Amharic Phrase Level Sign Language Recognition Using Deep Learning

Teferi, Girma Messele
BAHIR DAR UNIVERSITY

BAHIR DAR INSTITUTE OF TECHNOLOGY

SCHOOL OF GRADUATE STUDIES

FACULTY OF COMPUTING

Msc Thesis on:

Amharic Phrase Level Sign Language Recognition Using Deep Learning

By:

Teferi Girma Messele

July, 2021

Bahir Dar, Ethiopia
Amharic Phrase Level Sign Language Recognition Using Deep Learning

BY:

Teferi Girma Messele

A Thesis Submitted to the School of Graduate Studies of Bahir Dar Institute of Technology, BDU in Partial Fulfillment for the Degree of Master of Science in Information Technology in the Faculty of Computing

Advisor: Belete Biazen (Assi.prof.)

July, 2021

Bahir Dar, Ethiopia
DECLARATION

This is to certify that the thesis entitled “Amharic Phrase level sign language recognition using Deep learning approach”, Submitted in partial fulfillment of the requirements for the degree of Master of Science in Information Technology under Faculty of Computing, Bahir Dar Institute of Technology, is a record of original work carried out by me and has never been submitted to this or any other institution to get any other degree or certificates. The assistance and help I received during this investigation have been duly acknowledged.

Teferi Girma Messele

Name of the Candidate

Signature

Date

08/11/2013
BAHIR DAR UNIVERSITY
BAHIR DAR INSTITUTE OF TECHNOLOGY
SCHOOL OF GRADUATE STUDIES
FACULTY OF COMPUTING

Approval of thesis for defense result

I hereby confirm that the changes required by the examiners have been carried out and incorporated in the final thesis.

Name of Student: Teferi Girma Signature Date 07/11/2013

As members of the board of examiners, we examined this thesis entitled “Amharic Phrase Level sign language recognition using Deep learning” by Teferi Girma. We hereby certify that the thesis is accepted for fulfilling the requirements for the award of the degree of Masters of Science in “Information Technology”.

Board of Examiners

Name of Advisor Signature Date
Belete Biazen (Asst. Prof.) 08/11/2013

Name of External Examiner Signature Date

Adane Leta (PhD) 03/11/2013

Name of Internal Examiner Signature Date
Tesfa Tegegne (PhD)

Name of Chairperson Signature Date
Mr. Dagnachew Melesew

Name of Chair Holder Signature Date
Derejaw Lake

Name of Faculty Dean Signature Date
Mr. Belete Biazen 08/11/2013
ACKNOWLEDGEMENT

First of all, I would like to thank God to allow me to finalize the study. I would also like to extend my thanks to my thesis advisor Belete B.(Assistance prof.) for his supervision, advice, and guidance. Without his guide, continual follow-up, encouragement, and patience, this research wouldn’t have been completed. I greatly appreciate his support for everything to accomplish this thesis work.

Next, I would like to thanks Abraham D.(Assi.prof.) from Bahir Dar university computing faculty for his support, and starting from problem formulation up to the end of the thesis work he gives me direction and valuable help.

I also thank my wife. She has always loved me and supported me in everything. She was also sharing different ideas on how to ongoing my thesis works. Additionally, I want to thank all my friends who support me and help me by idea-sharing. I would also thanks IT, computer science, and software engineering department MSc students for their sharing of ideas for me.

Finally, I would like to thank Bahir dar university peda campus special need department students and Bahir dar yekatit 23 elementary school teachers and students, who volunteer in the data collection process. Especially, I would like to special thanks to teacher Fikrte, She is a special need teacher in yikatit 23 elementary school. I was asked everything related to special need information and she was also a volunteer.
ABSTRACT

Sign language is communication between hearing-impaired with hearing-impaired and hearing-impaired with normal people. It is also a way to communicate using hand and face gestures of phrases, words, or letters and it is a nonverbal communication language. However, there is a gap in communication between hearing-impaired with hearing peoples. This is due to the reason that most hearing people in Ethiopia do not know and understand the sign. So, a study aims to design a model that converts Ethiopian Amharic phrase signs to text to eliminate the communication gaps between the hearings disabled with hearing people because there is adifficulty of communication with Amharic phrase level that could not solve the previous works. We used deep learning techniques to solve a problem and achieve the objectives. Hybrid networks (CNN with LSTM) are applied in our study. Because a hybrid network is used to increase the performance of the model. For feature extraction, we used a convolutional neural network (CNN) algorithm and for classification, we used Long-short-term-memory (LSTM). LSTM has an ability to classify sequences of information and remember it for a long period of time. From the total dataset of 10500, 80% (8400) used for training, and 10% (1050) used for validation, and 10%(1050) also used for testing the model. The testing accuracy of the model is 96%. We used accuracy, precision, recall, f1-score, and confusion matrix to evaluate our model.

Keywords: Sign language recognition, Convolutional neural network, Long Short Term Memory, Deep learning
<table>
<thead>
<tr>
<th>CONTENTS</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>DECLARATION</td>
<td>I</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENT</td>
<td>IV</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>V</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>IX</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>X</td>
</tr>
<tr>
<td>ABBREVIATIONS</td>
<td>XI</td>
</tr>
<tr>
<td>CHAPTER ONE: INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Background</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Motivation</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Statement of the problem</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Objectives of the study</td>
<td>4</td>
</tr>
<tr>
<td>1.3.1 General objective</td>
<td>4</td>
</tr>
<tr>
<td>1.3.2 Specific Objectives</td>
<td>4</td>
</tr>
<tr>
<td>1.4 Research methodology</td>
<td>4</td>
</tr>
<tr>
<td>1.5 Scope of the study</td>
<td>6</td>
</tr>
<tr>
<td>1.6 Limitation of the study</td>
<td>6</td>
</tr>
<tr>
<td>1.7 Significance of the study</td>
<td>6</td>
</tr>
<tr>
<td>1.8 Organization of the Thesis</td>
<td>7</td>
</tr>
<tr>
<td>CHAPTER TWO: LITERATURE REVIEW</td>
<td>8</td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>8</td>
</tr>
<tr>
<td>2.2 Communication</td>
<td>8</td>
</tr>
<tr>
<td>2.3 Sign languages</td>
<td>8</td>
</tr>
<tr>
<td>2.3.1 Structure of sign language</td>
<td>9</td>
</tr>
<tr>
<td>2.3.2 Manual signs</td>
<td>10</td>
</tr>
<tr>
<td>2.3.3 Non-manual signs</td>
<td>10</td>
</tr>
<tr>
<td>2.3.4 Isolated Sign Recognition</td>
<td>10</td>
</tr>
<tr>
<td>2.3.5 Continuous Sign Recognition</td>
<td>11</td>
</tr>
<tr>
<td>2.3.6 Components of sign language</td>
<td>13</td>
</tr>
<tr>
<td>2.4 Ethiopian sign language</td>
<td>14</td>
</tr>
</tbody>
</table>
2.4.1 Signed Amharic .................................................................................. 15
2.4.2 Grammar of ETHSL............................................................................ 15
2.5 Sign language recognition....................................................................... 17
  2.5.1 Data Acquisition .................................................................................. 17
  2.5.2 Preprocessing ....................................................................................... 18
  2.5.3 Feature extraction ................................................................................ 24
  2.5.4 Recognition of patterns ........................................................................ 28
2.6 Evaluation of models ............................................................................... 36
  2.6.1 Confusion matrix .................................................................................. 37
2.7 Related works .......................................................................................... 38
  2.7.1 Foreign sign language recognition ....................................................... 38
  2.7.2 Ethiopian sign language recognition ..................................................... 39

CHAPTER THREE: SYSTEM DESIGN................................................................. 42
  3.1 Introduction ............................................................................................. 42
  3.2 System architecture .................................................................................. 42
  3.3 Data Acquisition ....................................................................................... 44
  3.4 Preprocessing ........................................................................................... 44
    3.4.1 Frame extraction ................................................................................ 45
    3.4.2 Resizing ............................................................................................. 47
    3.4.3 Noise removal ................................................................................... 48
    3.4.4 RGB image convert to grayscale ....................................................... 49
    3.4.5 Segmentation .................................................................................... 50
  3.5 Feature extraction ..................................................................................... 51
  3.6 Recognition of patterns ........................................................................... 54

CHAPTER FOUR: EXPERIMENT AND EVALUATION .................................. 60
  4.1 Introduction ............................................................................................. 60
  4.2 Dataset ...................................................................................................... 60
  4.3 Implementation and Experimental environment .................................... 63
  4.4 Evaluation of our model ......................................................................... 64
  4.5 Experimental result ................................................................................ 64
    4.5.1 Training and validation curve of our proposed CNN-LSTM model .... 64
LIST OF TABLES

Table 1: Related work summary ........................................................................................................41
Table 2: Frame extraction from video dataset of Amharic phrase የሰማ ምታት ..................................46
Table 3: Pseudocode for frame extraction ............................................................................................47
Table 4: Pseudocode for resizing image ...............................................................................................48
Table 5: Effects of Gaussian filter in Amharic phrase ወጻ ከጆን ..................................................49
Table 6: CNN hyperparameter setting for our model ............................................................................53
Table 7: LSTM hyperparameters we used in our model .................................................................59
Table 8: The collected dataset from 10 Amharic phrases .................................................................62
Table 9: Accuracy, Precision, Recall and F1-Score for CNN-LSTM model .................................69
Table 10: Correctly and incorrectly classified in CNN-LSTM model ............................................71
Table 11: General correctly and incorrectly classified of CNN-LSTM model .................................72
LIST OF FIGURES

Figure 1: Architecture of Deep neural network (Agrawal et al., 2014) .........................26
Figure 2: Sepp Hochreiter and Juergen proposed LSTM (Hochreiter & Sepp, 1997) ....32
Figure 3: The problem of the vanishing gradient (Graves et al., 2009) .........................33
Figure 4: The flow of LSTM for video frame training (Bhagat et al., 2019) ...............34
Figure 5: Internal Architectures of LSTM (Ng et al., 2015) ....................................35
Figure 6: Internal structure of LSTM (Hochreiter & Sepp, 1997) ............................36
Figure 7: General system architecture ........................................................................43
Figure 8: Original Video Acquired for Amharic phrase እድ ከመም ................................44
Figure 9: preprocess steps in our study .........................................................................45
Figure 10: Effects of Color conversion in Amharic phrase ከንት እት ..........................50
Figure 11: Effects of Image segmentation in Amharic phrase ከንት እት ......................51
Figure 12: Training and validation accuracy curve of CNN-LSTM model ..............66
Figure 13: Training and validation loss curve of CNN-LSTM model .....................67
Figure 14: Result of Accuracy, Precision, Recall, F1-score CNN-LSTM model .......68
Figure 15: Confusion matrix of CNN-LSTM model ..................................................70
## ABBREVIATIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARSL</td>
<td>Arabic Sign Language</td>
</tr>
<tr>
<td>ASL</td>
<td>American Sign Language</td>
</tr>
<tr>
<td>AUSLAN</td>
<td>Australian Sign Language</td>
</tr>
<tr>
<td>BSL</td>
<td>British Sign Language</td>
</tr>
<tr>
<td>CNN</td>
<td>Convolutional neural network</td>
</tr>
<tr>
<td>CSL</td>
<td>Chinese Sign Language</td>
</tr>
<tr>
<td>DSRM</td>
<td>Design Science Research Methodology</td>
</tr>
<tr>
<td>ENAD</td>
<td>Ethiopian National Association for Deaf</td>
</tr>
<tr>
<td>ETHSL</td>
<td>Ethiopian Sign Language</td>
</tr>
<tr>
<td>FT</td>
<td>Furor transformation</td>
</tr>
<tr>
<td>GT</td>
<td>Gabor transformation</td>
</tr>
<tr>
<td>HL</td>
<td>Hartly transform</td>
</tr>
<tr>
<td>LSTM</td>
<td>Long short term memory</td>
</tr>
<tr>
<td>MPL</td>
<td>Multi perceptron</td>
</tr>
<tr>
<td>NZSL</td>
<td>New Zealand Sign Language</td>
</tr>
<tr>
<td>PSL</td>
<td>Persian Sign Language</td>
</tr>
<tr>
<td>RNN</td>
<td>Recurrent neural network</td>
</tr>
<tr>
<td>SVM</td>
<td>Support vector machine</td>
</tr>
<tr>
<td>TSL</td>
<td>Taiwan Sign Language</td>
</tr>
<tr>
<td>WHO</td>
<td>World Health Organization</td>
</tr>
</tbody>
</table>
CHAPTER ONE: INTRODUCTION

1.1 Background

A language is a communication tool used by peoples to understand each other. Most people use voice for communication but some peoples are not lucky to use their voice for communication due to disability, one of these disabilities is an inability to hear and speak or hearing impairment (Masresha, 2010).

Sign language is a visually focused, natural language and non-verbal medium of communicating for the deaf population, and millions of hearing-impaired people use it as their first language. Hearing-impaired people communicate with other hearing-impaired individuals, and hearing-impaired people communicate with other people who can understand/learned sign language (Walelign, 2020). Sign language is developed as another language for everyday communication of hearing impaired people. The hands, head, face, and upper body are used to produce sign language, which is processed by the eyes, while the lips, mouth, and vocal cords are used to create verbal languages, which are processed by the ears. A sign language is composed of a standardized code of signs, gestures, and syntax, as well as a combination of hand placement, shapes, and motions (Omkar et al., 2019). Sign language is a nonverbal communication language used by people who are deaf or hard of hearing to express and transfer the meaning of their sign patterns to others (J & K, 2013; Kulhandjian et al., 2019).

Ethiopian Sign Language (ETHSL) is one of Ethiopia's under-researched languages. Even though it is used by over a million deaf people, there is a growing question of equality, participation, and use of rights which have been emerging among the users of the Deaf community. However, mostly not known about the ETHSL, the current status, and its use (Mulugeta, 2016).
According to a report published by the World Health Organization (WHO) in 2020, around 466 million people in the world are hearing loss. Most of the normal people can’t understand sign language and this is very difficult to easily communicate mute peoples to normal peoples. Hearing and speech impairment peoples get a challenge in different service sectors to communicate easily what they want.

1.1 Motivation

According to World Health Organization (WHO) report in 2020, around 466 million people worldwide have hearing disable. Among these, 34 million people are children. In addition to the current estimated number WHO also estimated that over 900 million people will be hearing disable by 2050, which implies that the communication gap will increase accordingly. So, the first motivation is to overcome such a communication gap because of the number of hearing-impaired persons is increase rapidly.

The second motivation is nowadays vision-based SLR is a highly active research area however, no one is done or considered on Amharic phrase-level Sign Language Recognition by vision-based methods. Due to such reasons, we wanted to study ETHSL on the Amharic phrase level.

1.2 Statement of the problem

Sign language is an independent language that is used to interact with hearing-impaired people. People must have to learn sign language first to use sign language. In Ethiopia Sign language has been serving in Deaf education for the last five decades, However, it does not get enough attention from researchers still now (Elizabeth, 2011).

The majority of normal people are unable to understand or do not know sign language. So, hearing-impaired people can not express their feelings with normal people. Therefore this leads to the following problem: Hearing-impaired persons are challenging to
accomplish their daily tasks easily, organizations are not cost-effective because they hiring sign language interpreters, in a health institution, the physician can’t treat hearing-impaired patients, minimize academic performance and job opportunities. Therefore, a system that recognizes ETHSL is necessary for deaf people to communicate with normal people.

Sign languages differ from one country to another country. Therefore there are so many sign languages in the world. For example, British Sign Language (BSL), Australian Sign Language (Auslan), American Sign Language (ASL), Taiwan Sign Language (TSL), New Zealand Sign Language (NZSL), Chinese Sign Language (CSL), Arabic Sign Language (ARSL), Persian Sign Language (PSL), Ethiopian Sign Language (ETHSL) and so on. So sign language is not universal and many research is conducted in different sign languages (Samuel, 2013).

Many studies are conducted in different sign languages; however, ETHSL has received little attention. When we compared the research conducted on Ethiopian and foreign sign languages, there were many studies on foreign sign language. This was the reason why we said there is little attention on Ethiopian sign language. For example, (Tefera, 2014), (Eyob, 2017), (Zerubabel, 2008), (Samuel, 2013), (Tamiru, 2018), (Abdi, 2011) and (Tamene, 2016) were researched on ETHSL recognition. They have gap in their study. The first gap was that they did not address the Amharic phrase level. Since Amharic phrases are longer than Amharic words and alphabets, Amharic phrase signs have an effect on the model's performance. Therefore, the Amharic phrase-level sign language has been considered in our study. At the end, the research answered the following questions.

- How to increase the quality of the image for better recognizing
- How to design CNN-LSTM model that could recognize Amharic phrase-level sign language.
- To what extent the recognition is achieved?
1.3 Objectives of the study

1.3.1 General objective

The general objective of this study is to design an Amharic Phrase level sign language recognition model.

1.3.2 Specific Objectives

The following specific objectives were achieved the general objective.

- To prepare phrase level sign language dataset
- To increase the quality of image for better recognizing Amharic phrase-level sign language
- To design a model to recognize the Amharic phrase sign language
- To test and evaluate the model.

1.4 Research methodology

To accomplish the thesis, we used Design Science Research Methodology (DSRM). DSRM focused on the design and construction of applicable artifacts, that is, systems, applications, methods, and others, that could potentially contribute to the effectiveness of Information Technology in organizations (Peffers et al., 2018). In recent years DSR has become a well-accepted research paradigm within IT and has an advantage in a discipline oriented to the creation of successful artifacts. DSRM process generally contains six stages or activities. First, identification of the problem-defining the research problem and justifying the value of a solution; second, the definition of objectives for a solution; third, design and development of artifacts (constructs, models, methods, etc.); fourth, demonstration and analysis by using the artifact to solve the problem; fifth, evaluation of the solution, comparing the objectives and the actual observed results
from the use of the artifact; and sixth, communication-communication of the problem, the artifact, its usefulness and effectiveness to other researches (Peffers et al., 2018).

**Data Collection**

Data were obtained at Bahir dar from yikatit 23 hearing impaired elementary schools and Bahir dar University's special needs department by capturing a video for each Amharic phrase sign. Frames are extracted from recordings after collected. For each video, a collection of sequences of frames is subjected to various image pre-processing algorithms such as resizing, noise removal, color conversion, and segmentation.

**Tools**

For data collection, we used a Sonny S400 with a full HD video recorder of 20 mega pixels with 1280x720 pixels image dimensions. Dell personal computer for implementation and other image preprocessing activities with Python Programming Language. Microsoft Visio for designing the system architecture and Microsoft office for preparing the document. We also used python programing language environment anaconda to run a code.

**Methods**

The overall method we were used includes pre-processing, feature extraction, and recognition. Pre-processing includes frame extraction from videos, resizing, and noise removal, color conversion, and segmentation. Frame extraction converts the acquired gesture video to frame images, resizing to normalize the image into a standard size. we used the state the art deep learning techniques for feature extraction and classification. The reason behind the selection of deep learning, they are efficient and accurate for image recognition easily. In machine learning, a person needs to identify and hand-code the applied features based on the data type and the probability of incorrect answers is high but in deep learning, tries to learn those features without additional human intervention and the probability of correct answer is high. Human programmers build algorithms for decoding and learning from data in machine learning, but in deep learning,
machines don't need a human programmer to tell them what to do with the data. As a result, deep learning trains using an artificial neural network that mimics the human brain and enables the computer to process data in a system in the same way that humans do. In general, deep learning outperforms machine learning when it comes to pattern recognition. CNN, RNN, and LSTM are examples of deep learning algorithms. We have selected CNN for feature extraction and LSTM for Recognition (Kshitij & Ying, 2018).

**Evaluation**

To evaluate our model, the sample data collected from different signers feed into the designed model. Then by using different types matrix, we evaluated our model. Some of matrix we used for evaluations is confusion matrix, accuracy, precision, recall, f1-score.

**1.5 Scope of the study**

The study is for only the Amharic language and focused on isolated Amharic phrase-level sign language of hand and face gestures. The study is not considered sentence Amharic sign languages

**1.6 Limitation of the study**

There was a time limitation to do extra work i.e not covered sentence level.

**1.7 Significance of the study**

The significance of the study is summarized as follows:

- Minimize the communication gap between hearing-impaired with normal peoples.
- Supports the hearing challenged person to accomplish their daily task easily
- It is cost-effective for the organization for hiring sign language interpreters.
- In a health institution, the physician can treat the hearing-impaired patient.
- Maximize academic performance and job opportunities for hearing-impaired peoples.
Supports for hearing-impaired peoples to run a business.

1.8 Organization of the Thesis

The thesis is organized into five chapters. The thesis general overview, the problem, and the study's general and specific objectives were all mentioned in the first chapter. The methods and procedures that we used to achieve the goal are also discussed in this chapter. Finally, in the last section of this chapter, the scope of the study and its key significance were discussed. In second chapter, examined the literature on the concept of sign language understanding in depth. We also discussed about image processing steps and pattern recognition approaches. Finally, CNN and LSTM detail building blocks also presented. Related research on sign language understanding, both on Ethiopian and other countries' sign languages, are explored in depth in chapter two. In third chapter, the system architecture of the proposed system are discussed in detail. The key components of system are preprocessing, feature extraction and classification. Many of the system's key components, as well as their operation, are extensively discussed. In forth chapter, simulation environment, experiment and evaluation of the proposed model for Amharic phrase level sign language recognition is addressed in detail. The proposed system's implementation, the dataset used, and the test results are all extensively discussed. In fifth chapter, the conclusion, contributions and future work of the study was summarized.
CHAPTER TWO: LITERATURE REVIEW

2.1 Introduction

A thoroughgoing study of related works and literature reviews are discussed in this chapter on sign language. Background and structure of the sign language, manual and non-manual sign language, components of sign language, Ethiopian sign language, signed Amharic, sign language recognition are presented. The steps in image processing and a detailed description of the building blocks of CNN and LSTM are presented.

2.2 Communication

A language is a tool that people use to interpret and communicate knowledge from one person or community to another (Tamiru, 2018). Human beings’ day-to-day activities and lifestyles are interconnected with social interactions with each other. These social interactions among human binges are done using communication. The process of conveying information and concepts from one person to another by mutually recognized signs, symbols, and verbal interaction is known as communication. It is crucial; without it, human presence on Earth will be impossible to conceive. The majority of people communicate through speech, but hearing disabled people are not blessed to use their voice as a communication tool, and the integration of mute people to other society is difficult due to lack of sign language knowledge on the normal peoples (Rivera-Acosta et al., 2017). Sign language is the only means for hearing impaired people communicate with others. The invention of sign language encourages them to express their thoughts, behaviors, and knowledge.

2.3 Sign languages

For people who are deaf or hard of hearing, sign languages are the most basic means of communication. It is made up of a well-organized system of signs. This involves facial
expressions and hand motions. Deaf people or hearing people who may communicate with deaf people typically use sign language (Eyob, 2017).

Sign language is a nonverbal communication language used by people who are deaf or hard of hearing to express and transfer the meaning of their sign patterns to others and due to lack of sign language knowledge on normal people it’s difficult to easily communicate hearing-impaired people with normal peoples (Kshitij & Ying, 2018).

Sign language is used all over the world, but it is not standardized by default. Generally, sign languages are defined by the country in which they are used, for example, American Sign Language, Ethiopian Sign. Due to the lack of an organized standard dataset, few studies in Ethiopian Sign Language have been published or are in progress (Tefera, 2014).

Sign language is a kind of communication used by deaf people. Spelling with your fingers is one kind of sign language that can be formed using either one or two hand gestures is sign language. A sign made with one hand is considered a one-handed sign, and a sign made with two hands is called a two-handed sign (FIKRTZ, 2014).

2.3.1 Structure of sign language

There is a fundamental distinction between spoken language and sign language in terms of sentence structure. For example, the structure of a sentence in Amharic is "Subject" + "Verb" + "Object," while the structure of sentences in Ethiopian Sign Language is "Topic" + "Comment," with "Time" inserted before "Topic" to signify if the event is past, current, or future (Tefera, 2014). The following examples would demonstrate the differences in sentence form between spoken and sign language.

አጠብኩ መኪና (እኔ መኪና በጠብኩ)  

In this example, the comment “አጠብኩ” (I wash) is used to illustrate the topic “መኪና” (car). The equivalent Amharic sentence is “እኔ መኪና በጠብኩ” (I wash a car).
2.3.2 Manual signs

According to (FIKRTE, 2014);(ABEBE, 2015) manual signs are the fundamental elements of sign language, and they are done solely by the hands and arms. In sign language, there are two types of signs: manual signs and non-manual signs. Manual signs, which convey meaning through hand movements and hand motions affecting other parts of the body, are the foundation of sign language.

2.3.3 Non-manual signs

In addition to manual signs, sign languages use non-manual signs to relay a lot of detail. Non-manual features such as mouth pattern, head and shoulder movement, and facial expression can be used in various variations to communicate a variety of information. For example, Lexical distinction, grammatical structure (questions, negation, sentence boundaries, and the argument structure of certain verbs), adjectival or adverbial content, and discourse feature. If you're angry at someone or about something, for example, you might not need to use even one sign. You may simply prove it by your facial expression. For anyone who asked you a "yes" or "no" question, you might simply shake your head (ABEBE, 2015). Even though manual signs make up a significantly large portion of the sign language vocabulary, non-manual signs are important for conveying information about signs. Unlike manual signs which are performed by hand and arms only, non-manual signs are conducted by facial expressions and also by body movements. These include head movement, facial expression (smile and anger), raised an eye to borrow, eye shift, and so on (Kshitij & Ying, 2018; Tamiru, 2018).

2.3.4 Isolated Sign Recognition

Isolated recognition concentrates on a single hand and face gesture made by the user and tries to recognize it. It also deals with the recognition of signs that are performed on their own, without any signs preceding or after them (Hetreed, 2008). Many researchers
concentrate on isolated sign language recognition since isolated words are considered the basic unit in sign language (Rashid & Albelwi, 2014).

According to Aran (Hetreed, 2008), isolated sign recognition research is important because it allows for the development of better mathematical models and features to represent the performed sign. Murakami and Taguchi (1991), cited in (Hetreed, 2008), proposed a signer-based (SD) recognition method that used recurrent NN to recognize 10 isolated Japanese Sign Language (JSL) words with 96 percent accuracy. Kadous (1996), cited in (Hetreed, 2008), developed a recognizer for 95 Auslan signs using instance-based learning and decision trees as a classification tool, achieving an accuracy of 80% and 15% for SD and SI, respectively.

Fang et al (2004), cited in (Hetreed, 2008), used a classification method of fuzzy decision tree and self-organizing function for SD and maps & hidden markov model (HMM) for SI to identify 5113 isolated Chinese Sign Language (CSL), with an accuracy measure of 91.6% for SD and 83.7% for SI.

Zhang et al. (2005), cited in (Hetreed, 2008), improved the system accuracy for 102 CSL by using Boosted HMMs, which improved the system accuracy to 92.7% for SD.

2.3.5 Continuous Sign Recognition

In continuous recognition, the user is required to execute gestures one after another, to recognize every gesture (Hetreed, 2008). Due to the multimodal nature of the cues (fingers, lips, facial expressions, body pose), extra-linguistic elements such as spatial references and pantomime, etc., recognition of continuous, natural signing is extremely difficult in terms of both video analysis and linguistics. Technical shortcomings such as spatial and temporal resolution, as well as inaccurate depth cues, add to these fundamental challenges (Piater et al., 2010). It also involves the issue of co-articulation (similar to speech recognition), in which the preceding sign influences the following one, complicating the recognition task since the changes between the signs must be
specifically modeled and integrated into the recognition scheme. During the transition between signs, additional movements or shapes can occur.

Starner and Pentland (1995), cited in (Hetreed, 2008), published one of the first experiments on a vision-based continuous sign language recognition method, experimenting with both colored glove and skin color monitoring. To understand the continuous signing, their method employs HMMs and grammar. They achieve a sign-level recognition accuracy of 99% using colored glove monitoring. Vogler and Metaxas (1997), cited in (Hetreed, 2008), discuss co-articulation effects and model the transition movements between signs that are automatically detected by using a k-means clustering technique on the sign's start and end points. They constructed an epenthesis model recognition method in which each sign is preceded by a transition cluster. They compare context-based and independent approaches with and without epenthesis modeling on a 53 sign vocabulary and 489 sentences. For sign recognition, they compare unigram and bigram models. The best accuracy, 95.8%, comes from epenthesis modeling with bigrams, compared to 91.7% from context-based bigrams without epenthesis modeling.

Fang et al (2004), cited in, proposed a divide-and-conquer method for SI continuous CSL recognition. The authors combine a Simple Recurrent Network (SRN) and HMMs. The SRN is used to separate continuous signs into discrete CSL recognition sub-problems, and the SRN outputs are used as HMM states. The SI tests are 85 percent accurate, while the SD tests are 92.1% accurate. Holden et al. (2005), cited in (Hetreed, 2008), introduced an automated vision-based Auslan recognition device that used HMMs to model each sign and included features that represented the relative geometrical positioning and shapes of the hands, as well as their directions of motion. On 163 test sign phrases from 14 separate sentences, their method obtained a 97 percent recognition rate at the sentence level and a 99 percent success rate at the word level. A technique based on Transition-Movement models (TMMs) for broad vocabulary continuous sign language recognition is proposed in Fang et al (2007), cited in (Hetreed, 2008). In continuous signing, TMMs are used to manage the transitions between two adjacent signs. The transformations are dynamically clustered and segmented, with the extracted
sections being used to train TMMs. A signed model is used to model continuous signing, which is then followed by a TMM. A Viterbi search is used, along with a language model, qualified sign models, and TMM. With a sensor glove and a magnetic tracker, 3000 test samples from 750 different sentences are used to capture the broad vocabulary sign data of 5113 signs. Their method is 91.9% accurate on average.

### 2.3.6 Components of sign language

Sign language is made up of multiple components of a sign, each of which has its effect on the interpretation of the sign. Hand shapes, fingerspelling, location, movement, palm orientation, and facial expression are all important components (Masood et al., 2018; Tefera, 2014).

**Hand Shape:** It is made by twisting and extending the fingers and palm. The hand form is the initial setup for making a symbol. Shapes can be made with two hands or one hand, with both hands having a different shape at the same time (Masood et al., 2018).

**Fingerspelling:** It is the method of spelling words by making their written form letter by letter in a manual alphabet, and it is a manual part of sign language that is used for full sign communication. It is useful for signing people's names, addresses, technical terminology, acronyms, initialized signs, and foreign language phrases (Masresha, 2010).

**Location:** It has the potential to alter the sense of a sign when it is done either touching a part of the body or not touching anything. Because of the place where the signing is made, certain signs have the same aspect but have different meanings. When two signs are identical, their position may be used to distinguish them (Rautaray & Agrawal, 2015).

**Movement:** Arcs, parallel lines, and curvy shapes are some of the gestures that express valid meaning in sign language (Zerubabel, 2008). Hand moments play the most critical
role in the sign language of these movements. Along with hand action, eye motions and body movement (movement across the chest) can be used (Masresha, 2010).

**Orientation:** The direction in which the hand rotates is called orientation (up, down, left, right). When a sign goes in one direction, it may have one sense and another when it moves in the opposite direction. The orientation of the fingers and the back of the hand are apparent until the palm is described. As a consequence, the position of the palm faces on the hand is a convenient way to explain the orientation (Tamiru, 2018).

**Facial Expressions:** Non-manual sign language relies heavily on these elements. During the signing, facial expressions have the potential to alter the context. For example for WH questions eyebrows are lowered, eyes are narrowed, and head forward with a slight tilt to the shoulder but for yes or no questions eyebrows are raised, eyes open wide and head and shoulder forwarded (Masresha, 2010).

## 2.4 Ethiopian sign language

American Sign Language was used to create Ethiopian sign language. Even if Ethiopian sign language is derived from American sign language, Nordic countries have had an impact on sign language, such as the Finish Sign language, which is used by Ethiopia's deaf population. One of Finland's minority sign languages is Finnish Sign Language. Deaf people from other parts of the world who use other sign languages as their mother tongue have also migrated to Finland (Eyob, 2017; Masresha, 2010).

Following the opening of deaf schools by American and Nordic missionaries, Ethiopia has formally begun to use sign language after the 1960s. The missionaries brought the sign language which was in use in their country and for more than 50 years foreign sign language was assimilated in Ethiopian deaf culture and community. The influence of the graduates of these schools is visible in the development of ETHSL (Tefera, 2014).
2.4.1 Signed Amharic

There is a language which is known as signed English which is different from ASL and it has some components of ASL but it adds additional signs. Similarly signed Amharic is used to facilitate the interaction between the hearing and the deaf people. It is similar to signed English and has the objective of producing signs that correspond to Amharic sentences, in Amharic order. It is used by Ethiopian Television to give information to the deaf people of Ethiopia (Masresha, 2010).

2.4.2 Grammar of ETHSL

The following points express the grammatical structure of ETHSL.

**Dominant hand**

The dominant hand can be the left or right hand, depending on the individual. If the signer is right-handed, the dominant hand is the right hand; if the signer is left-handed, the dominant hand is the left hand. A dominant hand normally performs signs that are made by one hand. There are three kinds of signals based on how the hands are used. These are one-handed signs:- use the only dominant hand, Two-handed symmetrical signs- use both hands and both hands move the same way, Two-handed non-symmetrical signs- use both hands, dominant hand moves and the other remains stationary (Tefera, 2014).

**Signing Area**

The signing area is an area where signing takes place and it starts from the area around the head, shoulders, and down to the waist. The signing should not go out of this space unless we are speaking for a large audience (Tefera, 2014).
Initialized Sign

Are signs that contain the shape of the first letter of the word. To describe the word the signer is expected to sign the first letter of the word (Tefera, 2014).

Gender

Location shows the gender of some signs. Most male signs are formed on the forehead and female signs in the chin or cheek area (Tefera, 2014). For example Father and mother.

Plurals and possessives

Plurals can be expressed by repeating signs or by including signs that indicate numbers such as “many” after the specific sign. The other one is forming the sign and then pointing to several locations in the signing area. A possessiveness is rarely used because context is usually used (Tefera, 2014).

Negatives

To indicate negative ideas in sign language the signer will include the word “not” after the sign. The other option is shaking the head back and forth while signing (Tefera, 2014).

Numbers

There are predefined signs for each symbol that represents a number (Ibrahim et al., 2018).
Repeating signs

In ETHSL, repetition is used to indicate continuous action in addition to plural forms. A continuous operation is done by making a sign with a slow circular movement that is repeated. When we need to demonstrate repetitive behavior, we make a sign with a series of fast, repeated movements (WOLDE, 2011).

2.5 Sign language recognition

Sign Language Recognition (SLR) is designed to understand hand and face signs and provide an interface for hearing-impaired individuals (Admasu & Raimond, 2010). Most of the time the following basic processes are necessary to make such a recognition system.

2.5.1 Data Acquisition

Data collection is the first task in sign language recognition. There are two types of sign language data collection methods: device-based and vision-based (Agrawal et al., 2014). To identify signs, a device-based approach requires several electromechanical instruments that are integrated with multiple sensors. In addition to this sensor-based process, smart gloves, which the signer should wear when signing and which cost $65 each, are required (Ibrahim et al., 2018). The cost of the sensors is very high in comparison to the cost of the glove. As a result, this method is completely not effective.

A visual-based approach used a camera that is less expensive and more mobile than the device-based method. It is a widely used method in the field of computer vision because it is a state-of-the-art approach. Nowadays, researchers choose this approach rather than device-based approaches because device-based approaches require costly sensors or hardware (J & K, 2013).
2.5.2 Preprocessing

Various methods are used to preprocess the obtained data. Images, recordings, audios, email, and numbers are all examples of data that can be obtained. In sign language recognition, the input data is usually images or recordings. Video data requires framing, but image data does not. After the video is collected from various signers, the video is converted to a series of frame images. Related processes such as resizing, noise removal, color conversion, and segmentation are now added to all images. The key goal of preprocessing is to improve images and eliminate unnecessary distortions to extract a good feature from the data since good preprocessing yields a good feature, which yields a good recognition rate. The following image processing principles are needed in the thesis work of video dataset.

I. Frame extraction

The conversion of video to frames or images is known as frame extraction. A video is made up of a series of images or frames. Frame rate refers to the number of images or frames taken in a second. The most popular frame rates are 24 frames per second (fps), 25 frames per second (fps), and 30 frames per second (fps), but slow-motion may use 50 frames per second (fps), 60 frames per second (fps), 100 frames per second (fps), and 200/240 frames per second (fps) (Bhagat et al., 2019).

II. Resizing

When an image is resized, the pixel size of the image is changed, but caution must be taken since pixels that contain important information can be lost. In image processing, there are a variety of resizing methods. The interpolation resizing technique uses known data to approximate values at uncertain stages. Image interpolation works in two ways, attempting to achieve the best approximation of pixel intensity based on the values of the surrounding area in one direction and attempting to achieve the best approximation of pixel intensity based on the values of the surrounding area in the other. The most widely
used interpolation algorithms are adaptive and non-adaptive interpolation algorithms. Non-adaptive methods (such as bilinear, bicubic, nearest neighbor, spline, Lanczos, and others) treat all pixels in the same way, while adaptive methods vary based on the data they’re interpolating (Lin et al., 2014).

**Bilinear interpolation:** A bilinear interpolation scheme allows you to estimate the value of a function at some point in the rectangle's interior if the function's value is determined at the four corners. Bilinear interpolation uses a weighted average of the values at the four corners of the rectangle. The interval between the point and the corners for \((x,y)\) position inside the rectangle is used to measure the weights. Corners that are closest to the pole are assigned more weight. The generic result is a saddle-shaped quadratic function, as seen in the contour plot to the right (Lin et al., 2014).

In bilinear interpolation, Linear interpolation in two directions is used to conduct bilinear interpolation. Although each step is linear in terms of sampled values and position, the interpolation as a whole is quadratic in terms of sample location. Bilinear interpolation interpolates pixel color values, resulting in a continuous transition in the display even though the original content has discrete transitions. This algorithm eliminates contrast or sharp edges, which is ideal for continuous-tone images, but it may be undesirable for line art (Lin et al., 2014).

**Bicubic interpolation:** Bicubic interpolation is an extension of cubic interpolation for interpolating data points on a two-dimensional regular grid. The interpolated surface is smoother than surfaces resulting from bilinear interpolation or nearest-neighbor interpolation. Bicubic interpolation can be done using Lagrange polynomials, cubic splines, and the cubic convolution algorithm. In image resampling, bicubic interpolation is often preferred over bilinear or nearest-neighbor interpolation when speed is not a problem. Unlike bilinear interpolation, which takes into account just four pixels \((2\times2)\), bicubic interpolation takes into account sixteen pixels \((4\times4)\). Bicubic interpolation results in images that are smoother and with fewer interpolation artifacts. It achieves
dramatically improved efficiency with just a small increase in computational complexity (Lin et al., 2014).

III. Noise removal

Unwanted information is applied to image as noise. Different filtering methods may be used to delete or reduce it. The followings are the noise removal methods:

**Median filter:** the median filter is a nonlinear optical filtering technique that is commonly used in the digital video frame extraction process and it protects edges while eliminating noise. The median filter replaces any value with the norm of nearby pixels as it moves through the image component by component (Kaur & Kaur, 2014).

**Mathematician filter:** it has a square measure that is built to limit and decrease the amount of time it takes to rise and fall without overshooting. A mathematician is a smoothing filter that is one of the twenty convolution operations used to eliminate noise and blur when extracting image frames from a video (Kaur & Kaur, 2014).

**Wiener filter:** Signal windowing is the best application for it. The original signal and noise spectra are required by the Wiener filter, which works best if the original signal is smooth (Ibrahim et al., 2018).

**Gaussian filter:** A linear filter is a Gaussian filter. It is most often used to distort an image or decrease noise. By itself, the Gaussian filter blurs edges and reduces contrast. When the noise level is too high, we can see that even if the number of noise pixels decreases as the Gaussian filter size increases, they remain in the image. With a 3 x 3 filter ratio, the median filter, on the other hand, already removes the majority of noise pixels. The Median filter will further exclude noise pixels by raising the filter size, but it removes a lot of image-structure information and image quality. In the video data due to light effects, there are object noises rather than pixel noises and these object noises also reduce by using the Gaussian filter technique. As the filter size is increase in Gaussian the amount of noise removal is increase (M. Wang et al., 2014).
IV. Color representation

Color is extremely important in digital image processing. RGB, binary, and, grayscale images are all possible. A binary image has a black or white value, indicating that each pixel has either Black (0) or White (1). It is the most basic image representation format, taking the least amount of computing time and memory. Where shape and outline details are needed, a binary representation is normally used. Grayscale images have a wider range of intensity than binary images. With pixel values ranging from 0 (black) to 255 (white), each pixel is a different shade of gray. A grayscale image has 8 bits per pixel (Kumar et al., 2017).

V. Segmentation

Segmentation aims to break down an image into more reflective and understandable regions. The method of locating points and borders (e.g. lines or curves) in images is known as image recognition. Specific surfaces, artifacts, or natural parts of objects may all be represented by such areas. The threshold strategy, edge detection-based techniques, area-based techniques, clustering-based techniques, watershed-based techniques, partial differential equation-based techniques, and artificial neural network-based techniques are all examples of image segmentation techniques. In terms of the segmentation mechanism, these methods differ from one another (Kaur & Kaur, 2014).

Thresholding is the method of turning a grayscale image into a black and white image by turning all pixels that have a value above a certain threshold white and leaving the rest black. The thresholding of a grayscale image can be used to produce binary images. If the image intensity src(x,y) is less than some fixed constant T (that is, src(x,y) T), each pixel in the image is replaced with a black pixel, and if the image intensity is greater than that constant, each pixel is replaced with a white pixel (Guruprasad, 2020).
Thresholding method: The most basic method for image segmentation is threshold methods. These techniques separate image pixels based on their intensity level. These techniques are applied to the image of lighter artifacts than the background. These methods can be chosen manually or automatically, and they can be based on prior knowledge or information about image features. This technique makes image segmentation in the foreground and background regions of an image based on different intensities or colors simple and effective. It takes a grayscale image and transforms it into a binary image (Yenegeta, 2020).

There are three types of thresholding (Rivera-Acosta et al., 2017).

Global Thresholding: Any appropriate threshold value/T can be used to do this. Throughout the image, this T value will remain constant. T can be used to extract the output images q(x,y) and p(x,y) from the original image (Rivera-Acosta et al., 2017).

\[
q[x,y] = \begin{cases} 
1, & \text{if } p(x,y) > T \\
0, & \text{if } p(x,y) \leq T 
\end{cases}
\] (2.1)

Variable Thresholding: In this type of thresholding, the value of T can vary through the picture. This can also be broken down into two groups: Local and adaptive.

Local Threshold: In this, the value of T depends upon the neighborhood of x and y.

Adaptive Threshold: The value of T is a function of x and y.

Multiple Thresholding: Multiple threshold values, such as T0 and T1, are used in this form of thresholding. The following formula can be computed using these outputs.

\[
q[x,y] = \begin{cases} 
m, & \text{if } p(x,y) > T1 \\
n, & \text{if } p(x,y) \leq T1 \\
o, & \text{if } p(x,y) \leq T0 
\end{cases}
\] (2.2)

The values of thresholds can be calculated using the peaks of image histograms, and simple algorithms can also be created to do so (Kulhandjian et al., 2019).

Edge Based Segmentation Method: Since a single strength value does not have a clear knowledge of edges, this approach is a well-developed image processing technique in and of itself. It reflects on the importance of an image's dramatic change in intensity. Edge
detection algorithms look for edges where the second derivative has zero crosses or where the first derivative of strength exceeds a certain threshold. In edge-based segmentation approaches, the edges are first detected, then added to form the entity boundaries, which are then used to segment the related regions. The two most basic edge-based segmentation methods are gradient-based approaches and gray histograms. Using one of the simple edge detection techniques, such as the Sobel operator, canny operator, or Robert's operator, to find the edges. These methods provide a binary image as a result. There are systemic strategies that are aimed at detecting discontinuities (Al-amri et al., 2010).

**Region-Based Segmentation Method:** Area-based segmentation methods are those that divide an image into multiple regions with the same characteristics. There are two simple approaches that are based on this approach i.e region growing and region splitting and merging methods (Al-amri et al., 2010).

**Region growing methods:** The methods that segment the image into different regions based on the growth of seeds are known as an area growing based segmentation methods (initial pixels). Manually (based on prior knowledge) or automatically, these seeds may be chosen (based on the particular application). The growth of seeds is then regulated by the connectivity between pixels, which can be prevented using prior knowledge of the problem (Yenegeta, 2020).

**Region splitting and merging methods:** For segmenting an image into different regions, area splitting and merging-based segmentation approaches use two basic techniques: splitting and merging. Splitting is the process of separating an image into regions with identical characteristics iteratively, and blending is the process of integrating adjacent similar regions.
2.5.3 Feature extraction

After segmentation, feature extraction is the second most critical method of digital image processing. This method is used to extract morphological features such as color, shape, and size from a segmented image to represent it in a compact feature vector. It is often used to extract morphological features such as color, shape, and size from a digital image. The result of feature extraction is a numeric feature vector, which is used to interpret relevant information from images and serves as the foundation for the classification process. The aim of image feature extraction is to increase recognition rates by extracting new features from raw pixel data to represent objects (Yenegeta, 2020).

One of the most important stages in image recognition, in general, and sign language recognition, in particular, is feature extraction (Rautaray & Agrawal, 2015). It is self-evident that the whole video or image cannot be used as a feature(s), as this will necessarily require a high level of computation difficulty for whatever recognition method is being used. As a consequence, function extraction is critical to system success in terms of reducing system complexity and achieving good recognition outcomes. In this respect, hand and face gestures, for example, are very rich in form variety, motion, and textures, so extracting basic features of those elements allows for automated sign recognition in most sign languages around the world (Samuel, 2013).

The feature is a function of one or more measurements, each of which determines a quantifiable property of an object and which is measured to quantify some of the object's important characteristics. Several types of image features, such as color, shape, texture, and size, have been proposed for image recognition, and choosing the correct selection of features is the vital key to avoiding confusion in all pattern recognition systems (Rautaray & Agrawal, 2015).
Color

When displaying an image, color is the most obvious and significant aspect that humans notice. Since color information is more sensitive to the human visual system than gray levels, color is the first candidate for attribute extraction. One popular approach for representing color contents is the color histogram. The algorithms follow a common pattern: color space selection, color feature representation, and matching algorithms (Kulhandjian et al., 2019).

Texture: Textures are distinct intensity differences that are normally caused by the roughness of material surfaces and may represent the nature of certain real-world images. It's a crucial image attribute that's used for image characterization and can help in object recognition and identification. It's the image's pattern repetition over a specific region and patterns differ in scale, form, and color from one area of the image to the next (Yenegeta, 2020).

Shape

Shape descriptors are classified into two categories: region-based, which uses the whole area of an entity to describe the structure, and contour-based, which uses geographic features as boundary fragments. In a region-based technique, all of the pixels inside a shape are used to provide a representation of the shape. The region-based feature extraction approach for image recognition is a better representation of form and is used to classify non-connected or isolated shapes, but it has a higher computational complexity than the contour-based approach. The contour or boundary-based approach to obtaining the image of a shape only takes into account the shape's contour detail (pixels). This strategy is easy or more common since it takes less time to compute (Eyob, 2017; Kshitij & Ying, 2018).

There are several feature extraction techniques available for the extraction of image features i.e handcrafted and deep feature extraction approach.
**Deep feature extraction approach:** Statistical properties are automatically identified to extract features from an image.

**CNN:** Among the various algorithms in deep learning, CNN is the state of the art for feature extraction in image processing. It is a multi-layer neural feed-forward network with a deep supervised learning structural architecture that can be thought of as a two-part mixture of an automated feature extractor and a trainable classifier that can extract topology features from images. CNN extracts features from the original image at the first layer. It takes a pixel from an image as input and transforms it on various layers to extract features (Sheena & Narayanan, 2015). It can be used to learn complex and high-dimensional details. The way the convolution and pooling layers are probed varies. Convolutional, pooling, activation, and dropout fully-connected layers are among the multiple parameters and layers that make up CNN. Basic visual features are extracted from the local receptive domain using a convolutional layer. It's organized in a plane called a basic unit of neurons, and it's also known as feature mapping.

![Architecture of Deep neural network](image)

*Figure 1: Architecture of Deep neural network (Agrawal et al., 2014)*
Convolutional neural networks are excellent at detecting local spatial patterns in data and using those patterns to distinguish images. Owing to the presence of pooling layers, CNNs are unaffected by rotation or translation of two identical images, resulting in an image, and its rotated image is known as the same image. Because of the vast advantages of CNN in extracting the spatial features of an image, they used the Inception model of the Tensor Flow library, which is a deep ConvNet, to extract spatial features from the frames of video sequences (Masood et al., 2018).

CNN has an operation of convolution to convolve image pixels. Convolution operation (denoted by * operator) over a two-dimensional input image I and two-dimensional kernel K is defined as:

\[
S(i, j) = (I * K)(i, j) = \sum \sum K(i + m, j + n) I(m, n) .
\]  

(2.3)

Where i and j are an image I coordinates and m, and n are kernel K coordinates. Mathematically, equation 2.3 is called cross-correlation. However, many neural network libraries implement it as a convolution operation and call it convolution. The output of the convolution layer after each convolution operation is then W output x H output x D output. Where

\[
W \text{ output} = \left( (W \text{ input} - F + 2 P) / S \right) + 1
\]  

(2.4)

\[
H \text{ output} = \left( (H \text{ input} - F + 2 P) / S \right) + 1
\]  

(2.5)

\[
D \text{ output} = K
\]  

(2.6)

Where F denotes the filter size, P denotes the number of zero paddings, S denotes the number of stride sizes, K denotes the number of filters applied.

Figure 2: CNN layer feature convolution process (Yenegeta, 2020)
Maxpooling: The main objective of maxpooling is to reduce the dimensionality of an input representation (image, hidden-layer output matrix, etc.) such that assumptions can be made about features contained in sub-regions. The process of max-pooling is just specifying the sliding size or filter size and taking the maximum of the values in the sliding matrix. For example when we take the $4 \times 4$ matrices and the sliding size or the filter size is $2 \times 2$ the step size is set to 2. For each slide, it’s taking the maximum value in the filter region.

![Maxpooling Diagram](image)

**Figure 3: max pooling with the filter size of 2 by 2 and stride of 2** (Yenegeta, 2020)

### 2.5.4 Recognition of patterns

Pattern is any distinguishable representation or interrelation of records, events, and concepts. And when heavily influenced by noise, patterns must be distinguishable. Classification, also known as pattern recognition, is a method for determining if a new entity belongs to a certain category or not, based on whether its characteristics fall within or outside of that group's scope. The classification task is to categorize or assign a new image (different from the training set) into one of the classes given images that are standard examples of the number of classes (the training set) and to improve accuracy, the training images should have as much a variety as possible (Yenegeta, 2020).

In image classification, there are two primary types of classification processes. There are two forms of classification: supervised and unsupervised. The idea behind supervised
classification is that a user selects sample pixels in an image that represent different groups, and then instructs image processing software to use these training sites as benchmarks for classifying the rest of the pixels in the image. Unsupervised sorting, on the other hand, analyses a huge number of unknown pixels and splits them into classes depending on the image's values' normal groupings (Walelign, 2020).

There are different algorithms to classify sign language signs. These are the followings:

**SVM:** SVM is the simplest type; it does not support multiclass classification, but it does support binary classification and dividing data points into two categories. The same principle is extended to multiclass classification after breaking down the multiclassification problem into multiple binary classification problems. It is a discriminative classifier with a hyperplane separator as its formal representation, and it aims to differentiate input patterns by minimizing the gap between the hyperplane's nearest vectors and correctly separating them. A nonlinear transformation based on a regularization parameter is used to position the input vectors in a high-dimensional feature space. Except for large data sets, it correctly divides groups of limited training samples and achieves the best generalization. As a consequence, SVM is a commonly used binary classification algorithm (Tamiru, 2018).

SVM, unlike conventional ANN, is focused on systemic risk minimization rather than analytical risk minimization. SVM has been suggested by several research institutes as a learning classifier for ability management in regression and binary classification problems. SVM is used in many applications, including face recognition, gender classification, SLR, text classification, and handwriting recognition (Kulhandjian et al., 2019).

**CNN:** The convolutional neural network (CNN) is a deep learning algorithm that can be used for feature extraction as well as classification at the first layer with the last layer. CNN separates attributes from the original image and classifies the pattern. The linked layer (dense layer) is one of CNN's final layers, and it is used for grouping. The final performance probabilities for each class are calculated using the fully connected layer.
Before adding the classifier layer, a fully connected layer is added at the end of the CNN model. The classification of CNN can be optimized through different optimization techniques. Optimization algorithms are used to decrease the loss and increase the accuracy of the model (Rathi & Gawande, 2017). There are two types of optimization algorithms; these are adaptive optimization algorithms and non-adaptive optimization algorithm. Stochastic gradient descent, Momentum, and Nesterov Momentum are examples of non-adaptive optimization algorithms. Ada Grad, Adadelta, and Adam are examples of adaptive optimization algorithms.

**RNN**: The most advanced algorithm for sequential data is recurrent neural networks (RNNs). They are the first algorithm to remember the input due to intrinsic memory, making them suitable for machine learning problems with sequential data. The series includes material that recurrent neural networks can use. (RNNs) use to perform recognition tasks. One drawback of RNN is that, in practice, RNNs are not able to learn long-term dependencies (Masood et al., 2018).

RNN is a kind of Neural Network in which the output from the previous step is used as input in the current step and all the inputs and outputs are independent of each other in previous approaches (before the implementation of RNN). So, when conditions like anticipating the next phase of a sentence are required, the previous phrases are required, therefore, remember the previous phrase is required. A recurrent neural network (RNN) is introduced with hidden layers to solve this issue. The most fundamental feature of RNN is the hidden state, which remembers some information about the sequential nature of the data. As expected, understanding context is a crucial ability for tasks such as sequence recognition and classification tasks. To evaluate a sequence of any size, an RNN is replicated throughout time flowing the contextual information learned to update the corresponding weights that represent its internal states. RNN has a “memory” which remembers all information about what has been calculated and store all details about the calculations. It uses the same parameters for each input and it produces the same output by doing the same task on all inputs or hidden layers. Unlike other neural networks, this decreases the complexity of the parameters (Drumond et al., 2018).
Consider a network of one input layer, three hidden layers, and one output layer. Then, as much as most neural networks, each hidden layer would have its collection of weights and biases. For example, the first hidden layer's weights and biases are \((W_1, B_1)\), the second hidden layer's weights and biases are \((W_2, B_2)\), and the third hidden layer's weights and biases are \((W_3, B_3)\). This implies that each of the layers is self-contained i.e they are unable to recall past outputs and RNN then completes the following tasks: 1) Independent activations are transformed to dependent activations by assigning the same weights and biases to all of the network's layers. This reduces the difficulty of raising parameters and recognizing previous outputs by making each output an input for the next hidden layer 2) Since the weights and bias of all the hidden layers are the same, we may join all three layers together to form a single recurrent layer (Y. Wang et al., 2000).

Long Short Term Memory (LSTM) is a combination of RNN with LSTM units, and LSTMs can learn to bridge time intervals of more than 1000 steps, even though the input sequences are noisy and incompressible.

**LSTM:** Long short-term memory was suggested by German researchers Sepp Hochreiter and Juergen as a solution to the vanishing gradient problem (LSTM). Learning to store information over prolonged time intervals via recurrent backpropagation takes a long time, mainly due to inadequate, decaying error backflow (Hochreiter & Sepp, 1997).

They used a new, effective, gradient-based approach called Long Short-Term Memory to solve this problem. LSTM can learn to bridge minimal time lags above 1000 discrete time steps by using constant error flow through constant error carrousels within special units. Multiplicative gate units figure out how to open and close access to a continuous error wave. LSTM is spatially and temporally local and has a numerical complexity of 0 per time step and weight. They put LSTM through its paces and compared it to other algorithms. LSTM
also solves complicated, artificially long time lag tasks that previous recurrent network algorithms have never been able to solve (Hochreiter & Sepp, 1997).

Figure 2: Sepp Hochreiter and Juergen proposed LSTM (Hochreiter & Sepp, 1997)

Standard RNNs are well known for having a narrow contextual knowledge set, making it difficult to learn long-term contextual dependencies. The issue arises from the amount of power given feedback receives in the hidden layer regrettably; the knowledge that regular RNNs can reach is very restricted in reality. If it loops through the network's recurrent connections, the power of a given input on the hidden layer, and hence on the network output, either decay or explodes exponentially. When using gradient descent to train machine learning algorithms, the vanishing gradient problem will occur. This is most prominent in deep learning systems, which have several cortical layers, but it can also happen in recurrent neural networks. The effect of the vanishing gradient is expressed on figure 3 below.

The short-term memory problem is the most common issue with RNN. This problem (vanishing gradient) is a possible issue that we need to address in our research. Long
Short Term Memory (LSTM) is a recurrent neural network that was created to solve the problem of vanishing gradients (Hochreiter & Sepp, 1997).

An LSTM hidden layer is made up of memory blocks, which are recurrently attached subnets. The input gate, forget gate, and output gate is three multiplicative gates that regulate the activation of internal units or cells in each block. The gates have the function of allowing cells to store and view information for extended periods. The activation of the cell, for example, would not be overwritten by new inputs entering the network as long as the input gate stays closed (has an activation close to zero) (Graves et al., 2009).

![Figure 3: The problem of the vanishing gradient](image_url)

As shown in the figure above the units are shaded according to how sensitive they are to the input at time 1 where black is high and white is low. As can be seen, the influence of the first input decays exponentially over time (Graves et al., 2009). These gates can learn which data in a sequence is important to keep or throw away. By doing that, it can pass relevant information down the long chain of sequences to make predictions. Almost all state-of-the-art results based on recurrent neural networks are achieved with these networks.
Since CNN is computationally intensive, the video frames were trained using LSTMs with a convolutional kernel. The Convolutional LSTM-based architecture in figure 4 below is used to train depth or RGB video frames. As seen in the diagram, there are two convolutional LSTM layers, each with a kernel size of (5,5), several filters of 32 and 54, and strides of 2. The output from these two layers is then passed into a fully connected layer with 100 nodes and ReLU activation until the final softmax layer classifies the video frames into one of the ten groups.

![Diagram of LSTM architecture](image)

**Figure 4: The flow of LSTM for video frame training (Bhagat et al., 2019)**

According to (Masood et al., 2018) Video frame sequences include both temporal and spatial features, and they were used to train the model on spatial features using a deep convolutional neural network (CNN) and temporal features using LSTM. Their dataset consists of 46 gesture types in Argentinean Sign Language, and the proposed model achieved a high accuracy of 95.2% over a vast range of images.

According to (Ng et al., 2015), because videos contain dynamic content, the variations between frames may encode additional information that can help anticipate outcomes more accurately. A basic recurrent neural network computes the hidden vector sequence $h = (h_1 \ldots h_T)$ and output vector sequence $y = (y_1 \ldots y_T)$ given an input sequence $x = (x_1 \ldots x_T)$ by iterating the following equations from $t = 1$ to $T$:

$$H_t = H(W_{ih}x_t + W_{hh}h_{t-1} + b_h) \quad (2.7)$$

$$Y_t = W_{oh}h_t + b_o \quad (2.8)$$
where W is for weight matrices (e.g. W_{ih} is for input-hidden weight matrix), b is for bias vectors (e.g. b_{h} is for hidden bias vector), and H is for the hidden layer activation function, which is commonly the logistic sigmoid function.

**Figure 5: Internal Architectures of LSTM (Ng et al., 2015)**

A single floating-point value C_{t} is stored in each LSTM cell (eq.2.8). This value can be reduced or deleted as a result of a multiplicative interaction with the forget gate F_{t} (Eq.2.7) or additively modified by the current input x_{t} multiplied by the input gate its activation (Eq. 2.9). The output gate O_{t} regulates the emission of h_{t}, which is changed by the hyperbolic tangent nonlinearity from the stored memory C_{t} (Eq. 2.9, 2.10).

Unlike standard RNNs, the Long Short Term Memory (LSTM) architecture uses memory cells to store and output information, allowing it to better discover long-range temporal relationships. The hidden layer H of the LSTM is computed as follows:

\[ I_{t} = (W_{xi}x_{t} + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_{i}) \]  \tag{2.9}

\[ F_{t} = (W_{xf}x_{t} + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_{f}) \]  \tag{2.10}

\[ C_{t} = f_{ct-1} + i_{t} \tanh(W_{xc}x_{t} + W_{hc}h_{t-1} + b_{c}) \]  \tag{2.11}

\[ O_{t} = (W_{xo}x_{t} + W_{ho}h_{t-1} + W_{co}c_{t} + b_{o}) \]  \tag{2.11}
\[ H_t = ot \tanh (ct) \]  

(2.12)

where \( \_ \) is the logistic sigmoid function, and I, F, O, and C are respectively the input gate, forget gate, output gate, and cell activation vectors. By default, the value stored in the LSTM cell C is maintained unless it is added to by the input gate I or diminished by the forget gate F. The output gate O controls the emission of the memory value from the LSTM cell.

Figure 6: Internal structure of LSTM (Hochreiter & Sepp, 1997)

2.6 Evaluation of models

Data collection, data exploration, pre-processing, feature extraction, and classification technique application are the general steps for designing a machine learning classification model. However, before the classifier model is finalized, we must ensure that it functions well, which is referred to as model evaluation. There are several methods for evaluating models, some of which are mentioned below.
2.6.1. Confusion matrix

The Confusion Matrix, also known as the Error Matrix, is used to evaluate/judge a model's performance on test samples where the actual values are already known. It is made up of a sum of wrong and accurate values that are broken down by class. It informs you of the model's inaccuracy as well as the sort of error the model produced. Simply said, it's a table that illustrates how our model got it wrong while predicting. True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) are the four terms that are usually used with confusion matrix (FN).

To find the errors in the prediction confusion matrix it is important to find other performance metrics such as Accuracy, Precision, Recall, and F1-measure (Novakovic et al., 2017). We briefly discussed these performance metrics one by one as the following.

**Accuracy**: as the name indicates, the value of accuracy metric submits the accuracy of our model in predicting results and defined by the following equation:

$$\text{Accuracy} = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (2.13)$$

**Precision**: is the measure of all properly identified (actual positive values) among all predicted positive values and calculated using the following equation:

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2.14)$$

**Recall**: is the measure of positive values that are predicted properly among all actual positive values and calculated using the following equation:

$$\text{Recall} = \frac{TP}{(TP+FN)} \quad (2.15)$$

High Recall Value states that the class is known correctly since a number of False Negative is small.

**F1-score**: is the weighted average of the precision and recall. The relative contribution of precision and recall to the F1-score are equal.

$$\text{F1-score} = \frac{2 \times (\text{precision} \times \text{recall})}{(\text{precision} + \text{recall})} \quad (2.16)$$

Micro-average, macro-average, and weighted-average for all the aforementioned performance matrix can also be calculated and used for additional analysis of results.
Macro-average precision or recall is just the average of the precision and recall (respectively) of the model on different classes.

\[
\text{Micro-\text{\text{average precision}} = \frac{(TP1+TP2+\ldots+TPN)}{(TP1+TP2+\ldots+TPN)+(FP1+FP2+\ldots+FPN)}} \tag{2.17}
\]

\[
\text{Micro-\text{average recall} = \frac{(TP1+TP2+\ldots+TPN)}{(TP1+TP2+\ldots+TPN)+(TN1+TN2+\ldots+TNN)}} \tag{2.18}
\]

2.7 Related works

From this, we discussed research works that have been done by different authors on sign language recognition. The discussion includes the research done for ETHSL and foreign Sign language.

2.7.1 Foreign sign language recognition

Many studies have been done on different sign languages. Therefore, some of the researches on non-Ethiopian and Ethiopian sign languages have to be tried to discuss as follows. (Sidig et al., 2017) used Hartly transforms (HT), Fourier transformation (FT), gabour transformation (GT) for image preprocessing, and support vector machine (SVM) for classification to recognize Arabic sign language recognition. (Ibrahim et al., 2018) proposed a system to recognize Arabic sign language recognition. In this article hand motion and head motion tracking has been done by skin blob detection to track the motion and Euclidian distance measurement is used for classification. (Kumar et al., 2017) used Kinect sensor and Hidden Markov Model algorithm to develop a multi-modal framework that can recognize a face, hand, and head motion to detect Indian sign language. They have tested their framework on different situations only hand motion, hand and head motion, and with face recognition.

According to (Bhagat et al., 2019), work was done on Indian sign language gesture recognition based on deep learning and image processing to recognize the static sign, dynamic sign, and numbers. Here for 36 static gestures 45,000 depth images and 45,000
RGB images are captured with Microsoft Kinect RGB-D camera and for 10 commonly used dynamic gesture words in Indian 1,080 videos were collected. The authors used a combination of RGB and depth images for training with 70% for training and 30% for testing. For static gestures, they use CNN with Softmax Classifier and LSTM was used to train video of dynamic gestures. An overall training accuracy of 98.81 was obtained for recognition of 36 static gesture signs and 99.08 training accuracy for 10 dynamic gesture signs. Because of sign language is different from one country to another country (Tamiru, 2018), we could not test the model by different images. So, every country sign language must be develop their model.

2.7.2 Ethiopian sign language recognition

Some works have been done on Ethiopian sign language. (Abdi, 2011) used hand motion trajectory detection (HMTD) to detect isolated Ethiopian sign languages. They have applied Modified Hausdorff Distance (MHD) and achieved better recognition accuracy. (Samuel, 2013) proposed a system to detect Ethiopian sign language recognition. (Tefera, 2014) used Hidden Markov model to detect Ethiopian sign language and KNN algorithms. Most of the time the above research was focused on Amharic text to Ethiopian sign language recognition and Ethiopian sign language to Amharic text on the alphabet and word level. These works were not considered Amharic phrase-level sign language.

Few works were focused on Ethiopian sign language recognition to speech conversion, however, they have some limitations. Research conducted by (Yigremachew & Endashaw, 2019) entitled Real-time Ethiopian sign language to audio converter. This work focused on recognizing the Amharic alphabet to audio form. This work did not address phrases. Mainly this work is focused on alphabet level only. Another work by (Admasu & Raimond, 2010) entitled Ethiopian Sign Language Recognition Using Artificial Neural Network was conducted to recognize Ethiopian sign language to text and speech. For feature extraction, they used Gabor Filter (GF) with Principal Component Analysis (PCA) and for recognition and classification used ANN algorithms.
Like the previous work, this study was focused on Amharic alphabet sign to voice and this did not consider Amharic phrase sign recognition.

Many research is conducted in different sign languages, however, little works have been made on ETHSL. For instance, research works for ETHSL recognition were done by (Tefera, 2014), (Eyob, 2017), (Zerubabel, 2008), (Samuel, 2013) (Tamiru, 2018) and (Tamene, 2016), (Abdi, 2011) and (Yigremachew & Endashaw, 2019). All of them have some limitations. The first limitation was, they were not considered Amharic phrase level. Because of Amharic phrase level has a greater length than the Amharic word and alphabet, Amharic phrase signs have an impact on the accuracy of the model. So, in this work, Amharic phrase level has been considered. The second limitation was, they were not used state-of-art deep learning LSTM for classification. Because there are spatial and temporal features in video frames Amharic alphabet, word, and phrase video dataset, the temporal feature can’t learn and classify by using traditional machine learning as well as CNN due to CNN has no internal memory to store information in some time interval. Temporal feature has its own sequence information and to train and classify it needs to store in internal memory in some interval of time. So, LSTM has internal memory and able to train, store and remember for long period of time. So, to learn and classify sequence of frames are better to use LSTM.
### 2.7.3. Summary

**Table 1: Related work summary**

<table>
<thead>
<tr>
<th>Title and author’s name</th>
<th>Algorithm used</th>
<th>limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indian sign language gesture recognition on words level (Bhagat et al., 2019)</td>
<td>LSTM for recognition Achieved 98.81%</td>
<td>✔ Not directly apply to ETHSL recognition.</td>
</tr>
<tr>
<td>Sign Language Recognition using Facial Expression</td>
<td>The Viola-Jones algorithm is used to detect the face, specifically the lip region. To extract features, use a Histogram of Oriented Gradients (HOG). For classification, ANN is used. Achieved 90.67%</td>
<td>✔ Not address hand signs. ✔ Not specifically applicable to ETHSL</td>
</tr>
<tr>
<td>Recognition of Isolated Signs in Ethiopian Sign Language (Tefera and Yaregal, 2014)</td>
<td>for training and recognition used Hidden Markov Models Achieved 86.9%</td>
<td>✔ Not consider Amharic phrase-level ✔ Consider only Amharic word-level</td>
</tr>
<tr>
<td>Ethiopian sign language recognition(Admasu &amp; Raimond, 2010)</td>
<td>GF together with PCA for feature extraction and ANN for recognition Achieved 98.53%</td>
<td>✔ Consider only Amharic alphabet ✔ Did not address Amharic word and phrase level</td>
</tr>
<tr>
<td>Automatic translation of Amharic text to Ethiopian sign language (Masresha, 2010)</td>
<td>eSIGN editor for creating gesture animation</td>
<td>✔ Considered only on the translation of Amharic text to ETHSL. ✔ Not include ETHSL to Amharic text.</td>
</tr>
</tbody>
</table>
CHAPTER THREE: SYSTEM DESIGN

3.1. Introduction

We propose to a design for the Ethiopian Amharic phrase-level sign language recognition model by using hand and face gestures. The system architecture that we design describes the overall tasks of the model and all components of the system architecture are discussed in detail. The model covers a series of different phases: Amharic phrase sign language video acquisition, video frame/image preprocessing, feature extraction, and Recognition. In the next section, we presented a general overview of the proposed system architecture.

3.2 System architecture

The following system architecture shows the overall architecture of the recognition model. The architecture describes the general flow of the system that contains: preprocessing (video to frame conversion, resizing, noise removal, convert to gray, segmentation), feature extraction with CNN, and recognition with LSTM models.
Figure 7: General system architecture
3.3 Data Acquisition

Our proposed study focused on visual-based Amharic phrase-level sign language recognition. We used data from Bahir Dar University's Special Needs Department and Yeakatit 23 primary school in Bahir Dar. The video was taken on a Sonny S400 camera with 20 mega pixels full HD video recorder and 1280x720 pixel image dimensions. All of the videos are MP4 files, and the converted frame images are png formats. To prevent the influence of sunshine and other external factors, all of the videos were taken in the same protected environment and a white uniform background.

![Original Video Acquired for Amharic phrase ከፍ ከመም](image)

3.4. Preprocessing

The collected data are preprocessing by using different techniques and the collected data may be images, videos, text, and numbers. In sign language recognition, mostly the input data can be images or videos. In this thesis, we collected video data and it needs framing but image data didn’t need framing. After we captured videos from different signers, the video is converted into the sequence of frames and after the video is changed to a sequence of the frame it is considered as images. From images, different preprocessing like resizing, noise removal, color conversion, and segmentation was applied to increase
the performance of CNN-LSTM model. Because the region of interest (ROI) is necessary for preparing feature extraction by CNN unless unnecessary features may be extracted by CNN. So, this ROI must be segmented. The main purpose of pre-processing is to increase image quality and remove unwanted distortions to get a good feature from the dataset. Because good preprocessing gives a good feature and a good feature gives a good recognition rate. We applied the following image processing in our study.

![Preprocessing Steps](image.png)

**Figure 9: preprocess steps in our study**

### 3.4.1 Frame extraction

Frame extraction is the conversion process of Amharic phrase sign video to frames. After we have captured the videos the next step is converting the video into image frames. The video is converted to the frame. At this stage, we have a set of sequences of frames for each video and we select appropriate frame images for the given sign and drop out meaningless images. From each video, we have taken 30 meaningful frames and the remaining frames are dropped. Because majority of phrases contained only 30 meaningful frames. Two of the phrases were extracted to 27 and 28 frames. So, in order to make 30 we were add the last frames 3 and two frames respectively. This technique is called padding (Choudhury & Kayas, 2012). The dropping of non-meaning frames are not affect the meanings of the phrase sign(Walelign, 2020). The figure below shows the sample of frame extraction from the Amharic phrase sign ከሣ የቀን.
Table 2: Frame extraction from video dataset of Amharic phrase የስ ምታት

<table>
<thead>
<tr>
<th>Original video dataset</th>
<th>Frames extracted from original dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Original video frames" /></td>
<td><img src="image2" alt="Extracted frames" /></td>
</tr>
</tbody>
</table>

Original video dataset

Frames extracted from original dataset
Table 3: Pseudocode for frame extraction

<table>
<thead>
<tr>
<th>Pseudocode for frame extraction</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> Video</td>
</tr>
<tr>
<td><strong>Output:</strong> Frame Images</td>
</tr>
<tr>
<td>Read video</td>
</tr>
<tr>
<td>Set currentFrame=0</td>
</tr>
<tr>
<td>While (True):</td>
</tr>
<tr>
<td>Frame= video_to_frames()</td>
</tr>
<tr>
<td>Write frames to disk</td>
</tr>
<tr>
<td>CurrentFrame increment by 1</td>
</tr>
<tr>
<td>until frame=30</td>
</tr>
<tr>
<td>Return frame</td>
</tr>
<tr>
<td>End</td>
</tr>
</tbody>
</table>

3.4.2 Resizing

Resizing the image is needed to make the size more appropriate for the further process. Because our data set contains images of large size i.e 1280x720 image dimensions. It is computationally heavy with the available processing resource needs to use of excessive memory and network parameters because of these reasons majority of researchers used 224x224 (Jerubbaal John Luke, Rajkumar Joseph & Cognitive, 2019). When we were fed our model image size 256x256 the model stop to train. That means train the model with large image size is difficult. Therefore, we used 224x224 image sizes. The selection of the image also depends on the types of images and the parameters we used in the model.

. We used bicubic interpolation methods to resize our images. Because images resampled with bicubic interpolation are smoother and have fewer interpolation artifacts. It yields substantially better results, with only a small increase in computational complexity and, the speed of the model also increased (Lin et al., 2014). When we experiment with the above methods bicubic gives better results. So we used bicubic interpolation techniques to resize our sample dataset.
Table 4: Pseudocode for resizing image

<table>
<thead>
<tr>
<th>Algorithm for image resize</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Data: original size image</td>
</tr>
<tr>
<td>Output Data: resized image</td>
</tr>
<tr>
<td>Begin:</td>
</tr>
<tr>
<td>For images in source directory</td>
</tr>
<tr>
<td>image = read image</td>
</tr>
<tr>
<td>resized image = resize(224, 224)</td>
</tr>
<tr>
<td>return resized image</td>
</tr>
<tr>
<td>Save resized image</td>
</tr>
<tr>
<td>End for loop</td>
</tr>
<tr>
<td>End:</td>
</tr>
</tbody>
</table>

3.4.3 Noise removal

We have used Gaussian filtering to reduce noises from the image that arises while capturing the videos. When we experiment, the gaussian noise removal technique is excellent to reduce noise in sign language recognition. Gaussian is used to remove large noise i.e noise in the form of object and median noise removal is also used to reduce small noise i.e noise in the form of pixels. When the video data is recorded, there is a shadow (object) due to the light effect, and this shadow/object noise is reduced by the Gaussian filtering technique because the noise is large (object noise) (Hambal et al., 2015). When we experiment the Gaussian noise removal technique yields the best results. The figure below shows the sample of Gaussian noise removal from the Amharic phrase sign ከወ. የጭ.

The table 5 shows original frames (frames with noisy) and effects of Gaussian filtering (frames after noisy removed).
### Table 5: Effects of Gaussian filter in Amharic phrase ከ不甘

<table>
<thead>
<tr>
<th>Frames</th>
<th>Effects of Gaussian noise removal</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Frame 1" /></td>
<td><img src="image2.png" alt="Effect 1" /></td>
</tr>
<tr>
<td><img src="image3.png" alt="Frame 2" /></td>
<td><img src="image4.png" alt="Effect 2" /></td>
</tr>
<tr>
<td><img src="image5.png" alt="Frame 3" /></td>
<td><img src="image6.png" alt="Effect 3" /></td>
</tr>
</tbody>
</table>

#### 3.4.4 RGB image convert to grayscale

The original extracted frame is in RGB (24-bit color) format, with Red, Blue, and Green components for each pixel. Processing an RGB image is computationally heavy as opposed to processing a grayscale image, which is an 8-bit image with only black and white shades. So, the input image has been converted to grayscale. Since grayscale images take less time to compute than RGB images. The figure below shows the sample of color conversion of the Amharic phrase sign ከ不甘 ከንይ.
3.4.5 Segmentation

We used segmentation techniques in our study. The purpose of segmentation is to make an image more meaningful and, easier to examine by simplifying and/or changing its representation. Objects and boundaries (lines, curves, etc.) in images are often located via image segmentation (Omkar et al., 2019).

**Thresholding method:** we used threshold methods for image segmentation. These techniques separate image pixels based on their intensity level. This technique makes image segmentation in the foreground and background regions of an image based on different intensities or colors simple and effective. It takes an image and transforms it into a binary image i.e white and black and white part is the region of interest (ROI) (Yenegeta, 2020). The figure below shows the sample of segmentation from the Amharic phrase sign ያንት ከት.
3.5 Feature extraction

In order to improve the recognition rates of our model we used feature extraction. Feature extraction is used to extract morphological characteristics such as color, shape, and size. A smaller set of more informative features is computed during feature extraction. Feature extraction is the process of reducing the number of resources that describe a large set of data. Analysis with a large number of variables generally requires a large amount of storage and computation cost (mostly time), also it may lead the classification algorithm to overfit training samples and generalize poorly to new samples. So, we used feature extraction in order to remove these problems.

There are various types of deep learning algorithms to extract features. From these we have used CNN for feature extraction algorithms because CNN automatically extracts
deep features from images. CNN detects edges from raw pixel data, uses these edges to detect shapes, and then uses these shapes to detect higher-level features. CNN extracts features by convolving the image with various filter sizes and shrinking the image's dimension with a pooling layer. Several Convolutional and pooling layers are used to extract important features from the image. Convolutional neural network (CNN) is a part of deep neural networks that are efficient for detecting patterns and make sense of them; this property makes CNN best for image analysis. CNN is a sequentially ordered layer and each of the layers has its unique functions on the data fed to it.

The segmented face and hand signs of our dataset is fed as image directly to the convolutional neural network (CNN) to extract features. The input for CNN-based feature extraction was 3x3 image, by conducting operations like convolution and filtering on this image, which was passed through several hidden layers. We were able to extract useful information for the recognition stage. The hyperparameters of CNN we used were the following:

**Convolution layer:** We used 2 convolution layers CNN networks. The input to the first convolution layer is 30,224 x 224 x 1 image. The convolution operation contains four parameters. The first parameter is the number of filters that are used to control the depth of the output volume. In our model, we have used 32, and 64 filters/kernels. A different number of convolution layers and filters are tested and those that achieve higher accuracy are selected. The second parameter is the kernel size, which determines the size of each filter (kernel) and is nearly always square. We have used 3x 3 filter sizes at a single layer. We have done convolution operations repeatedly before the input image is down-sampled with the pooling layers (operation). Functioning multiple convolution layers before applying a pooling layer allows the model to develop more complex features before the destructive pooling operation is done. An example of such convolution is described well in figure 2 of chapter two.
**Activation Layer:** We have used ReLU activation function in the activation layer throughout our model. We used the Relu activation function because it is so powerful, makes fast the training time, and increases the training accuracy of the model.

Relu is an activation function, the dimension of the output is always equal to the dimension of the input. The main function of ReLU activation operation is to introduce non-linear mappings from input to output. This makes CNN learn and model more complex functional mappings that existed in the input features. Hence, it is used to improve the accuracy or performance of the proposed model. Mathematically, it is defined as:

\[
\text{ReLU} = \max(0, x)
\]

As we have seen in the summary from table 6 below, we used two convolutions and flattens

**Table 6: CNN hyperparameter setting for our model**

<table>
<thead>
<tr>
<th>CNN feature parameter</th>
<th>Value we used for CNN feature extraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input shape</td>
<td>30, 224, 224, 1</td>
</tr>
<tr>
<td><strong>Convolution1</strong></td>
<td></td>
</tr>
<tr>
<td>filter</td>
<td>32</td>
</tr>
<tr>
<td>Kernel size</td>
<td>3, 3</td>
</tr>
<tr>
<td>Activation</td>
<td>Relu</td>
</tr>
<tr>
<td><strong>Convolution2</strong></td>
<td></td>
</tr>
<tr>
<td>Filter</td>
<td>64</td>
</tr>
<tr>
<td>Kernel size</td>
<td>3, 3</td>
</tr>
<tr>
<td>Activation</td>
<td>Relu</td>
</tr>
<tr>
<td><strong>Flatten</strong></td>
<td>1-dimension feature extracted here</td>
</tr>
</tbody>
</table>
3.6 Recognition of patterns

There are many algorithms used for sign language recognition/classification, as we have seen from chapter two literature sections. However, we used LSTM algorithms for classification/recognitions in our study. Because of our dataset is sequences of frames, LSTM also has an excellence of training and recognition of sequences of dataset (Bhagat et al., 2019). This is why we used LSTM for recognition.

The features extracted by CNN algorithm is given to the recognition algorithm LSTM in our model. After convolution and pooling operation in CNN, we got 1-dimesion features from flatten layers. We feed these 1-dimension features to LSTM algorithms. Our recognition was multi-class (10 class), allowing an object to be classified into several categories. We used 8400 frames to train our model, 1050 to validate and 1050 to test. We used different hyperparameters on LSTM algorithms in order to avoid overfitting problems and be our model is normal fit.

One of the most fundamental tasks of our thesis was preprocessing (video processing, image processing), feature extraction, and pattern recognition i.e to assign or categorize an image into one or more groups. Both feature extraction and classification/recognition networks are merged and trained end-to-end in our model. In video classification, the classification algorithm must be aware of the sequence of images' sequential characteristics. Video sign has both generic and Spatio-temporal characteristics. The time dimension of the data is very important when classifying video data into various groups (Masood et al., 2018). We used the vast amount of video data for the creation of methods for automatically recognizing video data into categories. Connections of CNN algorithm do not form cycles, which makes them insufficient for sequence labeling. For a better understanding of the temporal features of time series data, recurrent network structures have been introduced, leading to the emergence of recurrent neural networks. Recurrent neural networks (RNNs) allow cyclical connections to form loops, allowing the network's
internal state to retain a memory of previous inputs (Y. Wang et al., 2000; Wu et al., 2017).

These are what happens when we learn a recurrent neural network: First, the network is assigned a one-time phase of the input. Second, the network's current state is determined using the current input and the previous state. Third: the present situation for the next time stage, $h_t$ is changed to $h_{t-1}$. Fourth, depending on the type of the issue, we can go back in time and combine facts from all of the previous states. Fifth: Since all of the time phases have been completed, the output is determined using the final current state. Sixth: the output is then compared to the real output, i.e. the target output, and an error is produced. Seventh: then relay the error back to the network, which changes the weights, allowing us to learn the RNN.

Standard RNNs are well known for having a narrow contextual knowledge set, making it difficult to learn long-term contextual dependencies. The problem stems from the amount of power provided to a given input in the hidden layer. Regrettably, the knowledge that regular RNNs can reach is very restricted in reality. If it loops through the network's recurrent connections, the power of a given input on the hidden layer, and hence on the network output, either decay or explodes exponentially. When using gradient descent to train machine learning algorithms, the vanishing gradient problem occurs. This is most prominent in deep learning systems, which have several layers, but it can also happen in recurrent neural networks. The disappearing gradient effect was shown in the Literature review. The short-term memory problem is the most common issue with RNN. This problem (vanishing gradient) is a possible issue that we need to address in our research. Long Short Term Memory (LSTM) is a recurrent neural network that was developed to solve the problem of vanishing gradients.

An LSTM hidden layer is made up of memory blocks, which are recurrently linked subnets. The input gate, forget gate, and output gate is three multiplicative gates that regulate the activation of internal units or cells in each block. The gates have the function
of allowing cells to store and view information for extended periods of time. The activation of the cell, for example, would not be overwritten by new inputs entering in the network as long as the input gate stays closed (has an activation close to zero) (Bhagat et al., 2019).

Now let us describe the structure of the LSTM network how to work in our model: The cell state and its different gates are the basis of LSTMs. The cell state functions as a highway that transports important data all the way down the sequence chain. It can be thought of as the network's memory. During the processing of the sequence, the cell state holds relevant information. As a result, information from early time steps will find its way to later time steps, minimizing the short-term memory impact. Information is added or removed from the cell state through gates while the process is in progress. The gates are components of neural networks that determine whether or not the information is permitted to reach the cell state. The gates are in charge of learning and deciding what knowledge to keep and what to miss through the train.

The followings are the LSTM memory blocks.

**Input gate**: The cell state is modified using the input gate. In a sigmoid function, pass the previous hidden layer state and the current input. By expressing the values as being between 0 and 1, this is used to determine which values will be modified. If the value is 0, it indicates that it is not significant, and if it is 1, it indicates that it is important. Transfer the hidden layer state and current input to the tanh feature to squish values between -1 and 1, which may aid in network regulation. The sigmoid output is then multiplied by the tanh output. The sigmoid output will assess the information from the tanh output is necessary to retain.

**Output gate**: The output gate determines what the next hidden state should be. It's worth noting that the data from previous inputs are stored in the secret layer state. On the hidden layer, predictions are also made. First, the previous state of the hidden layer and the new feedback to the sigmoid feature. The changed cell state is then passed to the tanh feature. To determine what data the hidden layer state can hold, the tanh output is
multiplied by the sigmoid output. The output is matched to the hidden state's output. The next time stage contains a new cell state as well as a new secret value.

**Sigmoid layer:** In gates, there are sigmoid activations. The sigmoid activation mechanism is similar to the tanh activation function in appearance. Sigmoid function squishes values between 0 and 1, while tanh squishes values between -1 and 1. That any number multiplied by 0 equals 0, values are ignored or forgotten. This is crucial to remember for further processing or forgetting results. When we add a number by one, the result is the same, so the value remains the same or is held. As a result, the network will figure out which data to discard (throw) and which data to keep.

**Forget gate:** This gate makes the determination on which data should be discarded and which should be retained. The sigmoid function is used to combine the data from the previously hidden layer and the data from the current input. The numbers 0 and 1 represent the values. A value close to 0 indicates that the data should be ignored, while a value close to 1 indicates that the data should be retained.

**Cell state:** The forget vector is multiplied by the cell state point-wise. If the value is compounded by a value that is closer to 0, the value is lost from the cell state. The input gate's output is then used to perform point-wise addition, which is used to update the cell state to new values that the neural network finds essential. New cells are formed as a result of this.

LSTM architecture in which the output from one LSTM layer is input for the next layer. We experimented with various numbers of layers, Relu activation functions, and memory cells. Following the LSTM layers, a Softmax classifier predicts every frame.

The followings are the LSTM hyperparameters we used in our study.

**Activation:** Instead of tanh we used Relu activation function. Because Relu in hidden layer used to avoid vanishing gradient problem and better computation performance. In addition to overcoming the problem of vanishing gradients, Relu has a shorter run time and efficient activation. Relu is the most commonly utilized activation function, with a range ranging from (0 to infinity). The range of the tanh function is from (-1 to 1) and has an advantage of negative inputs will be mapped strongly negative and the zero inputs will
be mapped near zero. It has also the drawback of activation is Dense i.e. Costly, mainly used classification between two classes, it suffers from Vanishing gradient problem.

**Optimizer:** we used Adam (adaptive momentum estimation) optimizer. Because it is unaffected by the type of model and problem and has an ability to optimize. The optimization algorithm's main goal is to reduce the gap between actual and predicted output, often known as the cost function or loss function. Adam is an optimization algorithm that computes the network loss function by calculating the estimation of individual adaptive learning rate from the parameters of the first and second moments of the gradient (Kingma & Ba, 2015).

**Dense layers:** We have used only one fully connected layer to compute the final output probabilities for each class before applying it to the Softmax classifier.

**Dropout layer:** This is used to reduce overfitting by randomly disconnecting inputs from the previous layer to the next layer in the network architecture. Random disconnection ensures that no single node is responsible for “activation” when presented with a pattern. It enables multiple, redundant nodes to activate when given with similar patterns (inputs), which also helps our model to generalize. We have applied dropout layers with a dropping probability of 0.4 immediately before the fully connected layers, which is followed by the Softmax classifier.

**Learning rate:** we used a smaller learning rate to reduce the weight update. Decaying the learning rate helps to reduce overfitting. We have divided the initial learning rate by the number of epochs (1-50) each time the network is trained. In our case, the network at the first epoch is trained with the learning rate of 1e-5, while at the last epoch (epoch 50) the network is trained with 1e-5/50. As learning started to slow dramatically around epoch 50, we stopped training at epoch 50. Keeping the learning rate high will lead to overshooting of areas of the low loss since we are taking large steps to descend into these areas. It is better to decrease the learning rate progressively, and hence taking smaller
steps. A reduced rate enables the network to descend into a lower loss landscape (without missing it) effectively.

**Epoch:** we used 50 epochs to train our model.

**Batch size:** It refers to how many training instances were used in a single iteration. So, we used 32 training dataset in a single iteration.

**Softmax:** The output of the final fully-connected layer is given as input to the Softmax classifier. 10-way Softmax is used to recognize into a specific class.

The table below summarizes the hyperparameters we used in LSTM.

**Table 7:** LSTM hyperparameters we used in our model

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hidden layer</td>
<td>256</td>
</tr>
<tr>
<td>Dense layer</td>
<td>128</td>
</tr>
<tr>
<td>Dropout layer (dropping probability)</td>
<td>0.40</td>
</tr>
<tr>
<td>Batch size</td>
<td>32</td>
</tr>
<tr>
<td>Activation</td>
<td>Relu</td>
</tr>
<tr>
<td>Epoch</td>
<td>50</td>
</tr>
<tr>
<td>Optimizer</td>
<td>Adam</td>
</tr>
<tr>
<td>Learning rate</td>
<td>Le-5</td>
</tr>
</tbody>
</table>
CHAPTER FOUR: EXPERIMENT AND EVALUATION

4.1. Introduction

In this chapter, we have discussed the experimental conducted and an evaluation of the proposed model for Amharic phrase-level sign language recognition. Experiments were carried out to test the effectiveness of our proposed model and described in detail. Accordingly, the dataset used in training, validation and testing the proposed model, tools used for the implementation, the implementation of the proposed model, and the training and test results are described thoroughly.

4.2 Dataset

There is no ready dataset for Amharic phrase level sign language. Therefore, we have prepared our dataset to train, validation and test the proposed model. We gathered data for the study from Bahir Dar's Yeakatit 23 elementary school and Bahir Dar University's Special Needs Department. The dataset was prepared by five graduate students from Bahir Dar University's special needs department, one special need teacher, and one student from Yeakatit 23 elementary School and a total of seven signers have participated. Each signer is expected to perform a sign 5 times for each Amharic phrase sign. 35 videos were captured for each Amharic phrase sign and later these videos were converted into frame images. We collected data from 10 Amharic phrases. After collected the dataset, we remove the unnecessary part of the videos by using free video cutter software in manually. Then, extract videos into frames by using of frame extractor algorithms to have low computational complexity and we selected 30 important frames from each Amharic phrase video. Then the total prepared image datasets are 5x7x10x30=10500.

The video was captured by using the Sonny S400 camera, which has a 20 megapixel full HD video recorder and image dimensions of 1280x720 pixels. We kept the environment's lighting system uniform, and we used a white background for all of the videos we took. All of the videos are MP4 files, while the converted frame images are png files that have been resized to 224x224 pixels.
We considered health institutions and schools when collecting our sample data. Because when we asked special need experts, these are the most important areas in which people with hearing loss participate and facing problems of communication gaps. As previously explained above, we have obtained a dataset from seven signers. The Amharic phrases we used as a sample are listed in the table below. The overall amount of data gathered is then split into training, validation, and testing data. 10 percent of our data was used for testing and 10 percent used for validation, and 80 percent was used for training. The table below shows the collected dataset from ten Amharic phrases for the proposed model.
**Table 8: The collected dataset from 10 Amharic phrases.**

<table>
<thead>
<tr>
<th>No.</th>
<th>Amharic phrases</th>
<th>English meaning</th>
<th>Written in English alphabet</th>
<th>Number of video samples from each signer</th>
<th>Number of signers</th>
<th>Number of video sample datasets</th>
<th>Number of frame/image sample dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>የቅዶ የጥገና</td>
<td>Surgery</td>
<td>Kedo tigena</td>
<td>5</td>
<td>7</td>
<td>35</td>
<td>1050</td>
</tr>
<tr>
<td>2</td>
<td>መድሀኒቱ የመራል</td>
<td>The drug is bitter</td>
<td>Medihanitu yimeral</td>
<td>5</td>
<td>7</td>
<td>35</td>
<td>1050</td>
</tr>
<tr>
<td>3</td>
<td>የምሰጋና የገባል</td>
<td>Thanks</td>
<td>Misgan yigebal</td>
<td>5</td>
<td>7</td>
<td>35</td>
<td>1050</td>
</tr>
<tr>
<td>4</td>
<td>ልለ የምታት</td>
<td>Headache</td>
<td>Ras mitat</td>
<td>5</td>
<td>7</td>
<td>35</td>
<td>1050</td>
</tr>
<tr>
<td>5</td>
<td>እትምህርት በት</td>
<td>School</td>
<td>Timhert bet</td>
<td>5</td>
<td>7</td>
<td>35</td>
<td>1050</td>
</tr>
<tr>
<td>6</td>
<td>የማወስ አው</td>
<td>Dead man</td>
<td>Yemote sew</td>
<td>5</td>
<td>7</td>
<td>35</td>
<td>1050</td>
</tr>
<tr>
<td>7</td>
<td>የትምህርት አው</td>
<td>Sick person</td>
<td>Yetameme sew</td>
<td>5</td>
<td>7</td>
<td>35</td>
<td>1050</td>
</tr>
<tr>
<td>8</td>
<td>የሆል አው</td>
<td>Toilet</td>
<td>Shint bet</td>
<td>5</td>
<td>7</td>
<td>35</td>
<td>1050</td>
</tr>
<tr>
<td>9</td>
<td>የምሬርዓት አው-ት</td>
<td>Toothpaste</td>
<td>Yetirs samuns</td>
<td>5</td>
<td>7</td>
<td>35</td>
<td>1050</td>
</tr>
<tr>
<td>10</td>
<td>የሆድ ከመም</td>
<td>Abdominal pain</td>
<td>Hod himem</td>
<td>5</td>
<td>7</td>
<td>35</td>
<td>1050</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Total dataset</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>350</strong></td>
<td></td>
<td><strong>10500</strong></td>
<td></td>
</tr>
</tbody>
</table>
Accordingly, we have prepared our own data set that contains 350 and 10500 real videos and frames respectively of 10 different classes of Amharic phrases.

In the process of data collection, a lot of challenges have occurred. In yekatit 23 elementary school, signers of disabled students were not willing for recording. Those were the main challenges. Even if teacher Fikrte zenebe was asked her students to help us, they were not willing. But one student was a volunteer and we have taken data from teacher Fikrte Zenebe and her student Antnkut Muche.

4.3 Implementation and Experimental environment

The data is divided into training, validation, and testing datasets, with 80% of the data being used to train the model, 10% being used to validate, and 10% for testing. In our study made one experiments (one end to end experiments by using CNN and LSTM algorithms). According to (Dobbin & Simon, 2011) allocating the dataset for training is close to optimal for reasonably sized datasets (greater than 100 images). Our models are trained, validate, and tested on a Dell laptop computer with an Intel Core i5 processor and 4 GB of RAM, 1TB hard disk size, running Windows 10, and on the Google Collaborator cloud service. Data preprocessing, and document preparation have all been completed on the computer CPU and we used the Anaconda environment, which is an open-source delivery for the Python programming language. Jupiter Notebook was used to write the Python code.

Feature extraction and recognition activities have been completed on the Google collab/cloud server. The reason that we prefer to use google collaborator for the feature extraction and recognition tasks is that we can use any latest python packages simply and since we used a large number of datasets (10500 images) and google collaborator can afford GPU service. All of this service from google collaborator is afforded for free. The OpenCV library is an open-source image processing and computer vision library that can be used to perform image processing and computer vision operations when combined with Python. We used several Python packages both locally on the machine and in the cloud with the help of a Google collaborator.
4.4 Evaluation of our model

In our study, we analyzed the performance of recognizing CNN-LSTM models using various evaluation matrices. Accuracy, Precision (positive predictive value), Recall (True positive rate or Sensitivity), and F1-score are the performance evaluation matrices. The total number of correctly predicted signs for all test samples is called Accuracy.

The Confusion Matrix, also known as the Error Matrix, is used to evaluate/judge a model's performance on test samples with known actual values. It's made up of a total of incorrect and correct values broken down per class. It tells you about the model's inaccuracy as well as the type of error it made. Simply said, it's a table that illustrates how our model got it wrong while predicting. True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) are the four terms that are usually used with confusion matrix (FN). It is important to find other performance metrics such as Accuracy, Precision, Recall, and F1-measure, to find the errors in the prediction confusion matrix.

4.5 Experimental result

In this section, we discussed the experimental results of our model. We showed clearly the curves on the training and validation accuracy and loss of our model. From this curve, we identified whether the model is underfitting, overfitting, and good fitting. Good fitting model is a model without under-fitting and over-fitting problems. We also measured the model performance by using testing accuracy, precision, recall, and others. Before testing the model, we trained and validate it first.

4.5.1 Training and validation curve of our proposed CNN-LSTM model

We used CNN for feature extraction and a state-of-the-art LSTM for recognition/classification, as mentioned in the previous chapter. Both CNN and LSTM
support each other, in the sense that CNN extracts 1-Dimensional images that start from convolution layers and end flattens then, these 1-Dimensional images are fed into LSTM. We evaluate the CNN-LSTM model after it has been trained, and validate.

In our study, we made different video and image processing techniques. Such as frame extraction, resizing, color conversion, noise removal, and segmentation. Because the preprocessing of videos and images has a crucial impact on the performance increment of our model.

The accuracy curve for training and validation of the CNN-LSTM model is shown in figure 12 below. As clearly shown in the training and validation accuracy curve in figure 12, the training and validation accuracy increases when the number of epoch increase in parallel. But at epoch 28 the validation and training accuracy is almost similar because of the similarity between validation and training dataset. Training accuracy is most of the time greater than validation accuracy throughout the curve. There is only a small gap between training accuracy and validation accuracy. This indicates that our model is good fitting. There is no overfitting and underfitting in our models. When a model is overtrained on data to the point where it learns the noise from it, it is said to overfit (Sidig et al., 2017). Every example is learned using an overfit model. The overfitting model is a model that is difficult to generalize with a new dataset. Overfitting occurred when the model is trained with a small number of datasets. Underfitting occurs when a data model is unable to effectively represent the link between input and output variables, resulting in a high error rate on both the training set and unknown data (Jabbar & Khan, 2015).
Figure 12: Training and validation accuracy curve of CNN-LSTM model

As clearly shown in the training and validation loss curve in figure 13 below, the training and validation loss decreases when the number of epoch increase in parallel. Throughout the curve, training loss is most of the time less than validation loss. Between training and validation loss, there is just a tiny gap. This indicates that our model is good fitting. There is no overfitting or underfitting in our models. But at epoch 10 and 28 almost training and validation loss is similar because of the similarity of training and validation dataset.
4.5.2 Recognition accuracy of CNN-LSTM model

In this section, we discussed the recognition accuracy of the models and the different performance measurement matrix. After we trained and validated our model, then we testing and evaluated the model by different matrix. Such as accuracy, precision, recall, f1-score, macro avg, and weighted avg and confusion matrix.

There are 10 class in our study and we used 1050 datasets for testing. ወን እውነት: የመማርና የምስጋና የጥገና፣ መድሀኒቱ የማረረ፣ የተጋገር የምስጋና የጥገና፣ የትም የሆድ የመማርና የምስጋና and የጥርስ የመማርና are class names in the study.

As shown from table 9 and figure 14 below the testing accuracy of our model is 96%. The precision, recall, and f1-score of class 1 are 100%, 98%, and 99% respectively. Because we used deep learning CNN-LSTM algorithms and many datasets
that is why the testing or recognition accuracy increased. The other nine class’s precision, recall, and f1-score are mentioned on table 9 below.

Figure 14 below shows the testing accuracy of the CNN-LSTM model.

<table>
<thead>
<tr>
<th>Class</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-score</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>hodhimem</td>
<td>1.00</td>
<td>0.98</td>
<td>0.99</td>
<td>105</td>
</tr>
<tr>
<td>kedotigena</td>
<td>0.98</td>
<td>0.98</td>
<td>0.98</td>
<td>105</td>
</tr>
<tr>
<td>medihanituyamirial</td>
<td>0.94</td>
<td>0.93</td>
<td>0.94</td>
<td>105</td>
</tr>
<tr>
<td>misganayigebal</td>
<td>0.97</td>
<td>0.93</td>
<td>0.95</td>
<td>105</td>
</tr>
<tr>
<td>rasmitat</td>
<td>0.95</td>
<td>0.99</td>
<td>0.97</td>
<td>105</td>
</tr>
<tr>
<td>shintbet</td>
<td>0.93</td>
<td>0.96</td>
<td>0.94</td>
<td>105</td>
</tr>
<tr>
<td>timhertbte</td>
<td>0.95</td>
<td>0.95</td>
<td>0.95</td>
<td>105</td>
</tr>
<tr>
<td>yemotesew</td>
<td>0.97</td>
<td>1.00</td>
<td>0.99</td>
<td>105</td>
</tr>
<tr>
<td>yetamemesew</td>
<td>0.98</td>
<td>0.91</td>
<td>0.95</td>
<td>105</td>
</tr>
<tr>
<td>yetirssamuna</td>
<td>0.97</td>
<td>0.99</td>
<td>0.98</td>
<td>105</td>
</tr>
</tbody>
</table>

**Figure 14: Result of Accuracy, Precision, Recall, F1-score CNN-LSTM model**

As shown from figure 14 above result of Accuracy, Precision, Recall, F1-score and support(number of frames support in each class in testing/recognition) CNN-LSTM model are presented. ለታመመ በመው has relatively low recall accuracy because of dataset similarity(confusion) that are misclassified from its own class and classified from another class. The table 9 below shows the details of figure 14 above.
<table>
<thead>
<tr>
<th>Amharic phrase signs</th>
<th>Class</th>
<th>precision</th>
<th>recall</th>
<th>F1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>መግወለ ከመሃም</td>
<td>1</td>
<td>1.0</td>
<td>0.98</td>
<td>0.99</td>
<td>105</td>
</tr>
<tr>
<td>የተጆጆ ለም ከመሃም</td>
<td>2</td>
<td>0.99</td>
<td>0.94</td>
<td>0.97</td>
<td>105</td>
</tr>
<tr>
<td>እውሬ ከመሃም</td>
<td>3</td>
<td>0.91</td>
<td>0.95</td>
<td>0.93</td>
<td>105</td>
</tr>
<tr>
<td>ይወስወ ከመሃም ይጆጆ</td>
<td>4</td>
<td>0.94</td>
<td>0.97</td>
<td>0.96</td>
<td>105</td>
</tr>
<tr>
<td>ቀድሮ ከመሃም</td>
<td>5</td>
<td>0.96</td>
<td>0.99</td>
<td>0.98</td>
<td>105</td>
</tr>
<tr>
<td>የቃረት ከመሃም</td>
<td>6</td>
<td>0.97</td>
<td>0.93</td>
<td>0.95</td>
<td>105</td>
</tr>
<tr>
<td>ከጋራ ከመሃም</td>
<td>7</td>
<td>0.94</td>
<td>0.96</td>
<td>0.95</td>
<td>105</td>
</tr>
<tr>
<td>ይጆጆ ከመሃም ከተጆጆ</td>
<td>8</td>
<td>0.97</td>
<td>1.0</td>
<td>0.99</td>
<td>105</td>
</tr>
<tr>
<td>ያው ከመሃም</td>
<td>9</td>
<td>0.99</td>
<td>0.91</td>
<td>0.95</td>
<td>105</td>
</tr>
<tr>
<td>ይጆጆ ከመሃም</td>
<td>10</td>
<td>0.97</td>
<td>1.0</td>
<td>0.99</td>
<td>105</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>accuracy</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.96</td>
<td></td>
<td></td>
<td></td>
<td>1050</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Macro avg</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.97</td>
<td>0.96</td>
<td>0.96</td>
<td>1050</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Weighted avg</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.97</td>
<td>0.96</td>
<td>0.96</td>
<td>1050</td>
<td></td>
</tr>
</tbody>
</table>
The figure 15 above is the confusion matrix of CNN-LSTM model. Confusion matrix is an N x N matrix used to evaluate the performance of a recognition model, where N is the number of target classes. The number of correct and incorrect predictions are summarized with count values and broken down by each class in the above confusion matrix.

As we can see the tables below class 10 (የጥርስሳሙና” or ‘yetirs samuna’) and class 8 (የሞተሰው” or ‘yemote sew’) have no sign similarity with other class and we get 100% correctly classify i.e there is no misclassification.

Even if we get an acceptable result, there is some misclassification happened, this is due to the shape, orientation, and direction similarity between the Amharic phrases. For example, when we see class 3 and class 5 they are similar with facial expression. So, one image from ወይጠ ግን ን እነጋጆ is classified to ከጠ.ጠ.ሠ.ሠ.ሠ.

The table 10 below taken an idea from the confusion matrix of figure 15 CNN-LSTM model. The table describes details of each class test image number, correctly and incorrectly classify and testing accuracy. The Amharic phrase sign language that has similar signs made some confusion. That means the sign from one class is classified from another class. As we have shown from figure 15 የጠ.ጠ.ሠ.ሠ.ሠ.ሠ. Since of dataset similarity (confusion), it has a low recall accuracy because it is misclassified from its own class and classified from another class.
Table 10: Correctly and incorrectly classified in CNN-LSTM model

<table>
<thead>
<tr>
<th>Class</th>
<th>Number of test images</th>
<th>Correctly classify</th>
<th>Incorrectly classify</th>
<th>Testing accuracy(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ወድ በሆም</td>
<td>105</td>
<td>103</td>
<td>2</td>
<td>98.1</td>
</tr>
<tr>
<td>በያን ከጆ</td>
<td>105</td>
<td>99</td>
<td>6</td>
<td>94.3</td>
</tr>
<tr>
<td>ወድ ከጆጆ ከፋ ከፋ</td>
<td>105</td>
<td>100</td>
<td>5</td>
<td>95.2</td>
</tr>
<tr>
<td>ወድ ከጆጆ ከፋ ከፋ</td>
<td>105</td>
<td>102</td>
<td>3</td>
<td>97.1</td>
</tr>
<tr>
<td>ወድ ከጆጆ ከፋ ከፋ</td>
<td>105</td>
<td>104</td>
<td>1</td>
<td>99.0</td>
</tr>
<tr>
<td>ወድ ከጆጆ ከፋ ከፋ</td>
<td>105</td>
<td>98</td>
<td>7</td>
<td>93.3</td>
</tr>
<tr>
<td>ወድ ከጆጆ ከፋ ከፋ</td>
<td>105</td>
<td>101</td>
<td>4</td>
<td>96.1</td>
</tr>
<tr>
<td>ወድ ከጆጆ ከፋ ከፋ</td>
<td>105</td>
<td>105</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>ወድ ከጆጆ ከፋ ከፋ</td>
<td>105</td>
<td>96</td>
<td>9</td>
<td>91.4</td>
</tr>
<tr>
<td>ወድ ከጆጆ ከፋ ከፋ</td>
<td>105</td>
<td>105</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>1050</strong></td>
<td><strong>1013</strong></td>
<td><strong>37</strong></td>
<td><strong>96%</strong></td>
</tr>
</tbody>
</table>

4.6 Result Discussions

As shown in the training and validation accuracy and loss curve in figure 12 and 13, when the number of epoch increase training accuracy and validation accuracy increase linearly, and training and validation loss decrease throughout the curve. Both training accuracy and validation accuracy curve are more stable (no up and down). Training accuracy is almost greater than validation accuracy and training loss is almost less than validation loss throughout the curve. The gap between training and validation accuracy curves is very narrow. So the problem of overfitting and underfitting is not shown from training and validation accuracy as well as loss because our model fits with our dataset. From training and validation accuracy at epoch 28 validation and training accuracy is almost similar because of validation and training dataset similarity. By using 1050 images, we tested our model and the testing accuracy is 96%. Because of the total
number of dataset is high and we applied preprocessing, hybrid network for feature extraction and recognition, the accuracy of the model is increased. The model is tested by using confusion matrix, accuracy, precision, recall, f1-score. Almost all of the class scored the best accuracy, precision, recall, f1-score values. This implies that CNN-LSTM model has best performance. The confusion matrix also used to show correctly and incorrectly classified datasets in each class and the overall in the model of 1050 test dataset. The similarity signs are confusing each other i.e the dataset one class is misclassified from its own and classified to another (Walelign, 2020). But the sign has no similarity with another class classified 100% correctly. That means there is no misclassified. The table below shows the general correctly and incorrectly classified and model performance.

Table 11: General correctly and incorrectly classified of CNN-LSTM model

<table>
<thead>
<tr>
<th>Our model</th>
<th>Number of test image</th>
<th>Correctly classified</th>
<th>Incorrectly classified</th>
<th>Percent of correctly classified</th>
<th>Percent of incorrectly classified</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN-LSTM</td>
<td>1050</td>
<td>1013</td>
<td>37</td>
<td>96%</td>
<td>4%</td>
</tr>
</tbody>
</table>
CHAPTER FIVE: CONCLUSION AND FUTURE WORKS

In this chapter, we have discussed the conclusion of the proposed model for Amharic phrase-level sign language recognition. Major contributions and future works of the study also discussed in detail.

5.1 Conclusion

More than 1.5 million peoples are hearing impaired in Ethiopia. These people use sign language to communicating with each other. However, because of hearing people do not understand sign language; they cannot communicate with sign language. Therefore there is a communication gap between hearing-impaired people and hearing people. Some studies tried to solve the gap the communication among hearing impaired people and hearing people. Different researchers were conducted in Amharic alphabet and word-level sign language recognition but still, now there is no study in Amharic phrase-level sign language recognition. So, a study aims to design a model that translates Amharic phrase signs to text to avoid the communication gaps between the hearing disabled with hearing people.

We used deep learning techniques to solve a problem and achieve the objectives. Preprocessing (such as frame extraction, resizing, noise removal, and segmentation), feature extraction, and recognition were the three methods used in our proposed model for Amharic phrase-level sign language recognition. The proposed model was designed by a CNN feature extraction and the extracted features of the signs are then feed to LSTM and recognized by it. We used confusion matrix, accuracy, precession, recall, and f1-score to evaluate our model and achieved 96% testing accuracy.

5.2 Contributions

As a contribution to the scientific world or knowledge, the proposed deep CNN-LSTM model offered to easily recognize Amharic phrase signs. The main contribution of our study is described as follows. We classified the contributions as Scientific and
Methodological contributions. As scientific contributions, no one is study still now on Amharic phrase-level sign language recognition. So, the dataset is new i.e we prepared ourselves and this dataset is used for other researchers for the future. As we have discussed chapter three and four, there is no ready dataset in Amharic phrase sign languages. In the process of data collection, a lot of challenges have been occurred. The reason that signers are not willing for giving information. The challenges have been made their own impact on the success of our work, especially, during recording. As methodological contributions, we proposed CNN-LSTM model to recognize/classify Amharic phrase sign language. Because CNN is best for feature extraction and LSTM is best classifier/recognizer for sequentially ordered or Spatio-temporal data. So, we used the LSTM classifier, and we got best performance in our model. The resize, noise removal, color conversions, and segmentation also applied to increase the performance of our model.

5.3 Future works

In this study, we attained a good result in recognizing Amharic phrase-level sign language and it has its own impact on the ETHSL. However, Ethiopian sign language is still under developing stage. It needs more additional researches to support the Ethiopian deaf community by using a computer-aided system. There are gaps that should be filled in future researches because our work is not a full translation system but it has its own contribution on the area of Ethiopian sign language. Some of the recommendations for future work are listed below:

- Enhancing Amharic phrase level sign language recognition by developing Real-time Recognition for real-time translation of sign language.
- The proposed work can be further extended to sentence-level recognition (continuous Ethiopian sign language recognition)
- We recommend researchers to develop a mobile application to provide an easy way of communication among the hearing community.
- Our work presented about one-way communication which means it only translates the sign to text. Therefore, we suggest that other researcher will design the model
which should work like two-way communication to translate sign to text and vice versa.

- We used a uniform background (white color) while we captured videos of the signs. However, in real world, we are not able to find such uniform background. Therefore, future local researchers should consider complex backgrounds on videos and images of signs.
- We used a controlled environment when we collected our dataset. We recommend for the future researchers to use an uncontrolled environment
REFERENCE


https://doi.org/10.1109/ISDA.2010.5687057

https://doi.org/10.1504/IJCVR.2014.059361


https://doi.org/10.1109/DICTA47822.2019.8945850


https://doi.org/10.5220/0006585202150222

Elizabeth, D. (2011). Sign Language’ use as Medium of Instruction: The Case of Grade One and
Two at Mekanissa School for the Deaf. Addis Abeba Universty.


https://doi.org/10.21275/25031706


Science, 7(1), 39–46.


https://doi.org/10.1080/0960085X.2018.1458066

https://iis.uibk.ac.at/public/papers/Piater-2010-CSLT.pdf


https://doi.org/10.1109/CONFLUENCE.2017.7943247

https://doi.org/10.1007/s10462-012-9356-9


https://doi.org/10.1016/j.procs.2015.10.021


APPENDIX

APPENDIX A: Sample video data used for our model

![Sample video data](image)
APPENDEX B: The letter cooperation written to collect our dataset