ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

VISION BASED FINGER SPELLING RECOGNITION FOR ETHIOPIAN SIGN LANGUAGE

BY

EYOB GEBRETINSAE BEYENE

A THESIS SUBMITTED TO
THE SCHOOL OF GRADUATE STUDIES OF ADDIS ABABA UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTERS OF SCIENCE
IN COMPUTER SCIENCE

February, 2012

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

FACULTY OF COMPUTER AND MATHEMATICAL SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

VISION BASED FINGER SPELLING RECOGNITION FOR ETHIOPIAN SIGN LANGUAGE

BY

EYOB GEBRETINSAE BEYENE

Members of the Examination Board

NAME	SIGNATURE
1. Dr.Fitsum Admassu (Advisor)	_____
2. _____	_____
3. _____	_____

ACKNOWLEDGMENT

With the deepest gratitude, I would like to acknowledge and thank my advisor Dr Fitsum Admasu for giving me the opportunity to work with him and for his mandatory guidance in the development of this thesis. Dr. Fitsum, is an excellent mentor who pushed me to work well and to build up my confidence while making my research. I would also express my gratitude to my friends Tewodros Alene, Birhane Aregawi and Tesfu Geteye, who are helped me during data collection and document review.

Finally, I would like to thank Ato Eyasu Hailu (AAU, Department of Sign Language) for giving me important information regarding the language. My Last but not least appreciation goes to students of AAU, Kidane Admasu and Belay Tekle for their support.

Table of Contents Page

List of Figures iv
List of Tables v
List of Algorithms v
Acronyms and Abbreviations vi
Abstract vii
CHAPTER ONE 1
INTRODUCTION 1
1.1 Background 1
1.2 Statement of the Problem 2
1.3. Objective of the Study 3
1.3.1 General Objective 3
1.3.2 Specific Objectives 3
1.4 Application and Results 3
1.5 Scope and Limitation of the Study 4
1.6 Methodology 4
1.6.1 Data collection 4
1.6.2 Methods 5
1.7 Organization of the thesis 6
CHAPTER TWO 7
OVERVIEW OF SIGN LANGUAGE 7
2.1 Sign Language 7
2.2 Sign 7
2.3 Ethiopian Sing Language (ESL) 8
2.4 ESL Finger Spelling 8
2.5 Representation of ESL 11
CHAPTER THREE 12
LITRATURE REVIEW AND RELATED WORK 12
2.1 Literature Review 12
2.1.1 Methods of signed language recognition 12

2.1.2 Finite States Automata 13
2.2 Related Works 14
2.2.1 Ethiopian Sign Language Recognition Using Artificial Neural Network 15
2.2.2 Automatic translation of Amharic text to Ethiopian sign language 15
2.2.3 Continuous Gesture Recognition System for Korean Sign Language based on Fuzzy Logic and Hidden Markov Model 16
2.2.4 Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video 17
2.2.5 Real-Time Malaysian Sign Language Translation using Color Segmentation and Neural Network 18
CHAPTER FOUR 20
SIGNED HAND SEGMENTATION AND FEATURE EXTRACTION 20
4.1 Video Acquisition 20
4.2 Image frames Selection 21
4.3 Signed Hand Segmentation 28
4.3.1 Preprocessing of images 28
4.3.2 Segmentation Using Global Thresholding 35
4.3.3 Isolating Signed Hand 37
4.4 Feature Extraction 39
CHAPTER FIVE 42
RECOGNITION 42
5.1 Vowel Recognition 42
5.1.1 Motion Detection 42
5.1.2 Required Points for Recognition 45
5.2 Recommended frame for consonant recognition 52
CHAPTER SIX 56
EXPERIMENT AND RESULT 56
6.1 Test Data and Description 56
6.2 System Experimentation and Description 57
6.3 Recognition performance 58
CHAPTER SEVEN 61
CONCLUSION AND DIRECTIONS FOR FURTHER STUDY 61

7.1 Conclusion 61
7.2 Recommendation 62
REFERENCES 64
Appendix A: Main Program 67
Appendix B: Finite state of Automata 74
Appendix C: Segmentation and Center of mass 75
Appendix D: Detection and Recognition 77
Appendix E: Finding LD 79
Appendix F: Complete Test Data 80

List of Figures

Figure 2.1: Samples of ESL finger spelling with their seven movements [2] 9
Figure 2.2: The 34 ESL manual alphabet representations of Amharic consonant series [8] 10
Figure 2.3: Representation of tabernacle in ESL using arrows and signed symbols [2] 11
Figure 4.13: Pictorial representation of an automaton [20] 14
Figure 3.1: The desk-based and hat-mounted camera site [15] 17
Figure 4.1 Sample captured video of ESL finger spelling 20
Figure 4.2 Sample frame images of ESL finger spelling from a movie 21
Figure 4.3: Using only Initial seed variable and Initial seed variable plus distance based image selection criteria 26
Figure 4.4: Image selection process from a movie 27
Figure 4.6: Experiment results before and after color subtraction 30
Figure 4.8: Sample segmented images 37
Figure 4.9: Grouping technique using 4-connected neighborhood and 8-connected neighborhood [24] 38
Figure 4.10: Sample image result preprocessing technique 39
Figure 5.1: Pythagoras triangle used for estimation of LDx and LDy 44
Figure 5.3: Finite state of automata used for recognition of the vowels 52
Figure 5.4: Processed first and last frames of the seven vowels of ESL 54
Figure 6.1: GUI Experiment result of the proposed system 58

List of Tables

Table 4.1: Experiment results for different IV values 22
Table 5.1: Representation of ESL vowels by line and direction during signing 46
Table 5.2: Possible input alphabets grouped in to six directions 49
Table 5.3: Symbolize input alphabets by small letters 49
Table 6.1: Experimentation Result of vowels 59
Table 6.2 Experiment Result of each signer 60

List of Algorithms

Algorithm 4.1: Identify key frames relevant to the process 24
Algorithm 4.2: Human skin color subtraction algorithm 31
Algorithm 4.3: Global thresholding Algorithm 35
Algorithm 4.4: Algorithm for isolating hand sign 38
Algorithm 4.5: Algorithm for extract features 40
Algorithm 5.1: Finding distance L_D 43

Acronyms and Abbreviations

AmESL-T	Amharic text to Ethiopian Sign Language translation system
ANN	Artificial Neural Network
ASL	American Sign Language
AVI	Audio Video Interleave
EMA	Ethiopian Manual Alphabet
ESL	Ethiopian Sign Language
FPS	Frames per Second
GF	Gabor Filter
HMM	Hidden Markov Model
KSL	Korean Sign Language
MPG/MPEG	Moving Picture Experts Group
MSL	Malaysia Sign Language
PCA	Principal Component Analysis
RGB	Red, Green and Blue
WHO	World Health Organization
WMV	Windows Media Video

Abstract

In this paper we describe a method for automated recognition of Ethiopian Sign Language (ESL) finger spelling from a video. The method automatically selects images from a given movie. To understand the meaning of the selected images, it applied image preprocessing techniques, global thresholding, grouping neighborhood and calculating center of mass on the selected images. After applying these techniques, the method uses finite state automata to recognize the ESL finger spellings. The method recognizes the seven vowels of ESL.

The method is experimented using the 238 ESL finger spelling and achieves 90.76% recognition performance, through which each of the seven vowels have 34 representation from each of the 34 consonants. As a result, the method is appropriate to recognize the ESL finger spellings integrating with the previous or future works on ESL consonant recognition.

Index Terms—Sign Language Recognition, Finite State Automata and Video Understanding.

CHAPTER ONE
INTRODUCTION

1.1 Background

According to the World Health Organization (WHO) [1], up to tenth percent of the world's population is disabled. And Based on the 1994 Population and housing census, nineteen percent (19%) of the disabled population are assumed to be deaf and hard-of-hearing [2]. Based on this, in Ethiopia, deaf and hard of hearing population are estimated to be 1.52 million.

Sign languages are the basic means of communication between hearing impaired people. It is made up of an organized system of signs. This includes gestures, mimes and facial movements. Sign language is usually used by the deaf people, or the hearing people who can communicate with deaf people. A translator is usually needed when a deaf person wants to communicate with persons that do not speak sign language [3]. However, they cannot depend on interpreters every day in life mainly due to the high costs and the difficulty in finding and scheduling qualified interpreters [4].

According to [4], sign language is not universal. It varies from country to country or regions within countries. Ethiopian Sign Language (ESL) is the sign language of the deaf in Ethiopia. In Ethiopia, sign language began to be used formally after 1960's in connection with the appearance of American and Nordic missionaries who opened school for deaf [2]. For more than 50 years, the foreign sign languages were assimilated with Ethiopian deaf culture and sign language [2].

The Ethiopian National Association of the deaf in cooperation with the Finish Association of the deaf has produced Ethiopian sing language dictionary in 2008 [2]. Ethiopia has its own sign language. However, the Ethiopian Sign Language is not well studied; still it is at

the infant stage of development. A number of research works need to be done in order to address the special needs of the hearing-impaired community of Ethiopia [5].

ESL is complete in both signing and finger spelling. Signing includes the conceptual sign expressions which are dominantly applied to convey meaning in ESL [5]. However, the Ethiopian Sign Language, unlike other sign languages of the world has not yet get the opportunity to be developed, standardized and be a medium of instruction in schools. Because of this, the deaf students could not get access to equal educational opportunity as compared to their hearing counterparts. Besides, there is very small number of deaf students who could get access to join higher institution and develop their professional skill [6].

1.2 Statement of the Problem

Hearing-impaired people usually have communication problems when they want to communicate with hearing people without signing skill. At present, human sign language interpreters are contributing to remove the language barriers between people who are deaf and use sign language and people can hear and speak [5]. According to [4], hearing-impaired people do not depend on interpreters every day in life mainly due to the high costs and the difficulty in finding and scheduling qualified human interpreters. Therefore, it is essential to propose a method for ESL recognition system.

Previously, there was a study on designing and developing software for automatic translation of Amharic text into Ethiopian Sign Language [5] and recognition of Ethiopian Manual Alphabets (EMA) [6]. But, the Amharic text to ESL is one way communication. In addition to this recognition of EMA could not recognize all Ethiopian sign letters with their vowels. Consequently, spelled words are not recognized using the previous study [7]. Hence, in order to allow full communication (text to sign and sign to text), it is essential to include all the 238 finger spellings of the ESL and design a method of understanding of ESL vowels.

This thesis report will have contribution in the study of Ethiopian sign language translator. It introduces a new method on ESL finger spelling recognition.

1.3. Objective of the Study

1.3.1 General Objective

The general objective of the research is to develop a method for recognition of Ethiopian Sign Language finger spellings from a video.

1.3.2 Specific Objectives

The following are the specific objective of the research:

- Reviewing literature and related works
- To investigate algorithm used for selecting few image frames from a given video
- Researching method of segmenting hand sign from images
- Studying method for extracting features
- Examining and designing an automata to convert sign language into corresponding vowel text
- Exploring which frame from a given movie could be used for the recognition of the consonant

1.4 Application and Results

Previously, it has already been developed an automatic translator to convert Amharic text to Ethiopian sign language and Ethiopian Manual Alphabet recognition [5, 7]. This research work is a continuation of the previous work on ESL manual alphabet recognition [7]. The outcome of this study is used to understand vowels of the Ethiopian sign language. Therefore, hearing person can understand the ESL finger spelling by using the application. With this regard, an automated sign language to text translator helps to facilitate communication between deaf and hearing people. Thus, the application will contribute on

minimizing communication obstacles between hearing-impaired people and people who can read and understand text.

The thesis will have a contribution on the research areas of image processing, sign language study and use of finite state automata for image recognition. Moreover, the system has its own role on showing applications of physics and mathematics to image processing like center of mass and Euclidean distance.

1.5 Scope and Limitation of the Study

The thesis will recognize the seven vowels of each of the 34 Ethiopian sign language manual alphabets as we will discuss in Section 2.4 and it recommends which picture frame is used for the recognition of the consonant from set of frames. The video inputs for the system are one movie for each finger spelling. In other term, the study is not covered segmentation of words into individual sign letters from a video. In addition to this, the study will leave out facial expressions, sign words and sentences.

According to [5], implementation of a full translation tool which contains all ESL signs and finger spellings require more time to study and resources from the language side. Hence, the scope of this work is limited to understand the vowels of Ethiopian sign language finger spellings and to recommend which frame is best used for recognition of consonants.

1.6 Methodology

In this thesis, we investigate image processing techniques using Matlab programming language in parallel with the study of ESL finger spellings. Besides, we explore our own algorithms and numbers of Matlab built-in function in order to accomplish the study.

1.6.1 Data collection

The 238 (34*7) Ethiopian Sign Language manual alphabets were collected in terms of video. The data is collected by four signers with black background and white glove. The 34

consonant representation of ESL manual alphabets have a sequence of arrangement the same with the Amharic consonant series, which starts by 'He' {ሀ} and ends by 34th 'Ve' {ቨ}. The signs are captured by a digital camera. The movie for a single sign takes from one to three seconds, which means image frames represent for the ESL finger spelling are within the range of 30 and 90 frames.

1.6.2 Methods

Traditionally, the technology of gesture recognition was divided into two categories, vision-based and glove-based methods. In vision-based methods, computer camera is the input device for observing the information of hand signs [4]. According to [6], the vision-based regards to method used in building the recognition system and could be based on Neural Networks, Adaptive Neuro-Fuzzy inference Systems or Hidden Markov Models (HMM). On the other hand, glove based systems uses input sensors that have a direct contact with the hand. The glove devices worn by a signer and detect a degree of finger-bending and transmit hand posture. On this study, our concern is on the area of vision based method of recognition.

In continuous sign languages, motion detection is essential for recognition of the language. On this research we use center of mass to detect the motion and finite state automata to recognize the movements. In addition, global thresholding and other algorithms are also applied before recognition of the signs.

To achieve the objective of the research, we reviewed a number of international journals, thesis papers on sign language translation system and motion detection; books on computer vision & image processing and finite state automata. In addition to these electronic materials, Ethiopian sign language dictionary and other Ethiopian sign language materials are the basic resources. However, these documented materials are not the only resource for the study but also discussion was conducted with professional sign language

interpreters and students of Ethiopian sign language study. In addition observation on hearing-impaired people was also accomplished.

1.7 Organization of the thesis

This thesis is organized in Seven Chapters. Chapter Two introduces the overview of sign languages, signs and ESL manual alphabets. Chapter Three discusses literature review and related works on sign language recognition mainly on Ethiopian sing language. Chapter Four presents hand segmentation and feature extraction. Chapter Five deals with detail design of the finite states automata used for recognition of finger spellings. Experiment and results of the study are discussed on Chapter Six. Finally conclusions and direction for further study are presented in Chapter Seven.

CHAPTER TWO
OVERVIEW OF SIGN LANGUAGE

2.1 Sign Language

There are different sign languages all over the world, just as there are different spoken languages. These languages are developed in communities of deaf people across different countries. For instance, American and British use common spoken language English. However, American Sign Language and British sing language are different. Each sign language like spoken language, have vocabularies, grammatical structure and spelling (figure-spelling). According to [5] Sign language is a visual language consisting of various signs, gestures, finger spelling and facial expressions. It is developed as a language to meet the need of the deaf people to communicate with each other. Accordingly, sign languages are useful for the deaf community to learn the ways the hearing world understand using spoken language.

2.2 Sign

A sign is a movement of one or both hands and accompanied with facial expression, which corresponds to a specific meaning. Signs can be performed by one or two hands and they are called one-handed or two-handed respectively. In one-handed sign, the hand that performs the sign is always the same hand and is called dominant hand. A sign, in sign language is a movement of the dominant hand or of both hands [3].

Facial expressions add important information to the emotional aspect of the sign; however, in research works that have been developed they are excluded from the area of interest, since its analysis complicates the problem [6]. The term "sign" in sign language most often represents a whole word, unless it is differently clarified by some researchers in their work [3]. Signs can represent common words or meanings; however, they may not work properly for uncommon words like name. In cases of such meanings, it is common to spell it.

2.3 Ethiopian Sing Language (ESL)

In Ethiopia, sign language begun to be used formally after 1960s in connection with the appearance of American and Nordic missionaries who opened schools for the deaf [2]. The missionaries brought the sign language used in their own countries [2]. For more than 50 years, the foreign sign languages were assimilated with Ethiopian deaf culture and sign language [2]. Then after, Ethiopia has adapted its own sign language called the Ethiopian Sign Language (ESL) [2]. ESL consists of finger spellings and sign words developed from deaf community of Ethiopia.

2.4 ESL Finger Spelling

Ethiopian sign language has sign alphabets called Ethiopian manual hand alphabets [8]. These alphabets are used to spell scientific words, names and meanings which don't have single signed words. According to [5], ESL has 33 manual alphabets with their corresponding seven movements for each of the 33 alphabets. In 2009, additional one signed alphabet was set for the Amharic letter 'Ve/ቨ' [5]. Unlike American Sign Language (ASL), ESL finger spellings have movements that change the meaning of the sign letter. According to [9], ASL finger spellings do not require motions for most of the letters. Instead most of the letters are primary distinguished by the hand shape. Conversely, ESL finger spellings represent by hand shape and motion.

ESL finger spelling represents Amharic consonant series with hand configurations, seven movements correspond to the seven Amharic vowel orders [10] (ሀ፣ ሁ፣ ሂ፣ ሃ፣ ሄ፣ ህ፣ ሆ). Figure 2.1 shows the seven movements of ESL finger spellings that correspond to the seven Amharic vowel orders.

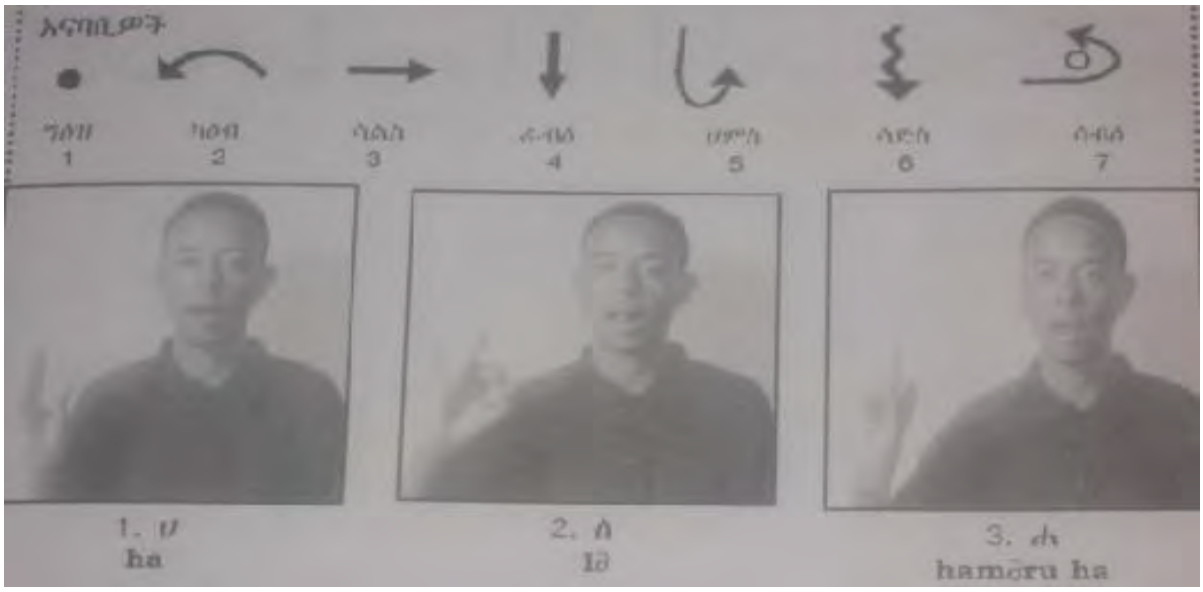


Figure 2.1: Samples of ESL finger spelling with their seven movements [2]

Ethiopian Sign Language finger spelling system uses hand shapes to represent the consonant series of the Amharic syllabary [10]. ESL has thirty-four hand shapes in the inventory of its finger spelling system. Like various other finger spelling systems, some Ethiopian hand shapes bear an iconic resemblance to the orthography of written Amharic [10]. Therefore, ESL manual alphabets are totally different with European, American and other sing language manual alphabets. As we can see from Figure 2.2 the 34 consonants of ESL manual alphabets have an iconic similarity with the Amharic fidel.

According [10], in Amharic there happens to be the first vowel-1 (**ግእዝ**), which changes nothing about the base consonant Fidel; the same thing happen to ESL vowel-1, which adds a null movement to the consonant hand shapes. On the other hand, the six vowels have different movements as shown in Figure2.1.

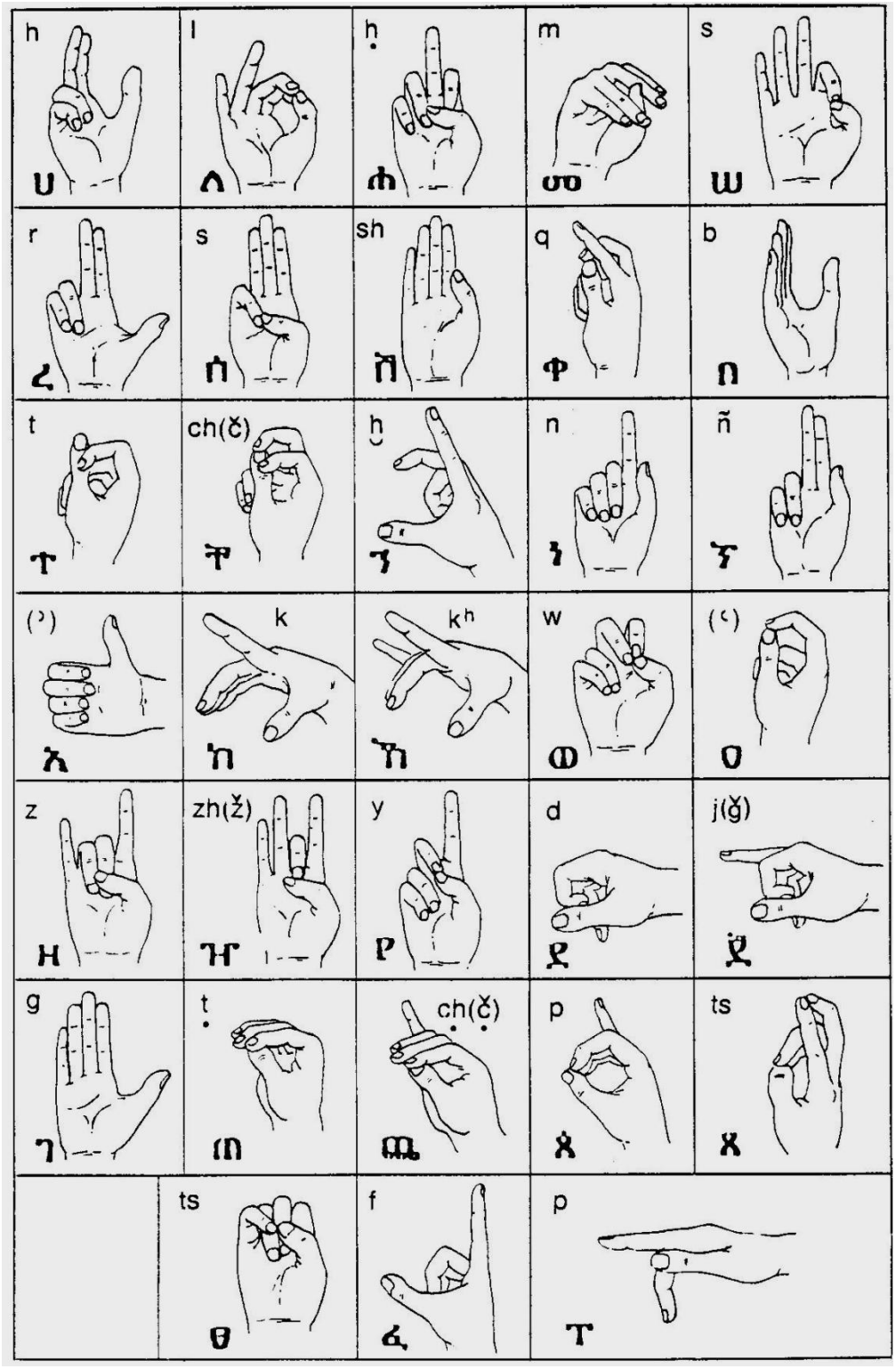


Figure 2.2: The 34 ESL manual alphabet representations of Amharic consonant series [8]

2.5 Representation of ESL

ESL uses signed symbol and arrows to represent the exact meaning of signed alphabets and signed words. The arrow shows how a particular sign can be moved, the location of the sign in relation to the body of the signer and how the sign is moved (slowly or fast).

For instance, Figure 2.3 shows ESL representation using signed symbol and arrow. The Ethiopian traditional way of caring tabernacle while moving to celebrate epiphany is in the head of the priest. Figure 2.3 shows the way of caring tabernacle and its representation is used for tabernacle. The movement of both hands start from the head and ends at shoulder.

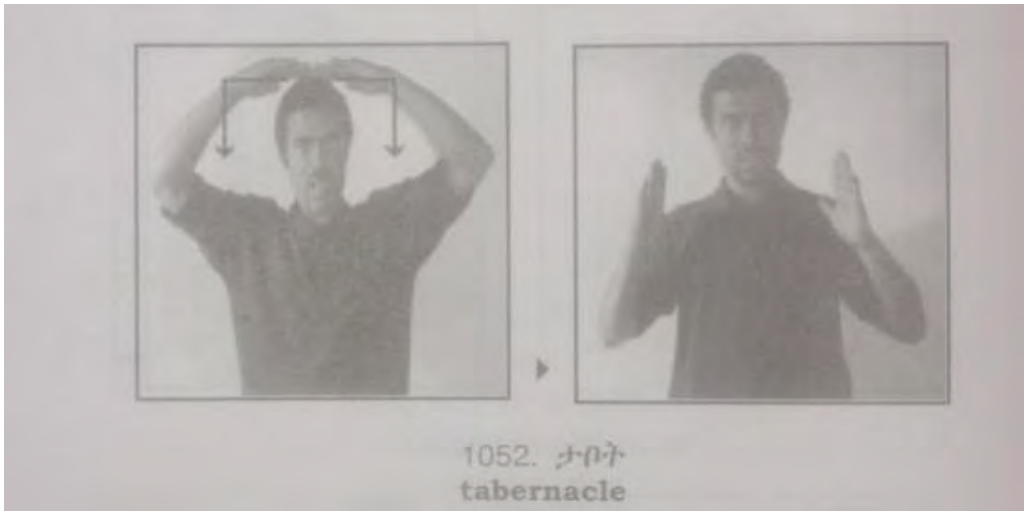


Figure 2.3: Representation of tabernacle in ESL using arrows and signed symbols [2]

CHAPTER THREE
LITRATURE REVIEW AND RELATED WORK

2.1 Literature Review

2.1.1 Methods of signed language recognition

Researches indicates that, automatic sign language recognition systems generally focused on two areas of interest glove-based methods [11, 12] and Visual based methods [6, 7, 13, 14, 15]. These two methods are described below.

a. Glove Based Method

Glove based systems use input sensors that have a direct contact with the hand. According to [11, 12, 15], glove has been commonly used as input device to recognize signs. The glove-based input device, used to measure hand shape or flexion/extension of hand or fingers [12]. In addition, the glove devices worn by a signer detect a degree of finger-bending and transmit hand posture. Although it can sense exactly, there are difficulties such as an uncomfortable wearing and limitation in installation [15].

Most commercial sign language translation systems uses the data-glove method, as it is easy to obtain information about the degree of finger flexing and 3D position of the hand using the gloves [14]. Thus, this system requires less computational power, and real-time translation is much easier to achieve [14]. However, these data gloves are expensive and trying to find cheaper data gloves reduced the amount of sensor which causes loses of important information [14]. Thus, glove based method causes losses of accuracy in translating signs.

b. Vision Based Method

With the recent development of computer vision and computer processing speed, there has been an incredible progress on image recognition. In vision based systems, digital camera is used as an input device to capture images of the signer. Vision based systems have been used Artificial Neural Network (ANN), Hidden Markov Model (HMM) and Image Processing Algorithms approaches [6].

The major advantage of this approach compared to the data-glove approach, is the flexibility of the system. It can be developed to include non-manual signals such as recognition of facial expressions and head movements as well as perform lip-reading [14]. However, its main disadvantage is, it requires a large amount of computation, just to extract the hand position before performing any analysis on the images [11].

2.1.2 Finite States Automata

Basically, finite states automata were used to describe the morphological process of natural language [16]. Nowadays, the method is used for image recognition and image indexing [17, 18, 19]. According to [20, 21] an automaton is an abstract model of a digital computer. As such every automaton includes some essential features.

- It has a mechanism for reading input, the input mechanism can move only from left to right and reads exactly one symbol on each step
- It will be assumed that the input is a string over a given alphabet
- It has a control unit, which is said to be in one of a finite number of "internal states"
- The transitions from one internal state to another are governed by the transition function δ.
- Has a temporary "storage" device,
- The automaton can also produce output of some form; an automaton whose output response is "yes" or "No" is called an "Acceptor".

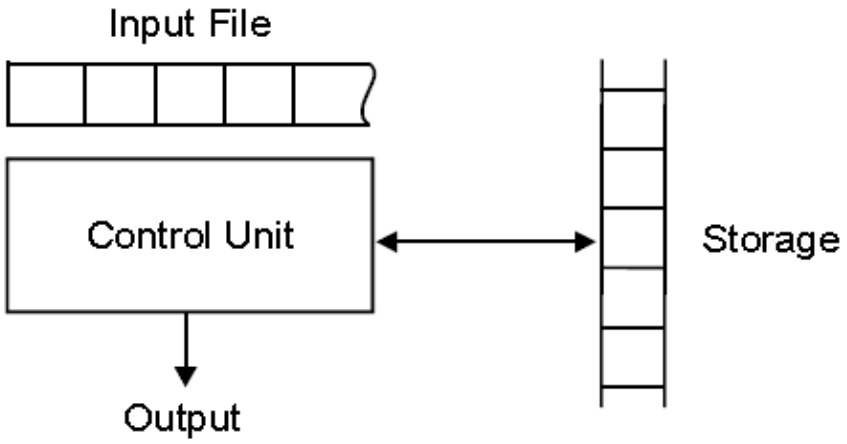


Figure 4.13: Pictorial representation of an automaton [20]

Automata can be deterministic or non deterministic [21]. A deterministic automata is one in which each move (transition from one state to another) is unequally determined by the current configuration [20].If the internal state, input and contents of the storage are known, it is possible to predict the future behavior of the automaton [20]. This is said to be deterministic automata otherwise it is nondeterministic automata [20]. Nor determinism means a choice of moves for an automaton or allows a set of possible moves [21]. We explained a summary of the basic theory of finite state automata. For those who are interested to explore more, details are presented in books of theory of automata, formal language and computation [20] and an introduction to formal language and automata [21].

2.2 Related Works

There have been a significant number of published papers on different types of sign language recognition and related topics. However, there are limited numbers of researches on Ethiopian Sign Language. In this topic, we will see the approaches and results of some chosen examples from different types of sign language recognition and thesis papers on Ethiopian sign language. From these different studies on different sign language we obtained some concepts and techniques to our study on ESL recognition.

2.2.1 Ethiopian Sign Language Recognition Using Artificial Neural Network

In the paper [6] the authors introduce, a hand gesture detection and recognition system for Ethiopian Sign Language (ESL). The system has been verified using 34 letters of the Ethiopian Manual Alphabets (EMA) [7]. The system uses five samples, and integrate 34 Ethiopian Manual Alphabet (EMA), totaling up to 170 images. The system also use white glove while collecting the data from ESL students. Moreover, the paper [7] describes non EMA images were collected from websites and the system is capable of rejecting these non EMA images.

Hand gestures detection and recognition system for EMA was developed using Gabor Filter (GF) together with Principal Component Analysis (PCA) for feature extraction and Artificial Neural Network (ANN) for recognizing the ESL of 34 letters of EMA [7]. The experiments conducted and proved that with sufficient data, ANN approach produces very good results and able to recognize an unknown input sign (non EMA) very fast [7].

However, the paper could not provide full translation functionality. The signed image taken was representing for the consonant ESL. It also uses isolated signed hands. These images could not be used to recognize finger spelling with their vowels.

2.2.2 Automatic translation of Amharic text to Ethiopian sign language

The paper presented on automatic translation of Amharic text to Ethiopian Sign Language (ESL) in other word simulating the Amharic text to ESL. The paper studied both Amharic and ESL grammatical structure. The paper also designed and developed Amharic text to ESL translation algorithm.

The authors developed an automatic translator, which translates Amharic text into Ethiopian sign language. Using the translator, it is possible to translate Amharic texts into gestures that a hearing impaired can understand. The tool is useful in enabling people who

don't know sign language to communicate with deaf individuals. The hearing-impaired can also utilize the software to develop written language skills [5].

Amharic text to Ethiopian Sign Language translation system (AmESL-T) was tested and evaluated in terms of the translation accuracy and user satisfaction. The evaluation results show that, the translation accuracy and the quality of the signing are acceptable and it satisfies the users' needs [5]. Finally, the author recommends on having a two way communications and it is better to design and develop ESL recognition system.

2.2.3 Continuous Gesture Recognition System for Korean Sign Language based on Fuzzy Logic and Hidden Markov Model

The paper studied on continuous Korean Sign Language (KSL) recognition using color vision. The study focuses on sentence-based recognition system. Because it is sentence-based recognition, the signer doesn't need to pause between sign words. Due to difficulty of segmenting continuous sign in to individual sign words, they disassembled the KSL into 18 hand motion classes according to their patterns and represent the sign words as some combination of hand motions or directions. They also try to reject unintentional gestures motions, such us preparatory motion and meaningless movement between sign words using fuzzy portioning and state of automata.

Continuous KSL include several KSL sentences and each sentence has a set of KSL words [15]. Before recognition of Korean sign language, the paper uses two kinds of gesture segmentation. The first one is segmenting compound or complex sentence into individual sentences, they call it sentence segmentation. Another is word segmentation that segments words from a single or simple KSL sentence. To recognize 18 hand motion classes they adopt Hidden Markov Model (HMM). Using these methods, the system recognizes 15 KSL sentences and obtain 94% recognition ratio [15]. However, the system works only for continuous KSL.

2.2.4 Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video

American Sign Language (ASL) uses approximately 6000 gestures for common words and finger spelling for communicating obscure words or proper nouns [13]. The paper states that, there was number of previous study to recognize those gestures. However, they focused on isolated signs or finger spelling. The paper studied on two real-time hidden Markov model-based systems for recognizing sentence-level continuous ASL. The first system observed the hands from desktop mounted camera and the second system observed from users hat-mounted camera. Both systems use a single camera to track the user's hands. The study uses a four state HMM with one skip transition for both systems. The figure in Figure 3.1a and Figure 3.1b shows the camera position of the two mounted locations [15].

a) View from the desk-based tracking camera.

b) The hat-mounted camera, pointed downward toward the hands, and the corresponding view.

Figure 3.1: The desk-based and hat-mounted camera site [15]

The system tested using 40 words lexicon. The first system, which observes the user from desktop mounted camera achieved 92% accuracy and second which observe the user from hat mounted camera achieved 97% accuracy.

2.2.5 Real-Time Malaysian Sign Language Translation using Color Segmentation and Neural Network

The paper presented an automatic visual-based sign language translation system. The automatic sign-language translator translates Malaysia Sign Language (MSL) in to English. According to [14], researches done before this study focused on finger spelling. On the other hand this study could recognize both finger spelling and sign gestures for both static and motion signs. The translator can translate alphabets, numbers and a few words from the MSL. The set of words chosen in the study are these commonly used words like he, I, you, close and telephone that can generate a simple sentence.

The study uses Logitech webcam to captured Malaysian Sign Language; this is good choice for practical applications because webcam can be widely available and easily obtained. It also uses colored gloves for both left and right hand with different colors for finger tips and palm. This is advantageous to extract data using color segmentation. The system uses five frames per second (fps) with frame resolution of 352x288 pixels. For the recognition, they used two-layer feed forward neural network. To train the sign database, they used resilient back propagation rule. The system applies three neural networks to recognize alphabet, number and word signs separately and achieved the recognition rate of over 90%.

Generally, the above related works [7, 13, 14, 15] use artificial neural network or hidden Marcov Model to recognize the different sign language across different countries. These papers focused on recognition of isolated signs by training the sign database. They used two different data used for train and test.

However, it is advantageous if it is developed a mechanism to recognize without training sign data. This possibly increase the processing time and do not need to have storage for the training data. In this study, we are going to understand the motion of signs. For this case we don't need to store any data for training and we don't need to cross check with some stored data. In order to understand these movements, it is necessary to detect the motion of the frames. We used a single point from a frame to detect the motion and finite state of automata to understand the motion.

CHAPTER FOUR
SIGNED HAND SEGMENTATION AND FEATURE EXTRACTION

This chapter describes the detail techniques on how signed hand is segmented and what features will be extracted. To segment the signed hand and to extract features, we use four basic steps. These are video acquisition, frame selection, signed hand segmentation, and feature extraction.

4.1 Video Acquisition

Captured video of Ethiopian sign language finger spellings are first acquired by the system. The system takes each movie from a file. Figure 4.1 shows sample video input to the system.

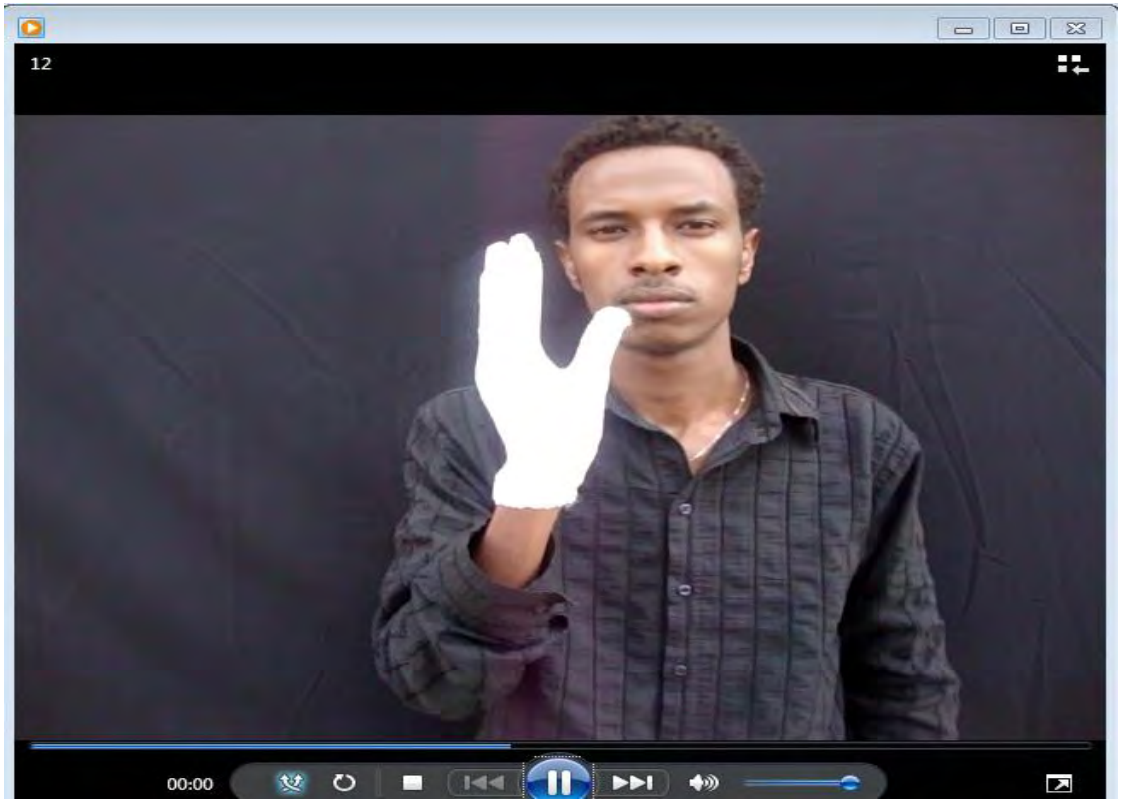


Figure 4.1 Sample captured video of ESL finger spelling

4.2 Image frames Selection

It is obvious that, movie is a sequence of images over time. For instance, the video represented in Figure 4.1 have 60 frames or images. Some of the frames from the above video are depicted in Figure 4.2. However, all the frames in the video will not use for recognition of the ESL finger spelling. Therefore, the next step is identifying key frames from a set of frames in a given movie.

Figure 4.2 Sample frame images of ESL finger spelling from a movie

Processing each frame from the movie is not required for the recognition of the ESL finger spellings. In addition to this, processing and analyzing all frames cause delay of processing time and unwanted space usage. Therefore, we have to select smaller number of image

frames than the actual number of frames in a given video. For this reason, we use a variable called *initial seed variable (IV)* for selecting sequence of images and we also applied a distance based selection technique as well. Initial seed variable have a scalar value and it is used to skip frames from the video during processing. It is useful to minimize the processing time. The value for the variable possibly can be one of the numbers {1, 2, 3… 9}. However, the default value is 9; this is the best value to skip number of frames from the movie. The value of the initial variable is limited to the above set. This is because, the required points (center of masses from frames) are four as we discussed latter on Section 5.1.2 and in our data we incorporate a one second length. Therefore, if we try to skip fames more than 9, it can produced error because we may select frame out of the total number of frames. The experiment result for the initial variable is depicted on Table 4.1. We experiment 56 videos to represent the seven vowels and we obtain the best recognition result when IV is 6, 7, 8, and 9.

Table 4.1: Experiment results for different IV values

	Vowel 1	Vowel 2	Vowel 3	Vowel 4	Vowel 5	Vowel 6	Vowel 7	Total
IV=1	8	7	8	7	7	4	5	46
IV=2	8	7	8	7	7	4	4	45
IV=3	8	7	8	7	7	4	5	46
IV=4	8	7	8	7	7	4	4	45
IV=5	8	7	8	7	7	5	5	47
IV=6	8	8	8	7	7	6	4	48
IV=7	8	8	8	7	7	5	5	48
IV=8	8	8	8	7	7	6	4	48
IV=9	8	8	8	7	7	5	5	48

However, the initial seed variable may not detect the motion alone. This is because of the variation in speed during signing in a single video. Hence, we used a distance based selection technique as we discussed on motion detection section on Section 5.1.1. Algorithm 4.1 describes the general frame selection algorithm. The algorithm select key frames to process. In order to make the algorithm easy to understand it is better to see these terms and ideas first. These helps, to make in mind some terms and ideas are explained in detail letter.

- *Center of mass:* This is the central value of a block. The block is obtained from a frame and particularly it is the sign hand. The detail how to get the block from the frame and how to compute center of mass is presented on Section 4.3 and Section 4.4.
- *CMI:* It is a variable and it is used as an index to store center of masses of different frames.
- *CenterOfMass:* is an array used to store different points or center of masses from different frames.
- *CofM()*: It is a function used to compute center mass of a block, from a frame
- *Frame (I)*: It is a function used to select a frame on index I
- *Distance ()*: It is a *function* used to compute the Euclidean distance between two points
- *LD*: It is a variable and it is used to store the length. This LD is important in order to assume whether the motion is detected or not. The detail how we obtained the value for LD and the usefulness of the variable is presented on recognition Chapter Section 5.1.1.
- *Recognition step*: It compute the recognition of vowel using the selected frames and it is presented on the recognition chapter Section 5.1. After applying IV and Distance based selection technique, we cross check whether the vowel is recognized or not. If

the system can't recognize using the default value of IV then it increases its value and tries to select other frames.

- *//* : Describes commented lines

Algorithm 4.1: *Identify key frames relevant to the process*

Input: Video recorded Ethiopian sign language finger spellings

Output: key frames for processing

1. *Set value for pixel length , LD* ***//****when the distance between two frames is greater than DP, the assumption is there is a motion between the frames*
2. *Set value for initial seed variable ,IV* ***//*** *IVis Used to select the next frame*
3. *Accept Video input, Vinput // Using Matlab function* `Vinput= mmreader(s);`
4. *Obtain total number of frames from the movie, TFrame // Using Matlab function //*`TFrame = Vinput.NumberOfFrames;`
5. *Select the first Frame image , F1 // first frame of Vinput*
6. *CMI→1* ***//*** *index used to store center of mass*
7. *Calculate center of mass of the first frame and store on CenterOfMass (1) , CenterOfMass (CMI)→CofM (CMI)* ***//*** *always stored and used as a first point*
8. *Access the next frame Image on index I→1+IV,*
9. *CMI→CMI+1*
10. ***If*** *(I>TFrame) Then Go to step 19* ***//*** *Index frame I, greater than Total frame*
11. ***Else*** *Continue to step 12*
    12. *Compute center of mass of the Frame on index I, CenterOfMass (CMI)→ CofM(Frame(I))*
    13. *D=Distance (CenterOfMass (CMI-1), CenterOfMass (CMI)) // Compute the //Euclidean distance D between adjacent Center of masses*
    14. ***If*** *(D>LD) Then* ***//*** *distance, D greater than motion detection pixel size, DP Go to Step 9, by changing the value of I, I→I+IV* ***//*** *If motion is detected //then finds another center of mass*
    15. ***Else***

Go to Step 10, by changing the value of I, I→I+1 // check if it increase the distance by accessing the next second frame

End if

End If

16. Recognition step

*17. **If (**Recognition of Vowel = True) Then Go to Step 19*

*18. **Else***

IV→IV+1 // try to recognize by increasing the value of the seed point, IV

***If** IV>=TFrame/3 Then*

Go to Step 19 // because it causes out of Total Frame

Else

Go to Step 5

End if

End if

19. Stop

To describe the above algorithm, the first image from the set of frames in a given video processed first. This image is possibly used for the recognition of the ESL consonants and it also uses as the first point selected for the recognition of vowels. The consonants can be recognized using the previous study on ESL recognition [7]. The next image is in the index *1+ initial seed variable*. The rest frames can be selected for process, using initial seed variable or by checking the distance among the previous selected frames. If the distance between previous two adjacent points satisfies the condition for motion occurrence, it uses initial seed variable to select the next frame, otherwise it uses the distance based selection technique.

Distance between selected adjacent points may vary if we only consider the initial seed variable. This happened, because of the speed in a given movie is different. As a result, the system may process frames in the same location or nearby. Distance based selection is

essential for having points in different location and approximately uniform distance between points. Three similar distances between adjacent points are used for the recognition of the ESL vowels. This is discussed in the feature extraction section, Section 4.4 in detail.

The frame selection process is used distance based selection technique, if the initial variable will not make a motion. This process can continue by selecting the next frame from the current indexed frame. This process persists until the motion detection assumption is satisfied. The usefulness of initial seed variable and distance based selection is depicted in the Figure 4.3. In the Figure 4.3 a, the distance between adjacent points didn't have uniform distance. This is because of the difference in velocity of the signer in a given movie. To make this difference uniform, we use distance based selection technique. In Figure 4.3 b, we observe uniform distance between selected points which represent frames.

a) Using only IV b) Using IV and Distance based

Figure 4.3: Using only Initial seed variable and Initial seed variable plus distance based image selection criteria

Distance based selection uses two adjacent points from two different frames. We find these two points by passing different image processing and analysis techniques like

preprocessing, and segmentation of signed hand. As an initial, we use frame *one* and *frame 1+ initial seed variable*. Then distance between two points p1(x_1, y1), and p2(x_2, y2) is computed using Euclidean distance in Equation 4.1.

$$D = \sqrt{(x2 - x1)^2 + (y2 - y1)^2} \qquad 4.1$$

If the distance between two points satisfies the set condition, it continues to find another point otherwise it calculates the distance with the next frame. This process will continue until the end of the number of frames in a given movie. As we can see from Figure 4.4, the red color circles are points which are used for the recognition purpose. These circles are points, which satisfy the condition. The blue color circles in Figure 4.4 show how it processed when the initial variable does not meet the condition. For instance, if we consider the fifth red point counting from the right side, it processed only one frame after the initial variable set. However, in the second point there are a number of frames were processed. This is due to different velocity of the signer in a single movie for a sign.

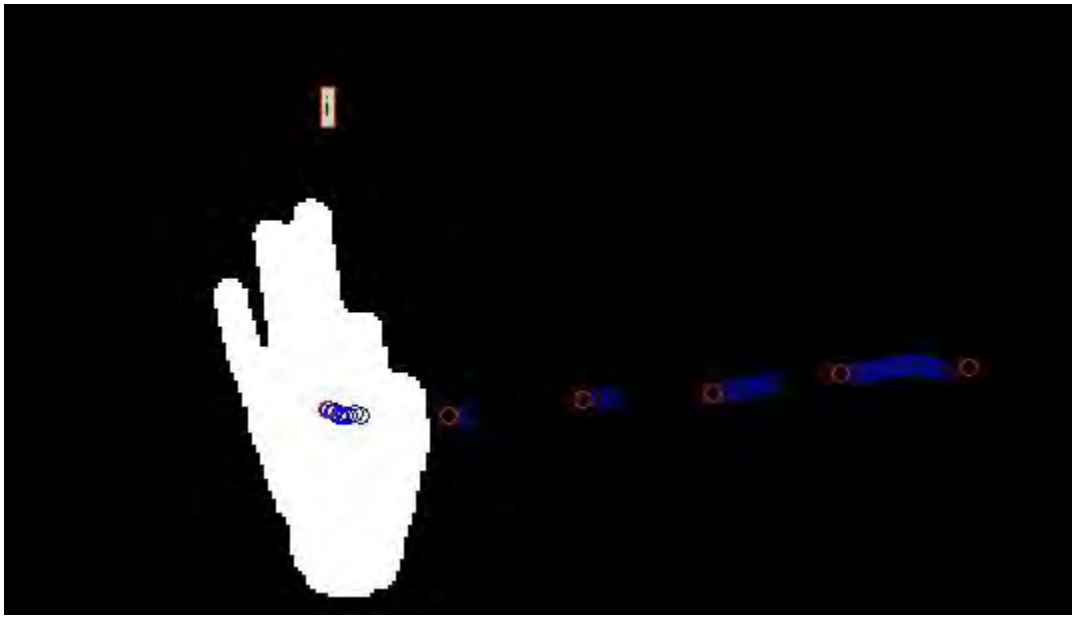


Figure 4.4: Image selection process from a movie

Finally, the initial variable probably iterates and increases its value. This happened, when all frames in the entire movie were processed and not yet recognized. This process may iterate until the condition *IV>Tframe/3* satisfies. This is due to the need of four points for recognition of the ESL as we discussed in Section 5.1.2 required points for recognition. If IV>*Tframe/3* is true, it produces error. This is because we can't select a frame, if the index is greater than the total number of frames in a movie. For instance, let's have a signed movie of one second which has 30 frames. If IV is 9, the possible frames used to find the four points can be frame 1, frame 10 frame 19 and frame 28. However, if IV is greater than or equals to 10, the fourth frame used to find the fourth point will not be in the range. For example, if IV is 10 then possible frames used to find the points are frame 1, frame 11, frame 21, and frame 31. This produced error because frame 31 is not found.

4.3 Signed Hand Segmentation

4.3.1 Preprocessing of images

In this study, the use of preprocessing is to prepare the frames easy for segmenting and isolating sign hand. In the frame selection section, movies are processed and selected image or frame was passed for the next step which is preprocessing of images. The selected images have colored image (combination of Red, Green, and Blue). Hence, every selected image should pass this step and this is essential for the segmentation of the sign hand. On this step, we have the following preprocessing procedures depicted in Figure 4.5.

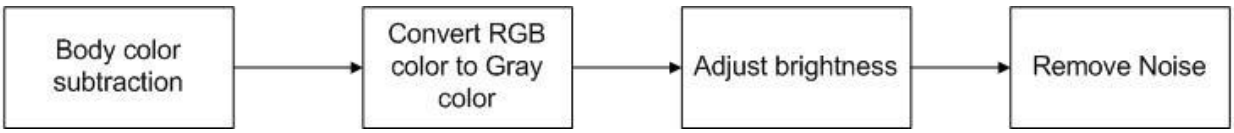


Figure 4.5: Preprocessing image procedures

a. Body color subtraction

As we can see from Figure 4.7a, the face and some part of hands of the signer have high RGB values (close to 255). The use of body color subtraction in this study is, to differentiate the property of the sign hand with other part of the body. The reason behind this concept is, our threshold value used to categorize the image is using a single threshold value (Section 4.3.2). This threshold value characterized the image in to two. Consequently, when this colored image is converted directly to gray color; part of the body have similarity with the signed hand which is difficult for the segmentation of signed hand. Accordingly, the signed hand that wears the white glove has each RGB value near to 255. If we are applying segmentation in the gray scale, the result includes the face and some body of the signer attached with the signed hand. Therefore, the new method subtracts the skin color from the entire image. This process is done using the Algorithm 4.2. The segmentation results of the four signers before and after color subtraction is depicted on Figure 4.6. The segmentation result before skin color subtraction indicates that, the parts of the body have similar property with the sign hand that wears white glove.

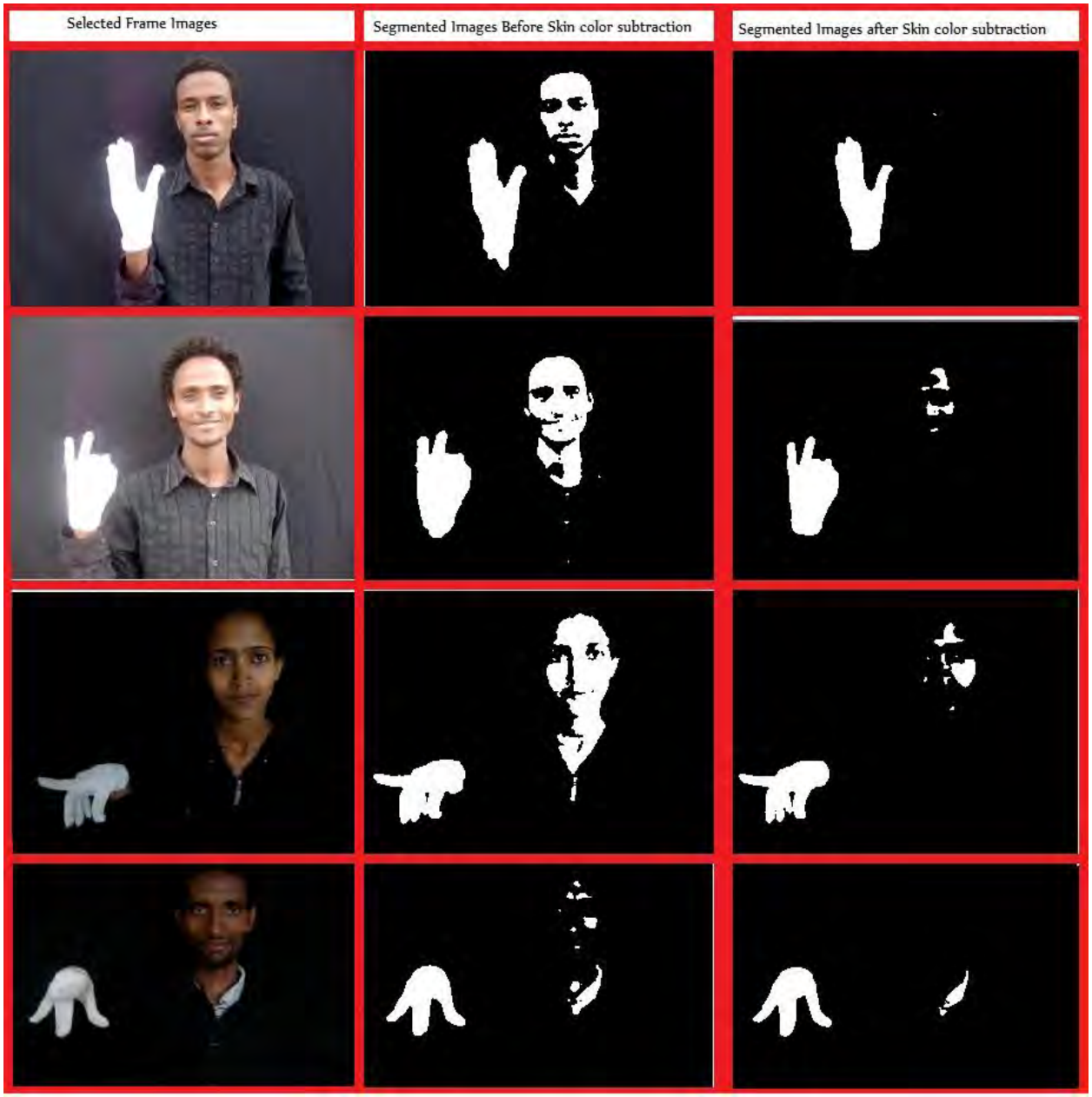


Figure 4.6: Experiment results before and after color subtraction

To subtract the skin color we obtain a single value from the face color. To detect the face color we use Macromedia Dreamweaver application software. Basically, it is possible to use any other application software to obtain the skin color. For example, the RGB skin color value we used to our data are *[93, 57, 55]* and *[242, 214, 192].*

Algorithm 4.2: *Human skin color subtraction algorithm*

Input: Colored or gray image

Output: Skin colored subtracted image

f ➔Read_image;

if *f is RGB colored image then*

r ➔ f(red); //have the same height and width with image f

g ➔ f(green)

b ➔ f(blue)

[R G B] ➔human Skin color; //RGB have scalar values

f ➔f(r-R,g-G,b-B); // each scalar value is subtracted from corresponding color value image

else

grayvalue ➔ sum of weighted value of RGB color of the face color

f ➔f-grayvalue

end if

After each RGB human skin color subtraction is done, the three different images are combined together. But if the images are grayscale the system subtracts the gray value of the human face color. We find the constant value of the gray image from Equation 4.2.

Figure 4.7b is the result of the color subtraction of the Figure 4.7a. In the figure, we see some part of the face is still appeared but with low value of RGB values. This happened, not because of the existence of white face rather it is the light effect with human skin color.

To convert RGB image to gray image, each RGB value has some weighted factor and adding them to have a single value. We convert RGB values to grayscale values by forming a weighted sum of the R, G, and B components. So, each pixel value in the gray scale

image is determined using the Equation 4.2 taken from the rgb2gray Matlab function. Figure 4.7c, is the corresponding gray level image for the true color image in Figure 4.7b.

$$P = 0.2989 \times \mathrm{R} + 0.5870 \times \mathrm{G} + 0.1140 \times \mathrm{B} \qquad 4.2$$

b. Adjust brightness

As we can see in Figure 4.7c, after subtracting and converting to gray level, the image result is too dark. For better recognition we adjust the brightness. According to [7] image adjustment is usually done on too dark or on too bright images in order to enhance image quality and to improve signed hand recognition performance. As a result, some important features become more visible

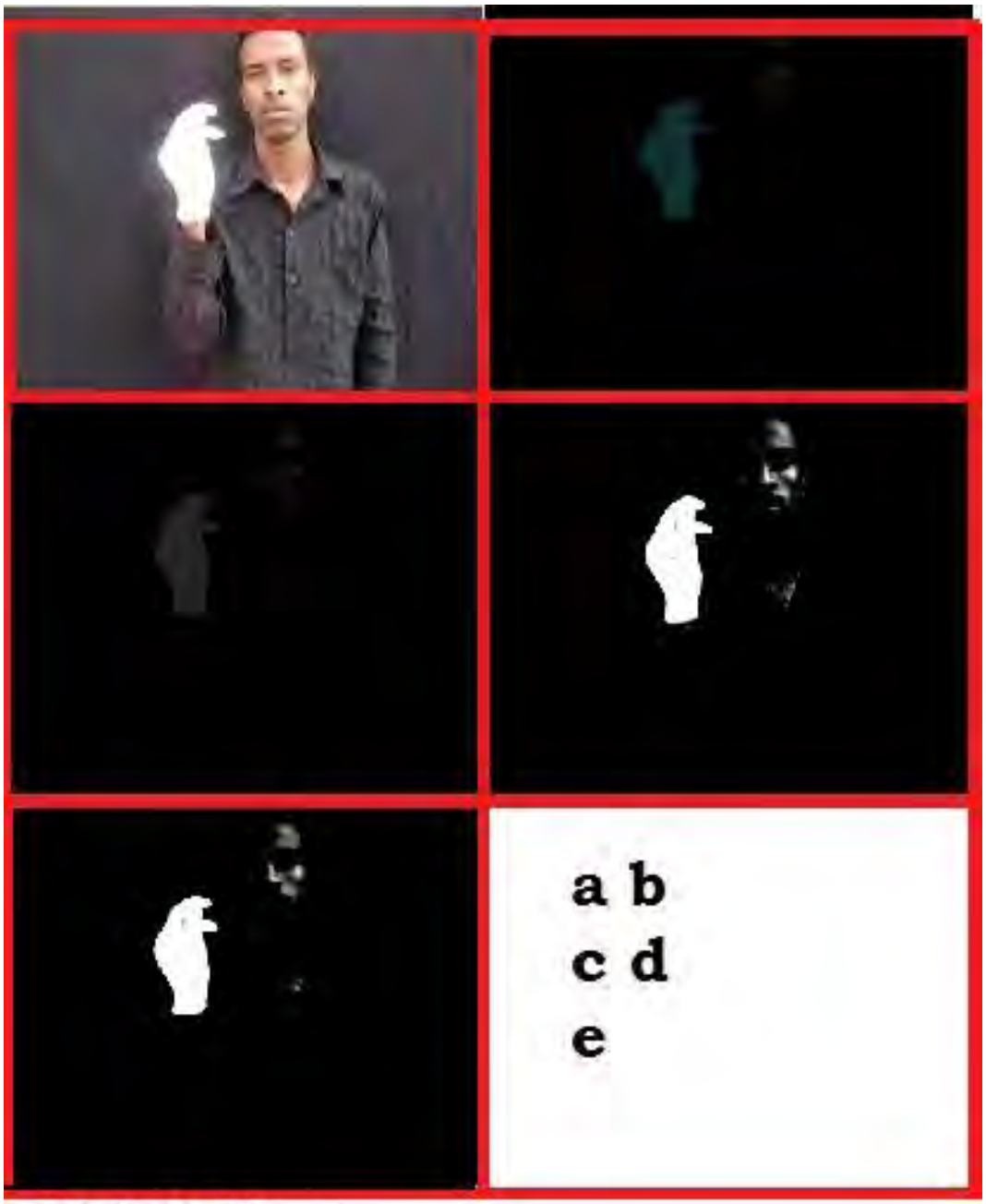
Intensity adjustment is a technique for mapping an image intensity values to a new range [23]. In our system, we use *imadjust* function to increase the contrast of the image. In Matlab *imadjust* uses to adjust intensity and the default range is *[0, 1].* In order to determine the actual value to use it will be multiplied by 255. Figure 4.7d, shows the result of Figure 4.7c after image adjustment.

c. Filter using median filter

The Median Filter block replaces the central value of an M-*by*-N neighborhood with its median value. Median filtering is a nonlinear operation often used in image processing to reduce "salt and pepper" noise [24].

After adjusting the brightness of the image in Figure 4.7d, we observe white noises that can affect the recognition of the required object. These small white particles possibly bond with other big objects, and this may cause to create large number of pixels than the signed hand. Hence, during segmenting the signed hand, the system may choose the wrong object. In addition to this, we observe black noise inside the signed hand. This may cause inaccuracy during matching with the consonant using the previous work ESL recognition

[7] or probably cause side effect on finding the center of mass. For this reason, median filter is our choice for filtering the system. In addition, as median filter minimally degraded edges and lines in the image, it is useful for breaking lines across the face and removes black noises in the signed hand. The median filter matrix used for our system is 3-by-3 neighborhood. Figure 4.7e shows after filtered by median filter.



a) Selected image
b) Color subtracted image
c) Converted to gray color image
d) Adjusted brightness
e) Noise filtered image

Figure 4.7: Sample preprocessing output images

4.3.2 Segmentation Using Global Thresholding

Because of its intuitive properties and simplicity of implementation, image thresholding enjoys a central position in applications of image segmentation [25]. A grayscale image is turned into a binary (black and white) image by first choosing a grey level threshold value (T) from the original image, and then turning every pixel black or white according to whether its grey value is greater than or less than T [26]. We already have a gray image from the prior steps, and now this image should be converted to binary image. Suppose that, the gray level image f(x, y) composed of lighted objects on dark background. In order to isolate the lighted objects from the background, we should set some value for T. And using that value it can be grouped in to two using Equation 4.3. The Value for T is obtained using Algorithm 4.3.

$$g(x,y) = \begin{cases} 1 & if\ f(x,y) \geq T \\ 0 & if\ f(x,y) < T \end{cases} \qquad 4.3$$

The system segments the images into background and foreground using threshold value T. However, choosing threshold value using visual inspection of image histogram or try and error is ineffectual due to the nature of non interactive system. In addition, the chosen threshold value should be used for the entire image. Moreover, the system also needs an automatic choosing of threshold values. Therefore, the system utilizes an algorithm from Gonzalez and Woods [2002] for selecting threshold value iteratively and automatically for global use. The algorithm is described in Algorithm 4.3.

Algorithm 4.3: *Global thresholding Algorithm*
Input: *Gray image f(x, y)*

Output: *Binary image*

1. Select an initial estimate for T (initial estimation is the midpoint between the minimum and maximum intensity values in the gray image, f(x, y))

2. Segment the image using T. This will produce two groups of pixel, G_1 consisting of all pixel values with intensity value f(x, y)>=T and G_2 consisting of pixel with values f(x, y) <T

3. Compute the average intensity values u_1 and u_2 for the pixel in the regions G_1 and G_2

4. Compute a new threshold value, T_n

$$T_n = \frac{(u_1 + u_2)}{2}$$

5. $T_s \rightarrow$ abs ($T - T_n$)

6. $T \rightarrow T_n$

7. Repeat step 2 through 4 until the difference in T, which (T_s) in successive iteration is smaller than a predefined parameter T_0 // Default value for $T_0 = 0.5$

Now using the algorithm, we convert the images in to binary images, in other words the image converted in to foreground value 1 and background value of 0. However, after segmenting the image using global thresholding, the foreground image contains objects (noises) less than the size of signed hand. Samples of segmented image with some noise are depicted in Figure 4.8. As we can see from Figure 4.8a, there are five objects exclusive to the signed hand. In order to extract information from the hand signed image only, we required a technique to remove these unnecessary particles or objects. In other term, we can apply a technique for isolating signed hand.

Figure 4.8: Sample segmented images

4.3.3 Isolating Signed Hand

Figure 4.8 shows the result of segmentation using global thresholding. On the other hand, the segmentation process using global thresholding couldn't isolate the required sign hand. This is occurred since the light effect with the face color makes bright and have high RGB value. Not only the light effect but also small white objects with the signer like cloth buttons and necklace could cause the segmentation process not satisfied yet. However, after segmentation, every created white object with the signed hand is smaller in area size. For this reason, we need a post processing technique for removing these objects; which are smaller in area size than the hand signed.

We can't remove all these unnecessary objects by the common filtering techniques like median filter used in the preprocessing part of this system. To isolate the signed hand from other objects, it is required to group connected objects in the entire image. On the other hand, grouping connected objects is better segmentation technique if we used 4-connected neighborhood. Grouping 4-connected neighborhood is useful to isolate objects connected

by the corner side of the adjacent pixel. Figure 4.9a and Figure 4.9b, show grouping technique for 4-connected neighborhood and 8-connected neighborhood respectively. After grouping each different object from the entire image, the largest object in area is selected for the next process. This means, the connected block with maximum number of pixels are selected as a sign hand. The algorithm used to segment this object is here in Algorithm 4.4.

Algorithm 4.4: *Algorithm for isolating hand sign*

Input: Segmented image using global thresholding, it has other objects with the sign hand

Output: Isolated sign hand image

1. *Get image from segmentation process*
2. *Group connected objects using 4-connected neighborhood*
3. *Calculate pixel size of the regions*
4. *Select the largest area*
5. *Put the selected area on its location with its previous frame size*

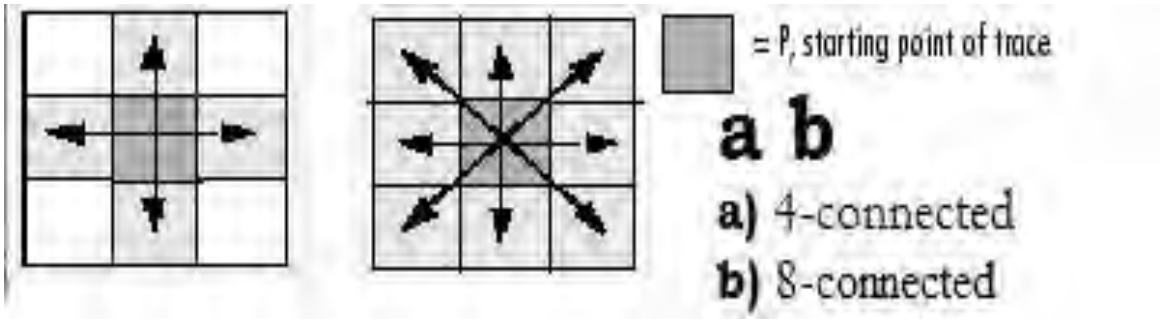


Figure 4.9: Grouping technique using 4-connected neighborhood and 8-connected neighborhood [24]

After selecting the largest area which is the sign hand, all unnecessary objects in the entire image is removed. Only the signed hand is presented with its previous frame size. Figure 4.10a, and Figure 4.10b shows before and after post processing respectively. The use of putting the image to the previous frame size on its location is useful for the recognition of

vowels of ESL. However, the selected image used for the consonant recognition can be resized according to the previous study on ESL recognition [7].

a) Before Isolating sign hand b) After isolated sign hand

Figure 4.10: Sample image result preprocessing technique

4.4 Feature Extraction

This step describes how to extract essential information from the selected images. This information is used to understand the sequence of images or in order to detect the different motion from a given videos. From the segmented signed hand, the vital information is a single dot which is the center mass of the object.

a. Computing Center of mass

To detect the motion of the frames, it is necessary to take a reference point of each segmented image. Therefore, we use center of mass of the segmented sign hand. The center of mass is the mean location of all the masses in a given environment. In other words, the center of mass is the point at which you can balance all the objects. The law of center of mass is defined by:

$$R = \frac{\sum_{i=1}^{n} MiRi}{\sum_{i=1}^{n} Mi} \qquad 4.4$$

Where M_i is for objects or masses and R_i is distance

In our case, the objects that we are assuming as individual masses are the number of pixels from an entire image. In addition, every object in the given image has one of the two values, which is the foreground weight of 1 and background weight 0. The objects also represented in a two dimension plane, x and y. For two dimensional image f, the pixels values are represented by f(x, y). Therefore, the mean location for R_x and R_y are:

$$R_x = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} i*f(i,j)}{\sum_{i=1}^{m}\sum_{j=1}^{n} f(i,j)} \quad 4.5$$

$$R_y = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} j*f(i,j)}{\sum_{i=1}^{m}\sum_{j=1}^{n} f(i,j)} \quad 4.6$$

Therefore the center of mass is:

$$R = f(R_x, R_y) \quad 4.7$$

Based on Equation 4.7, the algorithm used for finding center of mass is presented in Algorithm 4.5.

Algorithm 4.5: *Algorithm for extract features*

Input: Isolated signed hand, f (x, y)

Output: Point that represents the center mass of the sign hand

1. *Read image f*
2. $R_x \rightarrow 0$, $R_y \rightarrow 0$, $n \rightarrow 0$
3. *Find w →Width of f*
4. *Find h →Height of f*
5. *For i=1 to w*

6. *For j=1 to h*

7. *If f(i, j) →foreground image // if 1*

8. $R_x = \rightarrow R_x + i$

9. $R_y \rightarrow R_y + j$

10. $n \rightarrow n+1$

11. *End if*

12. *End for*

13. *End for*

14. $R_x \rightarrow R_x / n$

15. $R_y \rightarrow R_y / n$

16. $R \rightarrow f(R_x, R_y)$

Some of the center of masses from the set of frames for the ESL manual alphabets is depicted in Figure 4.11.

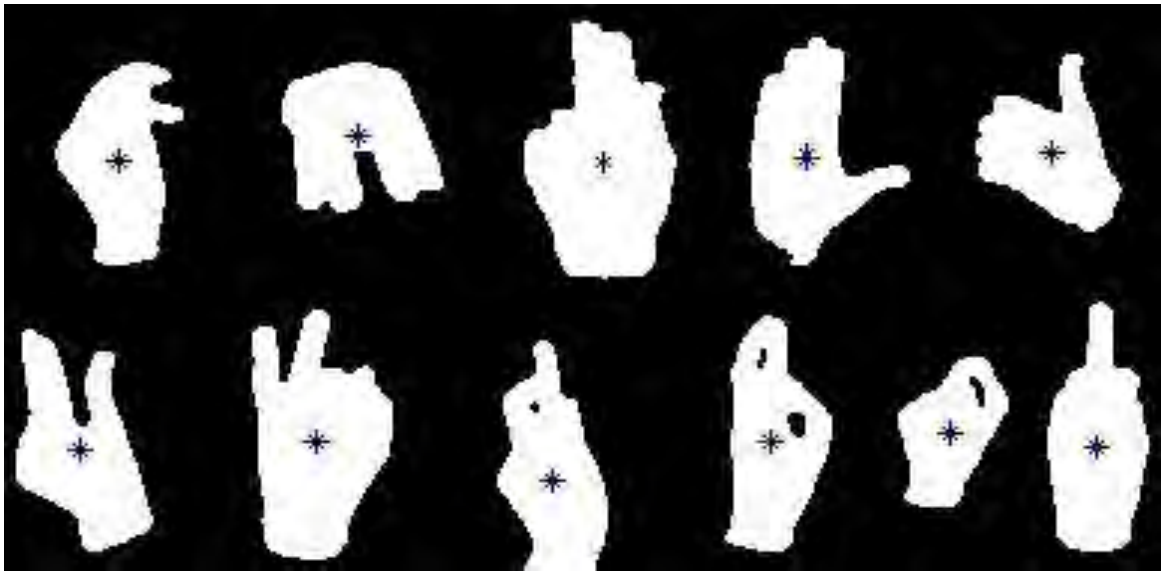


Figure 4.11: sample segmented signs with their center of masses

CHAPTER FIVE

RECOGNITION

After having point data from the video, it is necessary to design on how to understand the sequence of images using the representation of the frames. Accordingly, the method understands the vowel from the motion and it recommends which frame is useful for the recognition of the consonants.

5.1 Vowel Recognition

5.1.1 Motion Detection

The ESL finger spelling integrates hand form with movements. Hence, motion detection is essential before recognizing the ESL finger spellings. According to [2, 10], ESL manual alphabets have seven movements corresponding to the seven Amharic vowel orders. The first vowel order (Geez) is a motionless, others have six special movements. Therefore, to understand these movements it is practical to detect the motion state.

In order to identify these movements, first it is decisive to decide the length of the distance L_D, which is an assumption for no movement. The movement made by unconsciously hand shaking of the signer, error occurred in image processing, unconsciously body movement of the signer and camera vibration are assumed to be the motion is in a motionless state.

$$LD = unconsciously\ hand\ shaking + error\ in\ image\ processing + camera\ vibration + unconsciously\ body\ movement \quad 5.1$$

To choose the length of L_D, we used the maximum distance between the initial frame and others in a given movie of vowel-1 (Geez). Even if the first vowel is represented by stationary motion, there are motions made by L_D. Therefore, we have to assume the motion made less than or equals to L_D is in motionless. The algorithm used to find L_D is

presented in Algorithm 5.1. During motion detection, we always calculate the center of mass from the first frame and it is used as a first point for recognition. Hence, the first frame is the reference point to detect the motion. Therefore, in order to assume the motion made by $L_{D,}$ our starting point is frame one. However, during finding L_D we cannot assume that the maximum distance is between frame one and last frame. This is because of the motion is made by a kind of vibration. This is the reason why we need algorithm to find the L_D. In this algorithm, while saying frames we are indicating to the point that refers to the frame.

Algorithm 5.1: *Finding distance* L_D

Input: Video Clip for vowel-1 (Geez)

Output: The Maximum distance between frame 1 and others

1. *Read Video V_Geez*
2. *Find the total number of frames of V_Geez, Tframe*
3. $L_D \rightarrow 0$
4. ***For*** *I=2 to Tframe*
5. *Find distance, D =Distance(Frame 1, Frame I)*
6. ***If*** $D>L_D$ *Then*
7. $L_D=D$
8. ***End if***
9. $I \rightarrow I+1$
10. ***End For***
11. *Stop*

Using Algorithm 5.1, we test all of the ESL vowel-1. As a result, we obtain 34 different lengths of L_D representing motion made by the 34 ESL vowel-1 finger spellings. From the experiment result, we select the maximum one for the actual L_D. From our data, we

obtained the minimum value of L_D is 2.7869, the maximum value of L_D is 16.8525 and average value of L_D is 8.3247. The L_D results of the 34 manual alphabets are listed below.

LD=

Columns 1 through 8

10.8835 9.2358 6.3027 8.5060 7.1353 6.2195 6.2329 8.3150

Columns 9 through 16

8.2416 3.3995 4.6607 12.0966 15.4063 6.9343 10.2662 4.8235

Columns 17 through 24

5.9782 11.9140 8.0478 16.8092 10.4863 10.6279 9.0880 7.8957

Columns 25 through 32

4.9118 7.2965 2.9143 4.9468 13.0414 5.2121 2.7869 4.2025

Columns 33 through 34

11.3681 16.8525

From this we have chosen the maximum displacement value of L_D which is 16.8525. So our assumption is, if there is displacement less than or equals to the value of L_D, it is in a motion less state. Besides of this, the value of L_D is the calculated from the Pythagoras theorem value of L_{Dx} and L_{Dy} using the Equation 4.10. However, the input for recognition of the motion is decided using L_{Dx} and L_{Dy} as we discussed in Section 5.1.2.

$$LD^2 = LDx^2 + LDy^2 \qquad 4.10$$

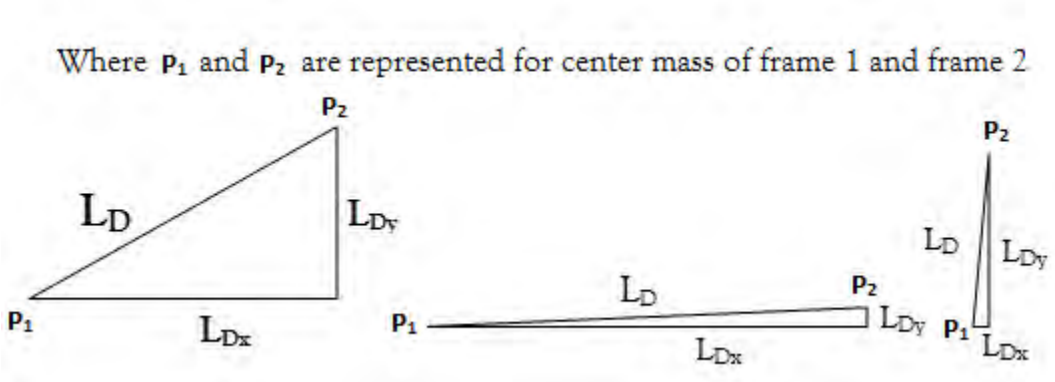


Figure 5.1: Pythagoras triangle used for estimation of LDx and LDy

From the Figure 5.1 we are seen that, when L_{Dy} is going to be zero ($L_{Dy} \rightarrow 0$), the value of $L_D = L_{Dx}$. The same condition will happen when ($L_{Dx} \rightarrow 0$), then $L_D = L_{Dy}$. Therefore our assumption is the value of L_D is used for both L_{Dx} and L_{Dy} during motion detection. On the other hand, we have to scale up the value of L_D based on the resolution of the frames. To formulate this, width of the frame has higher pixels than height of the frame which is 640 and 480 pixels respectively. Therefore,

$$LDx = LD \qquad 5.2$$

$$LDy = \frac{Height}{Width} * LD \qquad 5.3$$

From this we assumed, the signed hand is in a motion state if the displacement between two frames is greater than L_{Dx} in the x direction or L_{Dy} in the y direction.

5.1.2 Required Points for Recognition

To recognize the seven movements we use points that represent frames. In our case, the seven vowels of Ethiopian Sign language are characterized by at least three connected lines or four points. This is because, if the number of points is less than four points, it cannot differentiate all movements and if it is greater than four points it has an impact on decreasing the processing time. Let us see using a single point, two points and three points.

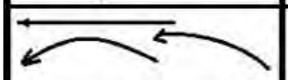
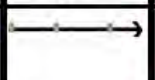
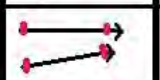
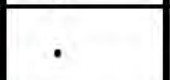
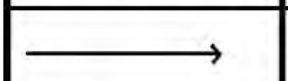
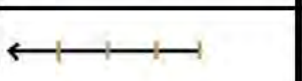
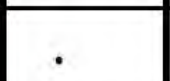
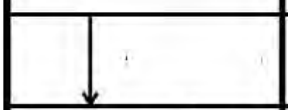
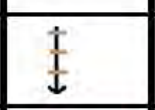
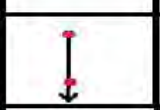
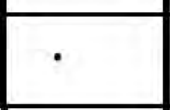
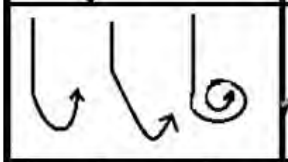
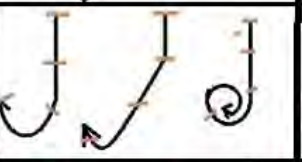
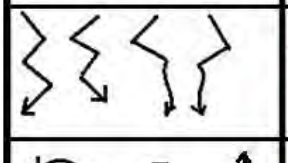
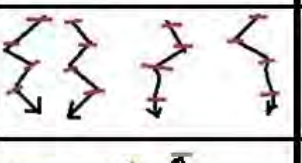
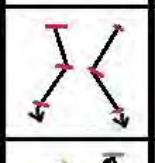
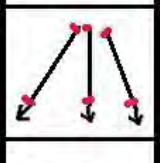
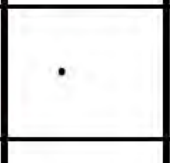
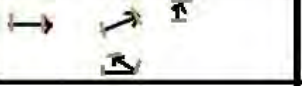
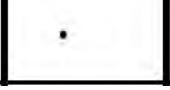
Using one point: We cannot recognize a line by a single point. Using this single point, we can only recognize vowel one, which is on stationary motion. Table 5.1 column 5 shows, the possible representation of the seven movements using a single point.

Using two points: By having different directions (four directions in our case) across movements, we can recognize vowel 1, vowel 2, vowel 3, vowel 4 and vowel 7 using a line or two points. As we can see in Table 5.1 column 4, these vowels can represent by a single line. However, it is difficult to represent vowel 5 and vowel 6 using a single line.

Using three points: This can produced two connected lines. Using these lines we can characterize vowel 5. However, it is impossible to characterize vowel 6 using two lines. As we can see in Table 5.1 column 3, vowel 6 cannot describe by two lines.

Therefore, the optimal points used for recognition of ESL vowels using finite state automata are four points or three connected lines. The displacement between two adjacent points should be greater than L_{Dx} in the x direction or L_{Dy} in the y direction. Table 5.1 shows the signer view movement of the sign hands in column 1, the possible four points used for recognition of the seven movements in column 2 and its corresponding vowel representation text in column 7. These points are represented in terms of x and y directions. In this case, P1 is represented by (x1, y1) which is the center of mass of frame 1 from the given movie, P2 is (x2, y2), P3 is (x3, y3), and P4 is (x4, y4).

Table 5.1: Representation of ESL vowels by line and direction during signing

Signer View	Reflection on Screen used for recognition , Using 4 points	Representation Using 3 points	Representation Using 2 points	Representation Using 1 point	Corresponding Vowel
					1
					2
					3
					4
					5
					6
					7

In Table 5.1, we have seen the properties of the seven movements using the center of masses. The actual movements of the ESL vowels of two signers are presented in Figure 5.2.

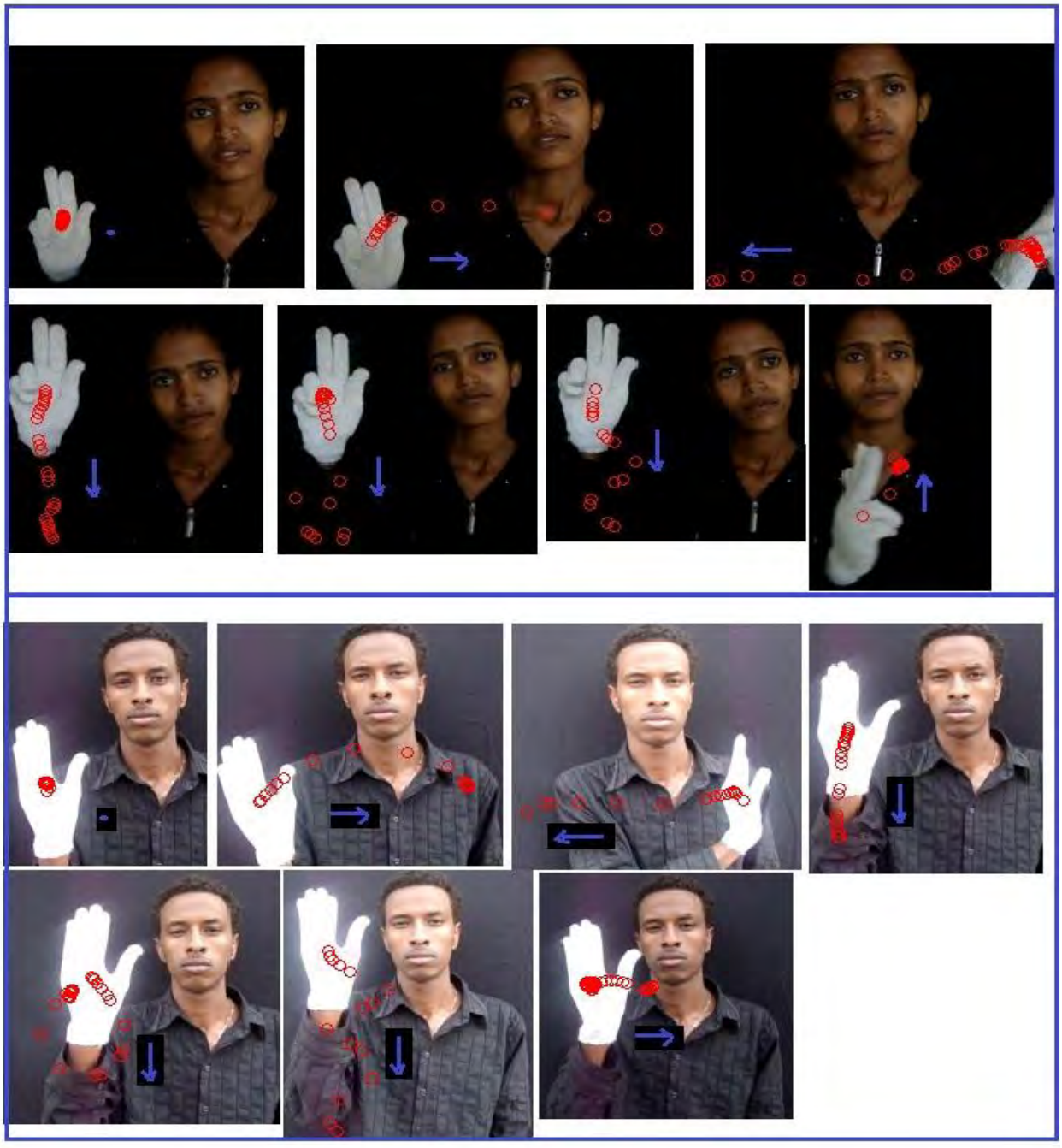


Figure 5.2: Actual results of the seven movements

Now it is essential to have an ideal machine, that takes properties of these points and produce the corresponding vowel number representation. Therefore, automata are our choice in this case. To understand the vowel or movements, we use a non deterministic finite state automaton. A non-deterministic that can make the best choice would be able to solve the problem without backtracking, but a non deterministic one can simulate non determinism with some work [21]. Non deterministic is an effective mechanism for describing some complicated language concisely and is helpful in solving problems easily [21]. Certain results are more easily established for Nondeterministic Finite Automata (NFA) than for Deterministic Finite Automata DFA [21]. Obviously, there is a difference in their definition, but this does not imply that there is a distinction between them [21]. Every NFA has an equivalent DFAs [20]. According to [20, 21], a non deterministic finite state of automaton is defined by:

M= (Q, Σ , ∂, q_0, F)

Q- Finite set of internal states

Σ-Finite set of input alphabets

∂- $Q\ X\ (\sum \cup \{\lambda\}) \rightarrow 2^Q$, where λ is an empty string

q_0- is the initial state $q0 \in Q$

F- is a set of final states F $\subseteq$ Q

The finite set of alphabets used in the recognition of ESL is derived from L_{Dx}, L_{Dy} and the four points. To find the required alphabets, it is necessary to find the difference between adjacent points. Accordingly, X_{21}=x2-x1, X_{32}=x3-x2, X_{43}=x4-x3, Y_{21}=y2-y1, Y_{32}=y3-y2, and Y_{43}=y4-y3. From this, movements we considered when X_{21}> L_{Dx}, X_{32}> L_{Dx}, X_{43}> L_{Dx}, Y_{21}> L_{Dy}, Y_{32}> L_{Dy}, Y_{43}> L_{Dy}, X_{21}<- L_{Dx}, X_{32}<- L_{Dx}, X_{43}<- L_{Dx}, Y_{21}<- L_{Dy}, Y_{32}<- L_{Dy}, and Y_{43}<- L_{Dy}. Along with, movements will not consider when abs(X_{21})$\leq$ L_{Dx}, abs(X_{32})$\leq$ L_{Dx}, abs(X_{43})$\leq$ L_{Dx},

abs(Y_{21})≤ L_{Dy}, abs(Y_{32})≤ L_{Dy}, and abs(Y_{43})≤ L_{Dy}. Therefore, we have 18 input alphabets for the recognition of the vowels. The minus (-) signs are used for the left and down direction. Table 5.2 shows the 18 inputs alphabets are grouped into six directions from the screen point of view.

Table 5.2: Possible input alphabets grouped in to six directions

Left direction	Right direction	Up direction	Down Direction	In stationary X-axes	In stationary Y-axes
X_{21}<-L_{Dx}, X_{32}<-L_{Dx}, X_{43}<- L_{Dx}	X_{21}> L_{Dx}, X_{32}> L_{Dx} , X_{43}> L_{Dx}	Y_{21}<- L_{Dy}, Y_{32}<- L_{Dy}, Y_{43}<- L_{Dy}	Y_{21}>L_{Dy}, Y_{32}>L_{Dy}, Y_{43}> L_{Dy}	abs(X_{21}) ≤ L_{Dx}, abs(X_{32}) ≤ L_{Dx}, abs(X_{43})≤ L_{Dx}	abs(Y_{21}) ≤ L_{Dy}, abs(Y_{32}) ≤ L_{Dy}, abs(Y_{43})≤ L_{Dy}

To form a general finite state of automata, it is important to discus strings that can generate vowels using the 18 input alphabets. For simplification reason, we symbolize the possible input conditions as shown in Table 5.3.

Table 5.3: Symbolize input alphabets by small letters

Letter	Condition	Letter	Condition	Letter	Condition	Letter	Condition
a	X_{21}<- L_{Dx}	f	X_{43}> L_{Dx}	k	Y_{32}> L_{Dy}	p	abs(Y_{21})≤ L_{Dy}
b	X_{32}<- L_{Dx}	g	Y_{21}<- L_{Dy}	l	Y_{43}> L_{Dy}	q	abs(Y_{32})≤ L_{Dy}
c	X_{43}<- L_{Dx}	h	Y_{32}<- L_{Dy}	m	abs(X_{21})≤ L_{Dx}	r	abs(Y_{43})≤ L_{Dy}
d	X_{21}> L_{Dx}	i	Y_{43}<- L_{Dy}	n	abs(X_{32})≤ L_{Dx}		
e	X_{32}> L_{Dx}	j	Y_{21}> L_{Dy}	o	abs(X_{43})≤ L_{Dx}		

Vowel 1(Geez)

In Table 5.1 row 1 column 2, vowel-1 didn't have any movements. In other term, it is in its stationary state in X and Y axes. Therefore, the possible string that represent for vowel-1 is:

$$Vowel\ 1 = \begin{cases} 1 & \{mnopqr\} \\ 8 & else \end{cases}$$

Vowel 2(KaEb)

In Table 5.1 row 2 column 2, vowel-2 has right direction in all possible moves of the x axes. However, in the Y-axes it has possibly moved up and then down or stationary using the specific condition. The possible strings for the vowel-2 are:

$$Vowel\ 2 = \begin{cases} 2 & \{(d|m)ef(q|p)(g|h)(r|l)\} \\ 8 & else \end{cases}$$

Vowel 3(Sals)

In Table 5.1 row-3 column 2, vowel 3 has left direction in all possible moves of the x axes and in stationary state across y axes. Therefore, string that could produce vowel-3 is:

$$Vowel\ 3 = \begin{cases} 3 & \{abcpqr\} \\ 8 & else \end{cases}$$

Vowel 4(RabE)

In Table 5.1 row 4 column 2, vowel-4 has down direction in all possible moves of the y axes and in stationary state across x axes. Therefore, string that can produce vowel-4 is:

$$Vowel\ 4 = \begin{cases} 4 & \{mnojkl\} \\ 8 & else \end{cases}$$

Vowel 5(Hams)

In Table 5.1 row 5 column 2, vowel-5 has down direction at the start moves and has left direction at the end of moves. Therefore, strings that can produce vowel 5 are:

$$Vowel\ 5 = \begin{cases} 5 & \{((m|d)(n|b|e)cjk(i|r|l))\} \\ 8 & else \end{cases}$$

Vowel 6(Sads)

In Table 5.1 row 6 column 2, vowel-6 has down zigzag moves in x axes and down moves in y axes. Therefore, strings that can produce vowel 5 are:

$$Vowel\ 6 = \begin{cases} 6 & \{(((\left((ae(o|c))\right|\left(db(o|f)\right))(j|p)kl)|(\left((me(o|c))\right|\left(mb(o|f)\right))jkl))\} \\ 8 & else \end{cases}$$

Vowel 7(SabE)

In Table 5.1 row 7, the move for vowel-7 is short distance which is only represented by two or three points, the rest have value of (0, 0). Therefore, strings that can produce vowel-7 are:

$$Vowel\ 7 = \begin{cases} 7 & \{(d|m)(n|e)(o|c)(p|g)\&\&(q|h)(r|i)\} \\ 8 & else \end{cases}$$

As a final point, the finite states of automata used to recognize the ESL finger spelling defined by:

M=({q0,q1,q2,q3,q4,q5,q6,q7,q8,q9q10,q11,q12,q13,q14q15,q16,q17,q18,q19,q20,q21, q22,q23,q24,q25,q26,q27,q28,q29,q30,q31,q32,q33,q34,qf}, {a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r}, ∂, q0, {qf}),

∂ is defined by the following transition graph:

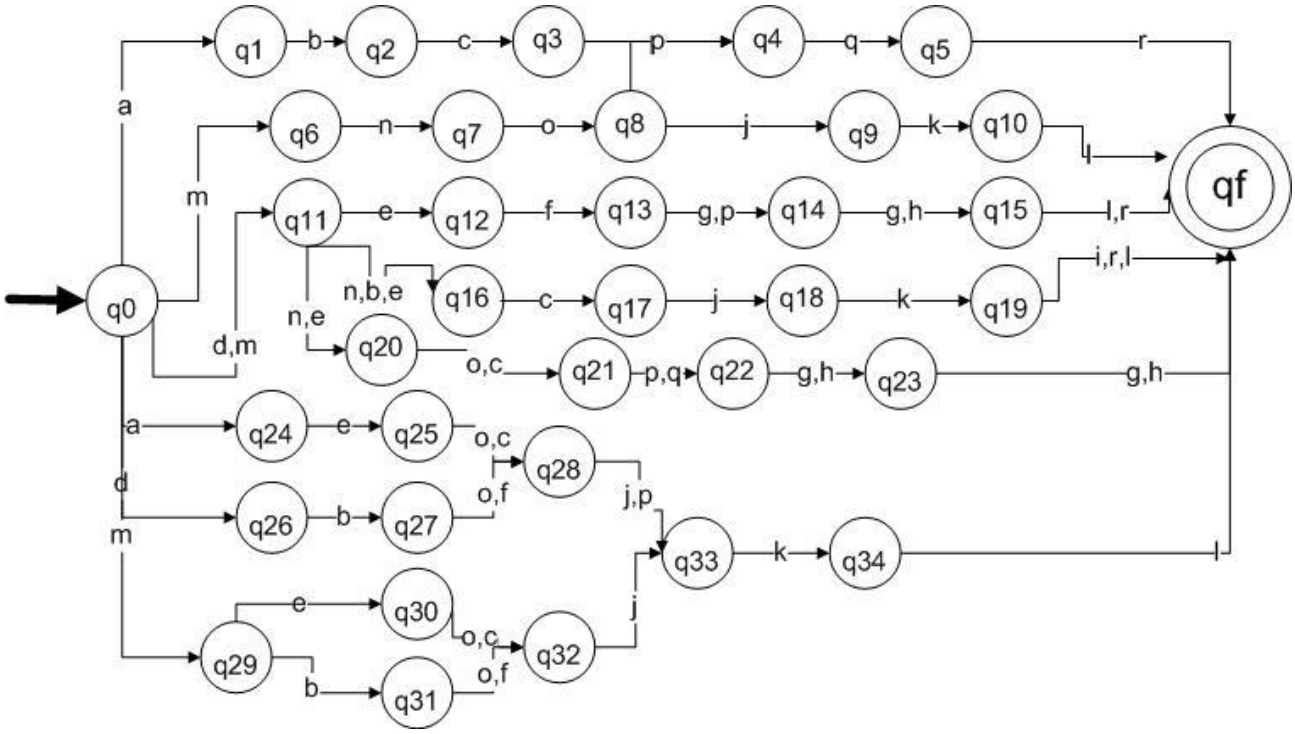


Figure 5.3: Finite state of automata used for recognition of the vowels

5.2 Recommended frame for consonant recognition

This study recognizes vowels of ESL finger spellings from a given movie. However, understanding the vowels without knowing consonants do not make sense. As a result, this thesis recommends which frame should be selected for the recognition of the consonant. And the selected frame will be used for the recognition of the consonant using [7] previous study or other in the future work.

ESL manual alphabet uses seven movements to identify vowels. However, during movement consonants may change its form. For that reason, choosing the right shape for recognition of the consonant is recommended. Figure 5.3 shows first and last frame of each the seven movements of the ESL. As we can see from the figures, the appearance of the sign in first frame and last frame are different.

This study also recommends, recognition of vowels should be done before recognizing the consonants. Thus, one frame can be selected in order to recognize the consonant. In this study we propose the following recommendations for each vowel of the ESL manual alphabets.

- The frames used to make vowel-1 are almost in a single station. Due to this reason, the sign hands in these frames have identical appearance. Therefore we can choose any of the frames from a given movie. However, it is better if it is the first frame. Figure 5.3 a, shows the appearance of the first and last frame for sign Fe {ﻉ}.

- The frames used for vowel-2 are located in different positions. The first frame located in the right most and the last frame is in the left most position from the screen point of view. While moving the signed hand from left to the right, the actual shape of the consonant is totally changed, Figure 5.3b. Therefore, frame used for accurate recognition of the consonant should be chosen frame one or near to frame one when the vowel is recognized as vowel-2.

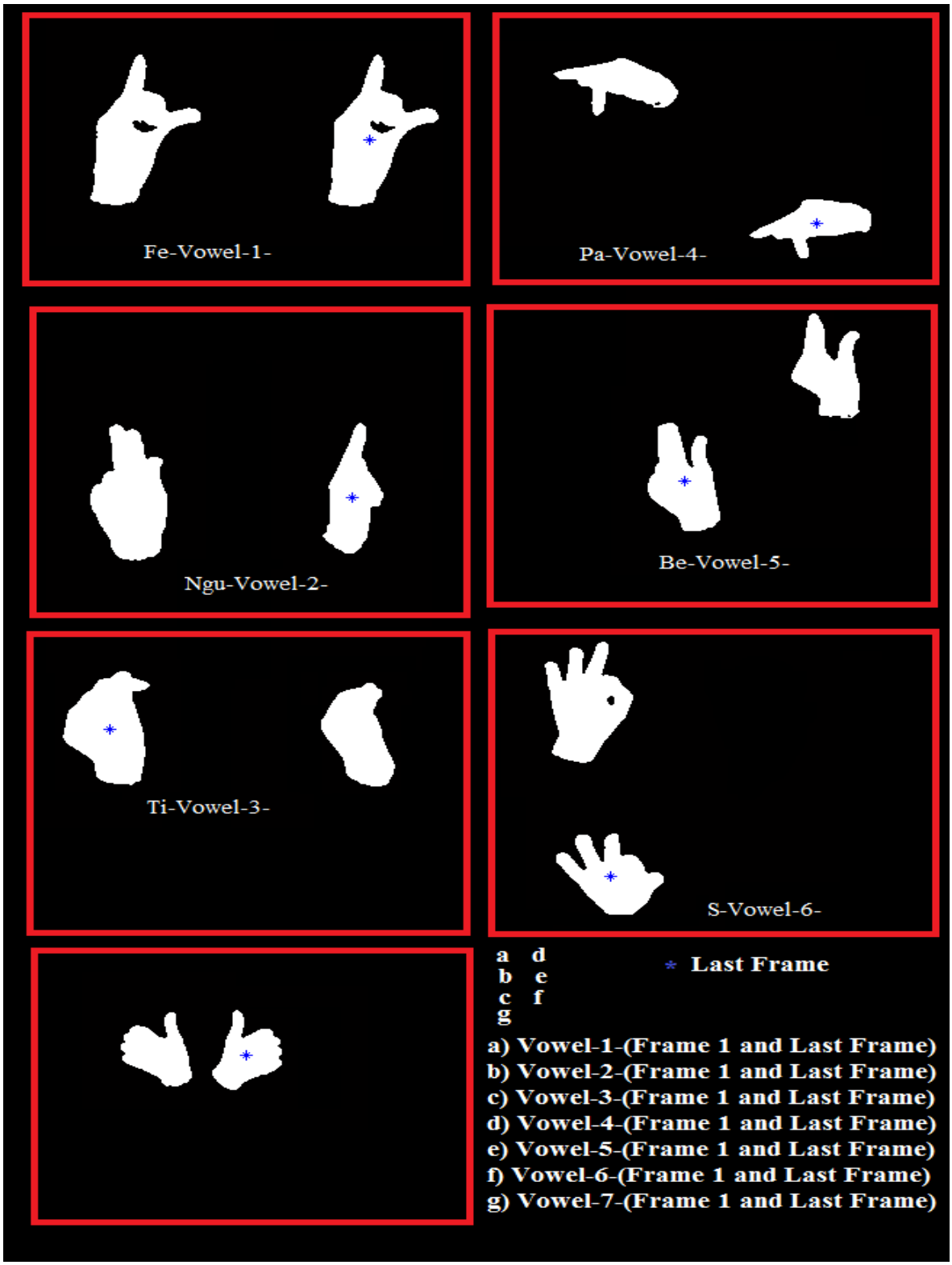


Figure 5.4: Processed first and last frames of the seven vowels of ESL

- Frames used for vowel-3 are also located in different positions. First frame is located in the left most and last frame is located in the right most position, Figure 5.3c. However due to the arrangement of sign hand and body of the signer, the actual appearance for consonant representation is in the last frame or close to the last frame.

- Frames used for vowel-4, vowel-5, and vowel-6 are located in different locations. However, these frames used to represent the vowels vary in the placement of x-axis. Figure 5.3 d, e, and f show the first frame on the top and last frame down. As we go from frame one to the last frame the appearance of the signed hand slightly changes its form. From this we can understand that, the actual consonant representation is presented in frame one or near to.

- The last vowel, vowel-7 rotates around its axis. Therefore, frame one have its real form of the consonant. However, the consecutive frames change its form. Moreover the last frame totally changes the shape of the signed hand. Figure 5.3 g, shows the first and last frame for the vowel-7-.

CHAPTER SIX

EXPERIMENT AND RESULT

This chapter presents the experimental result of the proposed method. It also shows the contribution of our study to the field of image processing and Ethiopian sign language recognition. To evaluate the performance of the system, we collect all possible ESL finger spellings as stated in the data collection Section (1.6.1). In addition to the examination of the whole system, the recognition performances of the cluster of vowels and recognition performance per users are also conducted.

6.1 Test Data and Description

In this research, we obtain the signs using a single digital camera with the help of white glove. The signer wear white glove for a reason of easy segmentation. The camera is taken by a person while the signers stand with black background. The camera is set in front of the signer. In vision-based sign language recognition, there are two mounting locations for the camera; the camera is in the position of an observer of the signer or from the point of view of the signer himself [13]. These two views can be thought as second-person and first-person viewpoints, respectively [13]. From this point of view, in this study the signed movies that we are going to recognize are captured as a view of second person. In other word, the direction of the signer view and the captured movie in the camera is opposite direction.

The system supports all video data types that Matlab could support. Basically, the system tested using MPG/MPEG (Moving Picture Experts Group) video format. The videos are captured using SONY 12.1Mega Pixel Digital Video camera. Originally the size frames of the captured movies are 640 X 480 resolutions and the type of data collected is a single movie for each alphabets of the ESL.

There are different types of video file format that organize sequence of images like Audio Video Interleave (AVI), MPG, Windows Media Video (WMV) and others. For video data, the term "file format" often refers either to the container format or the codec. A container format describes the layout of the file, while a codec describes how to code/decode the data. Many container formats support multiple codecs [24]. In our case, the system takes MPG file formats which contain 30frames/second.

The data used for this experiment is collected by four signers. The data is collected in terms of video mode. All signers spelled together all of the 238 ESL finger spellings. A single movie is represented to a single sign. Consequently, the 238 ESL manual alphabets are incorporated in our data. Furthermore, the data is clustered using the seven vowels of the ESL finger spelling and evaluated the performance of the vowels. Each clustered vowels have 34 signed videos.

The data is organized and incorporated in a single folder. The video is carefully renamed to its corresponding consonant followed by vowels. For example, sign ቡ {Bu} is renamed to 102, which is the first 10 is used for the consonant and 2 is for the vowel. This will be used for automated experimentation.

6.2 System Experimentation and Description

The system done for Video based Ethiopian sign language finger spellings integrates two goals. The main objective of the system is to show the experiment result of the study and second one is to show the process of the system and how it works. Figure 6.1 shows the user interface of the system. Left side of the picture shows the process and right one shows the experimentation result of the system.

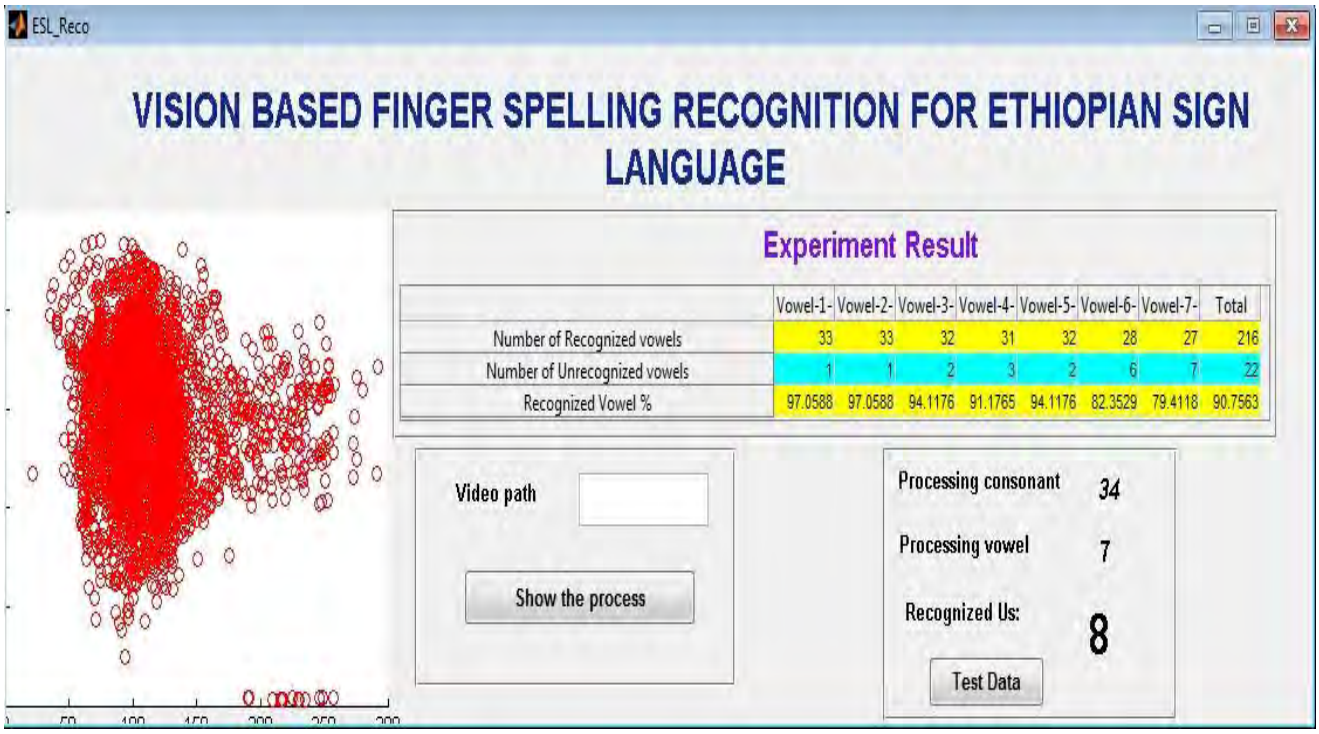


Figure 6.1: GUI Experiment result of the proposed system

The seven movements of ESL vowels are assigned to a value of one, two up to seven numbers. The validation of the recognition of the system works by crosschecking the vowel with its file name of the movie. If the result of the recognition of the sign is equal with the file name of the movie, then it counts as recognized otherwise it counts us unrecognized vowel.

6.3 Recognition performance

The experiment is done to evaluate the recognition performance of the new method. We can see this using the overall recognition performance of the system, recognition performance of each vowel group and recognition performance per signer. The overall recognition performance of the system is calculated using the Equation 6.1.

$$\text{System Performance} = \frac{\text{Total Number of Recognition of Vowel}}{238} \times 100 \qquad 6.1$$

Accordingly the overall system recognition performance is 90.75 %. This means 216 finger spellings are recognized from total of 238 finger spellings. The performance of the clustered vowels is calculated using Equation 6.2. The recognition performance results of each clustered vowel are listed in Table 6.1.

$$\text{Vowel clusterd Performance} = \frac{\text{Total Recognition of Vowel Clusterd}}{34} \times 100 \qquad 6.2$$

Table 6.1: Experimentation Result of vowels

Vowels	Total Number of Vowel	Recognized Vowels	Recognized Percentage (%)
Vowel-1-	34	33	97.0588
Vowel-2-	34	34	97.0588
Vowel-3-	34	32	94.1176
Vowel-4-	34	31	91.1765
Vowel-5-	34	32	94.1176
Vowel-6-	34	28	82.3529
Vowel-7-	34	27	67.4116

As we can see from Table 6.1 vowel 1, vowel 2, vowel 3, vowel 4, and vowel 5, have excellent recognition accuracy. Vowel 6 and Vowel 7has also satisfactory result but errors occur due to error in signing inputs. The move for vowel -6 is zigzag, and then signers can made this simple vibration of hand. Consequently, the assumption for movement of sign couldn't detect the motion in the x-axis. In case of Vowel-7, due to the input of some signs are rotated on its fixed point and created similarity with vowel-1-. However, if the data could be collected in a more accurate way and if the signers strictly follow the rule how to sign, better result can be obtained.

In addition to this, we experiment the recognition performance of the system per each signer. The result of this experiment is shown in Table 6.2.

Table 6.2 Experiment Result of each signer

Signer	**Total Number of vowels**	**Total Recognized Vowel**	**Recognized Percentage** (%)
A	10*7=70	63	*90.0*
B	8*7=56	54	*96.4286*
C	8*7=56	50	*89.2857*
D	8*7=56	49	*87.5*

From the experiment result of Table 6.2, we appreciate that, there is no significant accuracy variation among users. However, signer B has best accuracy result than others; this is because, signer B followed the rule of the vowels carefully. Finally, from the experiment result Table 6.2 we understand that, there is no significant accuracy different among signers and therefore the method is user independent.

Finally, this work can be possibly a continuation of yonas work [7]. The previous study on ESL recognition [7], works on EMA consonant recognition and it achieved 98.53% recognition performances. The work also has capable of rejecting non EMA signs. Therefore, if this work can integrate with yonas [7] work, the work on 34 EMA alphabets are increased to 238 ESL finger spellings. Accordingly, spell words can recognize using all possible finger spellings.

CHAPTER SEVEN
CONCLUSION AND DIRECTIONS FOR FURTHER STUDY

7.1 Conclusion

In this study we design and develop a method used to recognize vowels of Ethiopian sign language from a video. In addition to this we identified and recommend a single frame from a given movie that could be used for the recognition of consonants in each vowel. We also recommend that, it is better to recognize vowels before consonants. To achieve this all, we use different techniques of image processing techniques has been used.

In this work we applied image preprocessing algorithms, image segmentation, and post processing before trying to understand the signs. In addition, we used image selection algorithm to increase the efficiency of the system. Preprocessing and post processing are used for accurate recognition. We also used image segmentation to identify the sign hand from the entire image.

To recognize the vowels of ESL, we extract features from the segmented sign hand. The extracted feature from the entire image is the center mass of the signed hand. The extracted information was used to detect the motion and to recognize the vowels. Group of center masses are finally used for the recognition of vowels. To recognize these vowels, we employed finite state automata.

ESL finger spelling have 34 consonants, these consonants use seven movements to represent the seven vowels. This builds 238 total number finger spellings in the language. Our system understands the seven movements of the ESL vowels. Understanding only vowels of ESL will not give meaning if it could not integrate with the consonants. However,

recognizing of these vowels has its own contribution to understand the ESL finger spelling and the study of the language.

For the successful completion of the study, we collect all of the 238 ESL figure spellings by four signers in terms of video mode. The data is experimented using our system. As a result the overall system achieved 90.34% recognition performance. Therefore, it is possible to conduct projects and researches for more work on the language by using this study as a base. Moreover, the thesis has its own role on the study of motion detection and image processing applications.

7.2 Recommendation

The thesis has successfully achieved its objective. However, the study is not complete to understand the ESL, but it has its own contribution on study and recognition of the language. From this perspective, we recommend these ideas used for further study of the language.

- Previously there was a study on ESL recognition of the 34 consonant manual alphabets [7]. In addition, this thesis recognizes the vowels of the ESL and recommends frame used for the recognition of the consonants. Accordingly, we recommend a development tool on the recognition of 238 ESL finger spelling. This can be done by a project using the methods of the prior studies. And the tool can be used for learner of the ESL.
- This thesis and Ethiopian Sign Language Recognition Using Artificial Neural Network [7] have a good evaluation result for recognition of the ESL figure spelling. Thus, extending this study to word level or segmenting word in to finger spelling level is fundamental for the study of the language.
- We also recommend that the thesis can be done by other method and techniques to increase its recognition performance. It is also advisable if the system integrate both the consonant together with the vowels.

- The ESL has used signs to express meanings. These signs have been developed with the deaf community. The Ethiopian National Association of the Deaf has developed a dictionary for these signs [2]. As a result we recommend a study to be done on recognition of these signs.
- Finally we suggest that, the ESL needs more study work from the language side. Accordingly, technology could boldly facilitate the work.

REFERENCES

[1] International Day of Disabled Persons retrieved from, Press Release WHO/68 http://www.who.int/inf-pr-1999/en/pr99-68.html, February 13, 2012.

[2] Ethiopian National Association of the Deaf, Ethiopian sign language dictionary, Addis Ababa, Ethiopia, 2008.

[3] Vassilia N. and Konstantinos G., "Hidden Markov Models for Greek Sign Language Recognition," Proceedings of 2nd WSEAS International Conference on Speech Signal and Image Processing ICOSSIP, Skiathos, Greece, September 2002.

[4] Aleem Khalid Alvi, M. Yousuf Bin Azhar, Mehmood Usman, Suleman Mumtaz, Sameer Rafiq, Razi, Ur Rehman and Israr Ahmed, "Pakistan Sign Language Recognition Using Statistical Template Matching," Proceeding of the World Academy of Science, Engineering and Technology, Las Cruces, USA, 2005.

[5] Masresha tadesse, "Automatic translations of Amharic text to Ethiopian sign language" A thesis submitted to the school of graduate studies of Addis Ababa University, June 2010.

[6] Vassilia P.N. and Konstantinos M.G., ""Listening to deaf": A Greek sign language translator," IEEE, Information and Communication Technologies, 2006. ICTTA '06. 2nd , vol.1,pp.859-863.

[7] Yonas A. and Raimond K., "Ethiopian sign language recognition using Artificial Neural Network," Proceeding of the international confere`nce on Intelligent Systems Design and Applications (ISDA) IEEE, pp.995-1000, 2010.

[8] National Interpreter Resource Links: Ethiopian Manual Hand Alphabets, Retrieved from httpwww.terpslink.net\WSL\EtSL.html, December 10, 2011.

[9] Ricco, S. and Tomasi, C., "Fingerspelling Recognition through Classification of Letter-to-Letter Transitions", in Proceedings of ACCV (3). 2009, pp.214-225.

[10] Kyle Duarte, “The Mechanics of Fingerspelling: Analyzing Ethiopian Sign Language”, Gallaudet University Press, Sign Language Studies, vol. 11, no.1, pp. 5-21, 2010.

[11] M. Mohandes and S. Buraiky, “Automation of the Arabic Sign Language Recognition using the PowerGlove”, AIML Journal, Vol. 7, Issue 1, June, 2007.

[12] Tabata Y. and Kuroda T., “Finger spelling recognition using distinctive features of hand shape”, in proceeding of the 7th ICDVRAT with ArtAbilitation, Maia, Portugal 2002.

[13] Thad S. and Alex P., “Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video, IEEE ,transactions on pattern analysis and machine intelligence, vol. 20, no. 12, December, 1998 .

[14] Rini Akmeliawatil, M.P Ooi and Y.C Kuang, “Real-Time Malaysian Sign Language Translation using Colour Segmentation and Neural Network”, in the proceeding of IMTC 2007-Instrumentation and Measurement Technology Conference, Warsaw, Poland, May 2007.

[15] Jung-Bae Kim, Kwang-Hyun Park, Won-Chul Bang and Bien Z.Z. , "Continuous gesture recognition system for Korean sign language based on fuzzy logic and hidden Markov model," Proceedings of the 2002 IEEE International Conference on , vol.2, no., pp.1574-1579, 2002.

[16] Yael Cohen-Sygal and Shuly Wintner, “Finite State Registered Automata and their uses in Natural languages” In Proceedings of FSMNLP 2005, Finite-State Methods and Natural Language Processing. Helsinki, Finland, September 2005

[17] Marian Mindek and Michal Burda ,“Image Storage, Indexing and Recognition with Finite State Automata”, IAENG International Journal of Computer Science, Vol. 33, Issue 1, February 2007.

[18] Marian Mindek, “Finite State Automata and Image Recognition”, Proceedings of the Dateso Annual International Workshop on DAtabases, TExts, Specifications and Objects , Desna, Czech Republic, April 2004.

[19] Jarkko KariY and Cristopher Moore, “Rectangles and squares recognized by two-dimensional automata”, Research paper, Jun 2000, retrieved from http:// www.santafe.edu /media/workingpapers/00-06-032.pdf, On June 13, 2010.

[20] S.P.Eugene Xavier, Theory of Automata and Formal Languages and Computations, New Age International Publisher, 2005.

[21] Peter Linz, an Introduction to formal language and Automata, Jones and Bartlett publishers, 2001.

[22] Tarun Kumar and Karun Verma, “A Theory Based on Conversion of RGB image to Gray image”, International Journal of Computer Applications, Vol. 7, No.2, 2010, pp. 0975 – 8887.

[23] J. Lu, Q. Gu, K.N. Plataniotis, and J. Wang, "A Comparative Study of Skin-Color Models", in Proceeding of ICIAR, 2005, pp.729-736.

[24] The Math Works Ink. , Image Processing Toolbox for use with Matlab , User Guide, 1998.

[25] Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins, Digital Image Processing using Matlab, Prentice Hall, 2003.

[26] Alasdair McAndre, “An Introduction to Digital Image Processing with Matlab”, Notes for Digital Image Processing, 2004, retrieve from http://visl.technion.ac.il/ labs/anat/An Introduction To Digital Image Processing With Matlab.pdf, on November 15, 2010.

Appendix A: Main Program

```
function varargout = ESL_Reco(varargin)
% ESL_RECO M-file for ESL_Reco.fig
%      ESL_RECO, by itself, creates a new ESL_RECO or raises the existing
%      singleton*.
%
%      H = ESL_RECO returns the handle to a new ESL_RECO or the handle to
%      the existing singleton*.
%
%      ESL_RECO('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in ESL_RECO.M with the given input arguments.
%
%      ESL_RECO('Property','Value',...) creates a new ESL_RECO or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before ESL_Reco_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to ESL_Reco_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help ESL_Reco

% Last Modified by GUIDE v2.5 10-Aug-2011 19:10:03

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @ESL_Reco_OpeningFcn, ...
                   'gui_OutputFcn',  @ESL_Reco_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
```

```
    gui_State.gui_Callback = str2func(varargin{1});
end
 if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before ESL_Reco is made visible.
function ESL_Reco_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to ESL_Reco (see VARARGIN)

% Choose default command line output for ESL_Reco
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes ESL_Reco wait for user response (see UIRESUME)
% uiwait(handles.figure1);



% --- Outputs from this function are returned to the command line.
function varargout = ESL_Reco_OutputFcn(~, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;
```

```
% --- Executes during object creation, after setting all properties.

% --- Executes on button press in radiobutton1.
function radiobutton1_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of radiobutton1

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a
double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

 % --- Executes on button press in radiobutton2.
function radiobutton2_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton2 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of radiobutton2

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
     s1 = get(handles.edit1,'String');
     n=1;index_image=9;
     [cei cej]=RecognitionP(s1,index_image);
      %mov(n).cdata = read(xyloObj, n);
      %f=mov(n).cdata;
      %imshow(f);

% --- Executes on button press in btn_TestData.
function btn_TestData_Callback(hObject, eventdata, handles)
% hObject    handle to btn_TestData (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%clear all
vow_Rec=[0 0 0 0 0 0 0 0];
vow_UnR=[0 0 0 0 0 0 0 0];
tempVo=[0 0 0 0 0];
tt=1;
for con=1:1:34
   for vol=1:7
      vol1=int2str(vol);
      con1=int2str(con);
     s = ['C:\Users\Eyob\Desktop\Thesis work\tedyeyob\data1\' int2str(con)
int2str(vol) '.MPG']
    % s='C:\Users\Eyob\Desktop\Thesis work\Real Data\data\71.MPG';
     set(handles.txtvol, 'String', vol1);
     set(handles.txtcon, 'String', con1');
```

```
    xyloObj = mmreader(s);
    nFrames = xyloObj.NumberOfFrames
    voul='8';
    voul4='8';
    voul5='8';
    voul2='8';
    str=strcmp(voul, '8');          %Compare with unrecognized Vowel
    str1=strcmp(voul, '4');         %Compare with Vowel-4-
    str2=strcmp(voul,'5');          %Compare with Vowel-5-
    index_image=9;                  %Intial Variable

 while(str||str1||str2)             % While Recognion is complete
      if (index_image>nFrames/3)    % if it can not generate possible four
points
          break;
      end
      if((voul4=='8')&&(voul=='4'))
          voul4='4';
      end
      if((voul5=='8')&&(voul=='5'))
          voul5='5';
      end

    [cei cej]=Recognition(s,index_image); %Calling function Recognition
    voul=FinitAutomata(cei,cej);          %Calling function FiniteAutomata
    index_image=index_image+2;
    str=strcmp(voul, '8');
    str1=strcmp(voul, '4');
    str2=strcmp(voul, '5');

    set(handles.txtRecus, 'String', voul);

    % voul5
 end

  if((voul5=='5')&&((voul=='8')||(voul=='5')))
      voul='5';
```

```
elseif((voul4=='4')&&((voul=='8')||(voul=='4')))
    voul='4';
end
switch(vol1)
   case '1'
       if (int2str(vol)==voul)
          vow_Rec(1)=vow_Rec(1)+1;
       else
         vow_UnR(1)=vow_UnR(1)+1;

       end
    case '2'
       if (int2str(vol)==voul)
          vow_Rec(2)=vow_Rec(2)+1;
       else
         vow_UnR(2)=vow_UnR(2)+1;
       end
     case '3'
       if (int2str(vol)==voul)
          vow_Rec(3)=vow_Rec(3)+1;
       else
         vow_UnR(3)=vow_UnR(3)+1;
       end
    case '4'
       if (int2str(vol)==voul)
          vow_Rec(4)=vow_Rec(4)+1;
       else
         vow_UnR(4)=vow_UnR(4)+1;

       end
    case '5'
       if (int2str(vol)==voul)
          vow_Rec(5)=vow_Rec(5)+1;
       else
         vow_UnR(5)=vow_UnR(5)+1;
          tempVo(tt)=con
         tt=tt+1;
```

```
        end
       case '6'
        if (int2str(vol)==voul)
           vow_Rec(6)=vow_Rec(6)+1;
        else
          vow_UnR(6)=vow_UnR(6)+1;
         end
      case '7'
        if (int2str(vol)==voul)
           vow_Rec(7)=vow_Rec(7)+1;
        else
          vow_UnR(7)=vow_UnR(7)+1;
         end
     otherwise
 end
vow_Rec(8)=sum(vow_Rec(1:7));   %total recognized vowels
vow_UnR(8)=sum(vow_UnR(1:7));   %total unrecognized vowels
%each recognized vowel
VR1=vow_Rec(1);VR2=vow_Rec(2);VR3=vow_Rec(3);VR4=vow_Rec(4);
VR5=vow_Rec(5);VR6=vow_Rec(6);VR7=vow_Rec(7);VR8=vow_Rec(8);
%each unrecognized vowel
VUR1=vow_UnR(1);VUR2=vow_UnR(2);VUR3=vow_UnR(3);VUR4=vow_UnR(4);
VUR5=vow_UnR(5);VUR6=vow_UnR(6);VUR7=vow_UnR(7);VUR8=vow_UnR(8);
%Stor in the tabel
datavalue={ VR1 VR2 VR3 VR4 VR5 VR6 VR7 VR8;
            VUR1 VUR2 VUR3 VUR4 VUR5 VUR6 VUR7 VUR8;
            VR1*100/34 VR2*100/34 VR3*100/34 VR4*100/34 VR5*100/34 VR6*100/34
VR7*100/34 VR8*100/238};
set(handles.uitable1,'Data',datavalue);

  end
end
```

Appendix B: Finite state of Automata

```
function [voul]=FinitAutomata(cei,cej)
Ry=12.0;
Rx=16.0;
y21=(cej(2)-cej(1));
y32=(cej(3)-cej(2));
y43=(cej(4)-cej(3));
x21=(cei(2)-cei(1));
x32=(cei(3)-cei(2));
x43=(cei(4)-cei(3));
%right and down movements
d=x21>Rx;e=x32>Rx;f=x43>Rx;
j=y21>Ry;k=y32>Ry;l=y43>Ry;
%left and up movements
a=x21<-Rx;b=x32<-Rx;c=x43<-Rx;
g=y21<-Ry;h=y32<-Ry;i=y43<-Ry;
%~ no movement right and down
m=abs(x21)<=Rx;n=abs(x32)<=Rx;o=abs(x43)<=Rx;
p=abs(y21)<=Ry;q=abs(y32)<=Ry;r=abs(y43)<=Ry;
if(m&&n&&o&&p&&q&&r)
    voul='1';
elseif((d||m)&&e&&f&&(g||p)&&(q||h)&&(r||l))
   voul='2';
elseif(a&&b&&c&&p&&q&&r)
    voul='3';
elseif(m&&n&&o&&j&&k&&l)
    voul='4';
elseif(((m||d)&&(n||b||e)&&c&&j&&k&&(i||r||l)))
    voul='5';
elseif((((a&&e&&(o||c))||(d&&b&&(o||f)))&&(j||p)&&k&&l)||(((m&&e&&(o||c))||(m
&&b&&(o||f)))&&j&&k&&l))
        voul='6';
elseif((d||m)&&(n||e)&&(o||c)&&(p||g)&&(q||h)&&(r||i))
    voul='7';
else voul='8';
end
```

Appendix C: Segmentation and Center of mass

```
function [x y]=centermass(f);
%imshow(f)
% pause(1)


if size(f,3)==3 %RGB image
 r=f(:,:,1);
   g=f(:,:,2);
   b=f(:,:,3);
    r1=mean(mean(r));
    g1=mean(mean(g));
    b1=mean(mean(b));
    avg_rgb=(r1+g1+b1)/3;
    lighting none
    if(avg_rgb<30)
         f=cat(3,r-93,g-57,b-55);
    else
       f=cat(3,r-242,g-214,b-192);
    end
   % imshow(f)
   % pause(1)
        f=rgb2gray(f);%convert to Gray color
       % imshow(f)
      % pause(1)


end
f=imadjust(f);
  % imshow(f)
  %pause(1)
f=medfilt2(f,[3 3]); %filter using median filter
% imshow(f)
% pause(1)
T=0.5*(double(min(f(:)))+double(max(f(:))));
done=~true;
while~done
    g=f>=T;
```

```
    Tn=0.5*(double(mean(f(g)))+double(mean(f(~g))));
    done=abs(T-Tn)<0.5;
    T=Tn;
end


     g=f>=T;%threshold the image
     % imshow(g)
         CC = bwconncomp(g, 4);
         S = regionprops(CC, 'Area');%make regions by area
         S=struct2cell(S);
         m = cell2mat(S);
         [u v]=max(m);%select the largest region which is the fingure spelling
         g = false(size(g));
         g(CC.PixelIdxList{v}) = true;


            [M N]=size(g);
%imshow(g)
%find the center of mass of the pixels
ci=0;cj=0;k=0;
for i=1:M
    for j=1:N
        if g(i,j)==1
            ci=ci+i;
            cj=cj+j;
            k=k+1;
        end
    end
end
y=ci/k;
x=cj/k;
hold on
```

Appendix D: Detection and Recognition

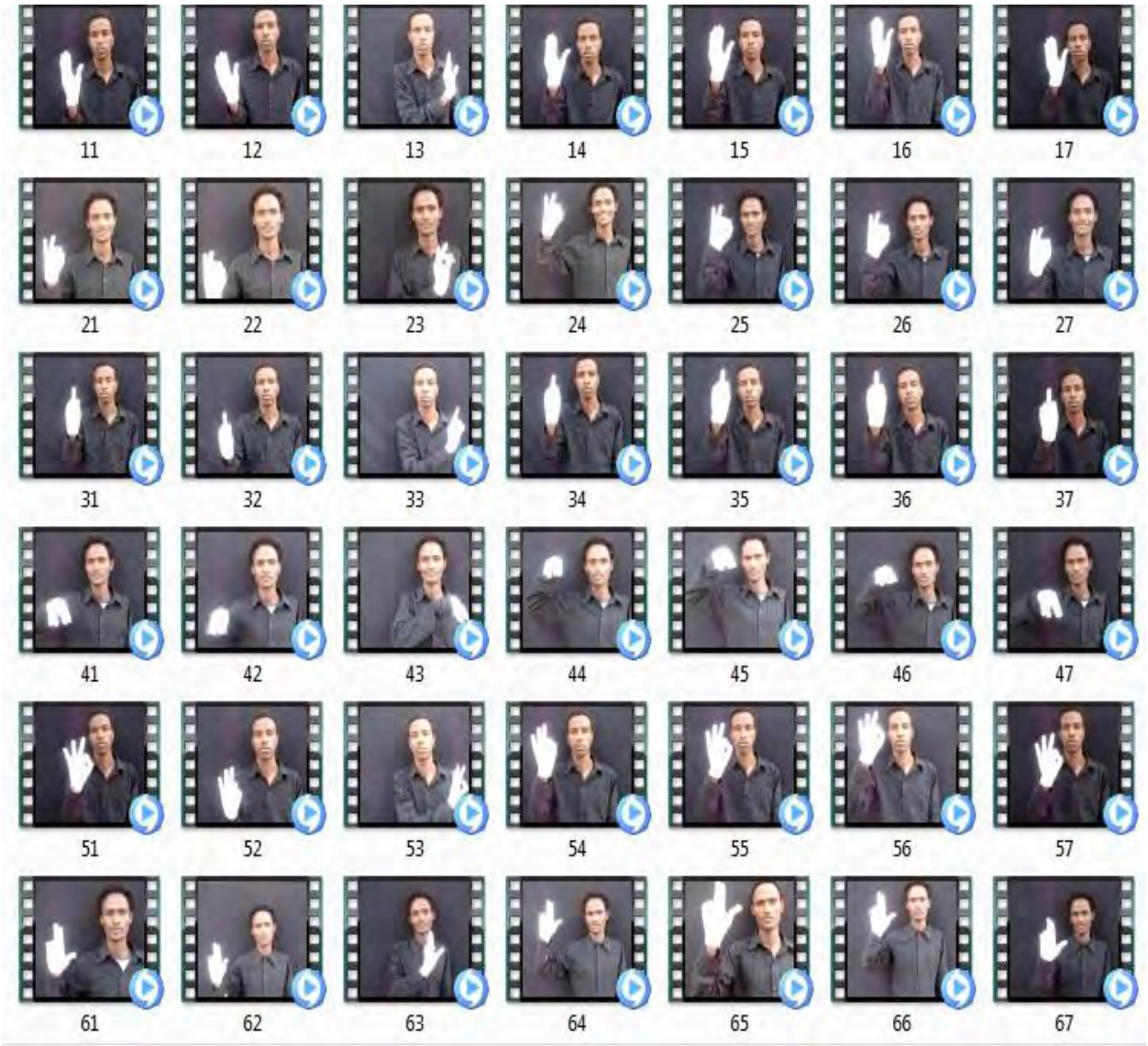
```
function [cei cej]=Recognition(s,index_image);
     xyloObj = mmreader(s);
     nFrames = xyloObj.NumberOfFrames;
     vidHeight = xyloObj.Height;
     vidWidth = xyloObj.Width;
     mov(1:nFrames) = ...
     struct('cdata', zeros(vidHeight, vidWidth, 3, 'uint8'),...
     'colormap', []);


  %points used for recognition
      cei=[0 0 0 0];     % X axis
      cej=[0 0 0 0];     % Y axis
      d=1;               %index for points to be stored
      n=1;               %Index for frames inside the loop
  % Read and Process Frame1 and find the center of mass in frame1
      mov(n).cdata = read(xyloObj, n);
      f=mov(n).cdata;
      f=imresize(f,[vidHeight*.5 vidWidth*.5]); %Resize the Frame
      [x y]=centermass(f);
      cei(d:4)=x;
      cej(d:4)=y;
  % find the rest three center of masses
       while(n<=nFrames)
             n=n+index_image;
             if(n>nFrames)
                  break;
             end
             d=d+1;
             LD=0; %intial distance length between two center of masses
             while(LD<=16.8525) % Until the condition satisfied, process each
next frame
                 mov(n).cdata = read(xyloObj, n);
                 f=mov(n).cdata;
```

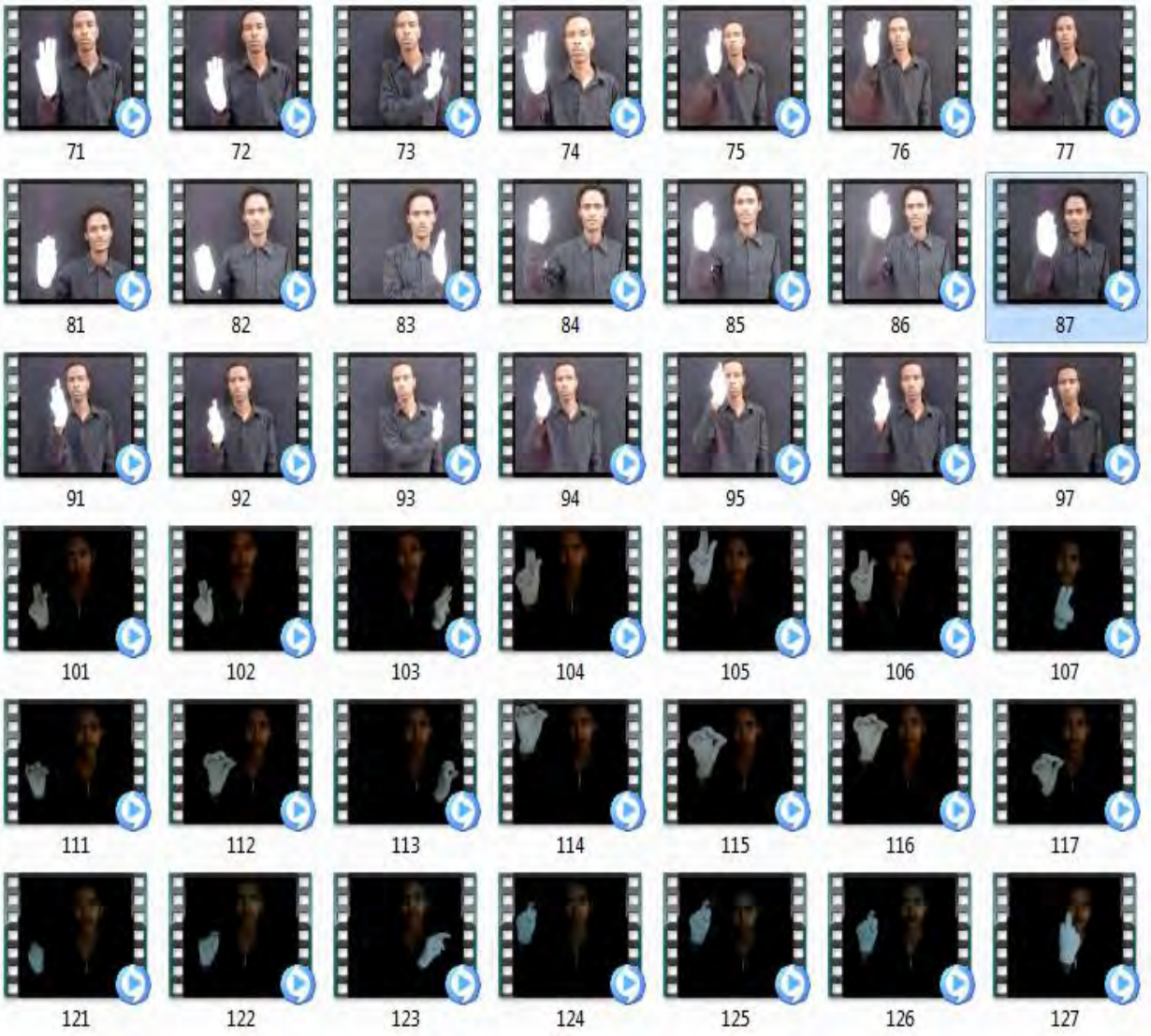
```
                f=imresize(f,[vidHeight*.5 vidWidth*.5]);
                [x y]=centermass(f);
                cei(d:d+2)=x;
                cej(d:d+2)=y;
                LD = sqrt((cei(d)-cei(d-1))^2+(cej(d)-cej(d-1))^2);%distance
                n=n+1;
               if(n>nFrames)
                   break;
               end
            end
               hold on
          plot(cei,cej,'or');
       end

end
```

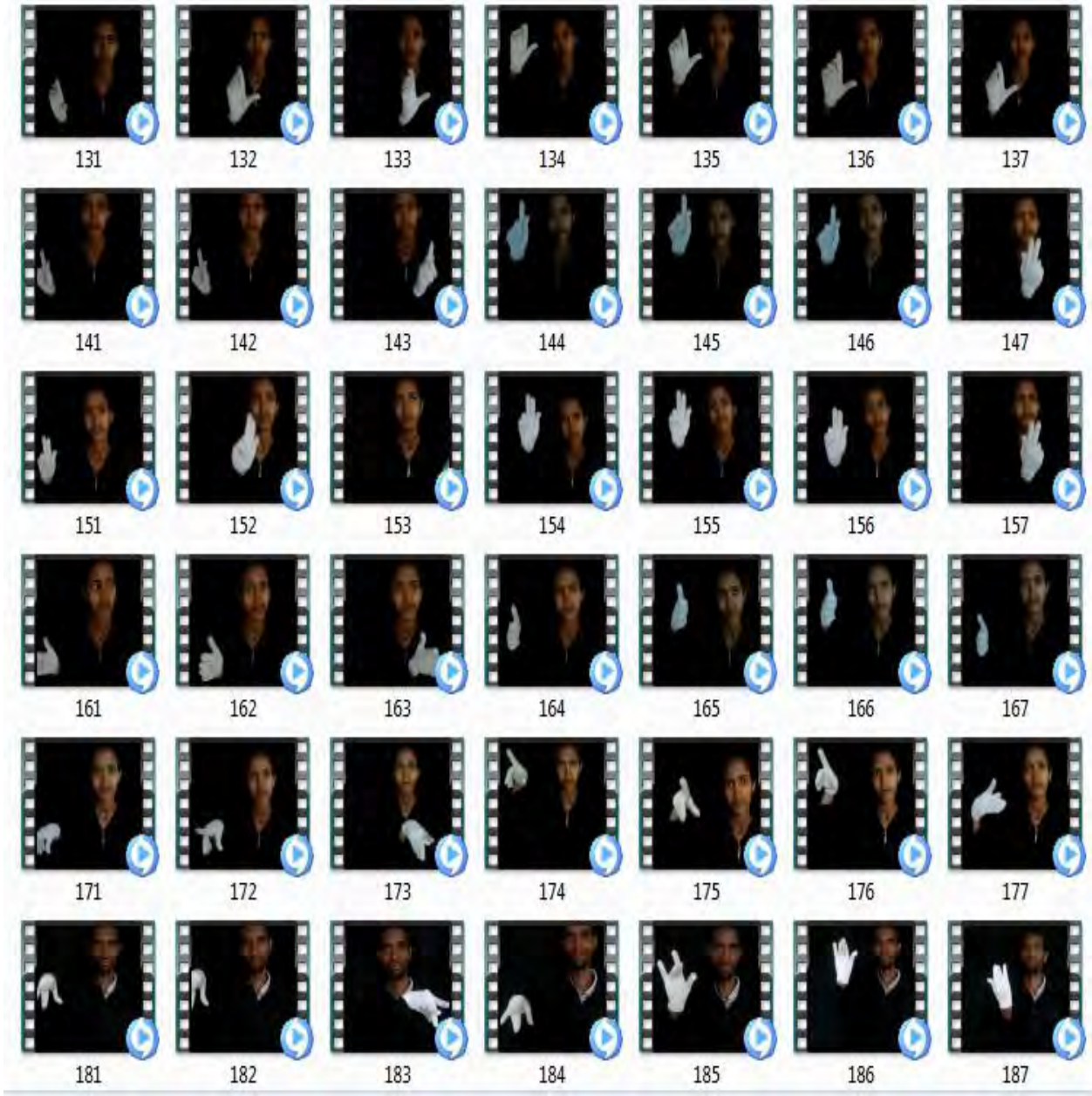
Appendix E: Finding LD

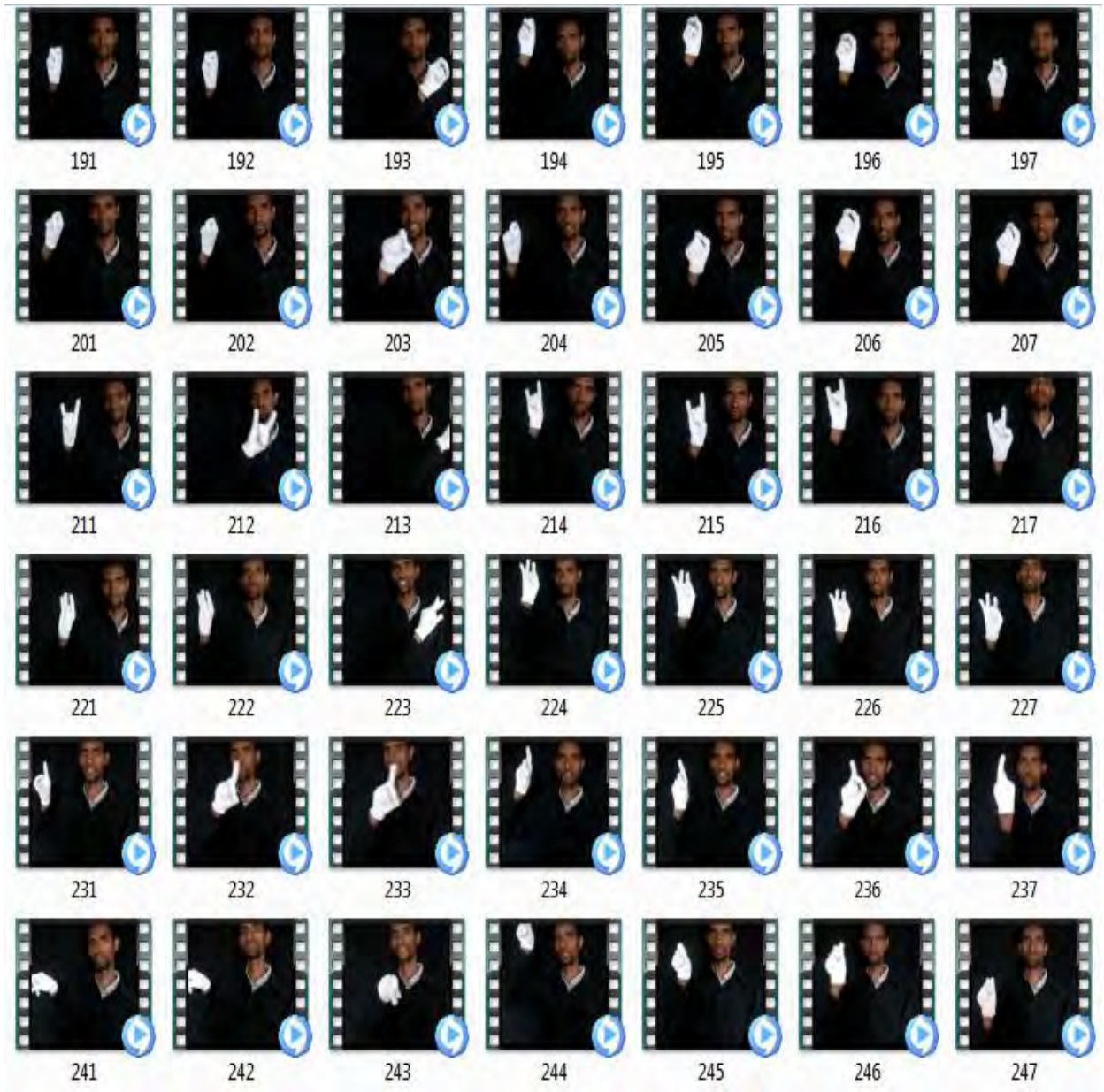
```
R=[];
for con=1:34
  for vol=1:1
       vol1=int2str(vol);
     s = ['tedyeyob\data\' int2str(con) int2str(vol) '.MPG'];
    xyloObj = mmreader(s);
nFrames = xyloObj.NumberOfFrames;
vidHeight = xyloObj.Height;
vidWidth = xyloObj.Width;
mov(1:nFrames) = ...
struct('cdata', zeros(vidHeight, vidWidth, 3, 'uint8'),...
'colormap', []);
 mov(1).cdata = read(xyloObj, 1);
f=mov(1).cdata;
f=imresize(f,[vidHeight*.5 vidWidth*.5]);
[x y]=centermass(f);
index_image=6;n=1;
maxR=0;
while(n<nFrames)
    n=n+index_image;
     if(n>nFrames)
n=nFrames;
     end
mov(n).cdata = read(xyloObj, n);
f=mov(n).cdata;
f=imresize(f,[vidHeight*.5 vidWidth*.5]);
[x1 y1]=centermass(f);
ds1 = sqrt((x1-x)^2+(y1-y)^2);
if (ds1>maxR)
    maxR=ds1;
end
 end
R(con)=maxR;
  end
end
```

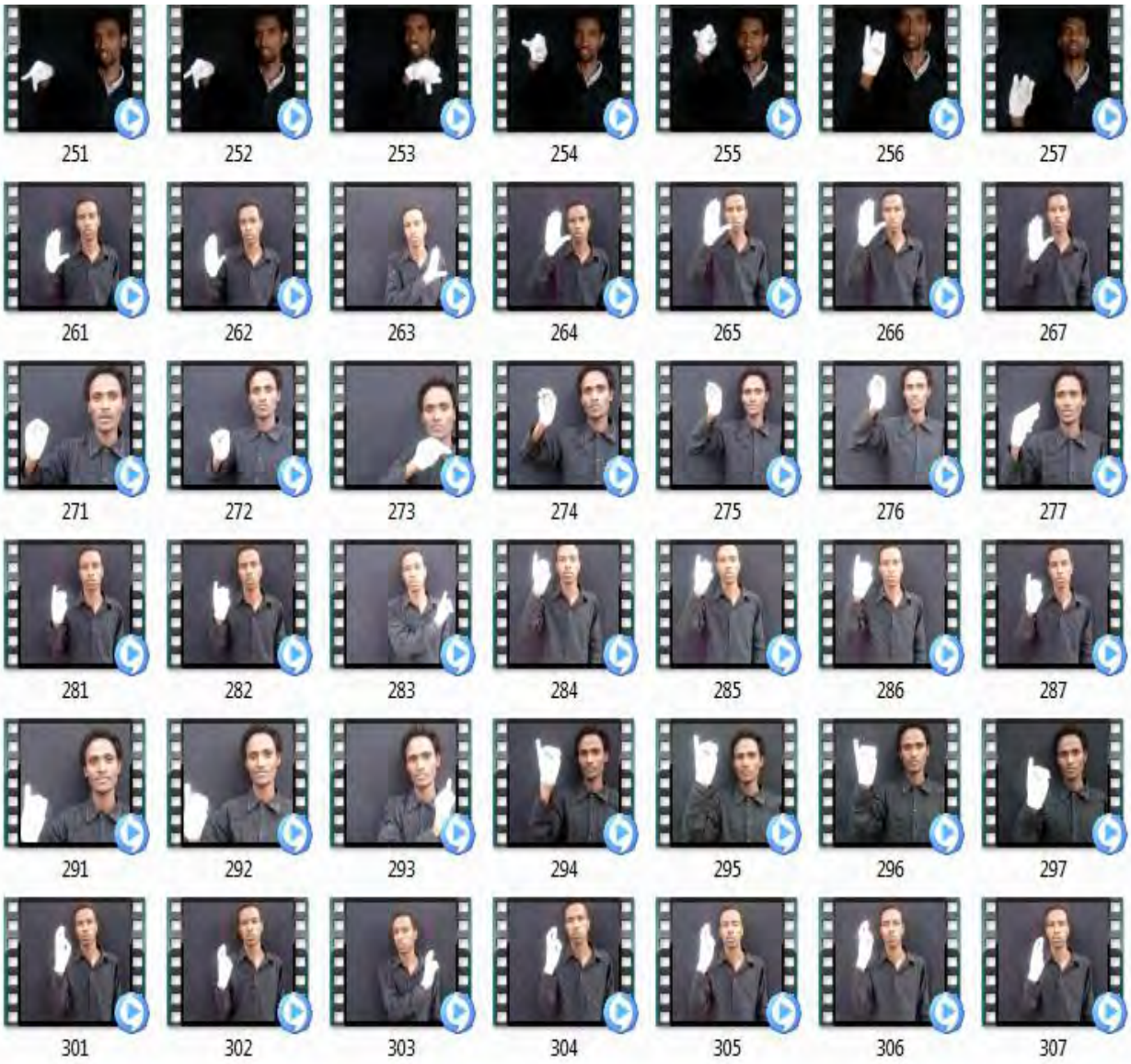
Appendix F: Complete Test Data




71
72
73
74
75
76
77
81
82
83
84
85
86
87
91
92
93
94
95
96
97
101
102
103
104
105
106
107
111
112
113
114
115
116
117
121
122
123
124
125
126
127


131 132 133 134 135 136 137
141 142 143 144 145 146 147
151 152 153 154 155 156 157
161 162 163 164 165 166 167
171 172 173 174 175 176 177
181 182 183 184 185 186 187


191
192
193
194
195
196
197
201
202
203
204
205
206
207
211
212
213
214
215
216
217
221
222
223
224
225
226
227
231
232
233
234
235
236
237
241
242
243
244
245
246
247


251 252 253 254 255 256 257
261 262 263 264 265 266 267
271 272 273 274 275 276 277
281 282 283 284 285 286 287
291 292 293 294 295 296 297
301 302 303 304 305 306 307


301
302
303
304
305
306
307
311
312
313
314
315
316
317
321
322
323
324
325
326
327
331
332
333
334
335
336
337
341
342
343
344
345
346
347

Declaration

This thesis is my original work and has not been submitted as a partial requirement for any degree in any university.

_____

EYOB GEBRETINSAE BEYENE

The thesis has been submitted for examination with our approval as university advisors.

_____ _____ _____

Dr. Fistum Admasu (Advisor)

February, 2012