech

The transcription of audio sentences into corresponding text was done manually on the Praat 6.1 software according to rule of Afaan Oromo grammar by consulting linguistics and different literatures. Therefore the researcher transcribed the audio to text manually and the transcriptions of each sentences with the audio was evaluated and corrected by linguistic experts. We paid for the experts to do so.

Praat is a free computer software package for speech analysis in phonetics. It was designed, and continues to be developed, by Paul Boersma and David Weenink of the University of Amsterdam [69]. It can run on a wide range of operating systems. The program supports speech synthesis, including articulatory synthesis.

Therefore 8,000 Afaan Oromo utterances where transcribed to corresponding text using Praat. But the verity of dialects, punctuations, and the rule of capitalization were not considered. In other words, all transcribed texts were in lower case and without punctuations (show diagram below).

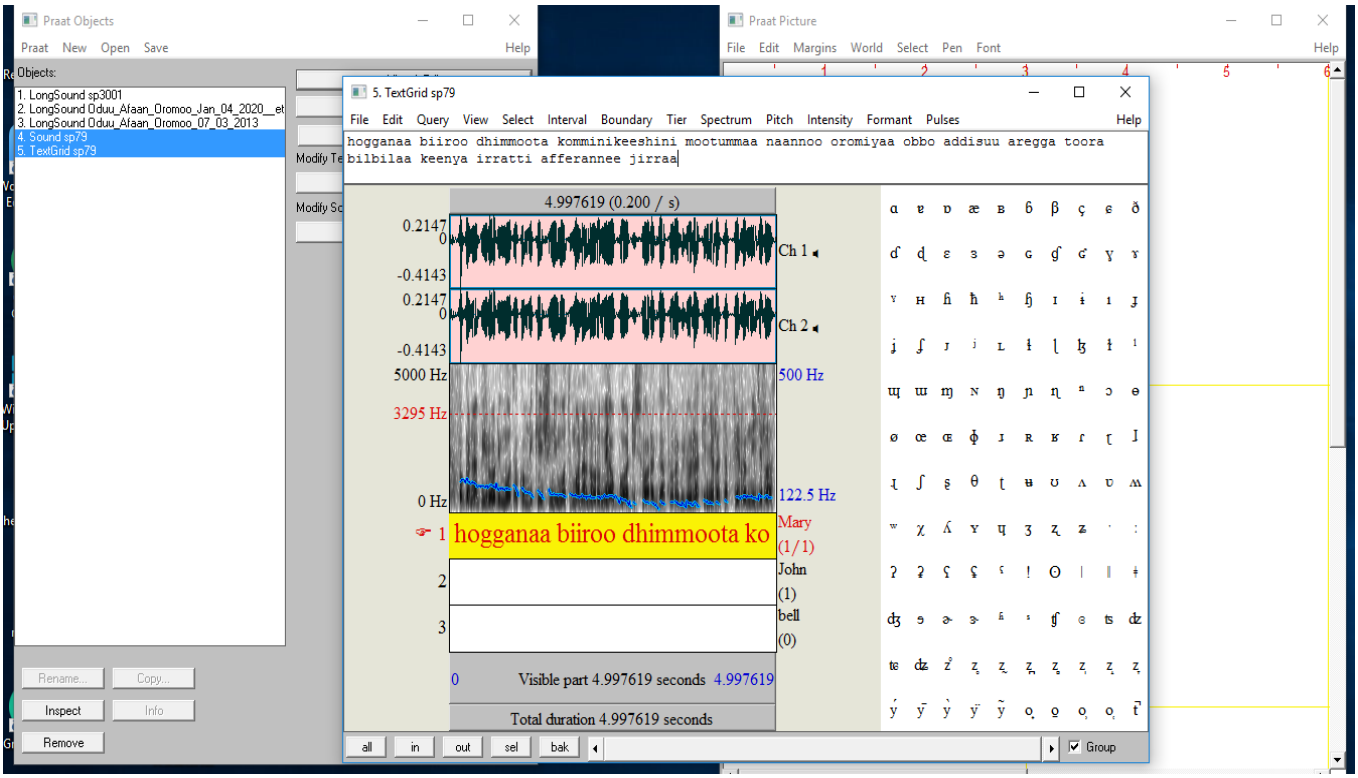


Figure 5.3 transcription of segmented speech in to corresponding Afaan Oromo sentence

3.6. Training and evaluation tools

There are various deep learning development tools for natural language processing that are open source and easy to adapt. The researcher chooses to use Anaconda because Anaconda is a free and open source distribution of Python programming language.

It is popular because it brings together many tools used in data science and deep learning with just one install, so it is great for having a short and easy setup.

Keras and tensor flow are used as tools to do this. Keras is a high-level neural network, written in Python and capable of running on tensor flow and enabling easy and fast prototyping supports CNN and RNN and It runs smoothly on CPU and GPU.

Tensor Flow is the combination of an open source platform for deep learning and it builds and trains deep learning models using a high-level keras API. Keras and tensor flow are used for construction, training and validation of model.

Chapter Four

Experiments and Result Discussions

Developing prototype for Deep learning based speaker independent large vocabulary continuous Afaan Oromo broadcast news speech recognition has the steps starting from data collection step, preprocessing the data, model training, recognition step and evaluation. Therefore, under this chapter details of experiments and their outcomes are discussed.

4.1. Training phase

The deep learning algorithm we used to train the recognizer is state of the art for continuous speech recognition [70]. The researchers tried to train two different models and based on their performance chooses the best one. These models or deep learning architectures are GRU RNN and CNN/RNN hybrid.

Tensor flow and keras deep learning tools was used for training the proposed model. Tensor Flow is flexible collection of tools and libraries that lets researchers push the state-of-the-art in deep learning. Tensor Flow is a Python library for fast numerical computing created and released by Google [71]. Therefore, we used tensor flow as backend for training our model.

4.1.1. Experimental setup

Intensive learning experiences require high processor and GPU supported computing. Then for this study, we first tried a computer with 8 GB of RAM, core i7 CPU with 2.41 GHz clock speed, and Ubuntu operating system as most literatures reveal they used personal computer for training.

However, because of audio data takes larger memory and computing resource than textual sources we could not continue with our laptop specially when we increase the size of our model so we use a freely available cloud computing (even if it is not enough still) by google which is called google colaboratory. Google colab or colaboratory has 12 GB RAM and single GPU and TPU for training. By using cloud computing and personal computer we trained our model.

4.1.2. Model construction and training

RNN GRU

After data preprocessing and extraction of important features from the preprocessed data the recognizer have to be constructed and trained. The first model the researchers select for training the recognizer is Gated recurrent unit (GRU).

As previously discussed under chapter three GRUs are type of RNNs that designed as a solution for the unstable gradient problem of RNNs as described under chapter three.

As many literatures reveals that RNNs are the most utilized and powerful deep learning architectures among all of the deep learning architectures in terms of error rates and speech recognition performance [68]. Therefore, we experimented Afaan Oromo speech recognition on GRU to see how it works for our case.

So the architecture of our GRU model is described using diagram as follows:

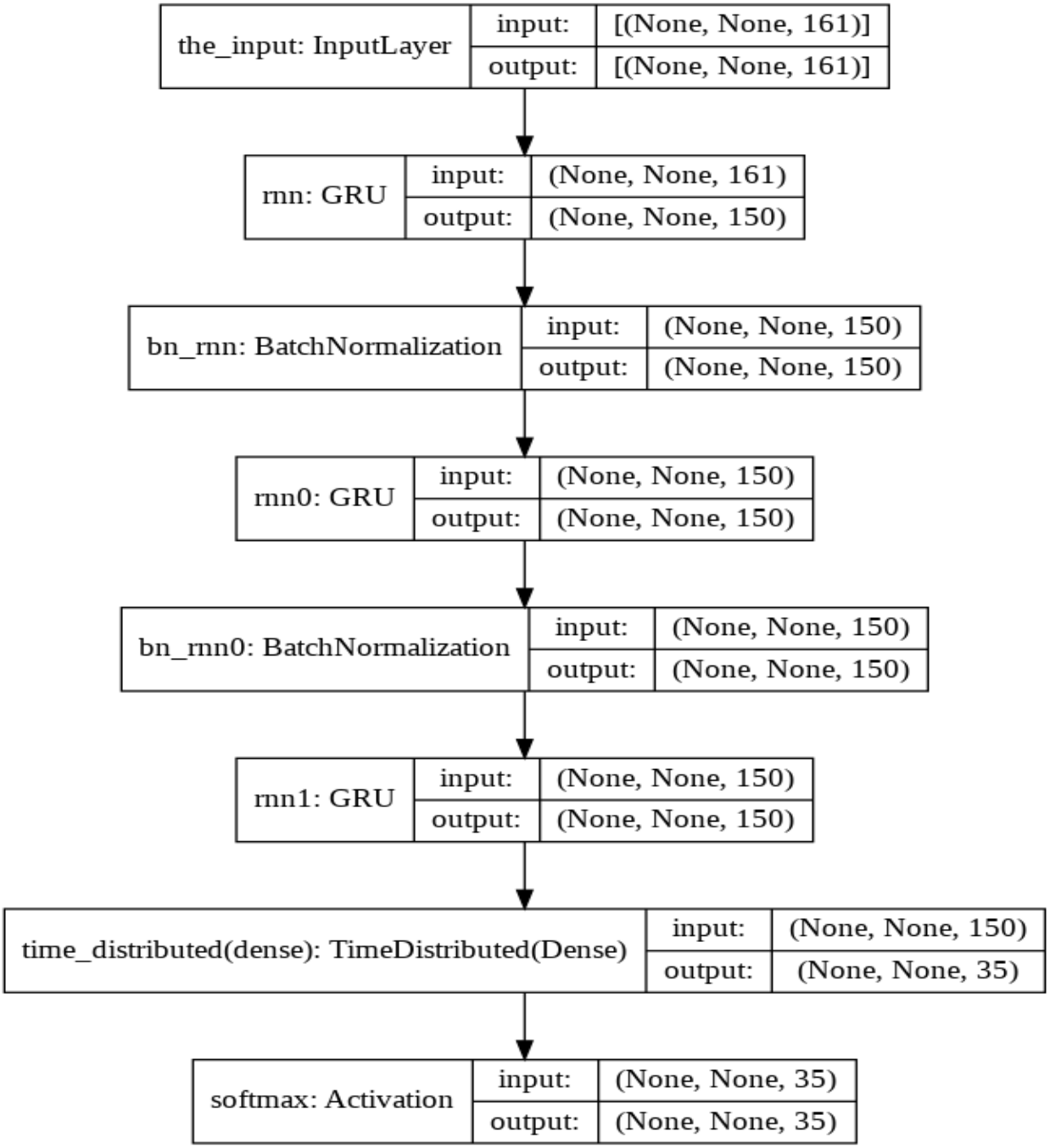


Figure 6.1 architecture of Deep RNN GRU model

As shown from the above diagram the model has simple GRU algorithm with three layers. Therefore we used RNN with three hidden layers and finally there is softmax for getting final output in words. We use the SoftMax activation function to generate the final output. Afaan Oromo has thirty-three alphabets and the softmax generates the most probable one including space and

blank character as final output. Therefore, it is used as the final layer in the speech recognition models.

We compiled the model and started the training. The loss function, the optimizer, and the metric that validate the model need to be specified. We used SGD as an optimizer. The learning rate determines the computational of the optimal weights for the model to be calculated. The metric is used for plotting the result after the history of the model is saved and we used WER as a metric for our model. The metrics contain training loss, validation loss and WER.

Below is summary of the model to visualize and generate each layer of the model with their corresponding output shape and number of parameters.

```
Model: "functional_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
the_input (InputLayer)       [(None, None, 161)]       0
_________________________________________________________________
rnn (GRU)                    (None, None, 150)         140850
_________________________________________________________________
bn_rnn (BatchNormalization)  (None, None, 150)         600
_________________________________________________________________
rnn0 (GRU)                   (None, None, 150)         135900
_________________________________________________________________
bn_rnn0 (BatchNormalization) (None, None, 150)         600
_________________________________________________________________
rnn1 (GRU)                   (None, None, 150)         135900
_________________________________________________________________
time_distributed_1 (TimeDist (None, None, 29)          4379
_________________________________________________________________
softmax (Activation)         (None, None, 29)          0
=================================================================
Total params: 418,229
Trainable params: 417,629
Non-trainable params: 600
_________________________________________________________________
None
```

Figure 6.2 summary of the RNN GRU model

CNN/RNN hybrid

The other deep learning architecture that we experimented with is hybrid of the two very common and powerful models CNN and RNN.

The reason we want to experiment with this architecture is that using CNN allows us to extract common and useful patterns from the MFCC or spectrogram features; this means that our RNN can then use this common features to extract information more efficiently. Therefore, the second recognizer was built using CNN/RNN.

Therefore, we combined a 1D convolutional layer with Bidirectional GRU layers. We also used Batch Normalizations, which is used to reduce overfitting by adding some noise, but alone it was not enough to avoid the overfitting so Dropout is used right after Bidirectional GRU layers also.

Bidirectional layer seems little slower but produces a better result according to our experiments and at the end softmax is used to calculate probabilities.

CNNs find spatial relationship between the features, while RNNs find temporal relationships that will allow us to decide what letter to associate with a sound. The RNNs used as hybrid of CNN/RNN are also RNNs that are bidirectional.

Connectionist temporal classification (CTC) with combination of the recognizers was also implemented for converting similar speeches which are spoken by different time durations or which are stretched and spoken rapidly to the same word.

So the architecture of our CNN/RNN model is described using diagram as follows:

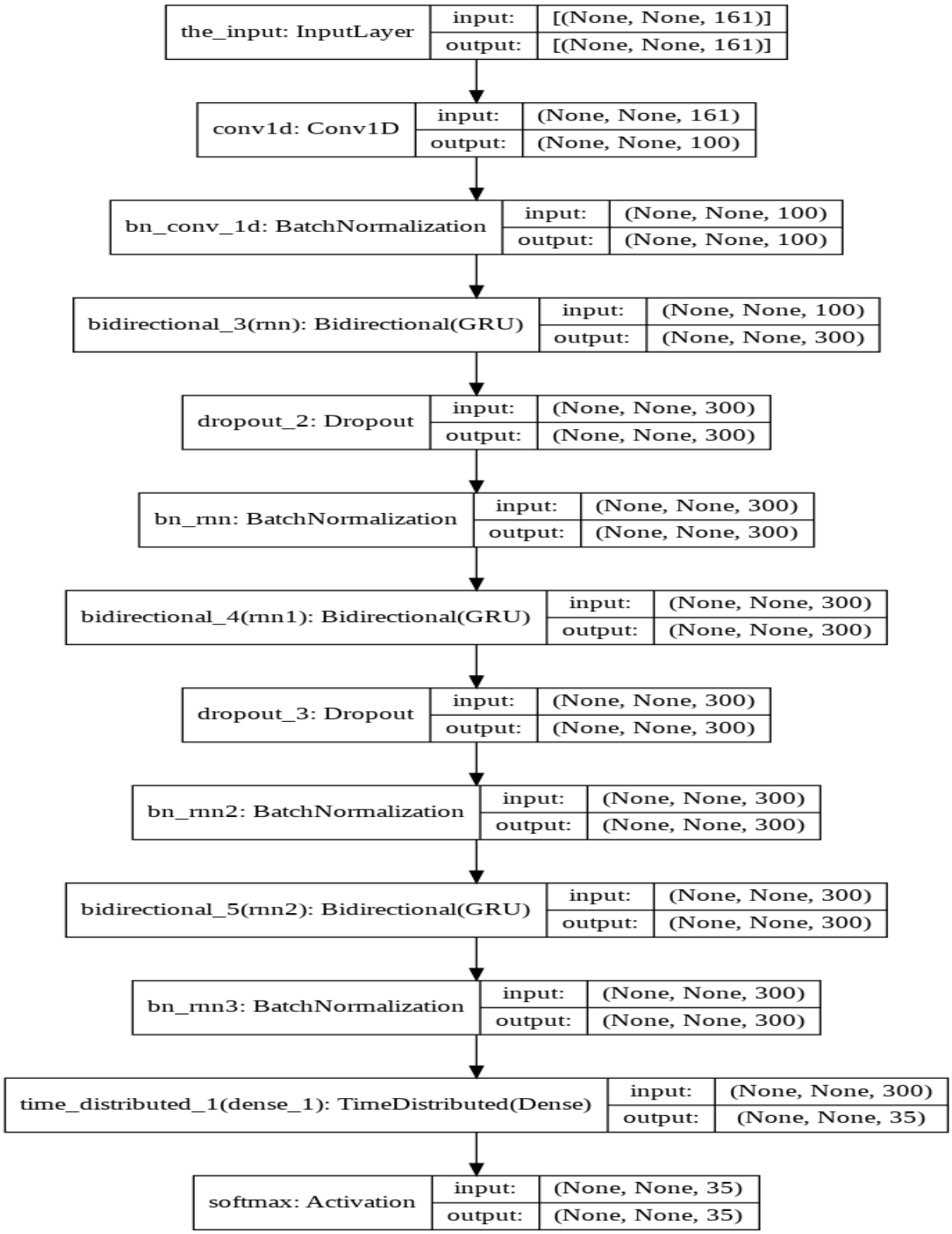


Figure 6.3 architecture of CNN/RNN model

As shown from the above diagram the model has 1D convolutional layer followed a Batch Normalization and three RNN hidden layers. Finally, there is softmax for getting final output in words. We use the SoftMax activation function to generate the final output. Afaan Oromo has thirty five outputs with 33 alphabets, space and black character and the softmax generates the most probable one as final output. Therefore, it is used as the final layer in the speech recognition models.

We compiled the model and started the training. The loss function, the optimizer, and the metric that validate the model need to be specified. We used 'SGD' as an optimizer. The learning rate determines the computational of the optimal weights for the model to be calculated. The metric is used for plotting the result after the history of the model is saved and we used WER as a metric for our model. The metrics contain training loss, validation loss and WER.

Below is summary of the model to visualize and generate each layer of the model with their corresponding output shape and number of parameters.

```
Model: "functional_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
the_input (InputLayer)       [(None, None, 161)]       0
_________________________________________________________________
conv1d (Conv1D)              (None, None, 100)         48400
_________________________________________________________________
bn_conv_1d (BatchNormalizati (None, None, 100)         400
_________________________________________________________________
bidirectional (Bidirectional (None, None, 300)         226800
_________________________________________________________________
dropout (Dropout)            (None, None, 300)         0
_________________________________________________________________
bn_rnn (BatchNormalization)  (None, None, 300)         1200
_________________________________________________________________
bidirectional_1 (Bidirection (None, None, 300)         406800
_________________________________________________________________
dropout_1 (Dropout)          (None, None, 300)         0
_________________________________________________________________
bn_rnn2 (BatchNormalization) (None, None, 300)         1200
_________________________________________________________________
bidirectional_2 (Bidirection (None, None, 300)         406800
_________________________________________________________________
bn_rnn3 (BatchNormalization) (None, None, 300)         1200
_________________________________________________________________
time_distributed (TimeDistri (None, None, 35)          10535
_________________________________________________________________
softmax (Activation)         (None, None, 35)          0
=================================================================
Total params: 1,103,335
Trainable params: 1,101,335
Non-trainable params: 2,000
_________________________________________________________________
None
```

Figure 6.4 summary of the model with 1D convulutions and three layers of RNN

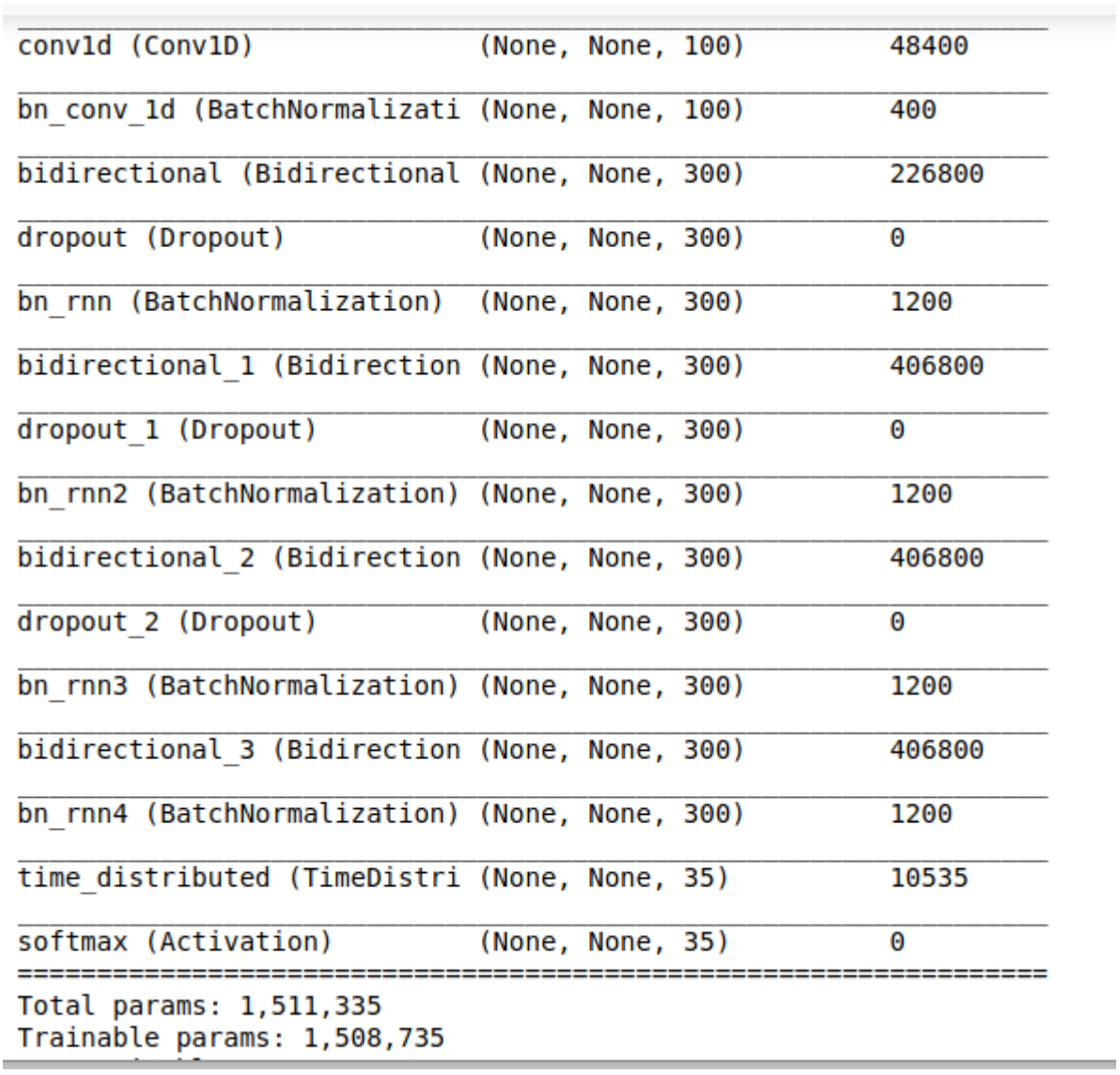


Figure 6.5 summary of the model with 1D convulutions and four layers of RNN (final model)

4.2. Parameters for training the model

To train the models we have to set training parameters as input to the model with our training datasets. As table 6.1 describes the number of epochs (the number of times the model goes over the entire dataset), and the batch size (how many training examples are processed at a time), learning rate, activation functions as well as the optimizers are specified.

No.	Parameters to train the recognizer	Values	Description of parameters
1	minibatch_size	20	No. of training examples that are processed at a time
2	Optimizer	SGD	algorithms or methods used to change the attributes of our network such as weights and learning rate in order to reduce the losses
3	Learning rate	0.005	a hyper-parameter that controls how much we are adjusting the weights of our network with respect the loss gradient
4	No. of epochs	20,30 and 50	No. of times the model goes over the entire dataset
5	Input features dimentions	13 and 161	MFCC and spectrogram features extracted from segmented data
6	Activation function	'relu'	help the network learn complex patterns in the data
7	dropout_rate	0.2	for preventing overfitting of the model 20% was used

8	No. of layers	3	number_layers of_GRU RNN model and CNN/RNN which we use for comparing purpose
9	Output_dimentions	35	which is output of final layer, it is no. of alphabets which are found in Afaan Oromo + space + blank character

Table 6.1 training parameters of recognizer models

After the model is trained, its performance should be evaluated with the validation dataset separately using the metric that we described under chapter four. Those metrics are, training loss, validation loss, and WER. Thus, the lower the WER and the lower the loss indicates the better the model performs. Seeing the loss and the WER values of the model we tried to adjust the parameters like batch size, number of RNN layers, and the value of dropout of the model to get a better model.

4.3. Results

Our RNN GRU and CNN/RNN hybrid speech recognition model is evaluated based on the nature of the GRU and CNN/RNN structure. We have training and validation dataset that pass though the preprocessing component. Thus, the model is trained using training datasets, the performance of the model is evaluated by testing datasets, and its WER is recorded besides the training and validation dataset. The evaluation held on the nature of the models structure is based on the amount of data, the number of epochs, and the number of hidden layers with constant batch size in which the model is built-in. Thus, we use 3000 total number of data for three hidden layer GRU and CNN/RNN with arbitrary 20 epochs. The evaluation processed is held on 2400 training and 600 testing dataset which are all sentences. Thus, the training WER and loss evaluates the training data in every learning rate in which the model learns the data once at a time (batch size =20), the model learns 20 training data at a time and it computes the training and testing loss. Then finally WER and loss evaluate the training performance of the model in each epoch.

Here is the results of our model:

	Type of model	No. of hidden layers	No. of epochs	Amount of data	Training loss	Testing loss	WER
1	RNN GRU	3	20	3000(80% training 20% testing)	27.8	217.6	1.019
2	CNN/RNN Hybrid	Single 1D convolutions of CNN with 3 layers of GRU RNN	20	3000(80% training 20% testing)	21.3	141.9	0.998
3	CNN/RNN Hybrid	Single 1D convolutions of CNN with 3 layers of GRU RNN	20	8000(80% training 20% testing)	21.2	22.0	0.957
4	Final model CNN/RNN	Single 1D convolutions of CNN with 3 layers of GRU RNN	30	8000(80% training 20% testing)	18.2	21.4	0.884
5	Final model CNN/RNN	Single 1D convolutions of CNN with 4 layers of GRU RNN	50	8000(80% training 20% testing)	13.4	19.2	0.693

Table 6.2 results of recognizer models

4.4. Discussions

As shown from above table deep RNN GRU model was not suitable for our case. That means the average loss which was 122.7 is much greater than our CNN/RNN hybrid model's average loss which was 81.6 with similar amount of hidden layers, similar epochs, similar amount of data and using common value of parameters. We also experimented with CNN/RNN hybrid model by increasing amount of data and number of hidden layers. So the performance of the model becomes increasing and increasing when we increase number of hidden layers that means complexity of model, and amount of data.

As shown from above table when amount of data increases from 3000 to 8000 but with similar value of the rest parameters both training loss and validation loss decreases to 21.2 and 22.0 respectively the WER was also much more improved. Therefore based on this result we decide to experiment with the hybrid model by increasing both amount of data and number of epochs. We increased number of epochs from 20 to 30 and with 8000 datasets and we get 18.1 training loss and 21.3 validation loss and again the performance of the model becomes increasing and increasing.

Finally, we want to see how our model performs when we increases amount of data, number of epochs and number of layers also. So we use all datasets that we prepare (8000) with 50 number of epochs and 4 number of hidden layers and we got 13.4 training loss and 19.2 validation loss with good WER of 69%. But the training process was slow because of we were not able to get high speed processors because of cost and totally it takes 4 days and half for finishing the training of final model.

Therefore as the experimental result from table 6.2 shows using larger models and huge amount of data is key thing in Deep Afaan Oromo speech recognition. However, because of resource and time constraints the researchers can experiment up to this for now.

The other thing that we see from our experiments was female speeches and speech dialects like Harari and Borena or which have smaller number of speakers from collected datasets were not well recognized by our model when we compare them with others. Lets see this example of female voice.

```
Output length: [374]
--------------------------------------------------------------------------------
True transcription:

olmaa da'imanitiin ala naanichatti adeemsi baruufi barsiisuu
--------------------------------------------------------------------------------
Predicted transcription:

olmadahimanitin ala nanichati ademsi baru barsisu
--------------------------------------------------------------------------------
```

When we see this particular sentence's WER it was 86% as calculated by our system. And lets compare this result with male speech and both the speakers' dialects was western dialects like wallaga dialect or dialect which is spoken and used widely (common).

```
output length: [259]
--------------------------------------------------------------------------------
True transcription:

seeratti dhiheessudhaf kan socha'an tahuus dubbatani
--------------------------------------------------------------------------------
Predicted transcription:

serati dhihesudhaf kan socha'an ta'us dubatani
--------------------------------------------------------------------------------
```

This particular sentences' evaluation result was 67% WER.

The same way we try to check five male and five female speakers speeches and in all cases female speeches recognition performance was less. In addition, our model less recognized speeches from Borena and Harari dialects even if the speakers were male. The reason for this was the amount of data that we use. This means the widely used dialects like wellega speakers were much more greater than Borena, kamisie and Harari speakers and number of female was also smaller than males as described in chapter Five.

To compare with previous Afaan Oromo ASR most of them used read speech as corpus for training their model so broadcast speech is more difficult than read speech as it was not recorded to become easy for machine and plus broadcast news speech is more noisy than read speech also. Broadcast news speech also did not consider noticeable pauses between words like read speech. However, 69% word error rate is still a good result as this work is the first of its type.

There are two previous works which were conducted on local languages using broadcast news. A work done by Yadeta [4] which was also continuous speech recognition for Afaan Oromo and a work done by Adugna [45] which was spontaneous speech recognition for Amharic. Both of the researchers used HMM for building the recognizer.

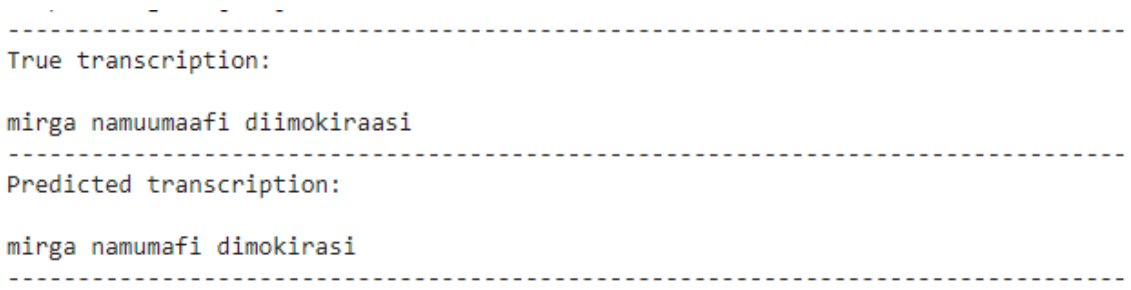
When we see our model with the first work [4], the researcher developed context independent (mono-phones based) and context dependent (tri-phones based) acoustic models and the best performance he obtained in terms of word error rate was 91.46% WER and 89.84% WER, for context-independent and context-dependent, respectively. And our models performance is 69% WER.

The performance result of Adugna [45] which was spontaneous speech recognition of Amharic, that the researcher got was 41.60% and 23.25% of word accuracy from speakers those are involved in training and speakers those do not involved in training, respectively.

Therefore these show us deep learning architectures definitely improves performance of speech recognizer for local languages specially Afaan Oromo. And further researches must also be based on deep learning architectures to get a more robust speech recognizer system for our local languages.

Finally, we were able to answer our research questions based on our findings as follows:

Accordingly, the researchers faces a lot of challenges during implementation of this system and collection of datasets of broadcast news from different media. As the main challenge using connectionist temporal classification(CTC) algorithm (for tackling sequence problem) was essential for our research, we also used it and it improved our result a lot. However because of CTC was developed and tested for foreign languages specially English previously it cannot understand Afaan Oromo's writing system during gemination of words and long and short vowels of Afaan Oromo. Because most of other languages like English does not use double consonants and double vowels for showing they are geminated and long respectively. Therefore, this algorithm considers these repetitions as same word, which are spoken slowly, and removed it. For instance these diagram shows prediction of our final model for different sentences.



```
--------------------------------------------------------------------------------
True transcription:

birrii kuma dhibba tokkoofi
--------------------------------------------------------------------------------
Predicted transcription:

bire kuma dhiba tokofi
--------------------------------------------------------------------------------
```

```
True transcription:

ibsa laatee jira
--------------------------------------------------------------------------------
Predicted transcription:

ibsa olate jira
--------------------------------------------------------------------------------
```

```
--------------------------------------------------------------------------------
True transcription:

seeratti dhiheessudhaf kan socha'an tahuus dubbatani
--------------------------------------------------------------------------------
Predicted transcription:

serati dhihesudhaf kan socha'an ta'us dubatani
--------------------------------------------------------------------------------
```

As seen from this examples of prediction our model almost for all datasets CTC is ignoring the repetitions even for training datasets themselves.

The first challenge was we could not get large amount of news data even from broadcast offices because of their lots of unnecessary and time-consuming bureaucracies. The other challenge was the nature of our dataset or preparing required corpus from broadcast news speech. Because, the broadcast news speech was difficult by their nature (i.e., they include other non-speech like music background and additional telephone reports with interviews and even mixture of other local and foreign language words).

The second thing was selecting feature extraction techniques. First we tried MFCC feature extraction techniques because of their wide usage, but their performance and amount of feature dimensions was small (13) so we also tried spectrogram as it’s feature dimension is 161.

Therefore, spectrograms gave us better result than MFCC for our case, so finally we choose spectrogram over MFCC.

The other a very challenging task was developing deep learning models which are suitable for relatively small amount of data and limited computing resource as most of already developed systems were for large amount of speech data and good computing resources. After reviewing a lot of literatures and conducting a lot of experiments we were able to construct one.

The other challenge was our personal computers were not enough for performing the training of recognizer, it becomes out of memory when we add number of layers and data to our deep learning models. Training of audio data on pc also takes a lot of times even multiple days and when there is no electricity our training was aborted several times. Because of using personal computer with no GPU was not enough for training different models with different architectures we used free cloud computing platform of google which is called google colab. Therefore because of unstable internet connection in our compound of Jimma University the training process was corrupted several times.

But with patience we try to finished the training. We have been prepared the speech corpus based on our scope and the resource we have for the study. By doing so we tried to overcome some challenges that happen in developing speech recognizer system for Afaan Oromo using deep learning.

The performance of the final model of recognizer achieved was 69% WER and 16.3 average loss which is very good as the recognition is at sentence level not phoneme as most of previous Afaan Oromo ASRs.

Chapter Five

Conclusions and Recommendations

In this chapter conclusions and recommendation are discussed. As the thesis work is only a good starting point for further researches for the other researchers we also put future works that we plan to conduct in future.

5.1. Conclusions

Investigating a Large Vocabulary, Speaker-independent, Afaan Oromo Continuous Speech Recognition Using Deep Learning (LVCSR) System from Broadcast News speech corpus was the overall work of this study. In order to do this thesis work, we had consulted different literatures about features of the language, development of ASR from broadcast news in general and the related materials were also reviewed. For this study, state of the art deep learning algorithms of RNN and hybrid CNN/RNN were used [70].

The corpus required for this study was collected from different sources. Accordingly, audio data from OMN, OBN, FANA, EBC, VOA, and BBC was collected and transcription and segmentation was done using Praat software. Because of the time constraints and scarcity of previous speech corpus for the language we only use 8000 total data sets with corresponding 8000 sentences.

For converting the audio data we collect to .wav form praat software was used. And we used jupyter notebook with python programing for feature extraction from the raw signal. Training of the system, recognizing test utterances, and for analyzing the result of the recognizer we used tensor flow and keras.

The speech recognizer system developed from 101 speakers (80 males and 21 females) using 8000 sentences which have 10:01:38 hours long.

We use 80% of the dataset for training and 20% for testing the speech recognizer system. Speakers who are involved in training does not involved in testing.

From several experiments done in this study, the best performance achieved was 69% WER and 16.3 loss after increasing number of hidden layers of CNN/RNN hybrid model with 50 epochs.

The strength of this thesis work is despite several challenges that we have faced in this study using deep learning for speech recognition of under resourced language Afaan Oromo we were able to prepare required corpus from broadcast news speech which was around 10 hours. Because, the broadcast news speech was difficult by their nature (i.e., they include other non-speech like music background and additional telephone reports with interviews).

This thesis work also shows the possibility of developing automatic speech recognition using deep learning algorithms for local languages.

As a weakness, our personal computers were not enough for performing the training on multiple and deeper recognizer models, it becomes out of memory when we add number of layers and data to our deep learning models. Training of audio data on pc also takes a lot of times and when there is no electricity our training was aborted several times. So problem of resource and time constraint can also be considered as weakness of this work.

From obtained results, we have a WER of 69% WER, and average loss of 16.3. Even if, we have gone to the most optimal level still our system needs improvements but we get improved results only when we can add huge amount of data and use high performing GPUs. However, very good improvements have been seen from our starting to final model. The researcher conducted several experiments in order to improve the performance of the recognizer; like using different architectures of deep learning algorithms, increasing the amount of data, tuning parameters like learning rate, number of epochs and number of hidden layers.

Therefore from our experiments show us deep learning architectures definitely improves performance of speech recognizer for local languages specially Afaan Oromo, and as the experimental result from table 6.2 shows using larger models and huge amount of data is key thing in Deep Afaan Oromo speech recognition.

5.2. Recommendations and Future works

The following are recommendations and future works that we will work on in future and for other interested researchers.

Our major recommendations and also future work is finding a way that we may modify CTC algorithm (because it's performance is amazing for sequence problems) for Afaan Oromo's writing system or finding other suitable way or algorithm.

The other is this study is the first of its kind to explore the possibility of developing continuous speech recognition system for Afaan Oromo using deep learning architectures. Nevertheless, this approach is state of the art algorithm for continuous speech recognition and other complex tasks the result obtained, (even if it is best among our local language ASRs' that we discuss in our related works) was not a good as other developed or well-resourced languages. Therefore, we recommend conducting a research on the area in order to increase the performance of ASR system using state of the art deep learning approaches using different speech corpus for Afaan Oromo.

In this research CNN/RNN model with four hidden layers was used, but making the model more deeper and larger can improve the result. So using deeper architectures on this data set is one recommendation. Because of budget constraints and lack of available payment way for cloud computing to use GPU we can't build deep and large enough model for the study.

Capitalization and punctuation rule is the other recommendation. We did not include these rules so interested researchers can do on it.

Collecting speech corpus and preprocessing it was very time consuming and also very costly. But, as our own experimental results indicates and many literatures reveal that using large corpus like 100 hour 3000 and in hundred thousands definitely improves the performance of the recognizer. So standardized and large corpus preparation is one of our future work and recommendations.

In this study, we have not tried to identify the variety of dialects. Because we do not have enough budget and time to collect the required speech with different dialects. Therefore, we set it as future works.

References

[1] D. R. K. Bhuvaneshwari Jolad, "DIFFERENT FEATURE EXTRACTION TECHNIQUES FOR AUTOMATIC SPEECH RECOGNITION: A REVIEW," *INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY,* February 2018.

[2] L. D. Xuedong Huang, "An Overview of Modern Speech recognition," in *Indurkhya/Handbook of Natural Language Processing,*, 2009, pp. 339-363.

[3] D. Mwiti, "KDnuggets News," [Online]. Available: https://www.kdnuggets.com/2019/09/2019-guide-automatic-speech-recognition.html. [Accessed 9 January 2020].

[4] Y. G. Gutu, "A large vocabulary, speaker-independent, continuous speech recognition system for afaan oromo: using broadcast news speech corpus," Addis Ababa University, AA, 2016.

[5] P. S. R. D. Suman K.Saksamudre, "A Review on different approaches for speech recognition system," *International Journal of Computer Applications,* vol. 115, no. 22, pp. 23-28, 2015.

[6] L. R. R. B.H. Juang, "Automatic Speech Recognition – A Brief History of the Technology Development," Atlanta, 2004.

[7] ".wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/speech_recognition#applications. [Accessed 9 January 2020].

[8] X. W. Dong Wang, "An overview of end-to-end automatic speech recognition," *Symmetry,* vol. 11, no. 8, p. 1018, 7 Augest 2019.

[9] "Quora.com," [Online]. Available: https://www.quora.com/why-are-hidden-markov-models-replaced-by-rnn-nowadays-in-many-applications-what-is-the-strength-of-rnn-over-hmm. [Accessed 10 January 2020].

[10] T. S. S. R. H.-K. J. K. George Saon, "The IBM 2016 English conversational telephone speech recognition system," *IBM T. J. Watson research center, Yorktown heights, NY,* no. 10598, 2016.

[11] S. Teferra, "automatic speech recognition for amharic," Germany , 2005.

[12] T. Degeneh, "The development of Oromo writing system," 2015.

[13] T. G. V. 3. |. N. 5, "HUMAN LANGUAGE TECHNOLOGIES AND AFFAN OROMO," *International Journal of Advanced Research in Engineering and Applied Sciences,* May 2014.

[14] A. Demissie, "speech recognition system for Afaan Oromo," Addis Ababa, Ethiopia, 2009.

[15] K. Gelana, "A Continuous, Speaker Independent Speech Recognizer for Afaan Oromoo," Addis Ababa, Ethiopia, 2010.

[16] D. D. Geleto, "Large vocabulary continuous speech recognition system for afaan oromo using hidden markov model (HMM)," Adama,Ethiopia, September 2016.

[17] T. Kebebew, "Speech Recognition for Afan Oromo using hybrid hidden markov models and artificial neural networks," Addis Ababa, Ethiopia, 2010.

[18] P. T. Chandralika Chakraborty, "Issues and Limitations of HMM in Speech Processing: A survey," *International Journal of Computer Applications ,* vol. 141 , no. 7, pp. 13-17, May 2016.

[19] L. Deng, "Three Classes of Deep Learning Architectures and Their Applications: A Tutorial Survey".

[20] T. Y. Alemu. [Online]. Available: http://ainsightful.com/index.php/2018/11/27/deep-learning-for-amharic-speech-recognition/. [Accessed 3 March 2020].

[21] W. M. B. T. Solomon Teferra Abate, "An Amharic Speech Corpus for Large Vocabulary Continuous Speech Recognition," in *INTERSPEECH 2005 - Eurospeech, 9th European Conference on Speech*, Lisbon, Portugal, 2005.

[22] H. Petkar, "A Review of Challenges in Automatic Speech Recognition," *International Journal of Computer Applications,* vol. 151, no. 3, pp. 23-26, October 2016.

[23] Z. S. Wubishet, "hidden markov model based large vocabulary, speaker independent, continuous amharic speech recognition," AAU, AA , June 2003.

[24] H. Bakanay. [Online]. Available: https://www.sestek.com/2020/08/the-advantages-of-speech-recognition-technology/. [Accessed 13 November 2020].

[25] J. M. N. a. C. R. Frankish, "Speech Recognition Technology for Individuals with Disabilities," *Augmentative and Alternative Communication,* 12 July 2009.

[26] G. R. M. Steven M. Ross, EXPERIMENTAL RESEARCH METHODS, Second ed., Researchgate, Ed., AECT, 2014.

[27] D. J. a. K. Warren. [Online]. Available: https://gradcoach.com/what-is-research-methodology/. [Accessed 13 November 2020].

[28] P. P. ,. C. S. Rajasekar, "Research methodology," *arXiv:physics/0601009 [physics.gen-ph],* 14 October 2013.

[29] D. J. Harland, "cemast.illinoisstate.edu," [Online]. Available: https://cemast.illinoisstate.edu/downloads/hsrs/types_of_research.pdf. [Accessed 24 April 2020].

[30] J. Le, "The 3 Deep Learning Frameworks For End-to-End Speech Recognition That Power Your Devices," Heartbeat.fritz.ai, 2019.

[31] "Stack Overflow.com," 1 may 2016. [Online]. Available: https://stackoverflow.com/questions/36963054/what-is-train-loss-valid-loss-and-train-val-mean-in-nns. [Accessed 27 march 2021].

[32] G. Gebregergs, "DNN-HMM Based Isolated-Word Tigrigna Speech Recognition System," AAU , Addis Ababa, Ethiopia, October , 2018.

[33] D. A. A. B. W. A. P. Y. Santosh K.Gaikwad, "A Review on Speech Recognition Technique," *International Journal of Computer Applications (0975 – 8887),* vol. 10, no. 3, November 2010.

[34] A. Hafte, "Hidden Markov Model Based Tigrigna Speech Recognition," Addis Ababa, Ethiopia, 2009.

[35] R. D. Shaikh Naziya S., "Speech Recognition System – A Review," *OSR Journal of Computer Engineering (IOSR-JCE),* vol. 18, no. 4, pp. 01-09, July 2016.

[36] M. D. G. Shreya Narang, "Speech Feature Extraction Techniques: A Review," *International Journal of Computer Science and Mobile Computing,* vol. 4, no. 3, pp. 107-114, 2015.

[37] N. M. Bill Swartz, "Feature Extraction for Automatic Speech Recognition (ASR)," *IEEE,* 1997.

[38] N. K. A. R. Sabur Ajibola Alim, Natural to Artificial Intelligence - Algorithms and Applications, Intechopen, 2018.

[39] N. K. A. R. Sabur Ajibola Alim, Some Commonly Used Speech Feature Extraction Algorithms, intechopen, 2018.

[40] A. A. R. O. Dr Hebah H. O. Nasereddin, "Classification Techniques for Automatic Speech Recognition (ASR) Algorithms used with Real Time Speech Translation," *Computing Conference,* 2017.

[41] L. R. R. M. K. Brown, "Dynamic time warping for isolated word recognition based on ordered graph searching techniques".

[42] D. K, "Top 4 advantages and disadvantages of Support Vector Machine or SVM," Medium daily, 2019.

[43] E. Lisowski, 21 July 2020. [Online]. Available: https://addepto.com/deep-learning-architecture/. [Accessed 25 November 2020].

[44] B.h. Juang & Lawrence r. Rabiner, "Automatic Speech Recognition, a brief history of the technology development," Atlanta, 2004.

[45] A. Deksiso, "spontaneous speech recognition for amharic using hmm," Addis Ababa, Ethiopia, 2015.

[46] S. Mahapatra, "Towardsdatascience.com," 21 March 2018. [Online]. Available: https://towardsdatascience.com/why-deep-learning-is-needed-over-traditional-machine-learning-1b6a99177063. [Accessed 21 November 20].

[47] "Oromo (Afaan Oromo, Oromiffa, Oromoo)," University of Cambridge Language Centre.

[48] L. Hardesty, "explained: Neural networks," MIT News Office, USA, 2017.

[49] B. R. Haohan Wang, "On the Origin of Deep Learning," RearchGate, February 2017.

[50] K. D. Foote, 7 February 2017. [Online]. Available: https://www.dataversity.net/brief-history-deep-learning/#. [Accessed 20 November 2020].

[51] M. T. Jones, "Deep learning architectures," *The rise of artificial intelligence,* 8 SEptember 2017.

[52] K. Some, "Analytics Insight.net," 31 October 2018. [Online]. Available: https://www.analyticsinsight.net/the-history-evolution-and-growth-of-deep-learning/. [Accessed 25 October 2020].

[53] S. Ahn, "Deep Learning Architectures and Applications," *Journal of Intelligence and Information Systems ,* vol. 22, no. 2, pp. 127-142 , 2016.

[54] S. Verma, "Towards Data Science," 20 September 2019. [Online]. Available: https://towardsdatascience.com/understanding-1d-and-3d-convolution-neural-network-keras-9d8f76e29610. [Accessed 30 january 2021].

[55] "Wikipedia, the free encyclopedia," [Online]. Available: https://en.wikipedia.org/wiki/Recurrent_neural_network. [Accessed 25 N0vember 2020].

[56] "Wikipedia, the free encyclopedia," [Online]. Available: https://en.wikipedia.org/wiki/Long_short-term_memory. [Accessed 26 November 2020].

[57] S. Rathor, 2 June 2018. [Online]. Available: https://medium.com/@saurabh.rathor092/simple-rnn-vs-gru-vs-lstm-difference-lies-in-more-flexible-control-5f33e07b1e57. [Accessed 26 November 2020].

[58] "From Wikipedia, the free encyclopedia," [Online]. Available: https://en.wikipedia.org/wiki/Gated_recurrent_unit. [Accessed 26 November 2020].

[59] [Online]. Available: https://en.wikipedia.org/wiki/Deep_belief_network. [Accessed 26 November 2020].

[60] D. N. S. A. L. Apeksha Shewalkar, "PERFORMANCEEVALUATIONOFDEEPNEURAL NETWORKSAPPLIEDTOSPEECHRECOGNITION:RNN, LSTMANDGRU," *JAISCR,* vol. 9, no. 4, pp. 235-245, 2019.

[61] A. Hannun, "Distill," 27 November 2017. [Online]. Available: https://distill.pub/2017/ctc/. [Accessed 19 december 2020].

[62] A. Fortes, "Hands-On Speech Recognition Engine with Keras and Python," 20 February 2019. [Online]. Available: https://medium.com/@fortes.arthur/hands-on-speech-recognition-engine-with-keras-and-python-c60488ac53cd. [Accessed 19 ddecembeer 2020].

[63] D. V. T. Urmila Shrawankar, "TECHNIQUES FOR FEATURE EXTRACTION IN SPEECH RECOGNITION SYSTEM : A COMPARATIVE STUDY," 2013.

[64] R. Vimala.C, "Suitable Feature Extraction and Speech Recognition Technique for Isolated Tamil Spoken Words," *International Journal of Computer Science and Information Technologies,* vol. 5 (1), pp. 378-383, 2014.

[65] A. T. ,. P. S. U. Koustav Chakraborty, "Voice Recognition Using MFCC Algorithm," *International Journal of Innovative Research in Advanced Engineering (IJIRAE) ,* vol. 1, no. 10, p. 158_161, November 2014.

[66] A. T. Rajeev Ranjan, "Analysis of Feature Extraction Techniques for Speech Recognition System," *International Journal of Innovative Technology and Exploring Engineering (IJITEE),* vol. 8, no. 7C2, pp. 2278-3075, may 2019.

[67] L. Roberts, "Analytics VIdhya," 6 march 2020. [Online]. Available: https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53. [Accessed 30 January 2021].

[68] Z. T. Yeşim Dokuz, "A Review on Deep Learning Architectures for Speech Recognition," *European Journal of Science and Technology,* pp. 169-176, April 2020.

[69] "Wikipedia, the free encyclopedia," November 2017. [Online]. Available: https://en.wikipedia.org/wiki/Praat. [Accessed 15 april 2021].

[70] R. Schl¨uter, "Automatic Speech Recognition: State-of-the-Art in Transition," RWTH Aachen University, California, 2017.

[71] [Online]. Available: https://data-flair.training/blogs/tensorflow-applications/. [Accessed 22 1 2021].

[72] T. Girma, "human language technologies and Afaan Oromo," *international journal of advanced research in engineering and applied sciences,* vol. 3, no. 5 , may 2014.

[73] T. S. S. R. a. H.-K. J. K. George Saon, "The IBM 2016 English conversational telephone speech recognition system," *IBM T. J. Watson research center,* 2016.

[74] S. Teferra, "automatic speech recognition for amharic," , Germany, 2005.

[75] H. Fayek. [Online]. Available: https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html. [Accessed 2 may 2020].

[76] D. F. G. T. G. L. A. R. Anirudh Raju, "Scalable Multi Corpora Neural Language Models for ASR," *Interspeech 2019,* pp. 3910--3914, 2019.

[77] D. N. S. A. L. Apeksha Shewalkar, "Performance Evaluation of Deep Neural Networks Applied to Speech Recognition: RNN, LSTM and GRU," *Journal of Artificial Intelligence and Soft Computing Research,* vol. 9, no. 4, pp. 235-245, 30 Augest 2019.

[78] A. Hannun, "Sequence Modeling," Stanford University, 2017.

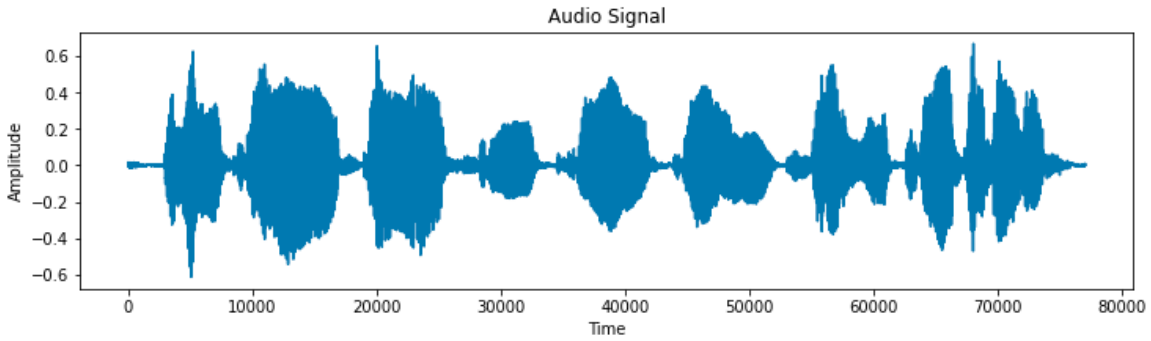
[79] "Oromo (Afaan Oromo, Oromiffa, Oromoo)".

[80] R. K. Bhuvaneshwari Jolad, "DIFFERENT FEATURE EXTRACTION TECHNIQUES FOR AUTOMATICSPEECH RECOGNITION: A REVIEW," *INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCHTECHNOLOGY,* vol. 3, pp. 181-188, February 2018.

Appendices

Appendix I: Sample of shape of raw audio signal its transcription and length in time

```
# plot audio signal
plot_raw_audio(vis_raw_audio)
# print length of audio signal
display(Markdown('**Shape of Audio Signal** : ' + str(vis_raw_audio.shape)))
# print transcript corresponding to audio clip
display(Markdown('**Transcript** : ' + str(vis_text)))
# play the audio file
Audio(vis_audio_path)
```

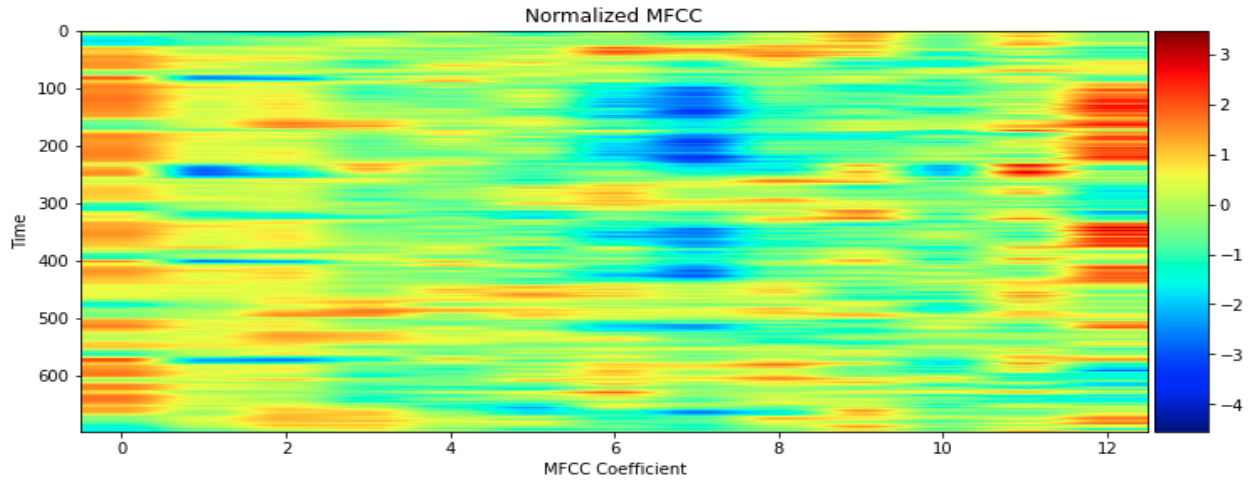


Shape of Audio Signal : (76998,)

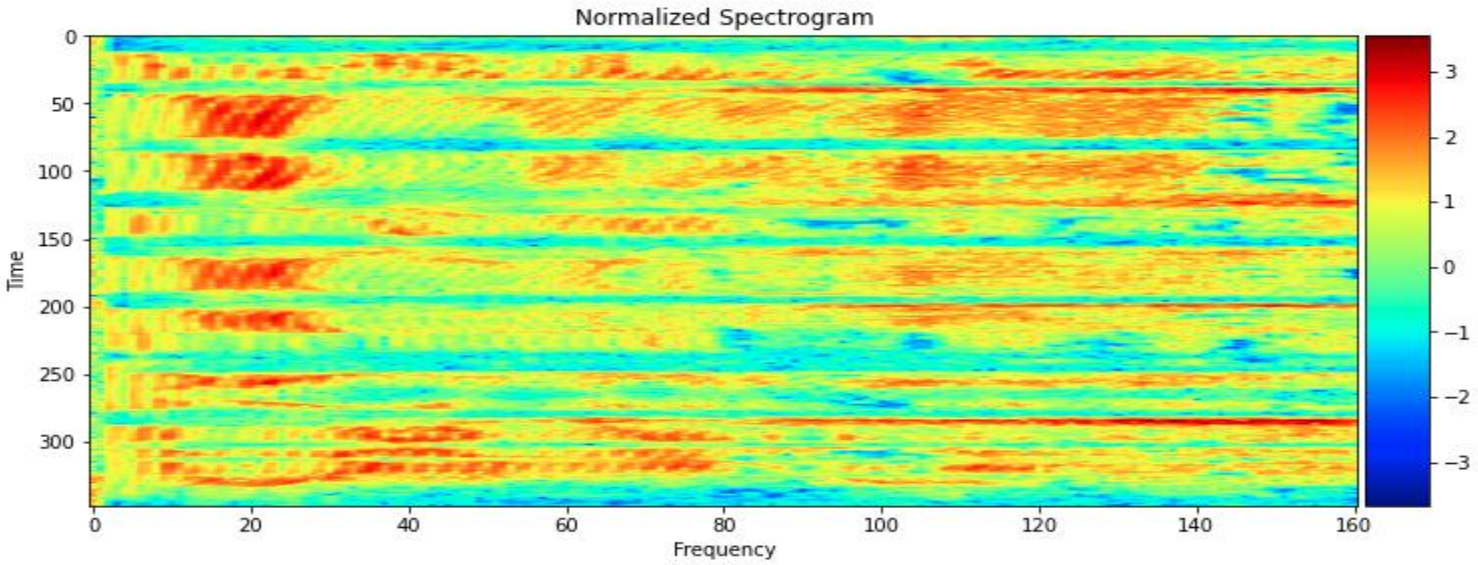
Transcript : biyyoota baha afriika ta'an daawwachuu eegaleera

0:00 / 0:03

Appendix II: Sample of MFCC and Spectrogram feature extractions from raw audio signals



Shape of MFCC : (697, 13)



Shape of Spectrogram : (348, 161)

Appendix III: Sample of collected datasets in Json format

{"key": "AfaanOromoSpeech/Traindata/101/28/sp4094.wav", "duration": 2.914761904761905, "text": "fooyya'iinsa seeroota diinagdeetis akka keessatti argaman ibsan"}
{"key": "AfaanOromoSpeech/Traindata/101/28/sp4095.wav", "duration": 3.578526077097506, "text": "haasaa ball'aa miseensoota mana maree bakka bu'oota ummataa biyyattitiif taasisaniin"}
{"key": "AfaanOromoSpeech/Traindata/101/28/sp4096.wav", "duration": 4.213424036281179, "text": "ijjiiramni biyyatti akka lafa hin qabannee gufachiisuu qofaa osoo hin taane"}
{"key": "AfaanOromoSpeech/Traindata/101/28/sp4097.wav", "duration": 2.250997732426304, "text": "jijjirama durayyuu biyyatti keessatti"}
{"key": "AfaanOromoSpeech/Traindata/101/28/sp4098.wav", "duration": 3.8959637188208616, "text": "jijjiiramni akka hin dhufne shira garaa garaa xaxaa akka ture"}
{"key": "AfaanOromoSpeech/Traindata/101/28/sp4099.wav", "duration": 2.8570521541950114, "text": "dhimmoota kanaan wal qabatan irrattis haasaa taasisaniiru"}
{"key": "AfaanOromoSpeech/Traindata/101/28/sp4100.wav", "duration": 3.607369614512472, "text": "wal ga'ii bara hojii ja'affaa walga'ii addaa lammaffaa"}
{"key": "AfaanOromoSpeech/Traindata/101/28/sp4101.wav", "duration": 2.1932879818594104, "text": "mana maree bakka bu'oota ummataa adeemsifame irratti"}
{"key": "AfaanOromoSpeech/Traindata/101/28/sp4102.wav", "duration": 2.828185941043084, "text": "gaaffilee garaagaraa miseensoota mana maree irraa ka'aniif"}
{"key": "AfaanOromoSpeech/Traindata/101/28/sp4103.wav", "duration": 1.9912698412698413, "text": "akka lafa hin qabanneef qofa osoo hin taane"}
{"key": "AfaanOromoSpeech/Traindata/101/28/sp4104.wav", "duration": 1.558390022675737, "text": "alqabumaa jala bultii irraa"}
{"key": "AfaanOromoSpeech/Traindata/101/28/sp4105.wav", "duration": 1.9624036281179138, "text": "jijjiiramichi biyyattiitti akka hin dhufneef"}
{"key": "AfaanOromoSpeech/Traindata/101/28/sp4106.wav", "duration": 2.3953061224489796, "text": "hirawwan gurguddoo raawwachaa turu himaniiru"}
{"key": "AfaanOromoSpeech/Traindata/101/28/sp4107.wav", "duration": 2.1932879818594104, "text": "jalqabumma irraa ajjeechaa jimlaa raawwachuun"}
{"key": "AfaanOromoSpeech/Traindata/101/28/sp4108.wav", "duration": 1.8758503401360545, "text": "ummanni itoophiyaa dhippo keessa galuun"}
{"key": "AfaanOromoSpeech/Traindata/101/28/sp4109.wav", "duration": 1.7315419501133786, "text": "jijjiirama jedhame akka hin hawwineef"}
{"key": "AfaanOromoSpeech/Traindata/101/28/sp4110.wav", "duration": 1.240952380952381, "text": "hojii hojjetaa tureen"}
{"key": "AfaanOromoSpeech/Traindata/101/28/sp4111.wav", "duration": 1.0100680272108844, "text": "jijjiirama booda"}
{"key": "AfaanOromoSpeech/Traindata/101/28/sp4112.wav", "duration": 3.0879138321995465, "text": "boollawwan namootni hedduun jimlaan itti awwaalaman"}
{"key": "AfaanOromoSpeech/Traindata/101/28/sp4113.wav", "duration": 2.048979591836735, "text": "argamaa turus

Appendix IV: Sample of training process of our model

```
Please use Model.fit, which supports generators.
Epoch 1/50
1800/1800 [==============================] - 10789s 6s/step - loss: 63.8691 - val_loss: 39.7284
Epoch 2/50
1800/1800 [==============================] - 11735s 7s/step - loss: 33.4896 - val_loss: 26.0371
Epoch 3/50
1800/1800 [==============================] - 11889s 7s/step - loss: 26.2847 - val_loss: 23.6654
Epoch 4/50
1800/1800 [==============================] - 11961s 7s/step - loss: 22.6405 - val_loss: 22.0587
Epoch 5/50
1800/1800 [==============================] - 12039s 7s/step - loss: 20.2787 - val_loss: 21.4577
Epoch 6/50
1800/1800 [==============================] - 11900s 7s/step - loss: 18.8217 - val_loss: 21.0832
Epoch 7/50
1800/1800 [==============================] - 11969s 7s/step - loss: 17.6754 - val_loss: 21.3848
Epoch 8/50
1800/1800 [==============================] - 11958s 7s/step - loss: 16.7864 - val_loss: 20.1684
Epoch 9/50
1800/1800 [==============================] - 12038s 7s/step - loss: 16.0808 - val_loss: 20.8356
Epoch 10/50
1800/1800 [==============================] - 11992s 7s/step - loss: 15.5032 - val_loss: 20.6590
Epoch 11/50
1800/1800 [==============================] - 11979s 7s/step - loss: 15.0333 - val_loss: 20.0802
Epoch 12/50
1800/1800 [==============================] - 12082s 7s/step - loss: 14.7786 - val_loss: 21.2216
Epoch 13/50
1800/1800 [==============================] - 11931s 7s/step - loss: 14.4132 - val_loss: 20.1866
Epoch 14/50
1800/1800 [==============================] - 11945s 7s/step - loss: 14.0345 - val_loss: 20.7583
Epoch 15/50
1800/1800 [==============================] - 12034s 7s/step - loss: 13.7432 - val_loss: 22.1490
Epoch 16/50
1800/1800 [==============================] - 12104s 7s/step - loss: 13.5372 - val_loss: 19.3224

1800/1800 [==============================] - 12104s 7s/step - loss: 13.5372 - val_loss: 19.3224
Epoch 17/50
1800/1800 [==============================] - 12017s 7s/step - loss: 13.1888 - val_loss: 20.0393
Epoch 18/50
1800/1800 [==============================] - 12006s 7s/step - loss: 13.2278 - val_loss: 20.3747
Epoch 19/50
1800/1800 [==============================] - 11985s 7s/step - loss: 13.0140 - val_loss: 20.9833
Epoch 20/50
1800/1800 [==============================] - 12064s 7s/step - loss: 12.9478 - val_loss: 20.0543
Epoch 21/50
1800/1800 [==============================] - 11996s 7s/step - loss: 13.0759 - val_loss: 20.7910
Epoch 22/50
1800/1800 [==============================] - 12007s 7s/step - loss: 12.8958 - val_loss: 21.2844
Epoch 23/50
1800/1800 [==============================] - 11941s 7s/step - loss: 12.6664 - val_loss: 21.3530
Epoch 24/50
1800/1800 [==============================] - 12267s 7s/step - loss: 12.5219 - val_loss: 20.9103
Epoch 25/50
1800/1800 [==============================] - 12157s 7s/step - loss: 12.4357 - val_loss: 21.8371
Epoch 26/50
1800/1800 [==============================] - 12147s 7s/step - loss: 12.9763 - val_loss: 20.2601
Epoch 27/50
1800/1800 [==============================] - 12140s 7s/step - loss: 12.9590 - val_loss: 21.7514
Epoch 28/50
1800/1800 [==============================] - 12211s 7s/step - loss: 13.1011 - val_loss: 23.3496
Epoch 29/50
1800/1800 [==============================] - 11887s 7s/step - loss: 13.1310 - val_loss: 19.2331
Epoch 30/50
1800/1800 [==============================] - 11727s 7s/step - loss: 13.1349 - val_loss: 19.6759
```

Appendix V: Sample of predictions of trained final model on test datasets

```
True transcription:

mootummaan naannoo oromiyaa dhiyeessatti fayyadamaa kan jiran ta'uu kan ibsan obbo geetuun
--------------------------------------------------------------------------------
Predicted transcription:

motuman nano oromiya dhihesati fayadhama kan jiran ta'u kan ibsan obo gethun
--------------------------------------------------------------------------------
```

```
|: import numpy as np

test_predictedText = get_predictions(index=101,
```

```
[ ] ____________________________________________________
None
('Failed to import pydot. You must `pip install pydot` and install graphviz (https://graphviz.gitlab.io/download/), ', 'for `pydotprint` to work.')
output length: [449]
WARNING:tensorflow:From /home/ly/anaconda3/lib/python3.8/site-packages/tensorflow/python/util/dispatch.py:201: sparse_to_dense (from tensorflow.python.ops.sparse_
Instructions for updating:
Create a `tf.sparse.SparseTensor` and use `tf.sparse.to_dense` instead.
--------------------------------------------------------------------------------
True transcription:

yeroo humna koree kanaati oli ta'e ammo sadarkaa biyyolessaatti koree ministeerota hunda'e
--------------------------------------------------------------------------------
Predicted transcription:

byero humna kode kanati oli ta'e amot tadrka biyoalesati kore ministerota hunda'e
--------------------------------------------------------------------------------
```

```
____________________________________________________
None
('Failed to import pydot. You must `pip install pydot` and install graphviz (https://graphviz.gitlab.io/downloa
d/), ', 'for `pydotprint` to work.')
output length: [149]
WARNING:tensorflow:From /home/ly/anaconda3/lib/python3.8/site-packages/tensorflow/python/util/dispatch.py:201: spa
rse_to_dense (from tensorflow.python.ops.sparse_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Create a `tf.sparse.SparseTensor` and use `tf.sparse.to_dense` instead.
--------------------------------------------------------------------------------
True transcription:

ibsa laatee jira
--------------------------------------------------------------------------------
Predicted transcription:

ibsa olate jira
--------------------------------------------------------------------------------
```

```
--------------------------------------------------------------------------------
True transcription:

seeratti dhiheessudhaf kan socha'an tahuus dubbatani
--------------------------------------------------------------------------------
Predicted transcription:

serati dhihesudhaf kan socha'an ta'us dubatani
--------------------------------------------------------------------------------
```

```
--------------------------------------------------------------------------------
True transcription:

kuma shantama kan hojjetame
--------------------------------------------------------------------------------
Predicted transcription:

kuma shantama kan hojetame
--------------------------------------------------------------------------------
```