Addis Ababa University

College of Natural Science

School of Information Science

ISOLATED WORD-LEVEL ETHIOPIAN SIGN LANGUAGE RECOGNITION

By

SAMUEL TESHOME ABEBE

October, 2013

Addis Ababa University

College of Natural Science

School of Information Science

ISOLATED WORD-LEVEL ETHIOPIAN SIGN LANGUAGE RECOGNITION

A Thesis submitted to the School of Graduate Studies of Addis Ababa University in partial fulfillment of the requirements for the Degree of Master of Science in Information Science

By

SAMUEL TESHOME ABEBE

October, 2013

Addis Ababa University

College of Natural Science

School of Information Science

Isolated Word-Level Ethiopian Sign Language Recognition

By

SAMUEL TESHOME ABEBE

APPROVAL BY BOARD OF EXAMINERS

Name	Title	Signature	Date
Melaku Girma (MSc.)	Chairperson	_____	_____
Million Meshesha (PhD)	Advisors	_____	_____
Dereje Teferi (PhD)	Examiner	_____	_____

Dedication

to

My Mother

Acknowledgement

First and foremost, I would like to take this opportunity to express my deepest gratitude toward my advisor, Dr. Million Meshesha for his commitment, patience, timely follow-ups, very constructive suggestion, and encouragement throughout the entire process of this thesis.

I would like to extend my special thanks to my mother who always gave me her heartfelt love and encouragement. And a special thanks goes to my beloved brothers Dawit, and Surafel, and to my sisters Tsegmariam, and Mekdese and to my aunt Seble Eshete.

I would like to pass my appreciation to Betsegaw Dereje for his cooperation, advice and unreserved help. I am very grateful to Ato Wondessen Teshome, Gedion Assefa, Kelemie Tebikew, Yidenkachew Amsalu, Henock Luleseged, Wondwossen Mulugeta, Melkamu Tamiru, and Ermias Abebe, who are always on my side. I am lucky to have you all.

I am also indebted to my friends and brothers Getachew Tadesse and Beidemariam Hailu for their continuous appreciation, valuable and unconditional suggestions and comments which were good inputs to the thesis work.

My heartfelt appreciation goes to the almighty God Jesus Christ and his holy Mother Virgin Mary.

Last but not list special thanks go to all staff members of School of Information Science and AAU, Budget and Finance Office.

Abstract

Hand gesture enables deaf people to communicate with each other and/or with hearing people on their daily lives rather than speaking. With regard to this there is still big communication gap between the deaf and hearing community because most of them don't know the language. Therefore, in order to narrow this gap teaching sign language to the hearing community is one solution, however, coming up with a computerized system that can translate sign language to text or sound and vice versa is a better solution. In this regard, a lot has been done for most of the sign languages all over the world. Even though, little attention was given to the Ethiopian Sign Language (EthSL), some attempts were made to come up with a system. Attempts to develop Ethiopian Manual Alphabet (EMA) recognition system from a static image were made. As an extension to this a recognition system that can recognize continuous gestures from sequence of video frames and that also determine hand movement trajectory was proposed. However, EthSL comprises not only EMA but also gestures that represent a whole word, so a recognition system that works at that level is required. In this thesis, a system that extracts hand gestures and motion trajectory for EthSL word is proposed. The system has three modules namely the data pre-processing, feature extraction and sign classification or recognition. The preprocessing starts off with the identification of key frames, followed by skin color detection to segment hand gestures. The feature extraction module is responsible for creation of manual features and determination of hand trajectories that combines them to create a feature vector. In which the classification module trains the system as well as build a model that can be used as a reference for the recognition of a sign. The proposed system is signer dependent and experimentations are conducted using real EthSL videos. The system achieves an overall recognition rate of 40%.

Keywords: Ethiopian Sign Language (EthSL); Hand Gesture; Hand Tracking; Feature Extraction; Classification.

Table of Contents

List of Figures

List of Tables

Acronym

ArSL	Arabic Sign Language
ASL	American Sign Language
Auslan	Australian Sign Language
BD	Block Division
BSL	British Sign Language
CGS	Candidate Gesture Selection
CS	Color Space
CSL	Chinese Sign Language
EMA	Ethiopian Manual Alphabet
EthSL	Ethiopian Sign Language
GF	Gabor Filter
HMM	Hidden Markov Model
HMTD	Hand Movement Trajectories Determination
KNN	K-Nearest Neigbour
MHD	Modified Hausdorff Distance
NMS	Non-Manual Signs
NZSL	New Zealand Sign Language
PCA	Principal Component Analysis
PSL	Persian Sign Language
SD	Signer Dependent
SI	Signer Independent
SL	Sign Language
SLR	Sign Language Recognition
SVM	Support Vector Machine
TSL	Taiwan Sign Language

CHAPTER ONE

INTRODUCTION

1.1. Background

According to International Labor Organization (ILO) [1], disability is defined as a state in which functional limitation and/or impairments are causative factors of the existing difficulties in performing one or more activities which, are generally accepted as essential and basic components of daily living, such as self-care, social relations and economic activity.

Disabilities can further be categorized into: Hearing and/or Speaking disabilities, Intellectual disability, Hand/Arm problems, Total/Partial Blindness, Leg problems, Leprosy, Overlapping, and other types [1]. Among these categories hearing and/or speaking disabilities is usually referred as 'deaf' with lower case 'd' to mean a person who has some hearing impairment and with upper case 'D' to describe a community whose members may or may not be able to hear but for which the medical condition of deafness is somehow relevant to them [2]. According to WHO report in 2001, 250 million people in the world have disabling hearing impairment (either before or after birth) in which two-third of these people live in developing countries like Ethiopia.

Speech in general has two components; auditory and visual [2]. Hearing impaired or deaf people depend on the visual component in verbal communication, using a technique called speech-reading. Hearing people also tend to use this technique in noisy environment where it's impossible to hear the auditory signal. However, speech-reading is a difficult task, since many speech-sounds either look alike or are only partly visible. This in return resulted in the creation of a language that can be used by deaf people to communicate with themselves and/or with other hearing people. This language is known as Sign Language [2]. Even though, there is considerable number of hearing impaired people around the world, there exists a sign language that is identified by the country where they are used. For example, British Sign Language (BSL), Australian Sign Language (Auslan), American Sign Language (ASL), Taiwan Sign Language (TSL), New Zealand Sign Language (NZSL), Chinese Sign Language (CSL), Arabic Sign Language (ArSL), Persian Sign Language (PSL), Ethiopian Sign Language (EthSL) and so on.

However, regardless of where they are being used there is still some similarity between sign languages. For instance, EthSL is believed to be originated from ASL with some influence from the Nordic countries [3].

In sign language there are different signs that represent letters, numbers and frequently used words. Sign language is not spoken or heard but it's a visual language seen by the eye and expressed by the movements of hand, face and the body [2]. In addition to this, there are four components of a sign language notation [2]: hand shape, location, movement and orientation or direction of signs.

- ✓ Hand Shape – particular orientation of the hand and fingers. In sign language, the actor is the right hand whereas the left hand is used only to help the right hand in some circumstances where one hand is not enough. Fig 1.1 (a) below depicts the hand shape that is used for the Ethiopian Manual Alphabet (EMA) ሠ. In cases where the signer is a left handed, he/she have to communicate that they are left hand, so that the other individual can adjust to the situation.

(a) (b)

Fig 1.1 EthSL Character '**ሠ**' and its Variants; (a) Hand shape for base character (ሠ),(b) Hand movement used to create variants of base (ሠ)

- ✓ Movement – the movement of the hand also makes differences in the character to be represented by a given sign. EthSL just like Amharic language uses seven orders of character which have the same sign but differentiated by the movement of the hand to create an alphabet. Fig 1.1 (b) above depicts the movement of hand to create the alphabet variants of EMA ሠ.
- ✓ Location – the position where you put your hand on. Usually you have to put your hands around your chest but sometimes it may be located in other parts, for instance: face, head, shoulder, etc. Location sometimes creates differences in the meaning of a sign.

- ✓ Orientation or direction of signs – in sign language, as visual means of communication, the palm of the hand should face the audience with the arm being held in an easy to see position near the body.

Any sign language in general and EthSL in particular uses two methods to create a word or sentence [4]. The first method is by concatenating each manual alphabet to one another and the second method is by using a single or multiple signs that stands for a given word form. The EthSL has 34 base alphabets where each base alphabet has 6 other variations. And the manual sign for the six variants are created with the same hand posture or form as the base alphabet followed by unique hand movement trajectories for each variants as shown in Fig 1.1 (a) & (b). It also uses gestures which represent whole words in the Amharic language as shown in Fig 1.2.
In addition to this the hearing impaired also use some natural signs that is used by other societies. For example, the sign used for "Good Bye" is used by both societies. Fig 1.2 below shows an example of Amharic word created using dedicated signs.

(a) (b)

Fig 1.2 EthSL Word created using dedicated Manual Signs; (a) Sign for 'Father', (b) Sign for 'Mother'

Sign Language Recognition is the translation of signed characters and words into their equivalent character or word text. In doing so, different researchers have attempted to come up with a recognizer based on static image or continuous video inputs. Sign language recognition is a non-trivial task because sign languages (SLs) are made up of thousands of different signs; each differing from the other by minor changes in motion, hand shape, location or Non-Manual features (NMFs). In this regard, there are challenges faced by recognition systems that are related to the multimodal nature of the visual cues (fingers, lips, facial expressions, body pose, etc.), as well as technical limitations such as spatial and temporal resolution and unreliable depth cues [5]. The challenge also comes in trying to recognize large lexicons of signs, for even the relatively straight forward task of citation-form, dictionary look-up [6]. The emergent solution

was to recognise the subcomponents, then combine them into words during the recognition process. A sub-unit based SLR was suggested by Cooper et al [6] that has two stages. In the first stage, sign linguistics sub-units are identified. In the second stage, these sub-units are combined together to create a sign level classification model.

1.2. Statement of the Problem

According to Ethnologue listing [7] there are a total of 1.2 million estimated users (both deaf and hearing people) of Ethiopian Sign Language (EthSL). Regardless of this number, still there is a clear communication gap between the deaf and hearing. In order to narrow this gap, different measures are considered; among those measures sign language training to hearing people and/or coming up with a system that facilitates communication between deaf and hearing people is worth mentioning.

With respect to the second option several researches have been conducted to come up with a system that converts sign language to text or vice versa for different sign languages all over the world. For example, Arabic Sign Language (ArSL) alphabet recognition system was built using polynomial classifiers as a classification engine for the recognition [8]. A neural network based system was also developed for recognition of Persian Sign Language (PSL) [9]. In the context of our country, Admasu and Raimond [10] have developed a recognition system for Ethiopian Manual Alphabet (EMA) which converts a given EMA into voice. The limitation of this research is that it only concentrates on recognition from static images; and hence further research needs to be done so as to recognize from continuous pictures or videos of sign languages. Accordingly, Tsegaye and Raimond [11] developed a recognition system which extracts candidate EMA gestures from continues pictures or video frames along with the hand movement required to finger spell EMAs, focusing on only EthSL alphabets (EMA). However, EthSL uses finger spelling or manual alphabets for names, technical terms, & sometimes for emphasis. In addition, most sign languages including EthSL use signs that stand for word forms such as noun, verb or adjective. Tsegaye and Raimond [11] recommend further research to come up with a recognition system that works at word or sentence level.

Words in EthSL dictionary are not only created by concatenating EMAs but also using signs that stands for a word. This necessitates the need to extend Amharic Sign Language recognition at

word-level. Therefore, the purpose of this study is to come up with a recognition system that converts EthSL word into equivalent text word representation.

To this end, the present study explores answers to the following research questions:

- ✓ What are the special features of word-level EthSL signs?
- ✓ How hand gesture is identified from sequence of video frames captured during sign communication?
- ✓ What are the suitable techniques for developing word-level SLR?
- ✓ What are the challenges in developing EthSLR system?

1.3. Objective of the study

1.3.1 General Objective

The main objective of this research is to design and implement Isolated word-level Ethiopian Sign Language Recognition System for EthSL words created by combining one or more signs with some sort of motion attached between them.

1.3.2 Specific Objectives

Specific objectives of this study are the following:

- ➢ To review literature on other sign language recognition systems for the identification of best practice that can be adopted and implemented for this work.
- ➢ To identify special features of word-level signing of Ethiopian Sign Language.
- ➢ To select and implement a technique that can identify hand gesture from a given sequence of video frames.
- ➢ To develop a prototype EthSL recognizer at word-level.
- ➢ To evaluate the performance of the prototype system on test dataset.

1.4. Scope and Limitation of the Study

The focus area of this study is to come up with a recognition system that recognizes EthSL gestures which represent a whole word into its equivalent Amharic word text. Since there is no standardized corpus that can be used to train and test the recognizer the researcher captured several videos of signing. For this study, isolated videos for each word were captured because of unavailability of gesture corpus, as well as the dynamic nature of continuous signing that need

advanced video and image processing tool. In addition to this, in order to deal with manual sign performed using either one hand or both hands, the body part above the abdomen were considered as the camera's field of view (FOV). So the proposed system has made use of the captured videos that contains manual sign articulators and recognizes the important ones accordingly. In addition to this, the researcher collected both the training and testing dataset from the same individual and hence the recognition system developed in this study is signer-dependent.

As pointed out in section 3.2 of this study only 10% of those signs that are frequently used are included in the training as well as testing datasets. In addition to this only those words with no occlusion what so ever were selected for this study. The other limitation of this system was that it only used single person to capture the dataset so it is signer dependent.

Because of the absence of standardized corpus the researcher was forced to capture the training as well as testing dataset videos from selected signer. During the course of action in coming up with the system, the main challenge faced was during the data collection where the selected signer's skin color was a bit tough for the skin detection module. In this regard the researcher was forced to change signer from time to time until the appropriate result is obtained.

In addition to this, the video capturing environment and the camera quality itself was also another challenge. The lighting condition of the room where the videos were captured was very challenging for the researcher. All this in return have affected the data quality of both the training and testing dataset which in return affected the performance of the system in terms of its accuracy in recognizing a given sign.

1.5. Significance of the Study

The research study is important for the following purposes:

- ✓ The main objective of this research is to come up with a recognition system for EthSL. This have a significant contribution towards narrowing the communication gap that exists between hearing and hearing impaired individuals.
- ✓ This extends the state-of-the art for designing a full-fledged SLR system for EthSL so that it can be integrated into portable devices such as smart phones, laptop, tablets, and others.
- ✓ As an academic exercise, this research will serve to fulfill the requirement of the program the researcher is engaged in, and increase the level of experience in the research area.

1.6. Methodology of the Study

The methodology of the study includes steps to be followed and tools to be used in doing the research work. Therefore, in this study an experimental research design is employed. The steps followed and tools used are discussed as follows:

1.6.1. Review of related Literature

Literature review was employed to get an overall understanding of both the linguistics of EthSL and recognition of sign language at large, and a review of related studies were made so that it is easier for the researcher to identify the gap. For this purpose, reference to books, journal articles, conference proceedings and other online scholarly materials were reviewed at each stage of the design.

1.6.2. Data collection and preparation

A pre-luminary observation on how hearing impaired individuals communicate to each other and with hearing individuals was conducted so that the researcher has a clear understanding of the communication process. Therefore, this activity is carefully employed in order to define the research objective and come up with the recognition system. In addition to this the researcher have made use of video camera to capture videos of the communication process which is used as data set to train as well as test the recognition system in the course of this study.

1.6.3. Video Processing and Segmentation

In this stage of the study the captured video passes through the process of video representation, video segmentation, and feature extraction in order for the data to be used in the proposed recognition system. In addition to this since this study is an extension of the works done by [11], it adopted techniques used by this researcher for activities related to word recognition that is created by finger spelling. But as it has been pointed out in previous topics there are also words that are created using gestures which represent a whole word. Handling this sort of words need to take into consideration not only the hand of the signer but also the whole body above the abdomen which comprises of signers hands, head & face, lips, and body pose.

In this regard, the researcher used two modules to implement the system. In the first module the researcher used skin detection and segmentation tools and techniques in order to extract manual

features from signer's hand. On top of the works done by [11], the second module is dedicated to hand tracking and hand motion determination using the trajectory of the hands' centroid as described in [12]. It combines a discriminative method for selecting skin-colored regions with a generative method for characterizing hand configurations and locating images of hands in various articulated poses. This permits a fairly robust estimation of hand trajectories [5].

1.6.4. Model preparation and Evaluation

The researcher used Matlab as an implementation tool for video processing and segmentation and train and tests the system using the data set collected from the signers. After the successful creation of the hand model using Principal Component Analysis (PCA), the next step is evaluating the model with different collection of inputs. The researcher considers the accuracy of the system in recognizing the EthSL as a performance measure and the error report is used as a feedback to refine the model again and again until better performance is obtained.

Given the total number of training set the accuracy is measured in terms of the total number of correct recognition out of the total dataset.

1.7 Outline of the thesis

This thesis is organized into five chapters. The first chapter gives an introduction about sign language in general and Ethiopian sign language in particular and it also provides brief information about the problem, objective, scope, and methodology of the study. The second chapter is all about review of literatures to grasp both theoretical and empirical understanding of sign language recognition. The overall system design and implementation is briefly discussed in chapter three. Experimental results and discussion of the results are presented in the fourth chapter in which every finding is explained based on the theories and the system architecture. Finally, conclusion and recommendation for future work is presented in chapter five.

CHAPTER TWO

LITERATURE REVIEW

A sign language is a language which, instead of acoustically conveyed sound patterns, uses manual communication and body language to convey meaning. It is a means of gestural communication either between deaf people, or between hearing people and the deaf community, in which postures[1] and gestures[2] have assigned meanings with a proper grammar. According to Wolde [13], sign language is a fully-fledged natural language with its own syntax and grammar rules. In linguistic terms, sign languages are as rich and complex as any oral language, despite the common misconception that they are not "real languages". Professional linguists have studied many sign languages and found them to have every linguistic component required to be classed as natural languages [14].

Like any other verbal language, its discourse comprises of well structured rendering and reception of non-verbal signals according to the context rules of the complex grammar. Postures are the basic units of a sign language, and when collected together over a time axis and arranged according to the grammar rules, they reflect a concrete meaning. For example, finger spelling in any sign language is communicated by making postures (static signs) for each letter (“a”, “b”, “c”, “*ʊ*”, ”*ơ*”, “*ȿ*”, etc) but in the case of continuous discourse, most signs comprise of gestures (dynamic signs) Rung-Hui et al (1998) cited in [4].

2.1. Sign Language

Sign languages can be categorized according to their major articulations [4]. Manual signs are performed using hands while non-manual signs (NMS) mainly include facial expressions, body movement and torso orientation. Although manual signs constitute a large proportion of sign language vocabulary, NMS also own a significant share to convey the whole context. Sign languages, like oral languages, organize elementary, meaningless units into meaningful semantic units. It is composed of three-dimensional (3D) manual and non-manual features. The elements of a sign are **H**and-shape (or Hand-form), **O**rientation (or Palm Orientation), **L**ocation (or Place

[1] is a static gesticulation of an articulator (hand, eyes, lips, body)

[2] is a physical movement of the hands, arms, face, and body with the intent to convey information or meaning.

of Articulation), **M**ovement, and Non-manual markers (or Facial **E**xpression), summarized in the acronym **HOLME**.

Hence, it differs from a spoken language in a way that a spoken language structure uses the words in a sequential manner but the SL structure allows manual and non-manual components to be performed in parallel. Another unique feature of SL over any spoken language is its capability to convey multiple ideas at a single instant of time. For example, a signer can tell about an incoming person with the help of his left hand while his right hand can be used in parallel to report the sitting people. Therefore, an ideal SL recognition system needs to follow Multi-Modal Approach, since it requires not only simultaneous observation of manual and NMS but also understanding situations shown by individual articulators [4].

2.1.1 Manual Signing

Sign linguists generally distinguish the basic components (or phoneme subunits) of a sign gesture as consisting of the hand shape, hand orientation, location, and movement. Different researchers have tried to come up with a phonological model based on the simultaneous and sequential occurrence of signs [15]. Stokoe (1960) cited in [15] proposed the first model that emphasized the simultaneous organization of these subunits. Whereas Liddell and Johnson (1989) cited in [15] come up with Movement[3] - Hold[4] model that emphasized sequential occurrences. The computational framework adopted for SL recognition must be able to model both structural occurrences; therefore, more recent researches such as Brentari (1995), Perlmuttar (1993), Sandler (1989), and Wilbur (1993) cited in [15] seem to aim in representing both the simultaneous and sequential structure of signs.

Manual signs can be used to create words and/or sentences using isolated signing and continuous signing respectively, in the later case, the hand(s) need to move from the ending location of one sign to the starting location of the next. In the same fashion, hand shape and hand orientation also change from the ending hand shape and orientation of one sign to the starting hand shape and orientation of the next. These inter sign transition periods are called movement epenthesis and are not part of either of the signs.

[3] defined as periods during which some part of the sign is in transition, whether hand shape, hand location, or orientation.

[4] brief periods when all these parts are static.

As described above, in continuous signing, transition from one sign to the next occur with the addition of movement epenthesis, hold deletion, metathesis, and assimilation that do not change the meaning of the sign. However, there are other systematic changes to one or more parts of the sign which affect the sign meaning (Klima and Bellugi (1979) as cited in [15]). These are:

- Temporal Aspect – the hand shape, orientation, and location of the sign are basically the same as in its lexical form but the movement of the sign is modified to show how the action is performed with reference to time.
- Aspectual Inflections – the meaning conveyed through these inflections are associated with aspects of the verbs that involve frequency, duration, recurrence, permanence, and intensity, and the sign's movement can be modified through its trajectory shape, rate, rhythm, and tension.
- Person Agreement (first person, second person, or third person) – the verb indicates its subject and object by a change in the movement direction, with corresponding changes in its start and end locations, and hand orientation.
- Number Agreement and Distributional Inflection – to show the number of persons in the subject and/or object, or show how the verb action is distributed with respect to the individuals participating in the action.

According to Ong and Ranganath [15] some aspects of signing impact the methods used for feature extraction and classification, especially for vision-based approaches. First, while performing a sign gesture, the hand may be required to be at different orientations with respect to the signer's body and, hence, a fixed hand orientation from a single viewpoint cannot be assumed. Second, different types of movements are involved in signing. Generally, movement refers to the whole hand tracing a global 3D trajectory where the hand moves in a circular trajectory. However, there are other signs which involve local movements only, such as changing the hand orientation by twisting the wrist or moving the fingers only. This imposes conflicting requirements on the field of view; it must be large enough to capture the global motion, but at the same time, small local movements must not be lost. Third, both hands often touch or occlude each other when observed from a single viewpoint and, in some signs the hands partially occlude the face. Hence, occlusion handling is an important consideration.

2.1.2 Non-Manual Signing

In addition to the manual sign, sign languages convey much of the information through non-manual signs. Non-manual features include position of upper torso, mouth pattern, head and shoulder movement, facial expression and eye-gaze, etc that can be used in various combinations to show several categories of information, including lexical distinction, grammatical structure (questions, negation, relative clauses, boundaries between sentences, and the argument structure of some verbs), adjectival or adverbial content, and discourse functions.. For example, if you are mad at someone, or about something, you may not have to use even one sign. You can just show it by the expression on your face. Or, if someone asked you a "yes" or "no" question, you could simply shake your head accordingly. Non-manual markers are those additional items listed below that are other than actual signs [16].

- ✓ Head nods
- ✓ Raised eyebrows
- ✓ Tilted head
- ✓ Pursed lips
- ✓ Eye shifts
- ✓ Eye gazes
- ✓ Facial expressions (smile, anger, frown, puzzled look, etc)
- ✓ Body shifts / movements

Facial expressions can be divided into two as lower and upper facial expressions. The lower facial expression provides information about a particular sign through the use of the mouth area (lips, tongue, teeth, check). On the other hand the upper face expression occur in conjunction with head and body movement which convey information indicating emphasis on a sign or different sentence types (i.e., question, negation, rhetorical, assertion, etc.), and involve eye blinks, eye gaze direction, eyebrows, and nose.

2.1.3 Integration of Non-Manual Signs

Non manual signs (NMS) are those visual signals which are not conducted by hand or arms but they are shown by facial expressions and body movements. They are an important asset of any sign language, hence, a complete sign language system must have provision to recognize NMS in

parallel with manual signs. Many researchers are trying to incorporate NMSs for a complete recognition system. For example, Starner et al (1998) cited in [4] detected some muscular movement based NMS (smile, disgust, and fear) and head movement by installing sophisticated sensors in a cap or eye glasses. As describe in [15] search for works in automatic NMS analysis revealed none that capture the information from all non manual cues of facial expression, head and body posture and movement. But Ming and Ranganath (2002), Sako and Smith (1996) classify facial expression only, while others Erdem and Sclaroff (2002), and Xu et al (2000) classify head movement only. Ma et al (2000) cited in [15] modeled features extracted from lip motion and hand gestures with separate HMM channels using a modified version of Bourlard's multi-stream model and resembling Vogler's Parallel HMM. The assumption was that each phrase uttered by the lips coincides with a sign/phrase in the gesture. However, in general NMS may co-occur with one or more signs/phrases, and hence a method for dealing with the different time scales in such cases is required. Also, in Ma et al (2000), the lip motion and hand gesturing convey identical information, while in general, NMS convey independent information, and the recognition results of NMS may not always serve to disambiguate results of hand gesture recognition.

2.2 Gesture Taxonomy

Webster's Dictionary defines a gesture as, (1) "a movement usually of the body or limbs that expresses or emphasizes an idea, sentiment, or attitude"; (2) "the use of motions of the limbs or body as a means of expression". Most of the gestures are performed with the hand, the face and the body. In the case of hand gestures, the shape of the hand, together with its movement and position with respect to the other body parts, forms a hand gesture. Gestures are used in many aspects of human communication. They can be used to accompany speech, or alone for communication in noisy environments or in places where it is not possible to talk. In a more structured way, they are used to form the sign languages of the hearing-impaired people [17].

In taxonomies of communicative hand/arm gestures, sign language (SL) is often regarded as the most structured of the various gesture categories [15]. It is used as a characterization for several classes of gesture. In this regard several taxonomies have been suggested in the literatures that vary from author to author, Kendon's (1992) continuum and Quek's (1994) taxonomy as cited in

[15] are worth mentioning. Fig 2.1 below shows the taxonomy that was developed by Quek which seems most appropriate for HCI purposes in general and SL in particular.

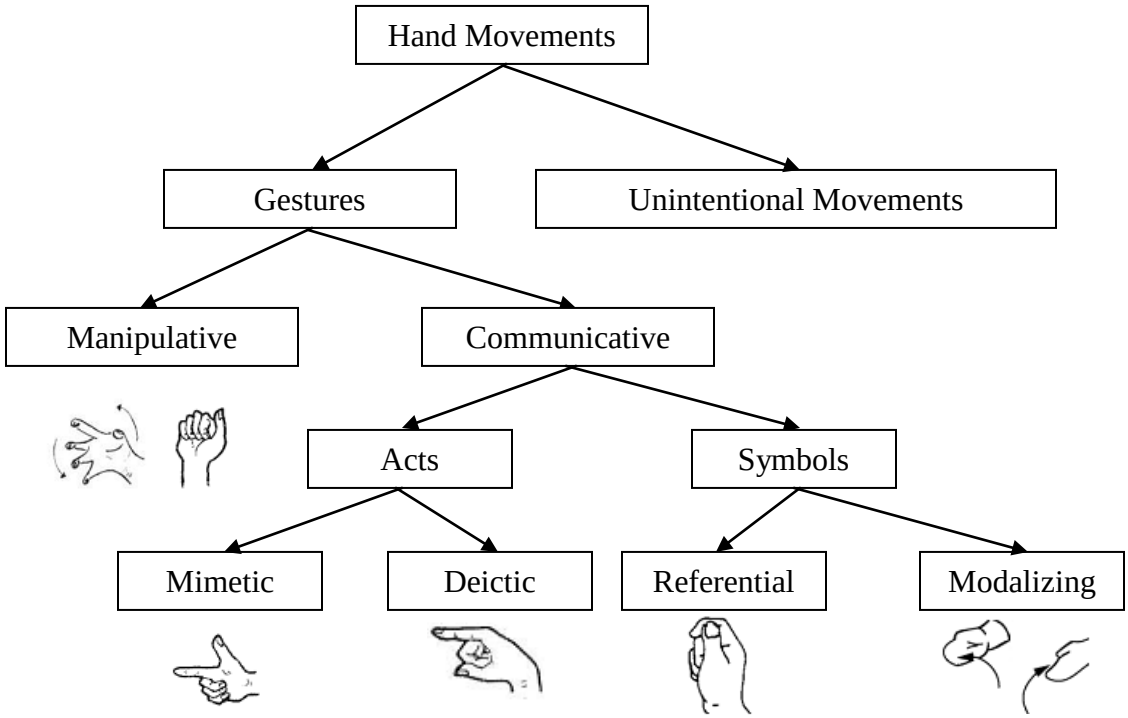


Fig 2.1 Taxonomy of Hand Gestures Taken from [18, 17]

As it can be seen from fig 2.1 Quek divides communicative gestures as *acts* and *symbols*, and SL is regarded as largely symbolic (i.e., the meaning is not transparent from observing the form of the gestures) and referential since modalizing gestures are defined as those occurring in conjunction with another communication mode, such as speech. Sign gestures are not all purely symbolic, and some are in fact *mimetic* or *deictic* (these are defined by Quek as *act* gestures where the movements performed relate directly to the intended interpretation). *Mimetic* gestures take the form of pantomimes[5] (these are described by Kendon's continuum) and reflect some aspect of the object or activity that is being referred to. For example, a smoker going through the motion of "lighting up" with a cigarette in his/her mouth indicates that s/he needs a light, or hand is like holding a gun. *Deictic* or pointing gestures are extensively used in SL as pronouns or to specify an object or person who is present or to specify an absent person by pointing to a

[5] gestures that display an invisible object or tool (e.g., making a fist and moving to indicating a hammer)

previously established referrant location. SL gesture can be used as an input to come up with a system that can automatically recognize SL(s) and it can also offer a useful benchmark for evaluating hand/arm gesture recognition systems [15]. As describe in [17] the next two types of symbol gestures are *referential* and *modalizing*. *Referential* gestures are used to refer to an object or a concept independently. For example, rubbing the index finger and the thumb in a circular fashion independently refers to money. *Modalizing* gestures are used with some other means of communication, such as speech. For example, the sentence “I saw a fish. It was this big.” is only meaningful with the gesture of the speaker.

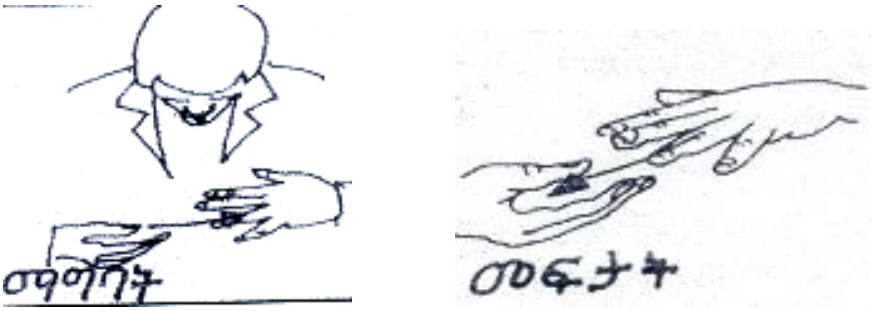
2.3 Ethiopian Sign Language

Ethiopian Sign Language (EthSL) is the natural language used primarily by about a million Ethiopian Deaf Community. EthSL, as other known Sign Languages (SLs), is accepted as minority language, which coexist with majority languages [19] and is native language for many deaf people. As any other sign languages ESL is a language which uses manual communication, body language and lip patterns instead of sound to convey meaning, simultaneously combining hand shapes, orientation and movement of the hands, arms or body and facial expressions to express fluidly a speaker's thought.

Since Amharic is the official language of Ethiopia, EthSL is based on the Amharic language. Amharic, a Semitic language of Ethiopia, has a phonetic inventory of thirty consonants (27 simple and 3 complex) and six vowels [20], including a number of consonant-vowel combination giving a total of 231 distinct symbols.

Accordingly, in EthSL, there are 34 base alphabets where each base alphabet has 6 other variations that are created with the same hand posture or form as the base alphabet followed by unique hand movement trajectories for each variation; this is referred to EthSL Manual Alphabet, in which words can be created using finger spelling. Finger spelling is a method of representing words using the signs of each letter the word comprises. If the word has seven characters, it will be represented by the combination of seven signs, one for each letter.

In addition to this, EthSL have around 1422 signs that are assigned to words or actions on different basis in Amharic Language [21]. Some signs are descriptive; they convey ideas (for instance Marriage and Divorce, shown in Fig 2.2. and other signs are of pointing to things rather than showing signs.



(a) (b)

Fig 2.2 Descriptive signs that convey ideas; (a) Sign for Marriage, (b) Sign for Divorce

The Deaf also use some natural signs along with sign-language. If we take “Good Bye” the sign that is used in the other society is also used in the deaf society. The other property of sign language is the use of arbitrary or local signs that are only used in small communities like a school, a village or an organization.

In many cases like names of persons, countries, cities and some other common words, the signs are first letter based. The sign for the name “**አበበ**” for instance, is the sign for “**አ**” and touching part of your face. There is hand location restriction in representing the name of males and females as depicted in Fig 1.2. It is not allowed to touch above or by the side of your eyes to represent a female name and below your eyes to represent a male name.

First letter based signing is not restricted to names only but common words like Enjera “**እንጀራ**” are also represented by using the sign of “**አ**” and adding some movement that shows method of baking “Enjera”.

2.3.1 Spatial Grammar and Simultaneity

Sign languages exploit the unique features of the visual medium. Oral language is by and large linear; only one sound can be made or received at a time. Sign language, on the other hand, is visual; hence a whole simultaneous scene/information can be taken in at once. Information can be loaded into several channels and expressed simultaneously. For example, in Amharic one could utter the phrase, “{mgb belahu} [**ምግብ በላሁ**]” (I ate food). To add information about the amount, one need to add a phrase “{beTam bzu}[**በጣም ብዙ**]” (too much). He/she would have to make even longer if he/she want to show the feeling, like satisfaction or dissatisfaction. But in ESL, all information can be conveyed at the same time by changing the way the hand moves

(fast, slow, strong, repeated) and at the same time by applying non-manual features such as body posture, and facial expression. Even though the Amharic phrase "{beTam bzu mgb belahu}[**በጣም ብዙ ምግብ በላሁ**]"(I ate to much food) is longer than "{mgb belahu} [**ምግብ በላሁ**] "(I ate food), in EthSL the two signs may have equal length [13], meaning to the difference in meaning of the sentence might be shown using non-manual sign while using the same manual sign for both sentences.

2.4 Sign Language Recognition

Deaf people have created and used signs among themselves. These signs were the only form of communication available for many deaf people. Within the variety of cultures of deaf people all over the world, signing evolved to form complete and sophisticated languages. Normally, there is no problem when two deaf persons communicate using their common sign language. The problem arises when a deaf person wants to communicate with a non-deaf person. Usually both will be dissatisfied in a very short time.

Communication in commonplace situations (for example, clinics, hospitals, post offices and police stations) is difficult if not impossible without Sign Language interpreters. To this end several researchers have come up with different computerized systems to narrow this communication gap by converting Sign Language to Text / Speech or vice versa. Therefore, sign language recognition is one area of study that focus on coming up with a system that converts a given sign language to text. The methodologies used in Sign Language recognition can be categorized into several types based on feature extraction methods, input type and the hardware dependency. Traditionally there have been three main types of sign language recognition: hand shape classification, isolated sign language recognition, and continuous sign classification Sandjaja and Marcos (2009) as cited in [23]. In addition to this sign language recognition can be either vision-based or device-based methods based on how the features of gestures are extracted. In this study the researcher follows the categorization used in [4] because of the fact that the categorization is somehow an integration of the categorization based on feature extraction methods, input type and the hardware dependency.

In general, a sign language recognition system requires the following components:

- ✓ Hand and body parts (face, shoulders, arms . . .) detection, segmentation, and tracking.
- ✓ Analysis of manual signals

- ✓ Analysis of non-manual signals
- ✓ Classification of isolated and continuous signs

2.4.1 Virtual Reality (VR) Gloves

In the glove based system, the signer wears some instrumented gloves equipped with a number of sensors (mechanical or optical/laser installed on the finger joints, wrist and palm) which generate a set of electrical signals that characterize the intended sign. Based on the collected input the processing unit compares the set of static sign samples with existing templates and generates output. Fig 2.3 shows examples of data gloves. This approach forces the user to carry a load of cables which are connected to the computer and hinders the ease and naturalness of the user interaction [18].

Fig 2.3 Data Gloves

According to Khan et al [4], these static signs are easy to incorporate due to their defined boundaries but in the case of continuous signing, a 2D motion trajectory is formed and matched with an existing template or a learnt model. Most of these researches involve recognition of a small set of words, mainly static postures or finger spellings.

Khan et al [4] noted that the performance and effectiveness of these methods is heavily dependent on the sensor density; for example, more sensors can be added to measure the elbow bends. Rung Hui et al (1998) as cited in [4] proposed a more sophisticated embedded sensor based solution for continuous signing in real time situations. A more complex hand glove, with 10 finger sensors for one hand along with a 3D tracker for orientation, is used for detecting continuous hand gestures by their motion trajectories. In this method, the motion trajectory of a gesture is divided into 10 vectors whose relative cosines, turning points and orientations are

matched with stored templates. Thad and Starner (2003) as cited in [4] suggested using accelerometers for providing complementary information about the hand/wrist rotation and orientation. Even though, the use of more sensors in a single design may increase the burden on the processor, the proposed accelerometers work in a master-slave topology with increased local computation and reduced load on the central processor. Perrin et al (2004) as cited in [4] proposed a low cost and short vocabulary single laser/optical based gesture recognition system in which a laser beam is transmitted and the amount of reflected energy is correlated with a reference signal to measure the displacement from the point of contact.

Cyber gloves have been widely used in most of previous works on sign language recognition [24]. Kudos (1996) cited in [24] reported a system using power gloves to recognize a set of 95 isolated Australian sign languages with 80% accuracy. Liang and Ouhyoung [12] employed the time-varying parameter threshold of hand posture to determine end-points in a stream of gesture input for continuous Taiwanese sign language recognition with the average recognition rate of 80.4% over 250 signs. In their system HMM was employed, and data gloves were taken as input devices. According to [23], Ibrahim Mohammad (2006) represented a pioneering work on the automation of the ArSL recognition using the Cyber Glove as an interface device and principal components analysis as the feature extraction algorithm.

The main advantage of glove based systems over vision based systems emanates in the fact that there is no need for hand image segmentation, a process that is complicated and computationally expensive [25]. Another advantage of the glove based system is that camera is not required as in the vision-based. And it is also efficient and robust due to smaller vocabulary set. But, for practical application of translation systems, it is inconvenient to wear gloves because it severely affects the user independence due to a dense mesh of installed sensors.

As it has been discussed in the previous topics, NMS play a great role in transferring important information to the other parties, through the use of facial articulators (eyes, eye-brows, etc) and body movement. They are also considered as an important asset of any sign language, therefore complete sign language recognition system must have a means to recognize NMS in parallel with manual signs. However, sensor based gloves or laser based tracking methods of data acquisition fails to detect NMSs that are one of the major components that needs to be considered for complete and effective communication with deaf individuals.

2.4.2 Vision-Based Methods

In order to increase signer independence and minimize the drawbacks of device/glove-based methods, researchers have come up with another method called Vision-based system in which a signer performs the different signs which are then captured with a video camera. In order to capture the whole signing space, the entire upper body needs to be in the camera's FOV [15]. Then from the acquired video stream, sign articulator (hand, face, mouth, etc) regions are segmented followed by a feature extraction stage and finally a classification stage [25]. This arrangement is termed as the second person view. Some researchers like Starner et al (1998) cited in [4] worked on a first person view, in which, the signer needs to wear the camera (in their hat, or eye glasses) covering the person's natural signing space.

According to Murthy and Jadon [18] computer vision based techniques are non invasive and based on the way human beings perceive information about their surroundings. Vision-based interaction is a challenging interdisciplinary research area, which involves computer vision and graphics, image processing, machine learning, bio-informatics, and psychology. To make a successful working system, there are some requirements which the system should have [18]:

1. *Robustness* – In the real-world, visual information could be very rich, noisy, and incomplete, due to changing illumination, clutter and dynamic backgrounds, occlusion, etc. Vision-based systems should be user independent and robust against all these factors.
2. *Computational Efficiency* – Generally, Vision-based interaction often requires real-time systems. The vision and learning techniques/algorithms used in Vision-based interaction should be effective as well as cost efficient.
3. *User's Tolerance* – the malfunctions or mistakes of Vision-based interaction should be tolerated. When a mistake is made, it should not incur much loss. Users can be asked to repeat some actions, instead of letting the computer make more wrong decisions.
4. *Scalability* – The Vision-based interaction system should be easily adapted to different scales of applications. For example, the core of Vision-based interaction should be the same for desktop environments, Sign Language Recognition, robot navigation and also for Virtual Environment.

2.4.2.1 Skin Color Detection Methods

Skin detection is the process of finding skin-colored pixels and regions in an image or a video, in which skin-tone color is an indication of the existence of humans in such media [26]. Since sign articulators such as hand and face have skin-tone color, detection and tracking based on skin color are achieving robust and accurate solution.

According to [4] hands are detected using their natural skin colors along with various other features (position, velocity etc) and matched with an existing set of templates. The signer is often required to wear long-sleeved clothing, with restrictions on other skin-colored objects in the background [15].

Skin color detection systems are severely affected by varying illumination, complex background, the signer's ethnicity (skin color), and articulator occlusion [4]. As was highlighted by [27] when building a system, that uses skin color as a feature for hand and/or face detection, one usually faces three main problems. First, what colorspace to choose; second, how exactly the skin color distribution should be modeled, and finally, what will be the way of processing of color segmentation results for face or hand detection. Skin color detection can either be pixel-based or region-based methods, in which pixel-based classify each pixel as skin or non-skin individually and independently from its neighbors. In contrast, region-based methods try to take the spatial arrangement of skin pixels into account during the detection stage to enhance the methods performance.

2.4.2.2 Colored gloves method

To avoid issues related to skin-color detection, which are caused by varying illumination, complex background, the signer's ethnicity, and articulator occlusion, color coded gloves are used by different researchers. Ken et al (1999) as cited in [4] applied different color on different parts of gesticulated hand (palm, fingers and back) so that each part can be identified independently. These methods are more robust as compared to skin color based method, but still restrict signer's independence.

2.4.2.3 Occlusion

As described in [4], occlusion in hand gesticulation is a normal phenomenon in which articulated parts (hands, face, and body) combine to form a posture, irrespective of the method employed.

Therefore, a recognition system is expected to keep track of articulators even when they are occluded (one hand with or behind the other hand, or face). In case of inter-hand occlusion, instead of individual hand detection, classical skin color based approaches yield a single larger blob. To solve this problem, Qiang et al (2004) as cited in [4] developed an adaptive skin model which classifies the targeted skin pixels out of a larger set of skin similar pixels. Skin similar pixels are those pixels which can belong to a wide range of skin colors. In an input image, true skin color is detected by Gaussian modeling discussed in section 2.5.2.2. Two separable Gaussians model both the classes with the prominent Gaussian for skin pixels and the weaker one for false skin pixels in skin similar space.

In case of occlusion, Morteza et al (2006) cited in [4] predicted the hand gesticulation based on previous hand position and the transitions (learnt in the training stage). In training, either different signers sign in front of the system or annotated sign language databases are used. These video databases contain a number of sign language sentences with different signers, lighting conditions and backgrounds. These benchmarks are recorded and annotated for training, development and testing of SL recognition systems.

As described in [15] some of the works avoid the occurrence of occlusion entirely by their choice of camera angle, sign vocabulary, or by having signs performed unnaturally. Another simplification is to use colored gloves, whereby face/hand overlap becomes straightforward to deal with.

2.4.2.4 Lack of depth information

Apart from occlusion, some manual signs involve the hands motion towards the observer which requires depth information to be incorporated. Similarly temporal information (i.e. past, future) of a sign can also be incorporated by performing it in spatially different location [4]. This is even very challenging for vision based system because determining 3D features from 2D images is impossible.

2.4.2.5 Stereo vision

Stereo-vision based method was suggested as a solution for issues related to lack of depth information [4]. This is because stereo imaging uses multiple images of the same scene taken from different camera locations. Disparity is defined as the relative movement of an object

between two or more views, with the disparity being a function of depth. Dreuw et al (2009) cited in [4] computed a dense disparity map between both acquired images by applying an affine transform on corresponding points in images. Seal et al (2005) cited in [4] used the assumption that objects closer to the camera have a greater disparity of movement between two images, to calculate the distance to the objects. If same techniques are applied in a SL recognition system, a large number of signs can be recognized by analyzing the 3D trajectory instead of 2D.

2.4.2.6 Time of flight camera

Since stereo vision is computationally expensive, the research direction has shifted to alternate range finding methods. Kolb et al (2008) as cited in [4] suggested state-of-the-art technological development called *Time of Flight* (ToF) cameras which can acquire not only intensity image but also a depth image of the scene with precision of a few millimeters. Although a ToF camera can produce intensity image, it normally has low lateral resolution which is unsuitable for processing fine details.

2.4.3 Other approaches to SLR

2.4.3.1 Hand Shape Classification

Previously during the period of 1994-1998, sign language was just string of sign represented by the hand-shape. Therefore, different researchers have put their effort to come up with a hand-shape classification system to recognize Signs. In order to capture the whole signing space, the entire upper body needs to be in the camera's field-of-view (FOV). And the hand (s) can then be located from the image sequence by using color, motion, and/or edge information.

Skin-color detection – Hands are detected using their natural skin colors with the assumption that they are the only skin regions in the camera view. When there are two skin regions resulting from the two hands of the signer, the two biggest skin-colored regions can be selected as the two hands. This approach will fail when the two hands are in contact, forming a single skin-colored region [17].

Motion cues – according to Cui and Wenge (1999) cited in [17] motion information can be highly informative when the hands are the only moving object in the image sequence. However, this assumption can further be relaxed by combining the motion cue with the color cue and assuming that the hand is the only moving object among the skin colored regions.

Edge detection – is the name for a set of mathematical methods which aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities. The points at which image brightness changes sharply are typically organized into a set of curved line segments termed *edges*. Edges typically occur on the boundary between two different regions in an image [28]. The goal of edge detection is to mark the points in an image at which the intensity changes sharply. Sharp changes in image properties usually reflect important invents and changes in world properties. As described in [29] commonly one can use one of the following operators for edge detection e.g. Binary morphology, Canny, Log and Differential operator.

2.4.3.2 Isolated Sign Recognition

Isolated recognition focuses on a single hand gesture that is performed by the user and attempts to recognize it. It also deals with the recognition of signs that are performed alone, without any signs before or after the performed sign [17]. Since isolated words are considered as the basic unit in sign language, many researchers focus on isolated sign language recognition [23].

According to Aran [17] the importance of the research on isolated sign recognition is that it enables the finding of better mathematical models and features that represent the performed sign. Murakami and Taguchi (1991) cited in [17] suggested signer dependent (SD) recognition system to recognize 10 isolated Japanese Sign Language (JSL) using recurrent NN and obtained 96% accuracy. Kadous (1996) cited in [17] also designed a recognizer for 95 Auslan signs using instance based learning and decision trees as classification method and obtained an accuracy of 80% and 15% for SD and signer independent (SI) respectively. On the other hand Fang et al (2004) cited in [17] applied fuzzy decision tree and self organizing feature for SD and maps & hidden markov model (HMM) for SI as a classification method to recognize 5113 isolated Chinese Sign Language (CSL) and the system returned an accuracy measure of 91.6% for SD and 83.7 for SI. Further improvement was also made by Zhang et al (2005) cited in [17] for 102 CSL using Boosted HMMs that in return improved the system accuracy to 92.7% for SD.

2.4.3.3 Continuous Sign Classification

In continuous recognition, user is expected to perform gestures one after the other and the aim is to recognize every gesture that the user performs [17]. The recognition of continuous, natural

signing is very challenging, in terms of both video analysis and linguistics, due to the multimodal nature of the cues (fingers, lips, facial expressions, body pose), extra linguistic elements such as spatial references and pantomime, etc. These fundamental difficulties are joined by technical limitations such as spatial and temporal resolution and unreliable depth cues [5]. As described in [17] it also includes the problem of co-articulation (similar to speech recognition), in which the preceding sign affects the succeeding one, which complicates the recognition task as the transitions between the signs should be explicitly modeled and incorporated to the recognition system. Additional movements or shapes may occur during transition between signs. Liddell and Johnson (1989) cited in [15, 17] called these movements *Movement Epenthesis*.

Starner and Pentland (1995) cited in [17] presented one of the first studies on vision based continuous sign language recognition system and experimented with both colored glove based and skin color based tracking. Their system uses HMMs and a grammar to recognize continuous signing. With colored glove based tracking, they achieve a sign-level recognition accuracy of 99%. Vogler and Metaxas (1997) cited in [17] address the co-articulation effects and model the transition movements between signs that are identified automatically by applying a k-means clustering technique on the start and end points of the signs. They constructed an epenthesis model recognition technique in which each sign is followed by a transition cluster. On a 53 sign vocabulary and 489 sentences, they compare context dependent and independent approaches with or without the epenthesis modeling. They also compare unigram or bigram models for sign recognition. The best accuracy, 95.8%, is obtained with epenthesis modeling with bigrams, in comparison to an accuracy of 91.7% which is obtained by context dependent bigrams without epenthesis modeling. Fang et al (2004) cited in [17] presented a SI continuous CSL recognition with a divide-and-conquer approach. The authors use a combination of a Simple Recurrent Network (SRN) and HMMs, SRN is used to divide the continuous signs into the sub problems of isolated CSL recognition and the outputs of SRN are used as the states of an HMM. The accuracy of the SI tests is 85% and 92.1% for the SD one. Holden et al (2005) cited in [17] presented automatic vision based Auslan recognition system using HMMs to model each sign, with features that represent the relative geometrical positioning and shapes of the hands and their directions of motion. Their system achieved 97% recognition rate on sentence level and 99% success rate at a word level, on 163 test sign phrases, from 14 different sentences. In Fang et al (2007) cited in [17], a methodology based on Transition-Movement models (TMMs) for large

vocabulary continuous sign language recognition is proposed. TMMs are used to handle the transitions between two adjacent signs in continuous signing. The transitions are dynamically clustered and segmented and these extracted parts are used to train the TMMs. The continuous signing is modeled with a sign model followed by a TMM. The recognition is based on a Viterbi search, with a language model, trained sign models and TMM. The large vocabulary sign data of 5113 signs is collected with a sensor glove and a magnetic tracker with 3000 test samples from 750 different sentences. Their system has an average accuracy of 91.9%.

2.4.3.4 Sub-unit Based Approaches

SLR is a non-trivial task. Sign Languages (SLs) are made up of thousands of different signs; each differing from the other by minor changes in motion, hand shape, location or Non-Manual Features (NMFs). While Gesture Recognition (GR) solutions often build a classifier per gesture, this approach soon becomes intractable when recognizing large lexicons of signs, for even the relatively straightforward task of citation-form, dictionary look-up. One of the solutions to this problem is to identify phonemes/subunits of the signs like the phonemes of speech. The advantage of identifying phonemes is to decrease the number of units that should be trained. The number of subunits is expected to be much lower than the number of signs [17] [6]. According to [6] sub-unit based SLR uses a two stage recognition system, in the first stage, sign linguistic sub-units are identified. In the second stage, these sub-units are combined together to create a sign level classifier.

As described in [6] linguists also describe SLs in terms of component sub-units; by using these sub-units, not only can larger sign lexicons be handled efficiently, allowing demonstration on databases of nearly 1000 signs, but they are also more robust to the natural variations of signs, which occur on both an inter and an intra signer basis. This makes them suited to real-time signer independent recognition. However, the phonemes of sign language are not clearly defined Vogler & Metaxas (2003) and Wang et al (2005) cited in [17] used hand shapes, motion types, orientation, or body location as the 4 main sub-unit/phonemes categories.

According to [6] Kim and Waldron (1993) were among the first adopters to use sub-units for SLR, they worked on a limited vocabulary of 13-16 signs, using data gloves to get accurate input information. Vogler and Metaxas (1998) cited in [17] used subunits instead of whole sign which were identified implicitly by looking at the geometric properties of the motion trajectory, such as

hold, line, or plane. Their system achieved 89.9% sign accuracy on a database of 53 signs and 486 sentences in a continuous signing task. In their later study (1999), they define a new model based on a sequential phonological model of ASL, the Movement and Hold model, which states that the ASL signs can be broken into movements and holds that are explicitly defined in the movement-hold model. And they also modeled transition between the signs with the movement epenthesis models. For each phoneme an HMM is trained and the combination of these phoneme models form a sign with the epenthesis models to define the transitions between the signs in continuous signing. Experiments with a 22 word vocabulary, modeled with 89 subunits (43 phonemes, 46 epenthesis models), yield similar word-level recognition results with word and phoneme-based approaches, 92.95%, 93.27% respectively.

In a recent work Vogler and Metaxas (2003) cited in [17], modeled the movement and shape information of the signs in separate HMM channels and used Parallel HMMs for this task. They modeled the right and left hand information in separate channels as well. Their system achieved 87.88% sign-level and 95.51% sentence-level accuracy. Fang et al (2004) cited in [17] presented an automatic subunit extraction methodology for CSL. They used HMMs to model each sign and assume that each state in the HMM represents a segment. The subunits are extracted from these segments by using a temporal clustering algorithm. Their system achieved a 90.5% recognition rate. Wang et al (2005) cited in [17] define *etymon* as the smallest unit in a sign language. The etyma are not automatically extracted and defined explicitly. 2439 etyma are defined for CSL and HMMs are trained for each etymon. They made a comparison between etyma-based and sign-based approaches on a 5100 sign CSL database, collected with a specialized capturing device. However, their analysis shows that the sign-based method has both better accuracy and lower recognition time.

2.5 Color Spaces and Skin Modeling

As it has been discussed in section 2.4.2 vision based SLR can make use of skin color detection in order to recognized skin-toned sign articulators. Skin color has proven to be a useful and robust cue for face and/or hand detection, localization and tracking. Beyond that it is an important cue that people use consciously or unconsciously to infer variety of culture-related (race, health, age, wealth, beauty, etc) aspects about each other [26, 27]. Therefore, applications such as HCI, surveillance cameras, robotics, and vision based SLR can use skin color detection

to design robust and accurate system that make use of human skin color found in an image or video.

According to Elgammal et al [26] even though detecting skin-colored pixels seems a straightforward easy task, it has proven to be quite challenging for many reasons. The appearance of skin in an image depends on the illumination[6] conditions (illumination geometry and color) where the image was captured. We humans are very good at identifying object colors in a wide range of illuminations, this is called *color constancy*. Color constancy is a mystery of perception. Therefore, an important challenge in skin detection is to represent the color in a way that is invariant or at least insensitive to changes in illumination. However, different literature suggested the use of color spaces as a solution to changes in illumination. Another challenge comes from the fact that many objects in the real world might have skin-tone colors. For example, wood, leather, skin-colored clothing, hair, sand, etc. This causes any skin detector to have many false detections in the background if the environment is not controlled. In addition to this, as it is pointed out in section 2.4.2.1 since model preparation is an issue in skin color detection, a researcher have to think of the kind of model to employ for skin color distribution. The next two sections talk about color space and skin classification/model respectively.

2.5.1 Color Spaces

As was highlighted by [26] colorimetry, computer graphics and video signal transmission standards have given birth to many color spaces with different properties. For skin color detection/classification color space choice is usually considered as the first step. According to Forsyth and Fleck (1996) cited in [26] the human skin color has a restricted range of hues and is not deeply saturated, since the appearance of skin is formed by a combination of blood (red) and melanin (brown, yellow). Therefore, the human skin color does not fall randomly in a given color space, but clustered at a small area in the color space. But it is not the same for all the color spaces. In the literature, there are many color spaces (with the aim of finding a color space where the skin color is invariant to illumination conditions), where each color space has its own advantage and disadvantage as compared to the other. Even though, RGB is the default color

[6] an observable property and effect of light.

space, any other color space can be obtained by a linear or non-linear transformation of RGB color space [11].

As was highlighted in [30] the choice of color space determines how effectively we can model the skin-color distribution. The choice of appropriate color space is often guided by the skin detection methodology and the application [31]. Having this in mind, [30] classified the available color spaces as basic color space (RGB, Normalized RGB, CIE-XYZ), perceptual color space (HIS, HSV, HSL, TSL), orthogonal color space (YCbCr, YIQ, YUV, YES), and perceptual uniform color space (CIE-Lab, CIE-Luv). On the other hand, [30] categorized the different color spaces as uniform color space (RGB, Normalized RGB, YCbCr, HIS, TSL) and perceptually uniform color space (CIELAB, CIELUV).

2.5.1.1 RGB

RGB is one of the most widely used color space for storing, representing, and processing of digital image data. It encodes colors as an additive combination of three primary colors: *red* (R), *green* (G) and *blue* (B). RGB Color space is often visualized as a 3D cube where R, G and B are the three perpendicular axes. One main advantage of the RGB space is its simplicity. However, it is not perceptually uniform, which means distances in the RGB space do not linearly correspond to human perception. In addition, RGB color space does not separate luminance and chrominance, and the R, G, and B components are highly correlated. The luminance of a given RGB pixel is a linear combination of the R, G, and B values. Therefore, changing the luminance of a given skin patch affects all the R, G, and B components. Because of this two reasons RGB is not a very favorable choice as a color space for color analysis and color based recognition algorithms.

2.5.1.2 Normalized RGB (rgb)

Normalized RGB (rgb) tries to reduce the dependence on lighting that is easily obtained from the RGB values by a simple normalization procedure:

$$\mathbf{r} = \frac{\mathbf{R}}{\mathbf{R+G+B}} \quad \mathbf{g} = \frac{\mathbf{G}}{\mathbf{R+G+B}} \quad \mathbf{b} = \frac{\mathbf{B}}{\mathbf{R+G+B}} \qquad (2.1)$$

As the sum of the three normalized components is known (r + g + b = 1), the third component does not hold any significant information and can be omitted, reducing the space dimensionality. The remaining components are often called ”pure colors”, for the dependence of r and g on the brightness of the source RGB color is diminished by the normalization [27]. It has been observed that under certain assumptions, the differences in skin-color pixels due to lighting conditions and due to ethnicity can be greatly reduced in normalized RGB (rgb) space. Also, the skin-color clusters in rgb space have relatively lower variance than the corresponding clusters in RGB and hence are shown to be good for skin-color modeling and detection [30].

2.5.1.3 HSI, HSV, HSL – Hue Saturation Intensity (Value, Lightness)

Hue-saturation based color spaces were introduced when there was a need for the user to specify color properties numerically. They describe color with intuitive values, based on the artist’s idea of tint, saturation and tone. These color spaces separates three components: *Hue* (H) – the property of a color that varies in passing from red to green, *Saturation* (S) – the property of a color that varies in passing from red to pink, and *Brightness* (I,V or L) – the property that varies in passing from black to white. The perceptual features of color such as H, S, and I cannot be described directly by RGB. Therefore, non-linear transformations are proposed to map RGB on to perceptual features. The transformation of RGB to HSV is invariant to high intensity at white lights, ambient light and surface orientations relative to the light source and hence, can form a very good choice for skin detection methods. One of the advantages of these color spaces in skin detection is that they allow users to intuitively specify the boundary of the skin color class in terms of the hue and saturation. And also explicit discrimination between luminance and chrominance properties made these color spaces popular in the works on skin color segmentation [26, 27].

$$H = \arccos \frac{\frac{1}{2}((R-G)+(R-B))}{\sqrt{((R-G)^2+(R-B)(G-B))}} \quad (2.2)$$

$$S = 1 - 3\frac{\min(R,G,B)}{R+G+B} \quad (2.3)$$

$$V = \frac{1}{3}(R+G+B) \quad (2.4)$$

2.5.1.4 TSL – Tint, Saturation, Lightness

TSL color space defines color as *Tint*—hue with white added, *Saturation* and *Lightness*. A normalized chrominance-luminance TSL space is a transformation of the normalized RGB into more intuitive values, close to hue and saturation in their meaning.

$$S = [9/5(r'^2 + g'^2)]^{1/2} \tag{2.5}$$

$$T = \begin{cases} \arctan(r'/g')/2\pi + 1/4, & g' > 0 \\ \arctan(r'/g')/2\pi + 3/4, & g' < 0 \\ 0, & g' = 0 \end{cases} \quad \text{Where } r' = r - 1/3,\ g' = g - 1/3 \text{ and r, g come from (2.1)} \tag{2.6}$$

$$L = 0.299R + 0.587G + 0.114B \tag{2.7}$$

Terrillon et al. (2000) cited in [27, 30] have compared nine different color spaces for skin modeling with a unimodal Gaussian joint probability distribution function (pdf) (only chrominance components of the color spaces were used). They argue that normalized TSL space is superior to other color spaces for this task. Brown et al. (2001) cited in [27, 30] has also employed this representation for their approach.

2.5.1.5 YCbCr

YCbCr is one of the color spaces under orthogonal color space class used in TV transmission. It is an encoded nonlinear RGB signal, commonly used by European television studios and also used in JPEG image compression and MPEG video compression. The YCbCr space represents color as luminance (Y) computed as a weighted sum of RGB values, and chrominance (C_b and C_r) computed by subtracting the luminance component from B and R values.

$$\begin{aligned} Y &= 0.299R + 0.587G + 0.114B \\ C_r &= R - Y \\ C_b &= B - Y \end{aligned} \tag{2.8}$$

One advantage of using these color spaces is that most video media are already encoded using these color spaces. Transforming from RGB into any of these spaces is a straight forward linear

transformation. Since this color space separate the illumination channel (Y) from two orthogonal chrominance channels (C_b and C_r), unlike RGB the location of the skin color in the chrominance channels will not affected by changing the intensity of the illumination. This facilitates building skin detectors that are invariant to illumination intensity and that use simple classifiers. Therefore, using such color spaces results in skin detectors which are invariant to human race. The transformation simplicity and explicit separation of luminance and chrominance components makes this color space attractive for skin color modeling (Phung et al. 2002, Zarit et al. 1999, Menser & Wien 2000, Hsu et al. 2002, Ahlberg 1999, and Chai & Bouzerdoum 2000) cited in [27]. Other color spaces that are categorized under the orthogonal color spaces with the same idea with YCbCr are YIQ, YUV, and YES.

2.5.1.6 CIE-Lab, CIE-Luv

The term"skin color" is not a physical property of an object, rather a perceptual phenomenon and therefore a subjective human concept. Therefore, color representation similar to the color sensitivity of human vision system should help to obtain high performance skin detection algorithm. To this end new perceptually uniform color spaces were proposed by G. Wyszecki cited in [27] and standardized by CIE (Commission Internationale de L'Eclairage). However, this uniformity is obtained at the expense of heavy computational transformations. In these color spaces, the computation of the luminance (L) and the chroma (*ab* or *uv*) is obtained through a non-linear mapping of the XYZ coordinates [30]. For skin detection, the CIE-Lab space has been used by Cai and Goshtasby (1999) and Kawato and Ohya (2000) cited in [30]. The CIE-Luv space has been used by Yang and Ahuja (1999) cited in [30].

2.5.2 Skin Classification/Modeling

The final goal of skin color detection is to build a decision rule that will discriminate between skin and non-skin pixels. This is usually accomplished by introducing a metric, which measures distance (in general sense) of the pixel color to skin tone. The type of this metric is defined by the skin color modeling method. From a classification point of view, skin-color detection can be viewed as a two class problem: skin-pixel vs. non-skin-pixel classification. In the literature different researchers have used different techniques to approach this problem. The following section gives a brief description of the most common methods used.

2.5.2.1 Explicitly defined skin region

One method to build a skin classifier is to define explicitly (through a number of rules) the boundaries skin cluster for different color space components. Single or multiple ranges of threshold values for each color space component are defined and the image pixel values that fall within these predefined range(s) for all the chosen color components are defined as skin pixels. In fact, most of the color spaces have been used with the explicit skin-color segmentation for various purposes. The simplicity of this method have attracted many researchers, for example, Peer et al. (2003) cited in [27, 11] classified skin color pixel based on the following combination of rules for RGB color space:

Under daylight uniform illumination, a pixel in (R, G, B) is classified as skin if:

$$\begin{aligned} &R > 95 \text{ and } G > 40 \text{ and } B > 20 \text{ and} \\ &\max\{R, G, B\} - \min\{R, G, B\} > 15 \text{ and} \\ &|R - G| > 15 \text{ and } R > G \text{ and } R > B \end{aligned} \tag{2.9}$$

And under flashlight or (light) daylight, a pixel is classified as skin if:

$$R > 220 \text{ AND } G > 210 \text{ AND } B > 170 \text{ AND } |R - G| \leq 15 \text{ AND } R > B \text{ AND } G > B \tag{2.10}$$

Dai and Nakano (1996) cited in [30] used a fixed range on *I* component in YIQ color space for detecting skin pixels from images containing mostly people with yellow skin. The *I* component includes colors from orange to cyan. All the pixel values in the range, $R_1 = [0, 50]$ are described as skin pixels in this approach. Sobottka and Pitas (1996, 1998) cited in [5] used fixed range values on the HS color space. The pixel values in the range $R_H = [0, 50]$ and $R_S = [0.23, 0.68]$ are defined as skin pixels. These values have been determined to be well suited for discriminating skin pixels from non-skin pixels on the M2VTS database, containing images of yellow and white skin people. Chai and Ngan (1999) cited in [30] proposed a face segmentation algorithm in which they used a fixed range skin-color map in the CbCr plane. The pixel values in the range $R_{Cb} = [77, 127]$, and $R_{Cr} = [133, 173]$ are defined as skin pixels on the ECU face and skin database.

The obvious advantage of this method is simplicity of skin detection rules that leads to construction of a very rapid classifier. The main difficulty achieving high recognition rates with this method is the need to find both good color space and adequate decision rules empirically.

According to Gomez and Morales (2002) cited in [27, 30] recently, there have been a proposed method that uses machine learning algorithms to find both suitable color space and a simple decision rule that achieve high recognition rates.

2.5.2.2 Gaussian Classifiers

Many of the representative works on skin-color distribution modeling have used Gaussian mixtures. The advantage of these parametric models is that they can generalize well with less training data and also have very less storage requirements. Whereas, non-parametric models require much storage space and their performance directly depends on the representativeness of the training image set.

2.5.2.2.1 Single Gaussian Models (SGM)

Under controlled illuminating conditions, skin colors of different individuals cluster in a small region in the color space. Hence, under certain lighting conditions, the skin-color distribution of different individuals can be modeled by a multivariate normal (Gaussian) distribution in normalized color space. Skin-color distribution is modeled through elliptical Gaussian joint probability distribution function (pdf), defined as [30]:

$$p(c) = \frac{1}{(2\pi)^{1/2}|\sum|^{1/2}} \exp\left[-\frac{1}{2}(c-\mu)^T \sum^{-1}(c-\mu)\right] \quad (2.11)$$

Where c is a color vector, μ and Σ are the distribution parameters (mean vector and diagonal covariance matrix, respectively). The model parameters are estimated from the training data by (2.12):

$$\mu = \frac{1}{n}\sum_{j=1}^{n} c_j, \quad \sum = \frac{1}{n-1}\sum_{j=1}^{n}(c_j-\mu)(c_j-\mu)^T \quad (2.12)$$

Where n is the total number of skin color samples c_j. The parameters, μ and Σ are estimated over all the color samples (c_j) from the training data using ML (Maximum Likelihood)) estimation approach. The probability p(c) can be used directly as a measure of skin-color likeliness and the classification is normally obtained by comparing it to a certain threshold value estimated

empirically from the training data [27, 30]. Alternatively, the **Mahalanobis** distance λ can be compared from the image pixel color vector c to mean vector μ, given the covariance matrix Σ:

$$\lambda = (c - \mu)^T \Sigma^{-1} (c - \mu) \quad (2.13)$$

2.5.2.2.2 Gaussian Mixture Models (GMM)

Though the human skin-color samples for people of different races cluster in a small region in the color space, it has been shown that different modes co-exist within this cluster and hence it cannot be modeled effectively by a single Gaussian distribution. A more sophisticated model, capable of describing complex-shaped distributions is the Gaussian mixture model. It is the generalization of the single Gaussian, the pdf in this case is:

$$p(c) = \sum_{i=1}^{N} w_i \frac{1}{(2\pi)^{1/2} |\sum_i|^{1/2}} \times \exp\left[-\frac{1}{2}(c - \mu_i)^T \sum_i^{-1} (c - \mu_i)\right] \quad (2.14)$$

Where c is a color vector and $\boldsymbol{\mu_i}$ and $\boldsymbol{\Sigma_i}$ are the mean and the diagonal covariance matrix. *N* is the number of Gaussians and the weight factor, $\boldsymbol{w_i}$ is the contribution of the $\boldsymbol{i^{th}}$ Gaussian. The parameters of a GMM ($\boldsymbol{\mu_i}$, $\boldsymbol{\Sigma_i}$, and $\boldsymbol{w_i}$) are approximated from the training data through the iterative expectation-maximization (EM) technique [27, 30]. To converge well, the EM technique needs a good initial guess of the parameters. These initial parameters can be obtained by *k-means* clustering of the training data.

2.5.2.3 Normalized Lookup Table (LUT)

This model is one of the non-parametric skin distributions modeling technique whose key ides is to estimate skin color distribution from the training data without deriving an explicit model of the skin color. Several face detection and tracking algorithms found in the literature use a histogram based-approach to skin pixels segmentation. The color space (usually, the chrominance plane only) is quantized into a number of bins, each corresponding to particular range of color component value pairs (in 2D case) or triads (in 3D case). These bins, forming a

2D or 3D histogram are referred to as the lookup table (LUT). Each bin stores the number of times this particular color occurred in the training skin images. After training, the histogram counts are normalized, converting histogram values to discrete probability distribution [27]:

$$P_{skin}(c) = \frac{skin[c]}{Norm} \tag{2.15}$$

Where *skin[c]* gives the value of the histogram bin, corresponding to color vector c and *Norm* is the normalization coefficient (sum of all histogram bin values, or maximum bin value present). The normalized values of the lookup table bins constitute the likelihood that the corresponding colors will correspond to skin.

Jones and Rehg (2002) cited in [30], built a 3D RGB histogram model with two billion pixels collected from 18,696 web images. They reported that 77% of the possible RGB colors are not encountered and most of the histogram is empty. They also computed two different histograms, skin and non-skin histograms. Given skin and non-skin histograms, the probability that a given color belongs to skin and non-skin class (also called class conditional probabilities) is defined as:

$$P\ (c/skin) = \frac{s(c)}{Norm_s} \qquad P\ (c/non\text{-}skin) = \frac{n(c)}{Norm_n} \tag{2.16}$$

Where *s(c)* is the pixel count in the color c-bin of the skin histogram, *n(c)* is the pixel count in the color c-bin of the non-skin histogram. $Norm_s$ and $Norm_n$ represents the total counts in the skin and non-skin histogram bins. From the generic skin and non-skin histograms, Jones and Rehg demonstrated that there is a reasonable separation between skin and non-skin classes. This fact can be used to build fast and accurate skin classifiers even for images collected from unconstrained imaging environments such as web images, given that the training dataset is sufficiently huge.

2.5.2.4 Bayesian Classifier model

Bayesian theory of decision (BTD) is a fundamental tool of analysis in Machine Learning. Several machine learning algorithms have been derived using BTD. According to Jain and Wu cited in [2] the fundamental idea in BTD is that the decision problem can be solved using probabilistic considerations. Bayesian networks are directed acyclic graphs that allow efficient

and effective representation of the joint probability distribution functions. Each vertex in the graph represents a random variable, and edges represent direct correlations between the variables [30].

The value of $P_{skin}(c)$ computed in (2.15) is actually a conditional probability $P(c|skin)$ – a probability of observing color c, knowing that we see a skin pixel. A more appropriate measure for skin detection would be $P(skin|c)$ – a probability of observing skin, given a concrete c color value. To compute this probability, the Bayes rule is used:

$$P(skin|c) = \frac{P(c|skin)P(skin)}{P(c|skin)P(skin) + P(c|\neg skin)P(\neg skin)} \qquad (2.17)$$

P(c|skin) and P(c|¬skin) are directly computed from skin and non-skin color histograms (11). The prior probabilities P(skin) and P(¬skin) can also be estimated from the overall number of skin and non-skin samples in the training set. Jones and Rehg (1999) cited in [27], an inequality $P(skin|c) \geq \Theta$, where Θ is a threshold value can be used as a skin detection rule. Receiver operating characteristics (ROC) curve shows the relationship between correct detections and false detections for a classification rule as a function of the detection threshold. It turns out, that the ROC curve for $P(skin|c) \geq \Theta$ is invariant to choice of prior probabilities, due to nature of the Bayes model. This means that P(skin) value affects only the choice of the threshold Θ.

One can avoid computing (2.17) explicitly, if what is really needed is the comparison of P(skin|c) to P(¬skin|c), not their exact values. Using (12) the ratio of P(skin|c) to P(¬skin|c) can be written as:

$$\frac{P(skin|c)}{P(\neg skin|c)} = \frac{P(c|skin)P(skin)}{P(c|\neg skin)P(\neg skin)} \qquad (2.18)$$

Comparing (2.18) to a threshold produces the skin/non-skin decision rule. That after some manipulations can be rewritten as:

$$\frac{P(c|\ skin)}{P(c|\neg skin)} > \Theta \qquad \Theta = K \times \frac{1 - P(skin)}{P(skin)} \qquad (2.19)$$

This shows, why the choice of prior probabilities does not affect the overall detector behavior – for any prior probability P(skin) it is possible to choose the appropriate value of K, that gives the same detection threshold Θ [27].

2.6 Feature Extraction

Feature extraction is one of the crucial steps to gesture recognition, in general and sign language recognition in particular [18]. It is obvious that the whole video or image cannot be used as a feature(s), because this would demand high calculation complexity for whatever recognition system it is used for. Due to this reason feature extraction plays an important role in system's performance in terms of reduced system's complexity and good recognition results. To this regard, for example hand gestures are very rich in shape variation, motion and textures therefore extracting specific features of those components allow in automatic identification of signs in most sign languages of the world. Although it is possible to recognize static hand posture by extracting some geometric features such as fingertips, finger directions and hand contours, such features are not always available and reliable due to self-occlusion and lighting conditions. But there are also many other non-geometric features such as color, Silhouette and textures, and other feature that exist due to the application of the non-manual articulators [18, 32].

As described in [33], recognition systems that operate in real world conditions require sophisticated feature extraction approaches in which the extraction stage aims for the robust segmentation of the signer's hands and face (which is needed as a reference point) from the input video/image sequence.

2.6.1 Extraction of Manual/Hand Feature

In sign language, hands convey a lot of information in different ways, including configurations, positions, trajectories, and instantaneous velocities of the two hands. In principle, hands are difficult to track, and their configurations (articulated pose) are difficult to estimate, because of their high number of degrees of freedom and their high level of self-occlusion, which give rise to an enormous variation of appearance and a high level of ambiguity [5]. In addition to this, sign might be one- or two-handed, therefore, sometimes it is impossible to know in advance whether the non-dominant hand will remain idle or move together with the dominant hand [33].

As described in [18] specifying features explicitly is not an easy task so, the whole image or transformed image is taken as the input and then features are selected implicitly and automatically by the recognizer. Thus, even if perfect image information were available, fitting an articulated model of a human hand to image data is computationally hard. These fundamental problems are exacerbated by technical issues. Most importantly, hands tend to move fast with respect to the frame rates and shutter times of typical video recording equipment, which results in substantial motion blur. Moreover, in typical recording settings, the structural determinants of the hands are small with respect to the pixel size, and imaging conditions are not optimized to enhance finger contrast. Consequently, the recovery of precise hand positions, let alone their articulated configurations, is very difficult in practice. However, one promising path towards a solution rests on two methodological pillars; (1) discriminative machine learning methods that identify systematic predictors of specific hand related parameters, and (2) the exploitation of redundancy [5]. Piater et al., designed hand tracking system that contains two steps that exploit these, skin-color region segmentation followed by PCA-based template matching. For the segmentation of the skin regions, the authors adopted Boykov et al. (2001) graph-cut algorithm. And they also incorporated two types of information; color and motion likelihood on the data or unary term based on histogram matching and image differencing, respectively [5].

In the literature different researchers have tried to come up with hand feature extraction methods using an approach that is optimal to their work. However, as described in [18, 34] these huge number of approaches can be grouped into three approaches by which manual or hand features are extracted and they are discussed on the next sections below.

2.6.1.1 Model based Approaches (Kinematic Model)

Model based approaches attempt to infer the pose of the palm and the joint angles. Generally, the approach consists of searching for the kinematic parameters that brings the 2D projection of a 3D model of hand into correspondence with an edge-based image of a hand. Rehg and Kanade (1994) cited in [26] propose one of the earliest model-based approaches to the problem of bare hand tracking in which image formation is modeled as a mapping of a 3D hand model into the image. The parameters of the mapping are the joint angles of the model (21 parameters) as well as the overall pose of the hand (6 parameters), yielding a total of 27 parameters. The approach can be thought of as a series of hypotheses and tests, where a hypothesis of model parameters at

each step is generated in the direction of the parameter space (from the previous hypothesis) achieving the greatest decrease in miscorrespondence. These model parameters are then tested against the image. This approach has several disadvantages that has kept it from real-world use. First, at each frame the initial parameters have to be close to the solution, otherwise the approach is liable to find a suboptimal solution (i.e. local minima). Secondly, the fitting process is also sensitive to noise (e.g. lens aberrations, sensor noise) in the imaging process. Finally, the approcach cannot handle the inevitable self-occlusion of the hand [34].

Huang and Jeng (2001) cited in [35] introduced a model-based hand gesture recognition system, which consists of three phases: feature extraction, training, and recognition. In the feature extraction phase, a hybrid technique combines hand edges and hand motions information of each frame to extract the feature images. Then, in the training phase, they use PCA to characterize spatial shape variations and HMM to describe the temporal shape variations. Finally, in recognition phase, with the pre-trained PCA models and HMM, the observation patterns can be generated from the input sequences, and then apply the Viterbi algorithm to identify the gesture.

A common problem with the model-based approaches is the problem of the feature extraction (i.e. edges). The human hand itself is rather texture less and does not provide many reliable edges internally. The edges that are extracted are usually extracted from the occluding boundaries. In order to facilitate extraction and unambiguous correspondence of edges with model edges the approaches require homogeneous backgrounds and high contrast backgrounds relative to the hand [18, 34].

2.6.1.2 View based Approaches

Due the above mentioned fitting difficulties associated with kinematic model based approaches, many have sought alternative representations of the hand. Gupta et al (2002) cited in [18] applied an alternative view-based approach that have gained significant focus in recent years. View-based approaches, also referred to as appearance-based approaches, model the hand by a collection of 2D intensity images. In turn, gestures are modeled as a sequence of views [18, 34].

Currently, eigenspace approaches represent the state-of-the-art for view-based approaches. The eigenspace approach provides an efficient representation of a large set of high-dimensional points using a small set of basis vectors [34]. Sirovich & Kirby (1987) cited in [34] proposed treating a set of images as a high-dimensional point set. An *m*-by-*n* image can be thought of as a

point in a $m \times n$ vector space by concatenating successive rows. The eigenspace approach seeks an orthogonal basis that spans a low-ordered subspace that accounts for most of the variance in a set of exemplar images. To reconstruct an image in the training set a linear combination of the basis vectors (images) are taken, where the coefficients of the basis vectors are the result of projecting the image to be reconstructed on to the respective basis vectors. The intent of the authors was to use this representation for compression purposes. Given the success in face recognition many have applied the eigenspace approach to hand gestures (e.g. (Black & Jepson, 1996; Gupta et al., 2002) cited in [34]).

2.6.1.3 Low Level Features based Approaches

As outlined in Section 2.6.1.1, model based approaches that dominated early attempts to solve the problem of hand gesture recognition are not robust in the presence of noise. In many gesture applications though all that is required is a mapping between input video and gesture. Therefore, many have argued that the full reconstruction of the hand is not essential for gesture recognition. Instead many approaches have utilized the extraction of low-level image measurements that are fairly robust to noise and can be extracted quickly. Low-level features that have been proposed in the literature include: the centroid of the hand region (e.g. (Starner et al., 1998; New et al., 2003)), principle axes defining an elliptical bounding region of the hand (e.g. (Starner et al., 1998)), and the optical flow/affine flow (e.g. (Cutler & Turk, 1998; Yang et al., 2002)) of the hand region in a scene [18, 34]. As an example, Starner et al. (1998) cited in [18, 34] demonstrate an American Sign Language recognition system based on the extraction of eight feature elements consisting of each hand's x and y positions, and respective principle axes. Coupled with a Hidden Markov Model based classification stage in which it reportedly achieved 99.2% word accuracy.

2.6.2 Extraction of Facial Features

Sign languages are multimodal languages, in which several channels for transferring information are used at the same time. One basically differentiates between the manual/gestural channels and the non-manual/facial channels and their respective parameters [33]. Non manual gestures such as facial expressions and head tilts play a very important role in sign language due to the fact that

many manual signs are ambiguous in isolation, and need to be accompanied by appropriate facial expressions in order to convey a specific message [5].

As described in [33] non-manual parameters are indispensable in signs language. They encode e.g. adjectives and adverbials and contribute to grammar. In particular, some signs are identical with respect to gesturing and can only be differentiated by making reference to non-manual parameters such as, *upper body posture*, *head pose*, *line of sight*, *facial expression*, and *lip outline*.

As described in [33] in the context of facial analysis, image preprocessing aims to the robust localization of the face region which corresponds to the rectangle bounded by bottom lip and the eyebrows. With regard to processing speed, image analysis is limited to a small search mask. This mask is devised to find skin colored regions with suitable movement patterns only. The largest skin colored object is selected and subsequently limited by contiguous, non skin colored region. Additionally, the general skin color model is adapted to each individual. Gray world color constancy is applied on the image sequence to reduce influences of the environment due to reflections and different lighting conditions.

The interpretation of facial expression is based on so called Action Units which represent the muscular activity in a face. In order to classify these units, areas of interest, such as the eyes, eyebrows, and mouth (in particular the lips) as well as their spatial relation to each other, have to be extracted from the images. For this purpose, the face is modeled by an active appearance model (AAM), a statistical model which combines shape and texture information about human faces [5, 33]. Shape and texture variations of the human face as well as the correlations between them are learned from a set of example face images, on which corresponding “landmark” points have to be marked priori (including our facial feature points of interest) and the trained appearance model must be adapted to the signer. And [33] used a front view image of the signer’s face and applied that to an artificial 3D head model for adaptation.

2.7 Classification Schemes for Sign Gestures

As described by Ong and Ranganath [15], there are two main approaches to sign gesture classification; that is, either to employ a single classification stage, or represent the gesture as consisting of simultaneous components which are individually classified and then integrated together for sign-level classification. The classification methods that have been proposed

generally fall under the following categories [18]: rule-based methods, where the gestures are manually encoded as a set of rules (preconditions) and machine learning-based approaches that use machines to infer models of gestures from a set of exemplars.

2.7.1 Rule based Approaches

Rule-based approaches consist of a set of manually encoded rules between feature inputs. Given an input gesture a set of features are extracted and compared to the encoded rules, the rule that matches the input is outputted as the gesture. As an example, Cutler & Turk (1998) and Su (2000), both cited in [18] defined predicates related to low-level features of the motion of the hands for each of the actions under consideration. When a predicate of a gesture is satisfied over a fixed number of consecutive frames the gesture is returned. A major problem with rule-based approaches is that they rely on the ability of a human to encode rules. In many cases the appropriate rules may not be intuitive especially when dealing with high-dimensional feature sets.

2.7.2 Machine Learning Based Approaches

As mentioned in Section 2.7.1 rule-based approaches are hampered by the ability of humans to find relations between features in a high-dimensional feature space. Given this limitation many have turned to machine learning approaches to find mappings between high-dimensional feature sets and gestures.

A popular machine learning approach is to treat a gesture as the out-put of a stochastic process. Of this class of approaches HMMs as applied by Starner et al (1998), Wilson & Bobick (1999), and Lee & Kim (1999) cited in [18], by far have received the most attention in the literature for classifying gestures.

2.7.2.1 Neural Networks

One efficient way of solving complex problems is following the lemma “divide and conquer”. A complex system may be decomposed into simpler elements, in order to be able to understand it. Also simple elements may be gathered to produce a complex system [36]. Networks are one approach for achieving this. There are a large number of different types of networks, but they all are characterized by the following components: a set of nodes, and connections between nodes.

One type of network sees the nodes as 'artificial neurons'. These are called artificial neural networks (ANNs).

Artificial neural networks are an attempt at modeling the information processing capabilities of nervous systems [37]. It is an extremely simplified model of the brain to transform an input into an output to the best of its ability. It is composed of many artificial neurons that are a computational model inspired in the natural neurons that co-operate to perform the desired function. Natural neurons receive signals through synapses located on the dendrites or membrane of the neuron. When the signals received are strong enough (surpass a certain threshold), the neuron is activated and emits a signal though the axon. This signal might be sent to another synapse, and might activate other neurons.

Neural Networks have the ability to learn because it can figure out how to perform its function on its own and it can also determine its function based on only sample inputs. It also has the ability to generalize, i.e. it can produce reasonable outputs for inputs it has not been taught how to deal with. Because of these reason researchers tend to use NNs and its variants for classification purpose such as Multilayer Perceptrons (MLP), Fuzzy Min-Max NNs, Adaptive Neuro-Fuzzy Inference System Networks, Hyperrectangular Composite NNs, and 3D Hopfield NN [15]. Figure 2.4 below depicts how neural networks work:

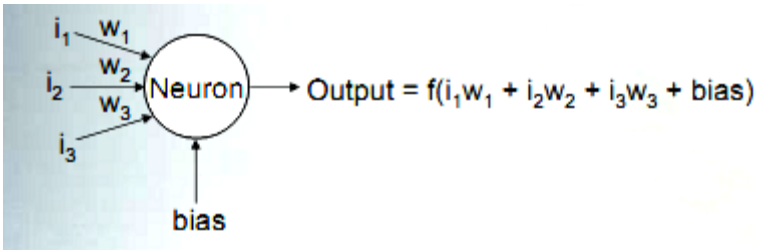


Fig 2.4 Neural network

Given a weighted input, the output of a neuron is a function of the weighted sum of the inputs plus a bias. The function of the entire neural network is simply the computation of the outputs of all the neurons. NNs do not execute programmed instructions; they respond in parallel (either simulated or actual) to the pattern of inputs presented to it. One of the methods to get the weights in a NN is using training. Training is the act of presenting the network with some sample data and modifying the weights to better approximate the desired function. It could be either supervised where the NN is supplied with inputs and the desired outputs or unsupervised where

the NN is supplied with inputs and the NN adjusts its own weights so that similar inputs cause similar outputs.

Erenshteyn et al (1996), Handouyahia et al (1999) and Wu & Gao (2001) cited in [13] employed MLP for classifying handshape. Additionally Waldron & Kim (1995) and Vamplew & Adams (1998) cited in [15] used MLPs to classify the hand location, orientation, and movement type from tracker data. As described in [15], Kim et al (1996) used fuzzy min-max NNs, Al-Jarrah & Halawani (2001) used adaptive neuro-fuzzy inference system networks, Su (2000) used hyperrectangular composite NNs for hand shape classification. Whereas Huang & Huang (1998) used 3D Hopfield NN for sign classification.

Time-series data, such as movement trajectories and sign gestures, consist of many data points and have variable temporal lengths. Therefore, NNs designed for classifying static data often do not utilize all the information available in the data points. But some attempts were made for example, Waldron & Kim (1995) cited in [15] used the displacement vectors at the start and midpoint of a gesture as input to the MLP, while Vamplew & Adams (1998) cited in [15] used only the accumulated displacement in each of the three primary axes of the tracker in classifying movement type. On the other hand, Yang et al (2002) cited in [15] used Time-Delay NNs which were designed for temporal processing, to classify signs from hand pixel motion trajectories. Murakami & Taguchi (1991) used Recurrent NNs which can take into account temporal context without requiring a fixed temporal length. They considered a sign word to be recognized when the output node values remain unchanged over a heuristically determined period of time.

2.7.2.2 Hidden Markov Models (HMMs)

Markov Models are a powerful abstraction for time series data, but fail to capture a very common scenario. How can we reason about a series of states if we cannot observe the states themselves, but rather only some probabilistic function of those states [39]? The Hidden Markov Model (HMM) is a powerful statistical tool for modeling generative sequences that can be characterized by an underlying process generating an observable sequence. HMMs have found application in many areas interested in signal processing, and in particular speech processing, but have also been applied with success to low level NLP tasks such as part-of-speech tagging, phrase chunking, and extracting target information from documents [39]. HMMs are able to

process time-series data with variable temporal lengths and discount timing variations through the use of skipped-states and same-state transitions [15].

There are three fundamental questions we might ask of an HMM. What is the probability of an observed sequence? What is the most likely series of states to generate the observations? And how can we learn values for the HMM's parameters A and B given some data?

1. Probability of an observed sequence: Forward procedure

In a HMM the assumption is that the data is generated by the following process: imagine the existence of a series of states $\vec{z}$ over the length of our time series. This state sequence is generated by a Markov model parameterized by a state transition matrix *A*. At each time step *t*, one can select an output x_t as a function of the state z_t. Therefore, to get the probability of a sequence of observations, one need to add up the likelihood of the data *x* given every possible series of states.

$$\begin{aligned} P(\vec{x}; A, B) &= \sum_{\vec{z}} P(\vec{x}, \vec{z}; A, B) \\ &= \sum_{\vec{z}} P(\vec{x} | \vec{z}; A, B) P(\vec{z}; A, B) \end{aligned} \tag{2.20}$$

HMM assumptions: the output independence assumption, Markov assumption, and stationary process assumption allow the possibility to further simplify the above expression (2.20). The bad news is that the sum is over every possible assignment to $\vec{z}$. Because z_t can take one of $|S|$[7] possible values at each time step, evaluating this sum directly will require $O(|S|^T)$ operations.

Fortunately, a faster means of computing $P(\vec{x}; A, B)$ is possible via a dynamic programming algorithm called the Forward Procedure. First, let's define a forward variable $\alpha_i(t) = P(x_1, x_2, \ldots, x_t, z_t = s_i; A, B)$ as the probability of the partial observation sequence until time t, with state S_i at time t. If we had such a quantity, the probability of our full set of observations $P(\vec{x})$ could be represented as:

$$\begin{aligned} P(\vec{x}; A, B) &= P(x_1, x_2, \ldots, x_T; A, B) \\ &= \sum_{i=1}^{|S|} P(x_1, x_2, \ldots, x_T, z_T = s_i; A, B) \\ &= \sum_{i=1}^{|S|} \alpha_i(T) \end{aligned} \tag{2.21}$$

[7] set of states.

2. Maximum Likelihood State Assignment: The Viterbi Algorithm

One of the most common queries of a Hidden Markov Model is to ask what was the most likely series of states $\vec{z} \in S^T$ given an observed series of outputs $\vec{x} \in V^T$. The Viterbi Algorithm is just like the forward procedure except that instead of tracking the total probability of generating the observations seen so far, one needs to only track the maximum probability and record its corresponding state sequence. Formally, we seek:

$$\arg\max_{\vec{z}} P(\vec{z} \mid \vec{x}; A, B) = \arg\max_{\vec{z}} \frac{P(\vec{x}, \vec{z}; A, B)}{\sum_{\vec{z}} P(\vec{x}, \vec{z}; A, B)} = \arg\max_{\vec{z}} P(\vec{x}, \vec{z}; A, B) \quad (2.22)$$

The first simplification follows from Bayes rule and the second from the observation that the denominator does not directly depend on $\vec{z}$. Naively, we might try every possible assignment to $\vec{z}$ and take the one with the highest joint probability assigned by our model. However, this would require $O(|S|^T)$ operations just to enumerate the set of possible assignments.

3. Parameter Learning: EM for HMMs

The final question to ask of an HMM is: given a set of observations, what are the values of the state transition probabilities *A* and the output emission probabilities *B* that make the data most likely? For example, solving for the maximum likelihood parameters based on a speech recognition dataset will allow us to effectively train the HMM before asking for the maximum likelihood state assignment of a candidate speech signal. If we know the state path for each training sequence, learning the model parameters is simple. We can compute the percentage of times each particular transition or emission is used in the set of training sequences to determine A_{ij}, the transition probabilities, and B_{jk}, the emission probabilities.

As described in [15], HMMs can also implicitly segment continuous speech into individual words trained word or phoneme HMMs are chained together into a branching tree-structured network and Viterbi decoding is used to find the most probable path through the network, thereby recovering both the word boundaries and the sequence. This idea has also been used for recognition of continuous signs, using various techniques to increase computational efficiency. These techniques include language modeling, beam search and network pruning, N-best pass,

fast matching, frame predicting, and clustering of Gaussians. Language models used by Gao et al (2000), Vogler (2003) and Wang et al (2002) cited in [15] include unigram and bigram models, on the other hand McGuire et al (2004) and Starner et al (1998) cited in [15] used strongly constrained parts-of-speech grammar. As an alternative to the tree-structured network approach, Liang and Ouhyoung (1998) and Fang et al (2001) cited in [15] explicitly segmented sentences before classification by HMMs.

To reduce training data and enable scaling to large vocabularies, some researchers define sequential subunits, similar to phonetic acoustic models in speech, making every sign a concatenation of HMMs which model subunits. Based on an unsupervised method similar to one employed in speech recognition (Jelinek (1998)), Bauer and Kraiss (2001) cited in [15] defined 10 subunits for a vocabulary of 12 signs using k-means clustering. Later, in 2002 they introduced a bootstrap method to get initial estimates for subunit HMM parameters and obtain the sign transcriptions. Recognition accuracy on 100 isolated signs using 150 HMM subunits was 92.5%. Encouragingly, recognition accuracy of 50 new signs without retraining the subunit HMMs was 81.0%. Vogler (2003), Yuan et al (2002) and Wang et al (2002) cited in [15] defined subunits linguistically instead of using unsupervised learning. The later achieved 86.2% word accuracy in continuous sign recognition for a large vocabulary of 5,119 signs with 2,439 subunit HMMs.

Kobayashi and Haruyama (1997) argue that HMMs, which are meant to model piecewise stationary processes, are ill-suited for modeling gesture features which are always transient and propose the partly hidden Markov model. In which the observable node probability is dependent on two states, one hidden and the other observable. And experimental results for isolated sign recognition showed a 73% improvement in error rate over HMMs.

2.7.2.3 Principal Component Analysis (PCA)

Principal component analysis (PCA) is a multivariate technique that analyzes a data table in which observations are described by several inter-correlated quantitative dependent variables [40]. It is also appropriate when you have obtained measures on a number of observed variables and wish to develop a smaller number of artificial variables called *principal components*, which may then be used as predictor or criterion variables in subsequent analysis. PCA is a useful statistical technique that has found application in fields such as face recognition and image

compression, and is a common technique for finding patterns in data of high dimension, and expressing the data in such a way as to highlight their similarities and differences [41].

As described in [40] the goals of PCA are to (1) extract the most important information from the data table; (2) compress the size of the data set by keeping only this important information; (3) simplify the description of the data set; and (4) analyze the structure of the observations and the variables. In order to achieve these goals, PCA computes *principal components* which are obtained as linear combinations of the original variables. The first principal component is required to have the largest possible variance (i.e., inertia and therefore this component will 'explain' or 'extract' the largest part of the inertia of the data table). The second extracted will have two important characteristics. First, this component will account for a maximal amount of variance in the data set that was not accounted for by the first component. Again under typical conditions, this means that the second component will be correlated with some of the observed variables that did not display strong correlations with component I. The second characteristic of the second component is that it will be uncorrelated with the first component. Literally, if you were to compute the correlation between components I and II, that correlation would be zero. The remaining components are computed likewise: each component accounts for a maximal amount of variance in the observed variables that was not accounted for by the preceding components, and is uncorrelated with all of the preceding components.

In the literature different scholars have applied PCA in sign gesture classification process. For example, Birk et al (1997) and Imagawa et al (2000) cited in [15] used PCA to reduce dimensionality of segmented hand images and the later one also applied an unsupervised approach where training images were clustered in eigenspace and test images were classified to the cluster identity which gave the maximum likelihood score. On the other hand, Kong and Ranganath (2004) cited in [15] classified 11 3D movement trajectories by performing periodicity detection using Fourier analysis, followed by Vector Quantization Principal Component Analysis suggested by Kambhatla and Leen (1997) cited in [15].

2.7.2.4 Support Vector Machine

SVM are supervised learning models with associated learning algorithm that analyze data and recognize patterns, used for classification and regression analysis. The basic SVM takes a set of input data and predicts, for each given input, which of two possible classes forms the output,

making it a non-probabilistic binary linear classifier. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

As described in [43] they have proposed a new technique for skin segmentation using SVM (support vector machine) active learning combined with region segmentation information and achieved above 90 % recognition rate using very few training samples. The researcher in [42] have designed tool for recognizing alphabet level continuous American Sign Language using Support Vector Machine to track the sign languages represented with hands is presented. Six letters are trained and recognized and got an efficiency of 92.13%.

2.7.2.5 K-Nearest Neighbors

The K-Nearest Neighbors (K-NN) algorithm is a nonparametric method in that no parameters are estimated as, for example, in the multiple linear regression models. Instead, the proximity of neighboring input (x) observations in the training data set and their corresponding output values (y) are used to predict (score) the output values of cases in the validation data set. These output variables can either be interval variables in which case the K-NN algorithm is used for prediction while if the output variables are categorical, either nominal or ordinal, the K-NN algorithm is used for classification purposes. In pattern recognition, the K-NN is a method for classifying objects based on closest training examples in the feature space. K-NN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computations is deferred until classification [44].

Among researchers that have applied the K-NN algorithms works done by [44] is worth mentioning because they have designed the algorithm to work as a first level detection upon a series of steps to bring the captured character images into actual spelling. They have applied K-NN algorithm with feature extraction as a guideline for the recognition system. In addition to this [23] have used K-NN as a static gesture classification tool to design a recognition system, in which the system achieved a recognition accuracy of 90.55%.

Further development for the proposed system will be conducted to extend the proposed system to build a complete Arabic sign language recognition system including all alphabets, numbers and most common gestures as well as enhance feature extraction methodology in order to achieve higher recognition rate.

2.8 Review of Related Works

In this section, a review of different local and international researches on sign language recognition is presented. The section starts by reviewing international research work done on word level ArSL recognition and sentence level ASL recognition and then moves to research works done by local researchers.

Among the global/international research reviewed Starner and Pentland [47] done a research work titled visual recognition of ASL using HMM. The proposed system describes sentence level ASL recognizer using HMM with the idea of it being popular in speech recognition as well as in handwriting recognition. In this recognition system, sentences of the form “personal pronoun, verb, noun, adjective, personal pronoun” are recognized. Six personal pronouns, nine verbs, twenty nouns, and five adjectives are included making the total lexicon number forty words. As described in this paper that HMM have intrinsic properties which make them very attractive for SLR. For the data collection they used a signer that wears distinctly colored gloves on each hand (a yellow glove for the right hand and an orange glove for the left) and sits in a chair before the camera. For the feature extraction they have used the position of the hands, some concepts of the shape of the hand and the angle of the hand relative to horizontal. Thus, an eight element feature vector consisting of each hand’s x and y position, angle of axis of least inertia, and eccentricity of bounding ellipse was chosen. The system attains a word accuracy of 99.2% without explicitly modeling the fingers.

Starner et al [48] continued their research entitled a wearable computer based ASL recognizer which is an extension of their previous work mentioned above. The paper describes a recognizer which uses one color camera pointed down from the brim of a baseball cap to track the wearer’s hands in real time and interpret ASL using HMM’s. As noted in the paper the hand tracking stage of the system does not attempt a fine description of hand shape rather the tracking process produces only a coarse description of hand shape, orientation, and trajectory. In this paper the authors have used two methods of hand tracking: one, using solidly-colored cloth gloves (a pink

glove for the right hand and a blue glove for the left), and two, tracking the hands directly without aid of gloves or markings in which the hand is tracked based on skin tone. They have come up with a cap camera mounted vision-based system that is capable of recognizing ASL through the use of HMM with low error rate on both training set and an independent test set. However, the cap camera mount is probably inappropriate for natural sign because signing involves facial gestures and head motion, which would have confounding effect on the hand tracking. So, the authors recommend a necklace that may provide a better mount for determining motion relative to the body, and another possibility is to place reference points on the body in view of the cap camera. The overall system performance in terms of accuracy is reported to exceed 97% per word on a 40 word lexicon.

Another word level automatic ArSL recognition system titled ArSL recognition system using HMM was done by Youssif et al [24]. The system is based on HMMs that have used large set of samples to recognize 20 isolated words from the standard ArSL. The data collected with no restrictions on the signer or word length, and the signers were glove-free and since the system is signer independent different signers with different skin color were considered. Skin detection, canny edge detection, and hand contours and fingertips tracking are used for hand tracking and recognition, based on which a model is produced using HMM. It was reported that the overall recognition rate is 82.22%. The authors recommend achieving higher recognition rates with larger data set. A psycholinguistic study on the structure of ArSL might be needed to choose the appropriate HMM model for each gesture. In addition they proposed to build a continuous sentence recognition system using sub-gesture word based recognition system.

The local research, which is entitled Ethiopian Sign Language using artificial neural network, was done by Admasu and Raimond [10]. The designed system focused on hand gesture detection and recognition technique for EthSL, for the recognition artificial neural network (ANN) has been employed to recognize the EthSL and translate in to Amharic voice. For the data collection one (right) hand finger spelling was used to capture 34 letters of EMA from ten volunteers to have a total of 340 hand gesture images. To reduce difficulty of segmentation caused by high variation in skin color, the signers were instructed to wear white glove in their right hand. For this research the authors applied two approaches for feature extraction. The first approach uses PCA and the second approach uses Gabor Filter (GF) together with PCA. The recognition process achieved a result of 95.588% for the first and 98.529% for the second approach.

Extension based on the second approach is the recommendation of the author in which there were three recommendations; considering EMA images taken without hand gloves, considering non-EMA images, and finally considering EMA images subject to changes in pixel size, background, brightness, noise level, and orientations.

In addition to the works done by [10] Tsegay and Raimond [11] proposed a research work titled offline candidate hand gesture selection and trajectory determination for continuous EthSL. This work describes a recognition system that extracts candidate EMA frames from the video sequence and that can also determine hand movement trajectories. The designed system has two separate parts; the Candidate Gesture Selection (CGS) and the Hand Movement Trajectory Determination (HMTD). A total of 144 videos were collected from 5 signers, where 70 of them were used to design the system and 74 of them were used to test the system. This paper used YCbCr as color space for the purpose of skin detection which is followed by hand isolation process to select signer's hand. For hand tracking the isolated hand is used as an input to collect centroid of each isolated hand gestures. The CGS combines two metrics speed profile of continuous gestures and Modified Housdorff Distance (MHD) measure and obtained an accuracy of 80.72% for this module. The HMTD is done by considering each hand gesture centroid from frame to frame and using angle, x- and y-directions, and it returned a result with an accuracy of 88.31%. In addition to this the system as a whole has a performance of 71.88%. This researcher recommends the idea of using the output of the research as an input to a recognition system where a word or sentence level recognition is required. And the addition of concept of digital image processing called occlusion can allow the signer to sign freely rather than being oriented to avoid overlapping between the signer hand and face.

CHAPTER THREE

SYSTEM DESIGN AND IMPLEMENTATION

This research aims to propose, design and implement a recognition system for EthSL. To this end previous chapters discussed issues related to sign language and its recognition and the steps one should follow to come up with an accurate and robust recognition system. And having this in mind after an extensive study on the subject, the system is designed and implemented accordingly.

3.1 System Architecture

Figure 3.1 below shows the architecture of the proposed word level recognition system for EthSL. The system accept sequence of AVI videos containing EthSL corpus, then using video and image processing important frames/images are extracted from the video that are referred as key frames which are assumed to contain important information regarding the sign in consideration. After this the next task is selection and segmentation of skin colored objects from the images so that manual articulators are easily identified for further processing. The next step is feature extraction in which the first sub task is hand detection based on the output returned by the skin color segmentation process. The second and the third sub tasks namely manual feature extraction and hand trajectory determination are done in parallel with each other, where manual features of sign articulator is identified and represented using PCA. Feature vector which is the output of feature extraction is used as input to the next step which is training and model preparation using KNN, where hand gesture and trajectory model is prepared so that it can be used by the recognizer whenever a new input that undergone through the mentioned steps come into the system.

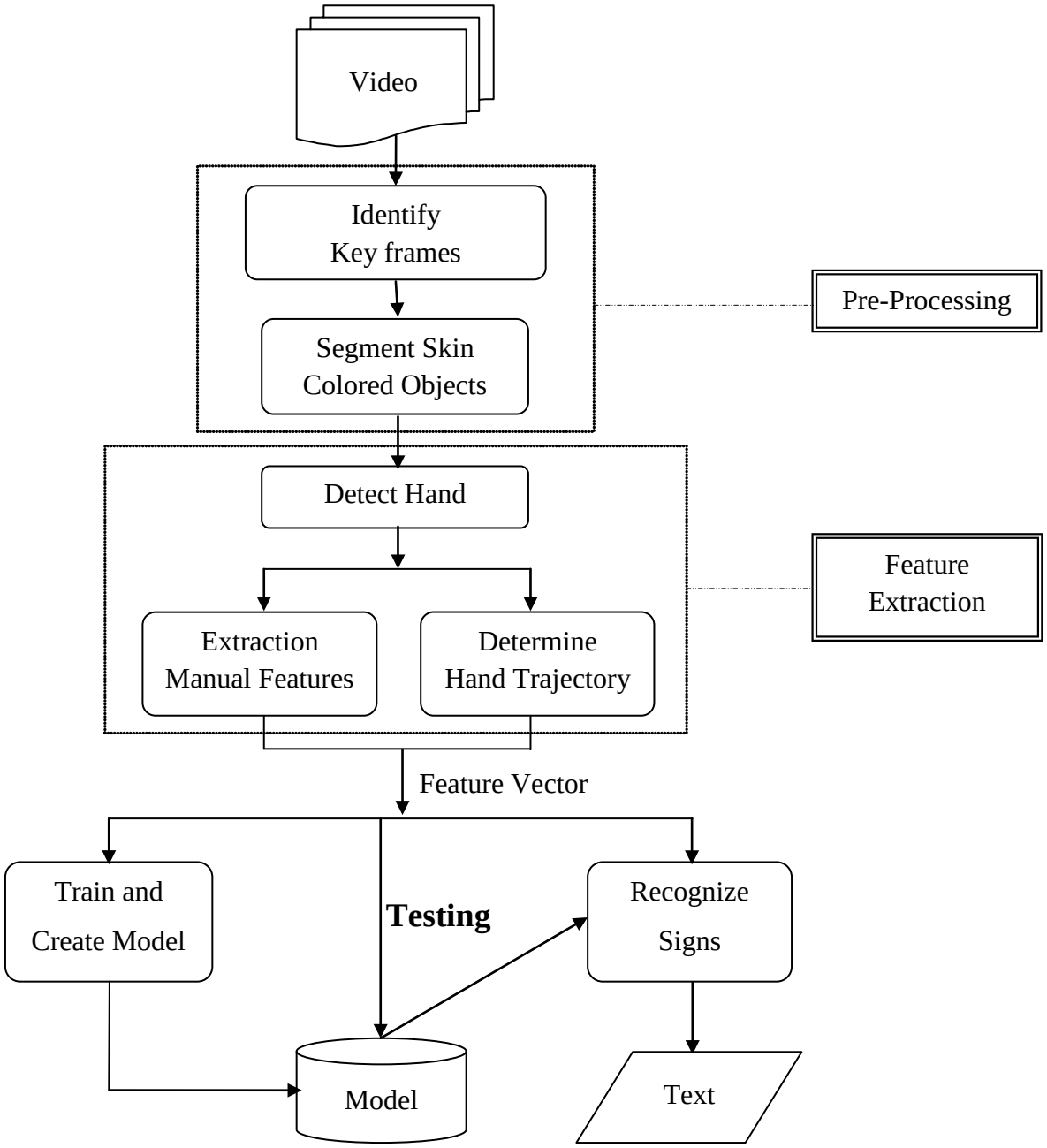


Fig 3.1 Overall System Architecture for Word-Level EthSL Recognition

3.2 Data Collection and Image Preprocessing

The EthSL comprises of 1422 [21] number of words that can be categorized into 23 themes, even though there was no scholarly material that can guide the researcher on which themes to select, with a proper consultation with EthSL experts, only 10 of the themes were selected for this

research as shown in table 3.1, with the criteria of them being frequently used by deaf community during their communication. Out of this 588 total population, purposive sampling was made by the researcher to select 10 % of the words from each theme as shown in Table 3.1.

Sign Theme	Vocabulary	Samples
Signing about Family and Family Relationship	61	6
Signing about Pronoun	27	3
Signing about Question	23	2
Signing about Food, Drink and related items	103	10
Signing about Business, Finance, Occupation & Organization	71	7
Signing about Emotion & Feeling Expressing issues	58	6
Signing about Mental Action	42	4
Signing about Communication & Government	64	6
Signing about Quality & Quantity	31	3
Signing about Descriptive signs	108	11
Total	**588**	**58**

Table 3.1 Frequently used EthSL themes

After deciding on how much to take from each theme the next decision was sample word selection for recognition. Therefore, in this regard first, words with no occlusion whatsoever and words that have no occlusion when they start but somehow have occlusion at the end frame are considered for this study. However, words with occlusion between the two hands and/or between the face and the hands are not included in this study. Out of the total samples selected all of the words were captured from one signer to be used for the training set whereas half of the samples were captured from the same signer to be used for the testing set to undertake the experimentation. Words selected for the training and testing phase of this study are shown below in table 3.2. A separate corpus was prepared for the training and testing dataset.

ህዝብ - People	**ፓስታ** - Pasta	**ፓርላማ** - Parliament
መተዋወቅ -Introduction	**ሊቀመንበር** - Chairperson	**መደወል** - Dialing
መጋባት - To Wed	**መቅጠር** - To employ	**መታወቂ** - Identity card
መውለድ - To give birth	**መቆጠብ** - To save	**መመለስ** - To answer
መፋታት - To divorce	**መክፈል** - To pay	**መተቸት** - To comment
ባዳ - Stranger	**ማባከን** - To waste	**መለካት** - To measure
እናንተ - You	**ተግባር** - Implementation	**መመዘን** - To weigh

እኛ - We	ደሞዝ - Salary	ሚዛን - Balance
የእኛ - Ours	መጥላት - To hate	ለምለም - Fertile
እስከ - Until	ለቅሶ - Mourning	ልዩ - Unique
የትኛው - Which	መረዳት - Understand	መበተን - To scatter
መክተፍ - To chop	መቆጣት - To be angry	መብለጥ - To surpass
ሳምቡሳ - Spring roll	መጠርጠር - Suspect	መፍጠን - To hasten
ስንዴ - Wheat	ነፍስ - Soul	ሙሉ - Complete
ቡና - Coffee	ክብር - Respect to God	ርካሽ - Cheap
አልኮል - Alcohol	መስማማት - Agree	ሰፊ - Wide
ኬክ - Cake	ህሊና - Conscience	ቀላል - Light
ዊስኪ - Whiskey	ክብር - Respect to man	አጭር - Short
ዳቦ - Bread	መግለፅ - To express	ጀግና - Brave

Table 3.2 Selected EthSL words for training and testing

The video was recorded in a slightly controlled environment, with 10.0 Mega pixel Canon digital camera. The recorded videos have a dimension of 640x480 pixels. Each video contains a collection of frames representing a gesture. At first, each video is pre-processed by applying Windows Movie Maker to remove frames from both sides of the video that are duplicate and not very important for the intended work. And then Free AVI video converter was used to convert the video into a format that is appropriate for the Matlab image processing toolkit. In addition to this the capture video was further sub divided into frames using Free Video to JPG so that the researcher was able to make use of each frames/Images when needed. Then before undergoing any other image processing activities designed either by the researcher or Matlab each frames contrast and brightness was adjusted using Matlab's `'imadjust'` built in function.

3.3 Skin Segmentation

As it has been briefly explained in section 2.4.2 in recent years, the study of skin color based segmentation is gaining popularity due to its active research in content-based image representation. Segmentation is the concept of subdividing an image into its constituent regions or objects [45]. In case of sign language recognition regions or objects can mean face and/or hand of the signer who does the articulation. Once the regions are located we can do different kinds of image processing activities to extract both manual and non-manual sign features.

Therefore, skin color detection is becoming one of the techniques used to detect signers hand in SLR so that hand features can be extracted and recognized accordingly.

The skin color of a person is dependent on some biological property like melanin, pigmentation etc. but this color range belongs to the subspace of the total color space. Therefore, the first task in skin color detection is selection of color space that is followed by skin color modeling or classification.

An HSV color space was selected as a color space for this research due to the fact that HSV color model is more popular when compared to RGB or YCbCr color model because it is compatible with human color perception [46]. In addition to this, in the field of computer vision one often want to separate color components from intensity for various reasons, such as robustness to lighting changes, or removing shadows. Note however, that HSV is one of many color spaces that separate color from intensity. These color spaces separates three components: the hue (H), the saturation (S) and the brightness (V). As V gives the brightness information, it is often dropped to reduce illumination dependency of skin color. In addition to this, while conducting the study the researcher have noticed that there were some clarity missing from the output and in return to this the researcher applied RGB color space on top of HSV to get the result shown in figure 3.2 (c).

Given an image, each pixel in the image is classified as a skin or non-skin using color information. If the H and S values of a pixel exceeds a threshold called skin threshold, this pixel is considered a skin pixel. Otherwise, the pixel is considered a non-skin pixel. A general image and its skin detected image can be seen such that white pixels represent the hand gesture and black pixels represent the background or any object behind the skin as shown in Fig. 3.2 (b).

(a) (b) (c)

Fig 3.2 Skin Detection; (a) Original Image, (b) result from HSV CS, and (c) result from hybrid of HSV and RGB CS

This module uses an explicit-definition skin color modeling technique discussed in section 2.5.2.1 with hybrid HSV and RGB color space to segment skin areas of the input images shown in Figure 3.2 (a). The function in Appendix A.4 is used for skin segmentation. The function has the form:

function [binaryImage] = skinSegmentation (rgbImage)

The function takes an RGB image as an input parameter and returns segmented binary images. First, the color space of the video frames was converted from RGB to HSV color space using `rgb2hsv()` matlab's built in function and each pixel was checked for skin or non-skin value on H-S plane based on a skin threshold. Then the binary image is converted back to RGB so that the rgb skin threshold can be applied to the image to finally acquire the skin detected images shown in Figure 3.2 (c). Both HSV and RGB skin threshold are used for skin detection [45]. The skin threshold used for this research is:

- (H, S, V) is classified as skin if:
  $0 \leq H \leq 0.25 \;\&\&\; 0.15 \leq S \leq 0.9$
- (R, G, B) is classified as skin if: (*Equation 2.9*)
  $R > 95$ and $G > 40$ and $B > 20$ and
  $Max\{R, G, B\} - Min\{R, G, B\} > 15$ and
  $|R - G| > 15$ and $R > G$ and $R > B$

After skin thresholding there were some non skin objects that were detected as a skin pixels due to reasons related to lighting condition, skin colored clothing and/or objects, and shadow on the background that somehow might affect the robustness of the skin detection or the system as a whole. Therefore, to solve this problem matlab's *'bwareaopen'* is used to remove connected components (objects) that have fewer than the specified threshold pixels. The result obtained from this operation is shown in Figure 3.3 below.

Fig 3.3 Binary image showing the before and after effect of function *'bwareaopen'* with threshold value of *'1890'*

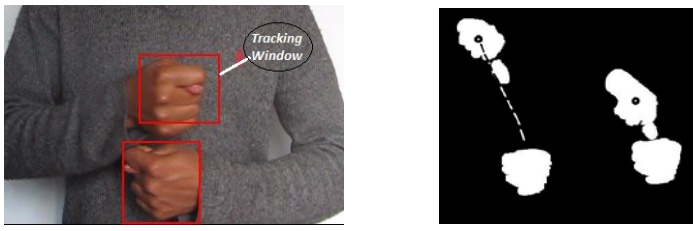
3.4 Manual Feature Extraction

Feature extraction is one of the major activities of any SLR system because it is the step where all manual and/or non-manual features of the signer are identified and further processing is undertaken based on the extracted information. Therefore, selecting the right set of features is the decisive key in order to increase the robustness of the recognition system. The features extracted can be either manual or non-manual or both, but for this study only manual features were considered for the design and implementation of the recognition system. The next three topics discuss about the things incorporated in the feature extraction.

3.4.1 Hand Detection and Segmentation

Although, the focus of this study is to extract manual features from a sign conversation, it was believed that identification and tracking of hand(s) is an ideally important part of the system. Therefore, based on the binary image obtained from the skin detection module this module tries to identify and segment hands with the assumption that the input binary image contains only three objects of interest the two hands and the signer head, with some exception to this related to hands of signer occluding to each other in some of the signing scenarios considered for this study. This module applies the concept of connected component analysis by labeling skin detected regions and by taking labeled region properties using matlab's `'bwlabel'` and `'regionprops'` built in functions respectively.

This module is sub-divided into two parts, where the first part focuses on drawing a tracking window on skin detected regions to show the hand detection, segmentation and tracking along the image sequence shown in Figure 3.4 (a) below. Whereas the second part focuses on finding the centroid of each skin detected regions to determine hand trajectory shown in fig 3.4 (b), which will further be discussed in section 3.4.3 below.



(a) (b)

Fig 3.4 Hand Detection and Segmentation; (a) Hand Tracking, (b) Hand Trajectory

For the hand tracking Mean Shift algorithm is used that is frequently used for visual tracking. It is a nonparametric clustering technique which does not require prior knowledge of the number of clusters, and does not constrain the shape of the clusters. The algorithm works in such a way that it creates a confidence map in the new image based on the color histogram of the object in the previous image, and use mean shift to find the peak of a confidence map near the object's old position. The confidence map is a probability density function on the new image, assigning each pixel of the new image a probability, which is the probability of the pixel color occurring in the object in the previous image. The algorithm implementation was adopted from the one written by Sylvain Bernhardt (2008) [49]. The algorithm accepts an AVI video file as input and allow the users to select the area they want to track, however the modification done on this part of the code is instead of selecting the area, the code automatically selects the hand of the signer using skin color detection [38].

3.4.2 Extraction of Manual Features

This module of the recognition system focuses on extracting manual features from skin detected objects. Basically since what we are dealing with is manual features involving the hand the focus is going to be on hand shape, hand location, hand orientation and as well as hand trajectory/ motion which will be discussed in the next section. Therefore, feature related to hand shape and hand location are extracted using one of the most popular dimensionality reduction and image feature extraction method; Principal Component Analysis (PCA) which was discussed in section 2.7.2.3. The function in Appendix A.7 is used to implement this module. The function has the form:

```
function [PCAFeature omega] = EthSLRPCA((ImgMat, nRows, nColumns,
                    ShowOutput, nEigValThres)
```

The function accepts five inputs: the first, an image vector generated from skin detected binary images of the first and the last frame of a video sequence. The second and the third inputs are used as a picture dimension to show PCA extracted features. The fourth input is a flag which tells this module to show the PCA extracted features in a form of image. The fifth input is a threshold value that is used to identify important features from the generated PCA feature. The function returns an M x N matrix containing PCA feature of all the training set and N x N weight

of each original symbol in the training set where N is number of training sets and M is picture dimension. The PCA algorithm has the following steps:

Step 1: Get data

This module would accept an image matrix containing the training or testing images with a dimension of 7240x20 and 7240x1, respectively, which was obtained by taking the first and last frames of the video as key frames with a dimension of 70x50 and centroid value of each hands on 120 frames of the given video.

Step 2: Calculate Covariance Matrix in terms of vectors

The function starts with the calculation of the covariance matrix obtained by multiplying the image vector with its transpose which is obtained by using matlab's function `'transpose'`, which returns an N x N square matrix.

Step 3: Generate Eigenvector and Eigenvalue

The second step is calculating the eigenvectors and eigenvalues of the covariance matrix using matlab's built in function shown below:

```
[v d] = eig(covariance);
```

Where v is the eigenvector and d is the eigenvalue.

Step 4: Sort and eliminate those values whose eigenvalues is less than threshold

The next step of PCA is choosing the most important principal component and creates a feature vector, in which the notion of data compression and reduced dimensionality comes in action, to do so the threshold value obtained through experimentation is used to store only eigenvector and eigenvalues above the threshold. This would also determine the total number of important features to take into consideration. Then this would be followed by sorting the eigenvector in ascending order using matlab's built in function `fliplr(v)`.

Step 5: Normalize the Eigenvector to unit Magnitude

This step is where eigenvector of dimension NxN is normalized into a unit magnitude by dividing each element by the square root of the sum of squared column elements.

Step 6: Find Eigenvectors of actual Covariance Matrix

The next step continues by finding the eigenvectors of actual covariance matrix. This is obtained by multiplying the original input image with the normalized eigenvector, which have MxN dimension. The result of this step undergoes through the process of normalization using step 5.

Step 7: Find weight of each original image

The next step finding the weight of each original image in the training set in transformed space by multiplying matrixes of the original image and the actual covariance matrix returned by step 6. The output of this result is going to be used as a training set for classification.

3.4.3 Hand Trajectory Determination

On this module different activities were performed to extract motion information by following the trajectory of the hands' centroid. As noted by Elmezain et al [12] a gesture path is spatio-temporal pattern which consists of centroid points (x_{hand}, y_{hand}) shown in Figure 3.5 (a) below. So, the trajectory is determined between two consecutive points from hand gesture path by Equation 3.1.

$$\theta_t = \arctan\left[\frac{y_{t+1} - y_t}{x_{t+1} - x_t}\right] \quad ; t = 1, 2, \ldots, T-1 \qquad (3.1)$$

Where T represents the length of gesture path. The trajectory θ_t is quantized by dividing it by 20^o in order to generate the code words from 1 to 18.

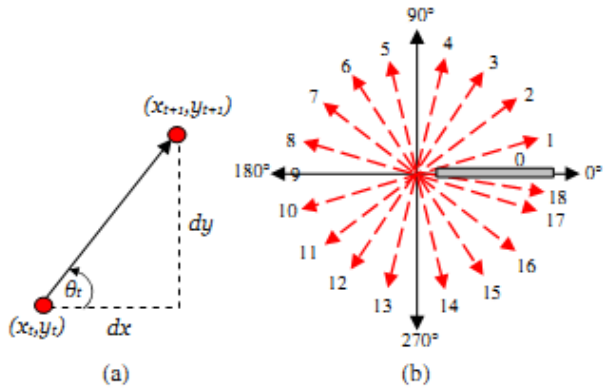


Fig 3.5 The trajectory and its code words (a) Trajectory between two consecutive points (b) directional Code words from 1 to 18 including also zero codeword taken from [20]

So having this in mind the code in Appendix A.5 is used to implement this module. The function has the form:

```
function thetaT = selectTrajectory(Movie, numFrames)
```

The function accepts a whole video file as an input, and total number of frames. First it extracts movie file and produce a frame/image and perform skin detection, hand contour detection to

obtain the centroid of each skin detected object and calculate the trajectory of each consecutive frames using equation 3.1. And concatenate the trajectory to get a vector of dimension 1 x 240.

After acquiring 7000x1 vector dimension of image feature for a single word and 240x1 hand trajectory angle for the two hands, a feature combination was made to acquire a total of 7240x1 feature vectors for a single word and 7240x58 for the whole training set. As described earlier PCA was applied to this feature vector for dimension reduction and feature representation.

3.5 K-Nearest Neighbor (KNN)

The task of classification is assigning a category to a new feature vector out of a given predefined categories, in order to recognize a given sign [23]. The category consists of a set of features obtained during the training phase using number of training images. Classification mainly concentrates on finding the best matching features vector for the new vector among the set of reference features.

For this research the researcher used K-Nearest Neighbor classification schemes and applied matlab's built functions '`knnclassify`' for both training and testing purpose.

In this case, this classification algorithm was given 58 data points for training and also new unlabelled data for testing. And the aim was to find the class label for the new point. The algorithm has different behavior based on k.

Case 1: k = 1 or Nearest Neighbor Rule – this is the simplest scenario, where given labeled point x the task is to find a point closest to x. Let it be y. Now nearest neighbor rule asks to assign the label of y to x.

Case 2: k = K or k-Nearest Neighbor Rule – this is a straightforward extension of 1NN, where we find the k nearest neighbor and do a majority voting.

The Matlab built in function for KNN classifications have the syntax of: `CLASS = KNNCLASSIFY (SAMPLE, TRAINING, GROUP)` – classifies each row of the data in `SAMPLE` into one of the groups in `TRAINING` using the nearest neighbor method. Whereas, `GROUP` is a grouping variable for `TRAINING`. Its unique values define groups, and each element defines the group to which the corresponding row of `TRAINING` belongs. The major task expected by the researcher is to prepare the needed inputs to the built in function and interpret the final result.

How KNN Works?

The general principle is to find the k training samples to determine the k-nearest neighbors based on a distance measure. Next, the majority of those k nearest neighbors decide the category of the next instance. Below are the step followed for KNN algorithm implementation:

Step 1: Determine k

Step 2: Calculate the distances between the new input and all the training data.

We can compute the distance between two scenarios using some distance function *d(x, y)*, where *x, y* are scenarios composed of N features, such that $x = \{x_1,\ldots,x_N\}$, $y = \{y_1,\ldots,y_N\}$.

Euclidean distance measuring: $d_E(x, y) = \sum_{t=1}^{N} \sqrt{x_t^2 - y_t^2}$

City block – sum of absolute difference: $d_A(x, y) = \sum_{t=1}^{N} |x_t - y_t|$

Cosine – One minus the cosine of the included angle between points $d_{CS}(x, y) = \dfrac{\sum_{i=1}^{n} X_i * Y_i}{\sqrt{\sum_{i=1}^{n} (X_i)^2} \times \sqrt{\sum_{i=1}^{n} (Y_i)^2}}$

Correlation – One minus the sample correlation between points $d_{Cor}(x, y) = \dfrac{d_{Cov}(x, y)}{\sqrt{d_{Var}(x)\, d_{Var}(y)}}$

Where d_{Cov} = Covariance distance & d_{Var} = Variance distance

Step 3: Sort the distance and determine k nearest neighbors based on the k-th minimum distance.

Step 4: Gather the categories of those neighbors.

Step 5: Determine the category based on majority or consensus vote.

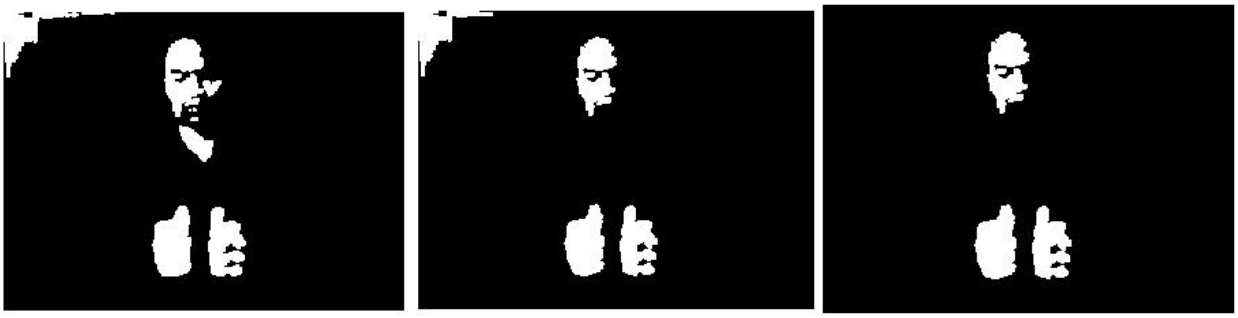
CHAPTER FOUR

EXPERIMENTAL RESULTS AND DISCUSSION

The design of the system is described in the previous chapter, and this chapter focus on the results obtained while conducting experimental analysis of the recognition system. The experimental results are also further discussed with some descriptive diagrams taken from the system. The performance of the system was measured in terms of its accuracy in recognizing a sign given the training set.

4.1 Detecting Skin Region

As described in section 3.3 a hybrid of HSV and RGB color space with explicitly defined skin threshold was used for skin detection. Even though, the skin detection is somehow constrained to red skinned individuals only. However, due to some lighting condition and reflection of shadow some regions were detected as skin regions. As explained in section 3.3 those non-skin regions were removed using matlab's *'bwareaopen'* function with a threshold determine while conducting an experiment on each training and testing videos. The threshold ranges between 1890 and 4500. This experimental result is shown in the Figure 4.1 below.



(a) (b) (c)

Fig 4.1 Skin Detection (a) both skin & non-Skin regions (b) Skin regions > 1890 (c) Skin regions > 4500

However, this threshold value differs from one video to another video in which the threshold value used for one image did not work for another image from the image collection as shown in the Figure 4.2 below.

(a) (b) (c)

Fig 4.2 Skin Detection (a) both skin & non-Skin regions (b) Skin regions > 3500 (c) Skin regions > 4500

Therefore, to resolve this issue a thorough experimentation on each video both from the training and testing set was conducted to determine the skin threshold. And the skin threshold used for one word might not be applicable for another word, in which either important detection might be removed or non-skin regions might not be remove that would increase the error rate of the system. Some of these values are shown in table 4.1 below.

	Words	Training		Testing	
		1st Frame	Last Frame	1st Frame	Last Frame
1	አልኮል	220	220	3500	2500
2	ዳቦ	220	220	2500	2500
3	ኬክ	220	220	900	900
4	ሊቀመንበር	220	220	900	900
5	ቡና	900	560	1200	1200
6	መተዋወቅ	255	365	1500	1500
7	መጋባት	100	270	250	250
8	የእኛ	360	450	155	1200
9	ሳምቡሳ	420	620	4200	1000

Table 4.1 Skin Threshold for the top 9 words

4.2 Extracting Image and Motion Features

4.2.1 Image Features

Manual component of sign language are mostly conveyed by the two hands. Typically, these components are characterized by hand shapes and movements. In order to identify those components an I x J image has to be considered that is often represented by a vector in an I ˙ J dimensional vector space. For this research an image file with a dimension of 480x640 was initially captured. However, handling high dimensional vector space is computationally inefficient. Therefore, as described in section 3.4.2 PCA was applied as a technique for dimension reduction and image feature representation.

In this study the assumption was to take the first and the last frames of the video sequence as images containing the starting and ending hand shape of a give word having a 70x50x2 dimension, in which both eigenvectors and eigenvalues were computed.

The challenge faced in this part of the study is that, as it can be recalled from the discussion of PCA in section 3.4.2, the plan was to use threshold values that is used to identify important features. With regard to this, the researcher took `[nEigValThres = 215.2502872]` as least value to represent the threshold and applying this threshold value changed the dimension of the eigenvector from 20x20 to 16x16. However, the problem came when going through step 7 of PCA algorithm execution, which was related to dimension incompatibility. And it was very challenging for the researcher. Therefore, threshold value of `[nEigValThres = 0.0001]` was considered for the research.

4.2.2 Motion Features

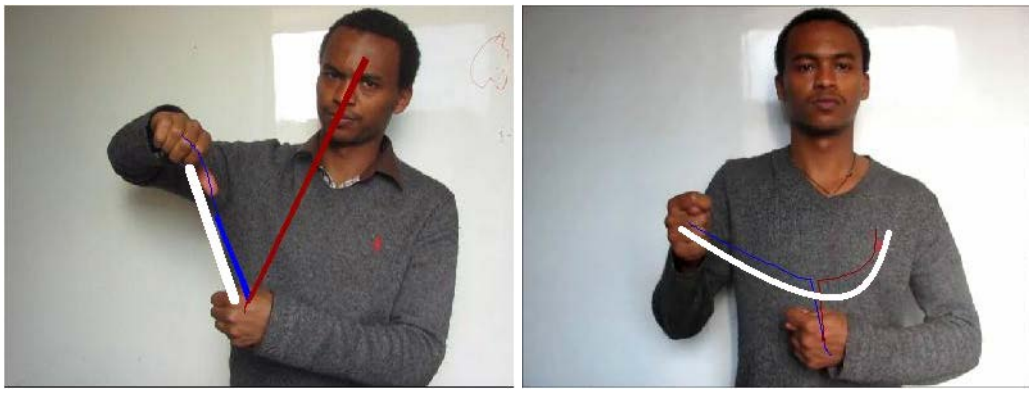
In addition to the manual features extracted from the captured images, features conveyed by the hand motions were also considered for this research. Therefore, hand trajectory information was extracted using the centroid of each detected skin region. The assumption was that only three skin regions were being considered in each frame of the video the signers face and his/her two hands. Therefore, with the use of each regions area and the assumption that the face region is the largest skin region image isolation was performed, so that centroid of the hand region can be taken for hand tracking. However, even though there was some correct hand isolation, there was

also incorrect isolation due to incorrect skin detection shown in Figure 4.3 below. Motion features was experimented using the function discussed in section 3.4.3.

(a) (b)

Fig 4.3 Hand Isolation for Hand Trajectory Determination (a) Correct Detection and Isolation (b) Incorrect Detection and Isolation

It was observed that due to the incorrect detection as well as isolation of the two hands keeping the robustness of the hand trajectory module was difficult. The result obtained for this particular case is show in Figure 4.4 below in which the correct hand trajectory was drawn using white line to show which one of the images is the correct one and which one is the wrong one.



(a) (b)

Fig 4.4 Hand Trajectory detection (a) Incorrect detection (b) Correct detection

After identifying centroid of each hand region as it has been discussed in section 3.4.3 the angle between two consecutive centroid points was calculated using equation 3.1. While conducting the experimentation the researcher have notice values such as NAN (not a number) of two points with the same x-coordinate, so the researcher have set a rule which will automatically assign a zero value for two consecutive points with the same x values. Meaning that in equation 3.1 there

is some sort of division between y coordinate and x coordinate of two consecutive points. So if two points have the same x value the subtraction is going to have zero value so division by zero is illegal. Therefore, in order to continue with the execution the researcher set the rule in which whenever subtraction of two x coordinate points is found to be zero, the angle is automatically assigned with a value of zero.

4.3 KNN Classifier

After acquiring features of the training set the next step was to use the test dataset and check whether the built feature vector returns the correct output or not. To do this task basically experimentation were done on KNN classification scheme. This scheme uses the training feature vector, the testing feature vector and class identifier. The experimentation was conducted on this classifier by changing the '`distance`' and '`rule`' argument of '`knnclassify`' built in function. The value from the '`distance`' argument includes '`Euclidean`', '`cityblock`', '`cosine`', and '`correlation`'; whereas the value for the '`rule`' argument includes '`nearest`', '`random`', and '`consensus`'. With regard to the 'rule' argument all the values returned the same result therefore, anyone of the rule can be used in the course of the recognition process. In addition to this the classification algorithm was also tested based on the input testing set in which the first input was original feature vector whereas the second input was after applying PCA on the testing set. Table 4.2 below show a summary of the output returned by the KNN classifier.

Given the original feature vector, the performance of the classifier differs depending on the distance used. Maximum accordingly achieved is 40% for Euclidean and City block and Correlation achieved a recognition performance of 20%. On the other hand, cosine achieved a recognition performance of 30%. So, we further tested by applying PCA for dimensionality reduction and selecting the best feature value for the input image.

Test Set	Expected Result	Euclidean		Cosine		City Block		Correlation	
		Before PCA	After PCA	Before PCA	After PCA	Before PCA	After PCA	Before PCA	After PCA
Alcohol	1	8	16	1	1	8	16	19	19
Bread	2	5	16	5	5	5	16	5	5
Cake	3	12	16	12	12	12	16	12	12
Chairperson	4	4	16	4	4	4	16	4	4
Coffee	5	5	16	4	4	5	16	4	4
Introduction	6	6	16	15	15	6	16	15	15
Marriage	7	7	16	7	7	7	16	7	7
Ours	8	16	16	16	20	16	16	20	20
Stranger	11	20	16	20	20	14	16	17	17
To Give Birth	14	12	16	12	12	12	16	12	12
Performance		**40%**	**0%**	**30%**	**30%**	**40%**	**0%**	**20%**	**20%**

Table 4.2 Summary of KNN classification result given an input before and after applying PCA

The classifier was given an input vector that have undergone through PCA analysis just like the training set. In this case Cosine achieved better than Euclidean and City Block with a recognition performance of 30%, and Correlation achieved the same result as the previous case. Therefore, based on the results obtained from this experimentation it is the researchers conclusion to use either Euclidean or City Block values to the distance parameter given an input before applying PCA.

4.4 System Outputs and Discussion

The final output of the recognition system is group of binary image containing the first and last frames of EthSL hand gesture for the test dataset, and the result returned by the KNN classification algorithms. Figure 4.5 below shows an output of the proposed design for a video clip of the word ‘CHAIRPERSON’ (**ሊቀመንበር**). For example, the first two images/frames in

Figure 4.5 represent the gestures extracted for the word ‘CHAIRPERSON’ from the video sequence of the test dataset. This is followed by the system outputs generated by the KNN classification algorithms having the distance value of Euclidean, Cosine, City block and Correlation, in doing so the result returned by KNN algorithm using Euclidean, City block and Correlation is the correct representation.

The outputs of the proposed design are used as inputs to EthSL word recognition system where the classification algorithms returns the group/class number where the returned gestures are recognized and then the system picks the underlying word as shown in Table 4.3. This table doesn’t show the exhaustive list of the words that exist in EthSL rather some of the words used for both the training and testing modules of the recognition system.

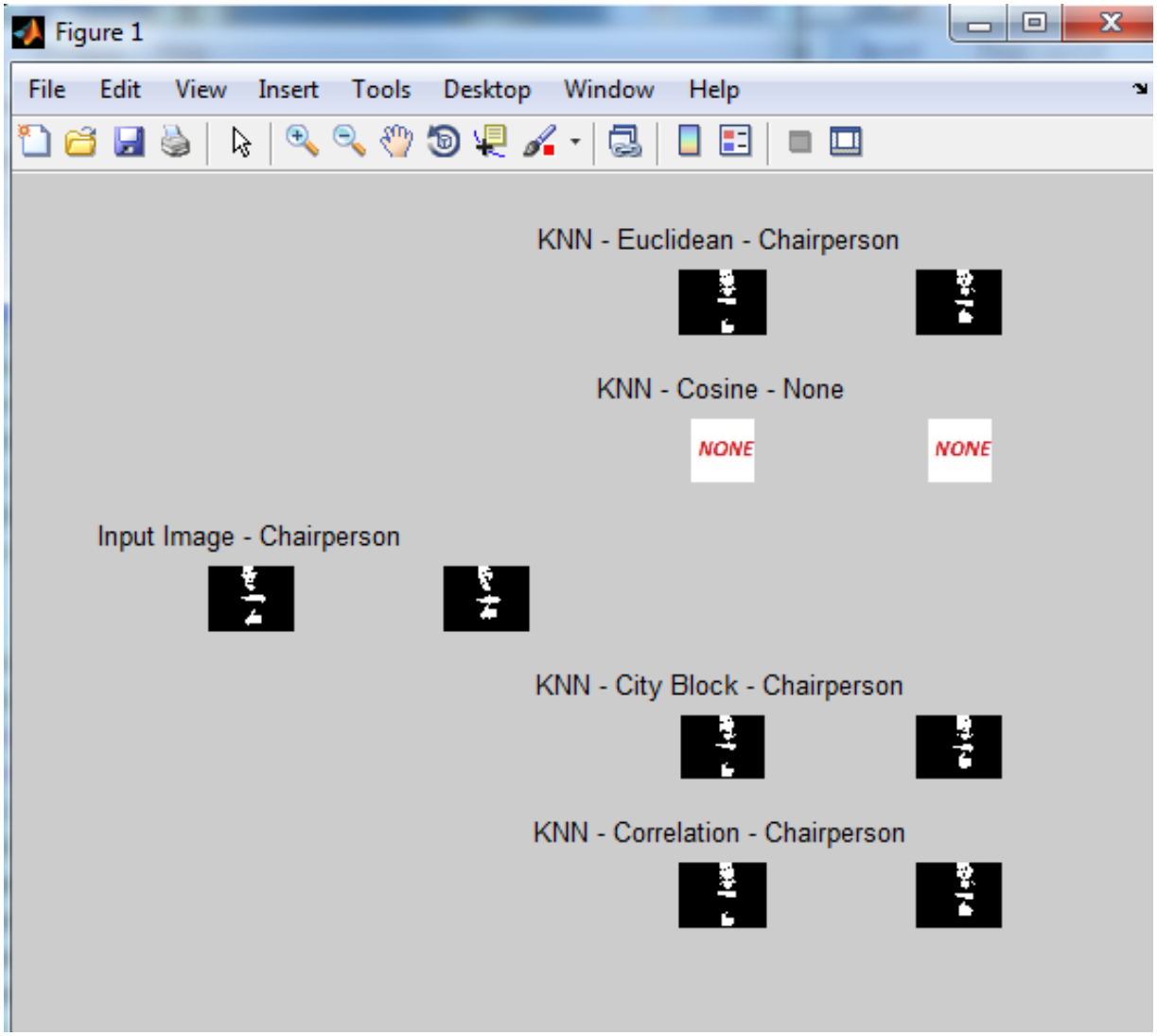


Fig 4.5 Sample output of the proposed design for the Word 'CHAIRPERSON' (**ሊቀመንበር**)

Table 4.4 shows 20 words with their underlying class/group number used to identify them in the recognition system. Therefore, the classification algorithm would give the numbers assigned to each word if there exist a similarity between the word found in the test set and the one found in the training dataset.

Word / ቃል	**Class / Group**	**Word / ቃል**	**Class / Group**
Alcohol / አልኮል	1	Stranger / ባዳ	11
Bread / ዳቦ	2	To Chop / መክተፍ	12
Cake / ኬክ	3	To Divorce / መፋታት	13
Chairperson / ሊቀመንበር	4	To Give Birth / መውለድ	14
Coffee / ቡና	5	Until / እስክ	15
Introduction / መተዋወቅ	6	We / እኛ	16
Marriage / መጋባት	7	Wheat / ስንዴ	17
Ours / የእኛ	8	Which / የትኛው	18
People / ህዝብ	9	Whiskey / ዊስኪ	19
Spring Roll / ሳምቡሳ	10	You / እናንተ	20

Table 4.3 List of Some EthSL Words used for training and testing

For example, Figure 4.6 below shows an output generated for the input test word 'COFFEE / ቡና' in which KNN classification algorithm returned 5, 0, 5, 4 as class or group. So, by refereeing Table 4.3 one can see that Euclidean and City Block returned 'COFFEE / ቡና' whereas Correlation returned 'CHAIRPERSON / ሊቀመንበር' and Cosine have returned none as an output for similar word input.

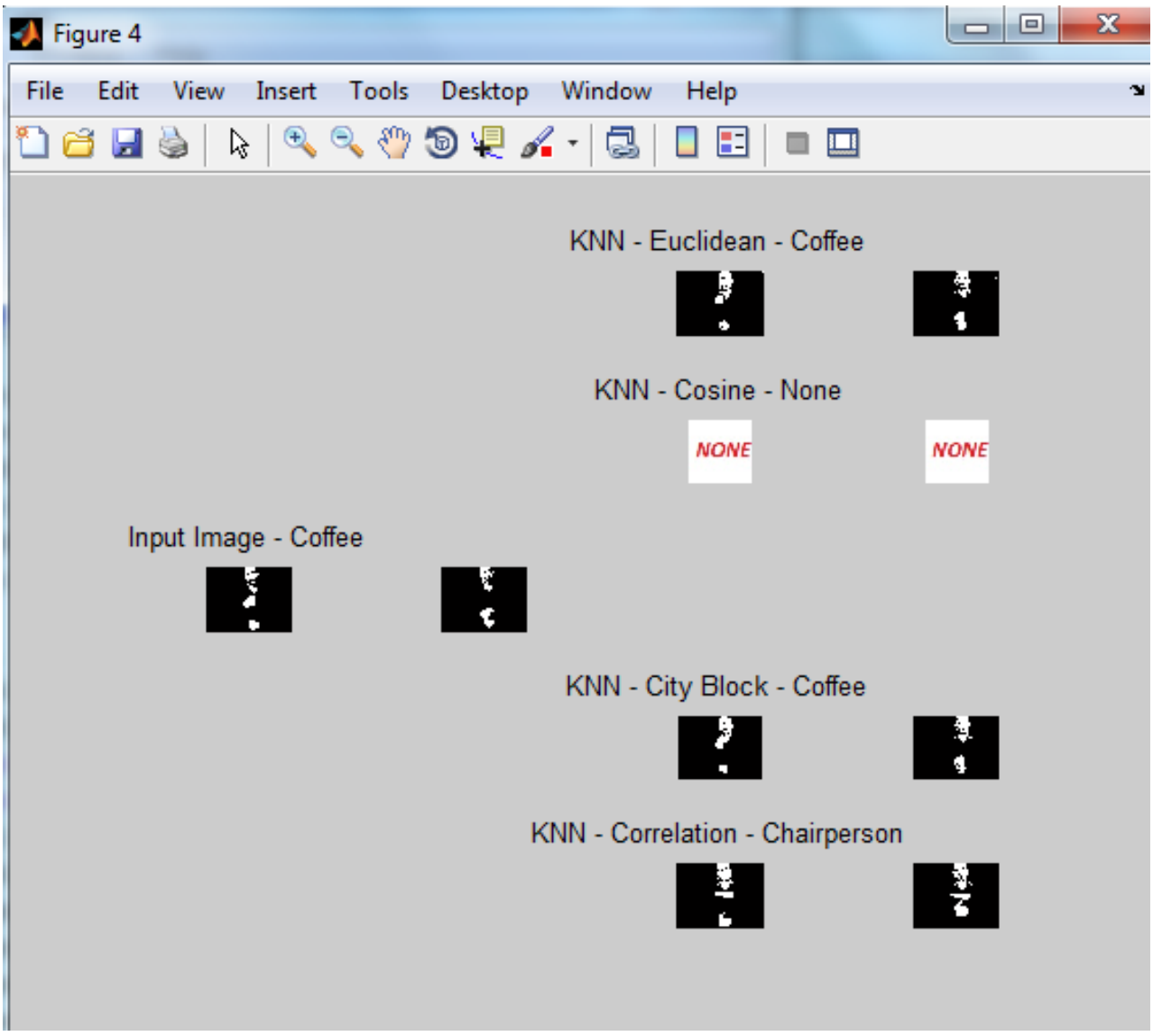


Fig 4.6 Sample output of the proposed design for the Word ‘COFFEE’ (ቡና)

Table 4.2 above show a summary of the results returned by the classification algorithms for 10 of the test images given as an input to the recognition system before and after applying PCA on the input image.

As it has been discussed above the recognition system is able to recognize word signs correctly with an overall accuracy of 40%. However, the remaining 60% is incorrect detection because of the reason related to image quality created because of the lighting condition, and camera motion. Figure 4.7 below shows the result returned for the input word ‘ALCOHOL / አልኮል’ and Euclidean returned ’OURS / የእኛ’, Cosine returned ‘To Chop / መክተፍ’, City block returned ’Whiskey / ዊስኪ’ whereas Correlation returned ’OURS / የእኛ’.

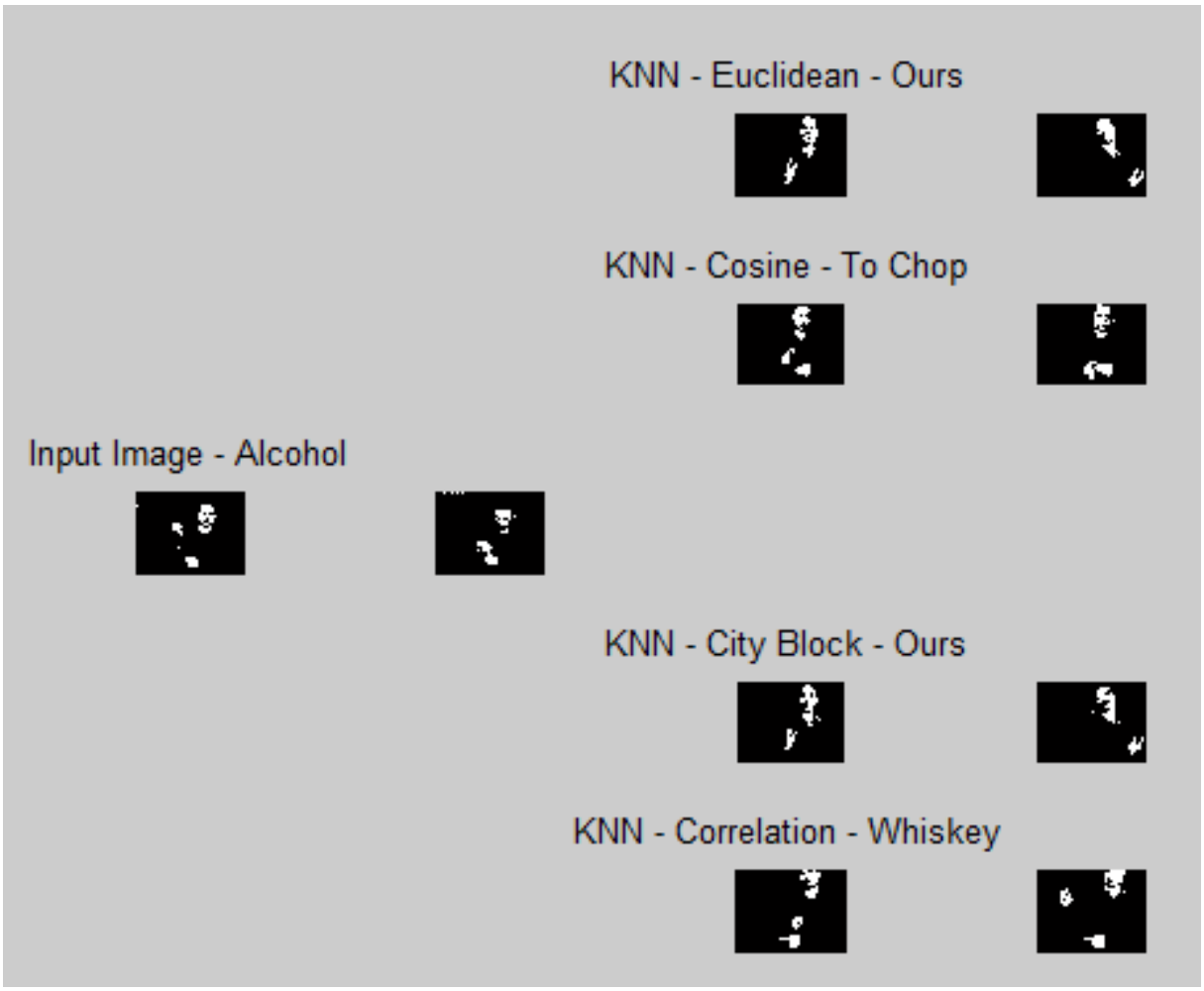


Fig 4.7 Sample output of the proposed design for the word ‘ALCOHOL’ (አልኮል)

If the RGB versions of the recognition result for EthSL word are required, the proposed design has outputs like shown in Figures 4.8 and 4.9.

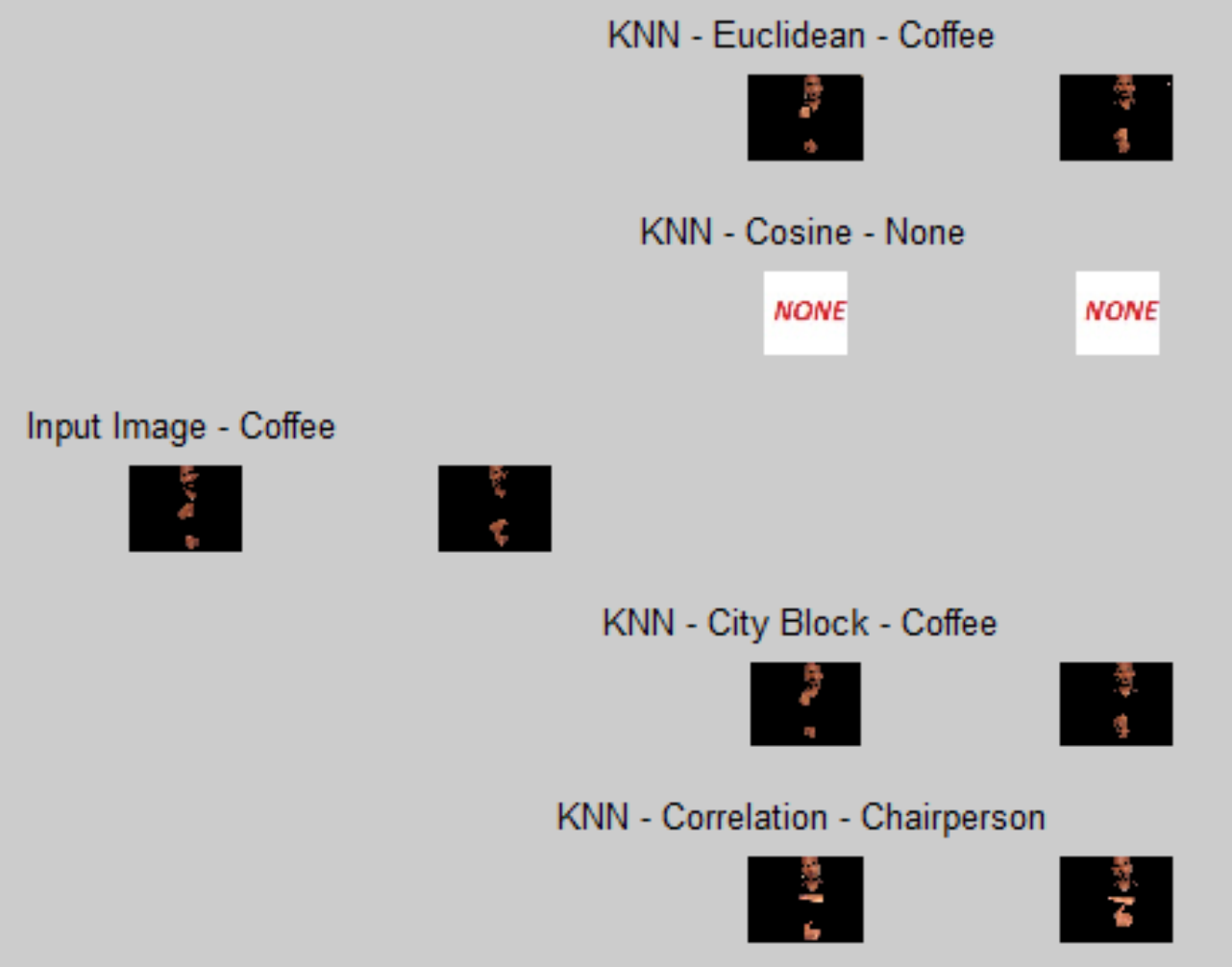


Fig 4.8 An RGB version output for a video clip of the Word ‘COFFEE’ (ቡና)

Fig 4.9 An RGB version output for a video clip of the word ‘ALCOHOL’ (አልኮል)

CHAPTER FIVE

CONCLUSION AND FUTURE WORKS

5.1 Conclusion

The purpose of this thesis was to design and implement the signer dependent word level EthSL recognition system for the conversion of EthSL sign to text. The end result expected from this research was a system that is capable of analyzing EthSL sign and recognize its equivalent Amharic text representation. To achieve this result, the system comprises of video pre-processing, a skin segmentation, hand detection and segmentation to extract both manual features and hand trajectory, and finally a feature classification modules.

A hybrid color space between HSV and RGB with explicitly defined skin threshold was used for skin detection, but the problem related to skin colored objects and lighting condition were bottleneck for the skin detection module in which objects with no interest to the research were detected, to resolve this problem Matlab's built in function with threshold which is determined through experimentation is used. The hand detection and segmentation module follows the skin segmentation to detect the articulator's hand based on the region property of the binary image. Then it is followed firstly by the manual feature extractor which extracts the manual hand shape, hand location, and hand orientation features and secondly by the hand trajectory detector which basically was used to extract motion features of the signer's hand.

A feature vector was created by combining manual feature such as hand shape, hand location, hand orientation and hand motion features of a given sign. For this study a video was captured with a 480x640 dimension. Therefore, handling each pixel one by one was computationally expensive and hence dimensionality reduction was done using PCA. For gesture classification KNN classifier is explored. Experimental result shows that the classifier achieved an overall accuracy of 40%. The reason for the low overall system performance is the non-existence of well constructed corpus for experimentation. To solve this problem, the researcher captured videos of signs for the experiment. But, during capturing a number of challenges happen, including: Poor video quality, Camera vibration, and lighting problem. This problem occurred due to the fact that

the researcher used a junior signer which in return created inconsistency in hand flinging during signing.

5.2 Future Work

While working on this thesis, there were many problems found that need future research. Some of them are listed below:

- ✓ For this work, the first and the last frames are considered to determine the manual features of the hand, but in the literature it is the recommendation of most of the scholars to use key frames that basically represent hand gesture of a signer. So, it is also my recommendation for future researchers to use key frame extraction techniques.
- ✓ In this thesis, an explicitly defined skin threshold was used in which there were some wrong detections, which in return made both the hand segmentation and motion trajectory modules to be a bit harder. Therefore, it is the researcher's recommendation to use adaptive skin detection method to gain a better result.
- ✓ For this study, isolated video were captured for each training and testing datasets. It is the researcher's recommendation to use continuous videos while signers are signing naturally so that the recognition system can be extended to sentence level.
- ✓ It is the researcher's recommendation to design a system that can recognize both EthSL EMA and word to come up with a full-fledged system.

5.3 Thesis Contribution

The previous sections of this chapter deals with the some of the features of EthSL and the technique that is going to be followed in order to come up with a recognition process. In doing so unique characteristics of EthSL signs that stands for a word have been identified. This study assumes that the first and the last frames of the video sequence are enough for feature extraction whereas for the hand motion trajectory unlike the hand trajectory that we use to EMA variant the hand trajectory that we use to create words using dedicated sign is a bit tough. So, the hand motion trajectory is assumed as part of the hand feature extracted from the selected two frames. The contributions of the thesis can be summarized as follows:

- ➢ For hand detection and segmentation, the researcher has used a hybrid of HSV and RGB color space to get a better skin detection result.

- For feature extraction, the researcher have used Principal Component Analysis (PCA) to extract both manual and motion trajectory features
- For gesture classification, the researcher experimented on one of the most known Matlab's classification algorithms i.e. KNN and selected it as the most appropriate one.

Reference

[1] *Country Profile on Disability, Federal Democratic Republic of Ethiopia.* Japan Internation Cooperation Agency Planning and Evaluation Department, March, 2002.

[2] D. Assefa, "Amharic Speech Training for the Deaf," M.S. thesis, Addis Ababa University, Addis Ababa, Ethiopia, 2006.

[3] *Ethiopian Sign Language & Deaf Culture Program Unit*, [online] http://www.aau.edu.et/index.php/sign-language-overview (Accessed: December 13, 2012)

[4] S. Khan, G.S. Gupta, D. Bailey, S. Demidenko, and C. Messom, "Sign Language Analysis and Recognition: A Preliminary Investigation," *24th International Conference on Image and Vision Computing IVCNZ*, 2009, pp. 119-123, New Zealand.

[5] J. Piater, T. Hoyoux, and W. Du, "Video Analysis for Continuous Sign Language Recognition," *4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies & 7th International Conference on Language Resources and Evaluation (LREC)*, 2010, Malta.

[6] H. Cooper, E.J. Ong, N. Pugeault, and R. Bowden, "Sign Language Recognition using Sub-Units," *Journal of Machine Learning Research*, vol. 13, pp. 2205-2231, 2012.

[7] M.P. Lewis, "Ethnologue: Languages of the World," 6th ed. Dallas, Tex.: *SIL International*. [online] 2009, http://www.ethnologue.com/ (Accessed: December 12, 2012)

[8] K. Assaleh, and M. Al-Rousan, "Recognition of Arabic Sign Language Alphabet Using Polynomial Classifiers," *EURASIP Journal on Applied Signal Processing*, vol. 2005, pp. 2136-2145, 2005.

[9] A.K. Sarkaleh, F. Poorahangaryan, B. Zanj, and A. Karami, "A Neural Network Based System for Persian Sign Language Recognition," *IEEE International Conference on Signal and Image Processing Applications*, 2009, pp. 145-149

[10] Y.F. Admasu, and K. Raimond, "Ethiopian Sign Language Recognition Using Artificial Neural Network," *10th International Conference on Intelligent Systems Design and Applications (ISDA)*, IEEE, 2010, pp. 995-1000

[11] A. Tsegay, and K. Raimond, “Offline Candidate Hand Gesture Selection and Trajectory Determination for Continuous Ethiopian Sign Language,” *Journal of Theoretical and Applied Information Technology*, vol. 36, no. 1, pp. 145-153, 2012.

[12] M. Elmezain, A. Al-Hamadi, J. Appenrodt, and B. Michaelis, “A Hidden Markov Model-Based Isolated and Meaningful Hand Gesture Recognition,” *International Journal of Electrical and Electronics Engineering*, vol. 3, no. 3, pp. 156-163, 2009.

[13] D.F. Wolde, “Machine Translation System for Amharic Text to Ethiopian Sign Language,” M.S. thesis, Addis Ababa University, Addis Ababa, Ethiopia, 2011.

[14] E.S. Klima, and U. Bellugi, *The signs of language*. Cambridge, MA: Harvard University Press, 1979.

[15] S.C.W. Ong, and S. Ranganath, "Automatic Sign Language Analysis: A Survey and the Future beyond Lexical Meaning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 873-891, June 2005.

[16] Paul, “ASLDeafinedBlog,” [online] July 31, 2011, http://blog.asldeafined.com/2011/07/non-manual-markers-in-american-sign-language-asl/ (Accessed: March 4, 2013)

[17] O. Aran, “Vision Based Sign Language Recognition: Modeling and Recognizing Isolated Signs with Manual and Non-Manual Components,” Ph.D. dissertation, Bogazici University, Turkey, 2008.

[18] G.R.S. Murthy, and R.S. Jadon, “A Review of Vision Based Hand Gestures Recognition,” *International Journal of Information Technology and Knowledge Management*, vol. 2, no. 2, pp. 405-410, 2009.

[19] C. Neidle, J. Kegle, D. MacLaoughline, B. Bahan, and R.G. Lee, *The Syntax of American Sign Language: Functional Categories and Hierarchical Structure*, Cambridge, MA. MIT Press, 2000.

[20] B. Yimam, “*አጭርና ቀላል የአማርኛ ሰዋስው*” *[short and simple Amharic Grammar]*. Addis Ababa: Alpha publishers, 2002.

[21] Ethiopian Sign Language Dictionary, Published by Ethiopian National Association of the Deaf (ENAD), 2008

[22] I.N. Sandjaja, Marcos N., "Sign Language Number Recognition," *NCM ’09 Proceedings of the 2009 5th International Joint Conference on INC, IMS and IDC*, pp. 1503-1508, 2009.

[23] N.R. Albelwi, and Y.M. Alginahi, "Real-Time Arabic Sign Language (ArSL) Recognition," *3rd International Conference on Communication and Information Technology*, 2012, pp. 497-501.

[24] A.A. Youssif, A.E. Aboutabl, and H.H. Ali, "Arabic Sign Language (ArSL) Recognition System Using HMM," *(IJACSA) International Journal of Advanced Computer Science and Applications*, vol. 2, no. 11, pp. 45-51, 2011.

[25] M. Mohandes, and M. Deriche, "Image based Arabic Sign Language recognition," *Proceeding of the 8th International Symposium on Signal Processing and Its Applications*, vol. 1, pp. 86-89, 2005.

[26] A. Elgammal, C. Muang, and D. Hu "Skin Detection - a Short Tutorial," *Encyclopedia of Biometrics*, Springer-Verlag, Berlin, Heidelberg, 2009.

[27] V. Vezhnevets, V. Sazonov, and A. Andreeva, "A Survey on Pixel-Based Skin Color Detection Techniques," . *In PROC. GRAPHICON '03*, 2003, pp. 85-92.

[28] S.K. Vaishali, and S.D. Lokhande, "Appearance Based Recognition of American Sign Language Using Gesture Segmentation," *International Journal on Computer Science and Engineering*, vol. 2, no. 3, pp. 560-565, 2010.

[29] S.K. Singh, and A. Kathane, "Various Methods for Edge Detection in Digital Image Processing," *International Journal of Computer Science and Technology*, vol. 2, no. 2, pp. 188-190, 2011.

[30] P. Kakumanu, S. Makrogiannis, and N. Bourbakis, "A survey of skin-color modeling and detection methods," *Pattern Recognition*, vol. 40, no. 3, pp. 1106-1122, 2007.

[31] K.K. Bhoyar, and O.G. Kakde, "Skin Color Detection Model Using Neural Networks and its Performance Evaluation," *Journal of Computer Science*, vol. 6, no. 9, pp. 963-968, 2010.

[32] V.N. Pashaloudi, and K.G. Margaritis, "Feature Extraction and Sign Recognition for Greek Sign Language," *Proceeding (385-045) Artificial Intelligence and Soft Computing*, ACTA Press, 2003.

[33] U.V. Agris, J. Zieren, U. Canzler, B. Bauer, and K.F. Kraiss, "Recent developments in visual sign language recognition," *Springer Journal on Universal Access in the Information Society*, vol. 6, no. 4, pp. 323-362, February 2008.

[34] K.G. Derpanis, "A Review of Vision-Based Hand Gestures," Internal Report, Department of Computer Science. York University, February 2004.

[35] Q. Chen, "Real-Time Vision-Based Hand Tracking and Gesture Recognition," Ph.D. dissertation, University of Ottawa, Canada, 2008.

[36] Y. Bar-Yam, *Dynamics of Complex Systems*. Addison-Wesley, 1997.

[37] R. Rojas, *Neural Networks: A Systematic Introduction*, Berlin: Springer-Verlag, 1996.

[38] D. Comaniciu and P. Meer. "Mean shift: A robust approach toward feature space analysis." *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, pp. 603–619, 2002. Available at http://www.caip.rutgers.edu/riul/research/papers/pdf/mnshft.pdf.

[39] P. Blunsom, Hidden Markov Models, August 19, 2004.

[40] H. Abdi, and L.J. Williams, "Overview Principal Component Analysis," *WIREs Computational Statistics*, vol. 2, pp. 433-459, July/August 2010, © John Wiley & Sons, Inc.

[41] I.S. Lindsay. *A tutorial on Principal Components Analysis*. February 26, 2002.

[42] M.S. Sinith, S.G. Kamal, B. Nisha, S. Nayana, S. Kiran, and P.S. Jith, "Sign Gesture Recongnition Using Support Vector Machine," *International Conference on Advances in Computing and Communications (ICACC)*, pp.122-125, 2012.

[43] G.M. Awad, "A framework for sign language recognition using support vector machines and active learning for skin segmentation and boosted temporal sub-units," Ph.D. dissertation, Dublin City University, Ireland, 2007.

[44] R. Naoum, H.H. Owaied, and S. Joudeh, "Development of a new Arabic Sign Language Recognition Using K-Nearest Neighbor Algorithm," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 3, no. 8, pp. 1173-1178, August 2012.

[45] S. Roy, and S.K. Bandyopadhyay, "Face detection using a hybrid approached that combines HSV and RGB," *International Journal of Computer Science and Mobile Computing*, vol. 2, no. 3, pp. 127-136, March 2013.

[46] I. Aldasouqi, and M. Hassan, "Human face detection system using HSV," *Proceeding CSECS '10 Proceedings of the 9th WSEAS international conference on Circuits, systems, electronics, control & signal processing*, pp. 13-16, 2010.

[47] T. Starner and A. Pentland "Visual Recognition of American Sign Language using Hidden Markov Models", *Proc. Intl. Workshop on Automatic Face and Gesture Recognition*, 1995.

[48] T. Starner, J. Weaver, and A. Pentland, "A Wearable Computer Based American Sign Language Recognizer" *Wearable Computers, 1997. Digest of Papers., First International Symposium on* , vol., no., pp.130,137, 13-14 Oct. 1997

[49] S. Bernhardt., "Mean Shift Algorithm Implementation Matlab code," Available at http://www.mathworks.com/matlabcentral/fileexchange/35520-mean-shift-video-tracking accessed on: June 2013

Appendix A : MATLAB Code

A.1 Main Program

```
% ADDIS ABABA UNIVERSITY
% COLLEGE OF NATURAL AND COMPUTATIONAL SCIENCE
% SCHOOL OF INFORMATION SCIENCE
% DESIGN AND IMPLEMENTATION OF :
        % ==== ISOLATED WORD-LEVEL EthSL RECOGNITION ====
clc;
clear all;
close all;

%% Add training and test directories to path

addpath([pwd '/EthSLR/Training/'])
addpath([pwd '/EthSLR/Testing/']);

%% Define variables
% No of Words used for Training
cWordTrain = [{'Alcohol'}, {'Bread'}, {'Cake'}, {'Chairperson'}, ...
           {'Coffee'}, {'Introduction'}, {'Marriage'}, {'Ours'}, ...
           {'People'}, {'springRoll'}, {'Stranger'}, {'toChop'}, ...
           {'toDivorce'}, {'toGiveBirth'},{'Until'}, {'We'}, ...
           {'Wheat'}, {'Which'}, {'Whiskey'}, {'You'}];
nTrainingSamples = 1;   % No of training videos used
nRows = 70;             % No of rows for each Frames/Images
nColumns = 50;          % No of columns for each Frames/Images
nEigValThres = 0.0001;  % Threshold below which to ignore eigen vectors
ShowOutput = 0;
ImgMat = [];

%% Perform preprocessing of Training images

for i = 1 : size(cWordTrain,2)
    for j = 1 : nTrainingSamples
        %% Training Video
        trainVideo = strcat(cWordTrain(i),'-train',int2str(j),'.avi');
        trainVideo = cell2mat(trainVideo);

        %% Perform preprocessing of Training Video
        colVect = preprocess(trainVideo);
        ImgMat = cat(2, ImgMat, colVect);
    end
end
```

```
%% Train the System

[PCAFeatures omega] = EthSLPCA(ImgMat, nRows, nColumns,...
     ShowOutput,nEigValThres);

%% Test Classifier
% No of Words used for Testing
cWordTest = [{'Alcohol'}, {'Bread'}, {'Cake'}, {'Chairperson'}, ...
           {'Coffee'}, {'Introduction'}, {'Marriage'}, {'Ours'}, ...
           {'Stranger'}, {'toGiveBirth'}];
nTestingSamples = 1;
PCACorrect_SVM = zeros(size(cWordTest,2),1);
PCACorrect_KNN = zeros(size(cWordTest,2),1);
classes = zeros(size(cWordTest,2),4);
for i = 1 : size(cWordTest,2)
    for j = 1 : nTestingSamples
        %% Input Video
        testVideo = strcat(cWordTest(i),'-test',int2str(j),'.avi');
        testVideo = cell2mat(testVideo);

        %% Perform preprocessing of Input Video
        procVideo = preprocess(testVideo);
        [PCAFeaturesIN omegaIN] = EthSLPCA(procVideo, nRows,...
            nColumns,ShowOutput, nEigValThres);

        InImWeight = EthSLRPCAget(procVideo,PCAFeatures');
        InImWeightPCA = EthSLRPCAget(PCAFeaturesIN,PCAFeatures');

        %% Perform KNN and SVM classification
        classKnn = EthSLknn(cWordTrain, nTrainingSamples,...
            InImWeight, omega);
        classSvm = EthSLsvm(cWordTrain, nTrainingSamples, ,...
            InImWeight, omega);
        classKnnPCA = EthSLknn(cWordTrain, nTrainingSamples, ...
            InImWeightPCA, omega);
        classSvmPCA = EthSLsvm(cWordTrain, nTrainingSamples, ...
            InImWeightPCA, omega);

        classes(i,1) = classKnn;
        classes(i,2) = classSvm;
        classes(i,3) = classKnnPCA;
        classes(i,4) = classSvmPCA;

        if classKnn == i
```

```
86              PCACorrect_KNN(i) = PCACorrect_KNN(i) + 1;
87          end
88          if classSvm == i
89              PCACorrect_SVM(i) = PCACorrect_SVM(i) + 1;
90          end
91      end
92  end
93  dispResult(classes, cWordTrain, cWordTest);
94  disp('The End');
```

A.2 Video Pre-Processing

The following function accepts a path containing an AVI movie file and returns a vector containing both the manual feature and hand motion trajectory of the given file.

```
1   function result = preprocess(vidPath)
2
3   readerobj = mmreader(path);
4
5   %% Read in all video frames.
6   vidFrames = read(readerobj);
7
8   %% Get the number of frames.
9   numFrames = readerobj.NumberOfFrames;
10
11  h = readerobj.Height;
12  w = readerobj.Width;
13  Movie(1:numFrames) = struct('cdata', zeros(h, w, 3, 'uint8'), ...
14          'colormap', []);
15
16  %% Read one frame at a time.
17  for k = 1 : numFrames
18      mov(k).cdata = vidFrames(:,:,:,k);
19  end
20
21  result = [];
22
23  colVect = selectVideoFrames(Movie, numFrames);
24  thetaT = selectTrajectory(Movie, numFrames);
25
26  result = cat(1, colVect, thetaT);
27
28  end
```

A.3 Select Video Key Frames

This function accepts a movie and total number frames and returns column vector containing manual feature of the first and last frames of the given video input.

```
function colVect = selectVideoFrames(Movie, numFrames)

colVect = [];
rgbImageFr = Movie(1).cdata;
binImageFr = skinSegmentation(rgbImageFr);
binImageFr = bwareaopen(binImageFr,620);
Mask = ones(2,2);
binImageFr = imerode(binImageFr, Mask);
Mask = ones(5,5);
binImageFr = imdilate(binImageFr, Mask);
binImageFr = imfill(binImageFr, 'holes');
binImageFr = imresize(binImageFr, [70 50]);
for i = 1 : 70
    for j = 1 : 50
        % Column Vector
        colVect = cat(1, colVect, binImageFr(i,j));
    end
end

rgbImageLs = Movie(numFrames).cdata;
binImageLs = skinSegmentation(rgbImageLs);
binImageLs = bwareaopen(binImageLs,620);
Mask = ones(2,2);
binImageLs = imerode(binImageLs, Mask);
Mask = ones(5,5);
binImageLs = imdilate(binImageLs, Mask);
binImageLs = imfill(binImageLs, 'holes');
binImageLs = imresize(binImageLs, [70 50]);
for i = 1 : 70
    for j = 1 : 50
        % Column Vector
        colVect = cat(1, colVect, binImageLs(i,j));
    end
end

end
```

A.4 Skin Color Segmentation

The following functions takes an RGB image as input and returns binary Image.

```
function [binImage] = skinSegmentation(rgbImage)

%=== Code For Skin Detection & Segmentation:

rgbImage = imadjust(rgbImage, [0 0.8], [0 1]);

hsvImage = rgb2hsv(rgbImage);
H = hsvImage(:,:,1);
S = hsvImage(:,:,2);
V = hsvImage(:,:,3);

[rows, cols, numberOfChannels] = size(hsvImage);

for r = 1 : rows
   for c = 1 : cols
       if 0 <= H(r,c) && H(r,c) <= 0.25 && 0.15 <= S(r,c)...
           && S(r,c) <= 0.9
           binImage(r,c) = 1;
       else
           binImage(r,c) = 0;
       end
   end
end

RGB = rgbImage;

for i = 1: rows
   for j = 1: cols
      if binImage(i,j)==1
           RGB(i,j,:) = RGB(i,j,:);
       else
           RGB(i,j,:) = 0;
       end
  end
end

R = RGB(:,:,1);
G = RGB(:,:,2);
B = RGB(:,:,3);

for row = 1 : size(RGB,1)
   for col = 1 : size(RGB,2)
       maxRG = max(R(row,col),G(row,col));
       maxRGB = max(maxRG,B(row,col));
```

```
45          minRG = min(R(row,col),G(row,col));
46          minRGB = min(minRG,B(row,col));
47          if R(row,col) > 95 && G(row,col) > 40 && ...
48              B(row,col) > 20 && (maxRGB - minRGB ) > 15 && ...
49              abs(R(row,col) - G(row,col)) > 15 && R(row,col) >...
50              G(row,col) && R(row,col) > B(row,col) ...
51               binImage(row,col) = 1;
52          else
53              binImage(row,col) = 0;
54          end
55      end
56  end
57  end
```

A.5 Hand Trajectory Determination

The following code is used to determine the hand motion trajectory. It accepts the movie file and the total number of frames in the movie. And the output of this function is a vector containing the angle value of both hands of the signer.

```
1   function thetaT = selectTrajectory(Movie, numFrames)
2   % Variable Declaration
3   cc1 = [];
4   cc2 = [];
5   cc3 = [];
6
7   for fr = 1 : numFrames
8       rgbImage = Movie(fr).cdata;
9       binImage = skinSegmentation(rgbImage);
10
11      %% Fill In The Holes:
12      binaryImage = imfill(binaryImage, 'holes');
13      binaryImage = isolateHand(binaryImage);
14
15      %% Putting Bounding Boxes Around Detected Blobs and Counting Them
16      binaryImage = bwareaopen(binaryImage, 250);
17      labeledImage = bwlabel(binaryImage, 8);
18      blobMeasurements = regionprops(labeledImage, binaryImage, 'all');
19
20      numObj = size(blobMeasurements, 1);
21      for i = 1 : numObj
22          centroid = blobMeasurements(i).Centroid;
23          x = round(centroid(1));
24          y = round(centroid(2));
25          if numObj == 3
26              if i == 1
27                  cc1 = cat(1, cc1, [x y]);
```

```
28            elseif i == 2
29                cc2 = cat(1, cc2, [x y]);
30            elseif i == 3
31                cc3 = cat(1, cc3, [x y]);
32            end
33        elseif numObj == 2
34            if i == 1
35                cc1 = cat(1, cc1, [x y]);
36            elseif i == 2
37                cc2 = cat(1, cc2, [x y]);
38            end
39        elseif numObj == 1
40            cc1 = cat(1, cc1, [x y]);
41        end
42    end
43 end
44 thetaTCC1 = zeros(120,1);
45 dimCC1 = size(cc1,1);
46 for k = 1 : dimCC1 - 1
47    cc1X = cc1(k+1,1) - cc1(k,1);
48    cc1Y = cc1(k+1,2) - cc1(k,2);
49    if cc1X ~= 0
50        thetaTCC1(k) = (atan(cc1Y / cc1X))/20;
51    else
52        thetaTCC1(k) = 0;
53    end
54    if thetaTCC1(k)< 0
55        thetaTCC1(k) = abs(thetaTCC1(k));
56    end
57 end
58 thetaTCC2 = zeros(120,1);
60 dimCC2 = size(cc2,1);
61 for k = 1 : dimCC2 - 1
62    cc2X = cc2(k+1,1) - cc2(k,1);
63    cc2Y = cc2(k+1,2) - cc2(k,2);
64    if cc2X ~= 0
65        thetaTCC2(k) = (atan(cc2Y / cc2X))/20;
66    else
67        thetaTCC2(k) = 0;
68    end
69    if thetaTCC2(k)< 0
70        thetaTCC2(k) = abs(thetaTCC2(k));
71    end
72 end
```

```
73
74 thetaT = cat(1, thetaTCC1, thetaTCC2);
75
76 end
```

A.6 Hand Isolation

This function is used to isolate the hand from a given binary image that has under gone through the process of skin detection using the matlab code in Appendix A.4, and the output of this function is a binary image containing only the hand of the signer taken from [11].

```
1  function segImage = isolateHand(segImage)
2
3  L = bwlabel(segImage);
4  [BB NO] = bwlabeln(L); % NO represents number of objects in segmented image
5  prop = regionprops(L,'Area');
6  binaryImage = bwareaopen(segImage, 250);
7  binLabel = bwlabel(binaryImage, 8);
8  blobMeasure = regionprops(binLabel, binaryImage, 'all');
9  numObj = size(blobMeasure, 1);
10 for ar = 1 : NO
11     blobArea(ar,1) = prop(ar).Area;
12 end
13 ind2 = 0;
14 ind = 0;
15 AREA = blobArea;
16 blobArea = sort(blobArea,'descend');
17 if numObj >= 3
18     for i = 1 : NO
19         if AREA(i) == blobArea(2)
20             ind = i;
21         elseif AREA(i) == blobArea(3)
22             ind2 = i;
23         end
24     end
25 elseif numObj < 3
26     for i = 1 : NO
27         if AREA(i) == blobArea(1)
28             ind = i;
29         elseif AREA(i) == blobArea(2)
30             ind2 = i;
31         end
32     end
33 end
34 for aa = 1 : size(L, 1)
```

```
    for bb = 1 : size(L, 2)
        if L(aa,bb) ~= ind
            if L(aa,bb) ~= ind2
                segImage(aa,bb) = 0;
            end
        end
    end
end
end
```

A.7 Feature Extraction

The following code is used to extract manual features from a given image matrix, and the input image goes under PCA process so that the function can return an MxN PCA feature and NxN eigenvalues.

```
function [PCAFeature omega] = EthSLPCA (ImgMat, nRows, nColumns,
                ShowOutput, nEigValThres)
%% Carry out PCA to extract features
% Find Covariance matrix in terms of vectors
L = ImgMat'*(ImgMat); % Covariance Matrix

[vv dd] = eig(L); % vv is eigen vectors, dd = eigen values
% Sort and eliminate those whose eigenvalue is less than threshold
v = zeros(size(vv));
d = zeros(1,size(dd,1));
NoOfFeatures = 0;
for i = 1 : size(vv,2)
   if dd(i,i) > nEigValThres
       v(:,i) = vv(:,i); % Store only vectors above threshold
       d(i) = dd(i,i);   % Store only Eigne values above threshold
       NoOfFeatures = NoOfFeatures + 1; % Count no of vectors saved
   end
end

%% Sort the Eigen Vectors from ascending to descending sequence
v = fliplr(v);

%% Normalize Eigen vectors to unit magnitude
for i = 1 : NoOfFeatures % access each column
   kk = v(:,i);
   temp = sqrt(sum(kk.^2)); % Calculate Magnitude
   v(:,i) = v(:,i)./temp; % Normalize each vector
end

```

```
%% Find Eigenvectors of actual Covariance matrix = ImgMat*ImgMat'
% This is for using v2 = X * v1
u = ImgMat * v;

%% Normalize the Eigen vectors
for i = 1 : NoOfFeatures
   kk = u(:,i);
   temp = sqrt(sum(kk.^2)); % Find magnitude
   u(:,i) = u(:,i)./temp; % Normalize Eigen vectors
end

%% Show the PCA extracted features
if(ShowOutput == 1)
   for i = 1 : NoOfFeatures
       % Display Extracted features
       f = figure();
       set(f,'name','Extracted Eigen features')
        % Reshape each vector to image
       Img = reshape(u(:,i),nRows,nColumns);
       imagesc(Img);
       axis equal;
       colormap('gray');
       set(gca, 'fontsize', 28);
   end
end

%% Find the weight of each original symbol in the training set in
   transformed space
omega = zeros(NoOfFeatures,NoOfFeatures);
for i = 1 : NoOfFeatures
   omega(:,i) = u(:,i)' * ImgMat(:,i);
end
end
```

A.8 Weight Calculation of Feature Space

This function will calculate weight of the feature space give the whole training set and the test set in consideration. And it return a matrix containing weight that is used as input for the KNN classifier.

```
function InImWeight = EthSLRPCAget(NormImage,u)
%Calculate weights in Transformed Feature space
InImWeight = zeros(1,size(u,1));
for i=1:size(u,1)
    %Calculate weight in Transformed basis
```

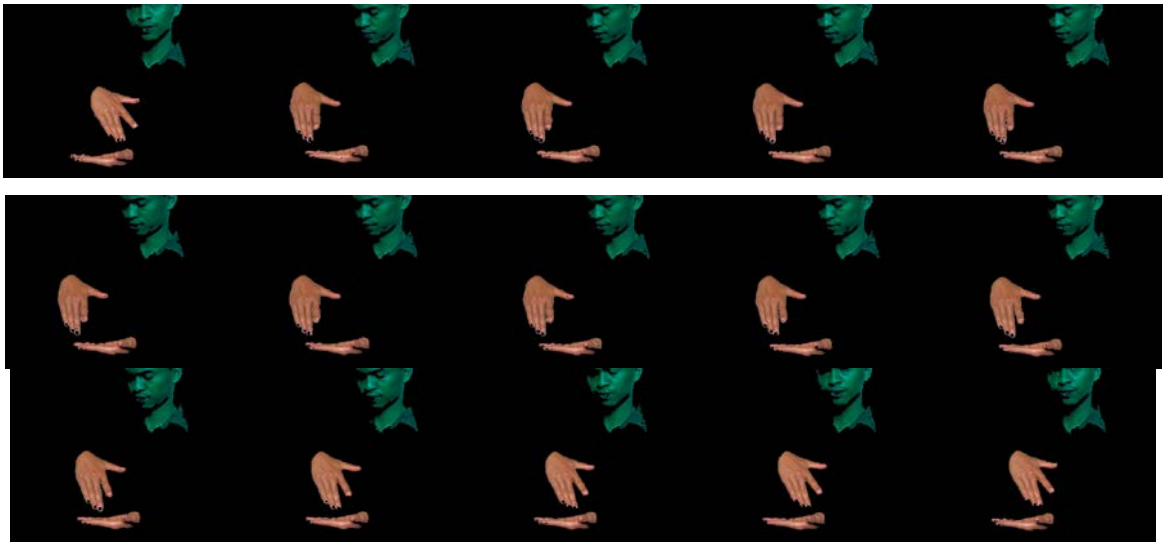
```
6       WeightOfInputImage = dot(u(i,:)',double(NormImage'));
7       %Store image weights
8       InImWeight(i) = WeightOfInputImage;
9   end
10  end
```

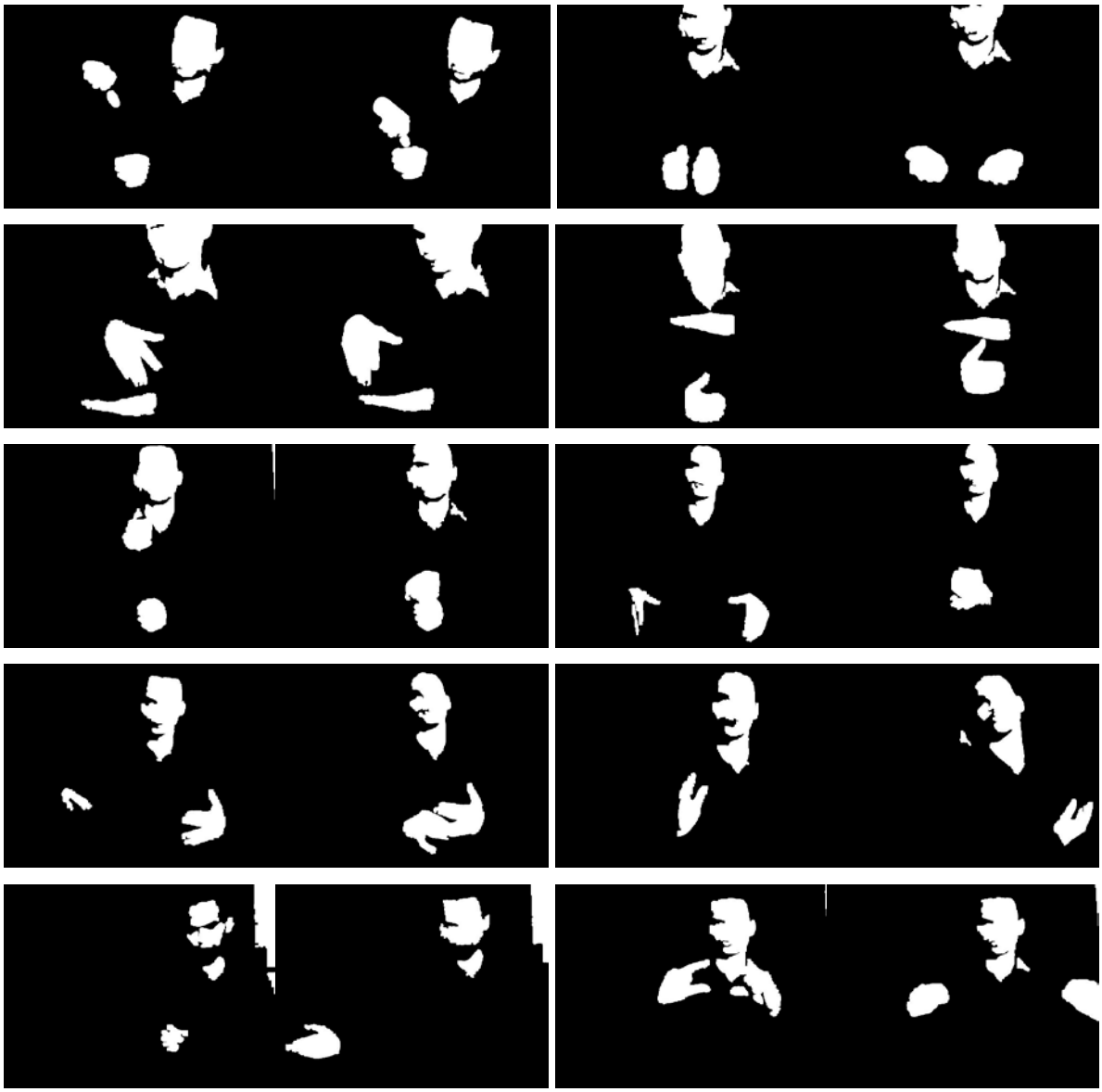
A.9 KNN Classification

This function calls the matlab's built in function for KNN classification, it accepts list of training words, number of training samples, the weight calculated between the training set and the current testing video and finally the weight of each original symbol in the training set. And its output is the class or group of the input video with respect to training set.

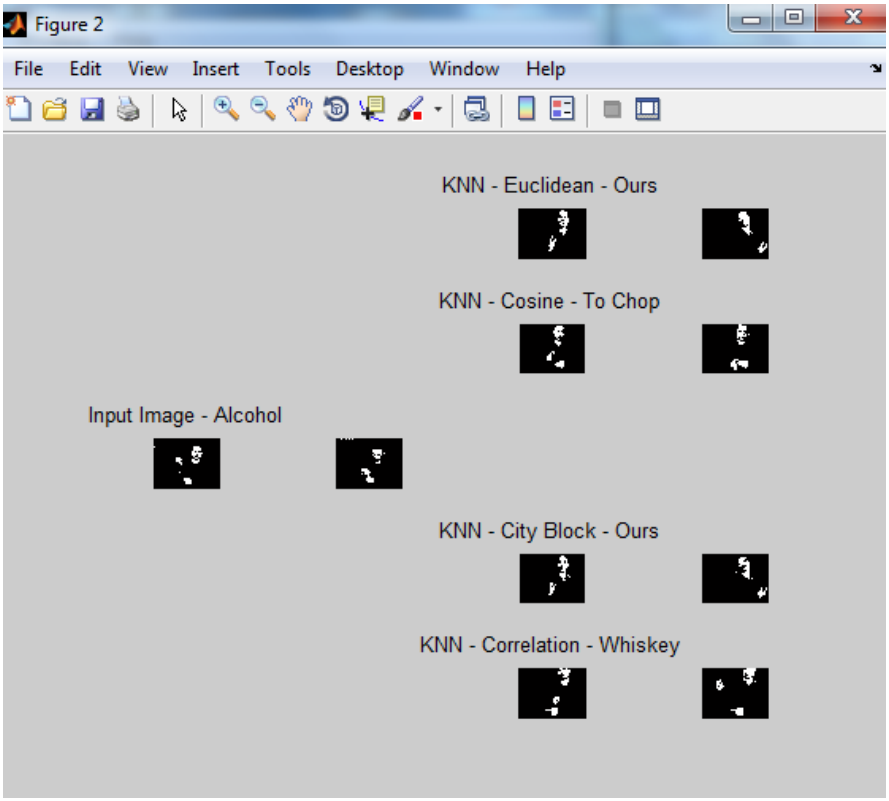
```
1   function Class = EthSLknn(cWord,nTrainingSamples,InImWeight,omega)
2   Training = omega';%store features in training matrix
3   Group = zeros(size(cWord,2)*nTrainingSamples,1); % Store group details
4   ll = 1;
5   for ii = 1:size(cWord,2)
6      for jj = 1:nTrainingSamples
7          Group(ll) = ii;
8          ll = ll + 1;
9      end
10  end
11
12  % KNN-Eucilidean
14  Class = knnclassify(InImWeight, Training, Group, ...
15          nTrainingSamples, 'cosine', 'random');
16  % Perform knn classification with Euclidean norm as basis
```

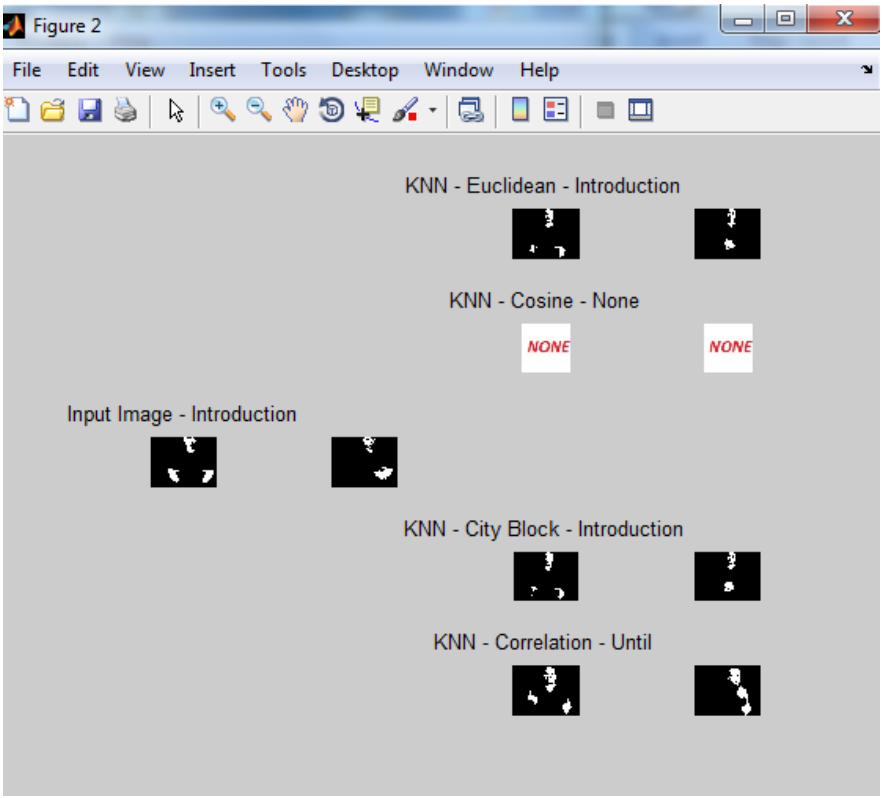
Appendix B : Sample Data Used for System Design





Appendix C : Sample Results of the Proposed Design





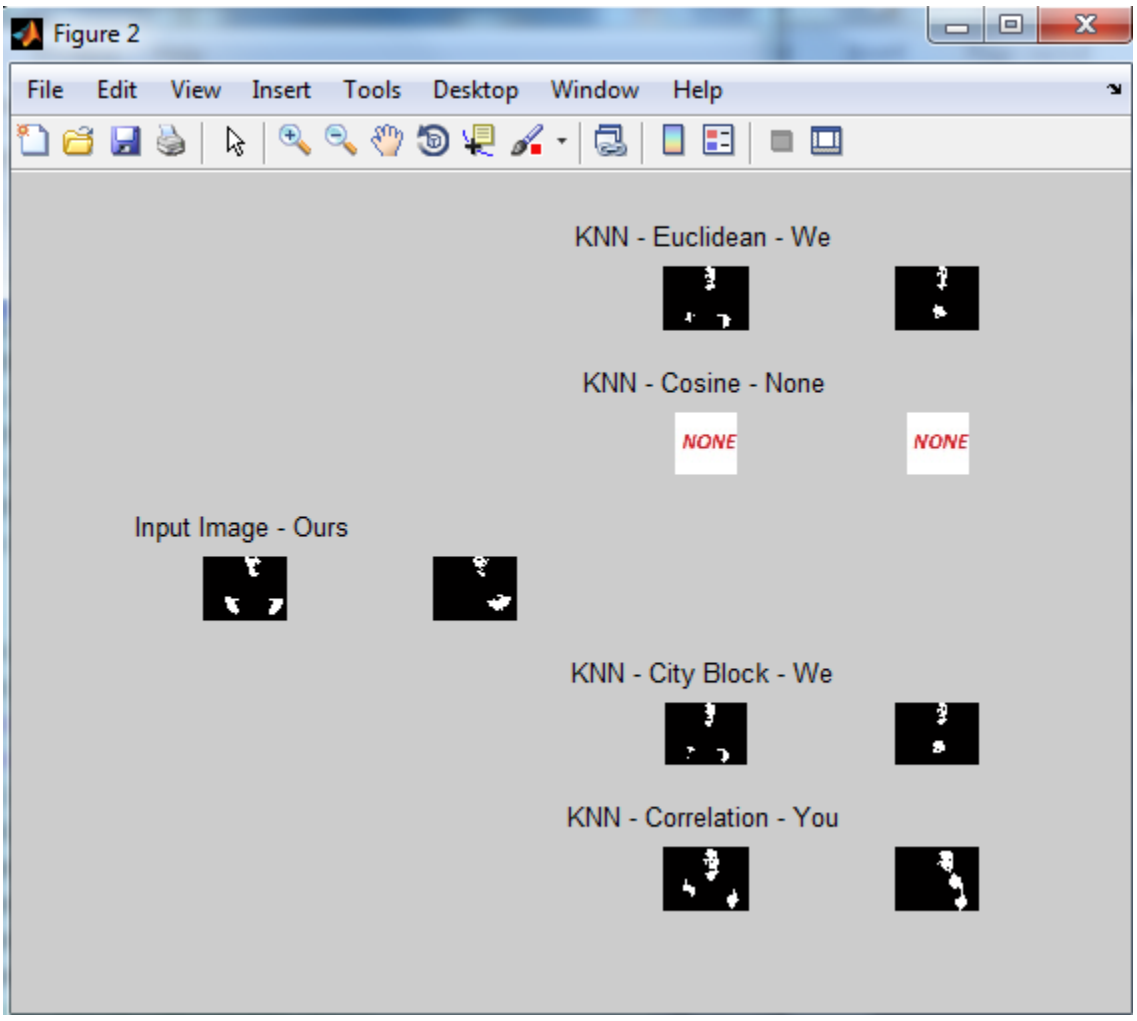


Figure 2
File Edit View Insert Tools Desktop Window Help
KNN - Euclidean - We
KNN - Cosine - None
NONE
NONE
Input Image - Ours
KNN - City Block - We
KNN - Correlation - You

Declaration

I, the undersigned, declare that this thesis work is my original work, has not been presented for a degree in this or any other universities, and all sources of materials used for this thesis work have been fully acknowledged.

Samuel Teshome Abebe

October 5, 2013

This thesis has been submitted for examination with my approval as a university advisor.

Million Meshesha (PhD)

October 5, 2013