ARABIC SIGN LANGUAGE RECOGNITION

BY

ALA ADDIN ISMAEIL SIDIG

A Dissertation Presented to the

DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

In

COMPUTER SCIENCE AND ENGINEERING

DEC 2017

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **ALA ADDIN ISMAEIL SIDIG** under the direction of his thesis adviser and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE AND ENGINEERING**.

Dissertation Committee

Mahmoud
Prof. Sabri A Mahmoud (Adviser)

Marwan
Dr. Marwan Abu-Amara (Member)

M. Mohandes
Dr. Mohamed Mohandes (Member)

Prof. Shokri Selim (Member)

Prof. Tarek Sheltami (Member)

عادل
Dr. Adel F. Ahmed
Department Chairman

Prof. Salam A. Zummo
Dean of Graduate Studies

20/3/18
Date

عمادة الدراسات العليا
DEANSHIP OF GRADUATE STUDIES

Dedication

This dissertation is dedicated to my family who has patiently waited for this moment from afar for five long years.

ACKNOWLEDGMENTS

All praises due to Allah who has permitted us and has given us the ability to complete this research work.

I am grateful to Professor Sabri A. Mahmoud for his advice and support during my PhD studies. He has been a motivating and encouraging figure, as well as a never ending source of knowledge. I thank him for his continuous reminding me to keep intention to serve the Ummah, and for being my second father in my immigration.

I want to thank the other members of my dissertation committee: Prof. Shokri Selim, Prof. Tarek Sheltami and Dr. Mohammed Mohandes, and Dr. Marwan Abu-Amara for their advice,help and valuable suggestions regarding this thesis. It is a privilege to have each of them serve in my examination committee. Special thanks to Dr. Abu-Amara for his detailed input and valuable comments. I am also grateful to KFUPM for providing support for all these years. And for supporting this research work through the project number IN151008.

Many thanks to my friends and colleagues at KFUPM. It was a pleasure to engage with them in fruitful discussions about various research topics. Special thanks to Hamza Lugman for his encouragement and support in publishing many papers together. Thanks to Mustafa Ghaleb, Baligh M. Al-Helali for their help in *collecting the database.*

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

THESIS ABSTRACT

NAME:	Ala addin Ismaeil Sidig
TITLE OF STUDY:	Arabic Sign Language Recognition
MAJOR FIELD:	Computer science and engineering
DATE OF DEGREE:	Dec 2017

Sign language is the major mean of communication for the deaf community. It uses body language and gestures such as hand shapes, lip patterns, and facial expressions to convey a message. Sign language is geographically specific as it differs from country to another. Arabic Sign language is used in all Arab countries. A sign language recognition system acts as a translator of these gestures into a form of spoken language such as text.

The availability of a comprehensive benchmarking database for Arabic sign language is one of the challenges of the automatic recognition and translation to a spoken language. This thesis introduces KArSL database for Arabic sign language consisting of 500 signs of numbers, letters, and words related to different domains such as health, religion, and common verbs. Signs in KArSL database *are performed by four professional signers and each sign is repeated fifty times by each signer. The database is recorded using state-of-art multi-modal Microsoft Kinect V2. This database will be made freely available for interested researchers.*

This thesis also propose different approaches for sign language recognition using this database and other databases. The proposed systems cover the stages of sign language recognition pipeline. Including segmentation of video into still key frames, hands trajectory processing, features representation techniques. Recognition accuracies of these systems indicate their suitability for such a large number of Arabic signs.

ABSTRACT ARABIC

المستخلص

درجة الدكتوراه في الفلسفة

الاسم	علاء الدين اسماعيل صديق صبر
عنوان الرسالة	التعرف الآلي على لغة الإشارة العربية
التخصص	علوم و هندسة الحاسوب
تأريخ التخرج	نوفمبر 2017

لغة الاشارة هي الوسيلة الرئيسية للتخاطب بين الصم و ضعيفي السمع . تستخدم فيها لغة الجسد مثل اشاراة الأيدي، حركة الشفاه، و تغيير تعابير الوجه. لغة الإشارة ليست موحدة عالمياً بل تختلف من بقعة لأخرى. لغة الإشارة العربية الموحدة تستخدم في جميع الدول العربية. التعرف الآلي على لغة الإشارة يعمل على ترجمة الإشارات الى شكل من أشكال اللغة المنطوقة كالكتابة مثلاً.

إحدى الصعاب التي تواجه الباحثين في مجال التعرف الآلي على لغة الإشارة العربية هي عدم توفر قاعدة بيانات كافية لتصميم مترجمات آلية يمكن الإعتماد عليها. هذة الرسالة تقدم قاعدة بيانات تحتوي على خمسمائة إشارة تشمل الأرفام، الحروف الهجائية، وكلمات من مختلف المجالات كالصحة و الدين و الأفعال اليومية و غيرها. تم تسجيل هذة الإشارات بواسطة أربعة أشخاص كل منهم قد كرر كل إشارة خمسين مرة على جلسات مختلفة. تم تسجيل الإشارات باستخدام كاميرا متفدمة متعددة الوسائط من إنتاج شركة مايكروسفت تسمى الكينكت. قاعدة البيانات هذه ستكون متوفرة مجاناً للباحثين في مجال التعرف الآلي على لغة الإشارة العربية.

هذة الرسالة تقدم طرقاً متنوعةً للتعرف الآلي على لغة الإشارة العربية باستخدام قاعدة البيانات آنفة الذكر. النمازج المقترحة تشمل المراحل المختلفة لبناء نظام للتعرف الآلي على لغة الإشارة. فتشمل تقطيع الفديو الى صور مفتاحية تختصر الفديو في عدد قليل من الصور، معالجة مسارارت حركة الأيدي عند الإشارة، و تقنيات

لإستخراج مميزات للإشارات عن بعضها. الدقة العالية لهذه الأنظمة تعكس مناسبتها للتعرف على هذا العدد الكبير من الإشارات العربية.

CHAPTER 1

INTRODUCTION

Communicating thoughts and feelings is an essential need for human beings. Hearing disabilities hinder the natural speech based communication. To communicate with each other and with speaking people, deaf has invented nonverbal languages that use descriptive gestures to convey their thoughts. These languages are developed by the deaf communities in different regions of the world.

Unfortunately, speaking people find it hard to learn these languages and many of the deaf population are not able to read and write spoken languages to use it for communication with hearing people. These problems increase the isolation of deaf people from the society. To communicate with deaf, speaking people need skilled professional translators that knows the spoken and signed languages. These skilled translators are few and can‘t be available all times.

Sign language recognition systems tries to fill this gap by exploiting the advanced technologies to automatically translate signed language to a form of spoken language such as text or speech. These systems are equivalent to speech-recognition systems used by speaking people. A sign language recognition system acquires the signs and converts them into other forms of language such as text or speech. Speech recognition automation has now advanced and became commercially available, while the automation of sign language recognition and especially Arabic Sign Language (ArSL) is still in its early stages of maturity. Also, there is no publicly available database ArSL, which makes it hard to compare different proposed recognition systems . The aim of this thesis is to contribute to the research in Arabic sign language recognition by providing a benchmark dataset and developing techniques to recognize signs.

1.1 Sign Language Recognition

Sign languages are full featured languages with their own vocabularies and grammar. They make use of hands-motion, fingers-configurations, facial-expressions, and body lane in parallel to express different terms. Sign language has thousands of words that form the language vocabulary and it uses a dedicated sign for each word. Facial expressions such as eye gaze direction, eye blinks, eyebrows, mouth, and tongue are used in sign language to express the emotions and feeling such as anger, happiness...etc [4]. For example, moving the hand opened and facing the ground down may have different meanings. If the signer is looking down, it means a child but, if the signer is looking front, it means relax. Fingers configurations are used to spell names or out of language vocabulary words. To effectively translate a signed language all its components need to be considered.

Sign language differs from one country to another and sometimes within the same country that has a unique spoken language. In the Arab countries, several localized sign language dialects exist like Saudi, Yemeni, Jordanian, Egyptian [5]. To unify them, an effort has been made by the Arab League Educational, Cultural and Scientific Organization (ALECSO) in 1999 to standardize the Arabic Sign language (ArSL) which resulted in a dictionary consisting of about 3200 words published in two parts [6].

1.1.1 Challenges in Sign Language Recognition

Sign language is a complete language with its own grammar and syntax. There are several challenges in the research of sign language recognition. Some of these challenges are imposed by the language characteristics while others are imposed by technologies' limitations, lack of resources, etc.

When signs are performed in a continuous sentence, hand needs to move from the end location of one sign to the start location of the next sign, this includes change of hand shape and orientations. This transient sign is not part of either of the signs which adds complexity to the segmentation of signs for recognition.

During the performance a sign, the hand may take different orientation with respect to the body of the signer. Hence, researchers can't assume fixed orientation for the hand. In addition, different motion patterns are used, some signs involves circular movement, while others involve local movement like wrist twisting. Hence, no fixed field of view can be assumed. Another challenge comes from the occlusion, where in some signs hands occlude each other or occlude the face.

Language grammar, some times, changes the sign. When the subject of a verb is pronoun, the gesture will vary depending on this pronoun, that by mixing the sign of the verb and the sign of the subject (I, You Him ...). This leads to different forms of one verb. Adverbs also change the sign, for example, to say ”run quickly” the speed of hand when performing the sign will be faster. Non-manual signs are important parts of the sign language. These includes facial expressions and body poses, which are used to modify the meaning of signs. Eye , eyebrows and mouth are used for example to change the verb into question or to show wonder or surprise.

All these mentioned challenges adds complexities to the recognition of sign languages.

1.1.2 Online vs Offline recognition

There are two categories of sign recognition, viz: online and offline recognition. These categories are based on the nature of the data processing. In online recognition, signs are collected and recognized in real time, i.e. at the same time it is produced. Online recognition provides real time response. There are few works in the literature that addressed it, Shanableh's work in [3] is an example.

In off-line recognition signs are performed and video of the signs is recorded and then a computer vision algorithm is applied to it later, or data is collected by sensors and stored for later processing. Off-line recognition make it easier to experiment and test many algorithms.

1.1.3 Levels of Sign Language Recognition

Alphabet signs are used to spell names letter by letter. Systems used to recognize alphabets are relatively simple and limited in vocabulary size, a lot of work is done in this area and many approaches are tried, Mohandes in [7] listed about 12 of them.

Isolated words sign recognition is more practical than alphabet but more complex as it deals with sequence of images. This additional time component is important in the analysis of this sequence. Signer performs signs in isolation starting and ending in static position. The vocabulary size can be very large. For Arabic sign language, more investigation is needed to determine the practical size of vocabulary.

Continuous sign language recognition systems are more practical than Alphabets and isolated word recognition systems. The main challenge here is the determination of the words boundaries. If this was successfully automated, then the isolated words recognition systems can be used to recognize each word of the sentence. This requires identification and removal of transient signs. Different approaches are used to locate the transient signs. For example, an assumption is made on the hand movement acceleration to be high in the transient sign than in word sign. Many other approaches are listed in [8]. Research in continuous Arabic sign recognition is still limited compared to alphabet and isolated words.

Although continuous sign recognition is more desirable in practical application, it depends on the isolated word recognition system for recognition. This increases the importance of accurate isolated word sign recognition.

1.1.4 Signer independence

The ideal sign recognition system would give high recognition accuracy for new signers that it has not seen during training. Different signers may have different signing styles. Similar to speech, different person may pronounce the same word differently or may speak in different speeds, signs can be performed differently by signers. To design a signer independent recognition system, the number of signers in the training database should be large enough to capture these differences. Some works in signer independent Arabic sign language recognition is present in the literature [9, 10, 11, 12], the highest number of signers used for training and testing, to our knowledge, is 18 in AL-Rousan in [9].

1.2 Motivation

The deaf population is increasing and many of them can't read and write spoken languages, and few non deaf people learn sign language. These problems lead to isolation of deaf people from the society. Thus building a sign language recognition system helps in addressing this need.

The advancement in automation of speech recognition make it now mature enough to be adopted by different applications to serve the speaking people. In contrast, these applications can't be used by deaf due to weakness in automation of sign language recognition. Currently, sign language translation is expensive because it requires a professional human translator.

In addition, no publicly available data set for Arabic sign language which delays the advancement in building recognition systems. Due to the absence of mature recognition systems, most public services are not translated into sign [13].

Considering the above mentioned facts, there remains a lot to do in the recognition of Arabic sign language. The aim of this dissertation is to contribute in elimination of these limitations , improve the state of the art in the Arabic sign language recognition, and build a database for Arabic sign recognition.

1.3 Model of Sign Language Recognition System

Generally sign language recognition and translation system follows the pipeline pictured in figure 1.1. This thesis implements the sign recognition part while the other translation part is being addressed by my colleague Hamzah Luqman in his PhD thesis work. Here, we introduce each stage briefly.

1.3.1 Sign Acquisition

The first stage is signs acquisition. Signs acquisition techniques can be broadly classified into two categories: vision based and sensor based. In vision based, cameras are used to capture the sign while in sensor based other sensors are used to acquire the features that represent the sign. The sensor-based approach employs a glove of electronic sensors to be worn by the signer. The sensors track and detect hands and fingers‘ motion. The drawback of this approach is that the signer may be uncomfortable and can’t naturally perform the sign as he does without wearing it.

In vision based acquisition typically cameras are used to capture the signs and store them as sequence of images for processing. Most of the reviewed works uses single video camera. Some researchers has employed more than one camera to capture different views of the sign. This is helps in dealing with difficulties imposed by the environment, but also adds computational overhead to process the stereo images.

In this thesis, we use an acquisition device that combine the best features of both vision and sensor based while freeing the signer from wearing any instrument. We use Kinect-2 for recording signs which provides color images, depth images and skeletal joints‘ locations of the signer body. We developed a recording tool that synchronously records the color, depth and skeletal joints locations. A sample frame from each channel is shown in figure 1.2.

1.3.2 Preprocessing

The raw acquired data may need some preprocessing before being suitable for the recognition system. Depending on the recognition system, this stage may be used to segment the video of the sign into a sequence of still images, locate the position of some key components of the sign such as the hands and head, or smooth the

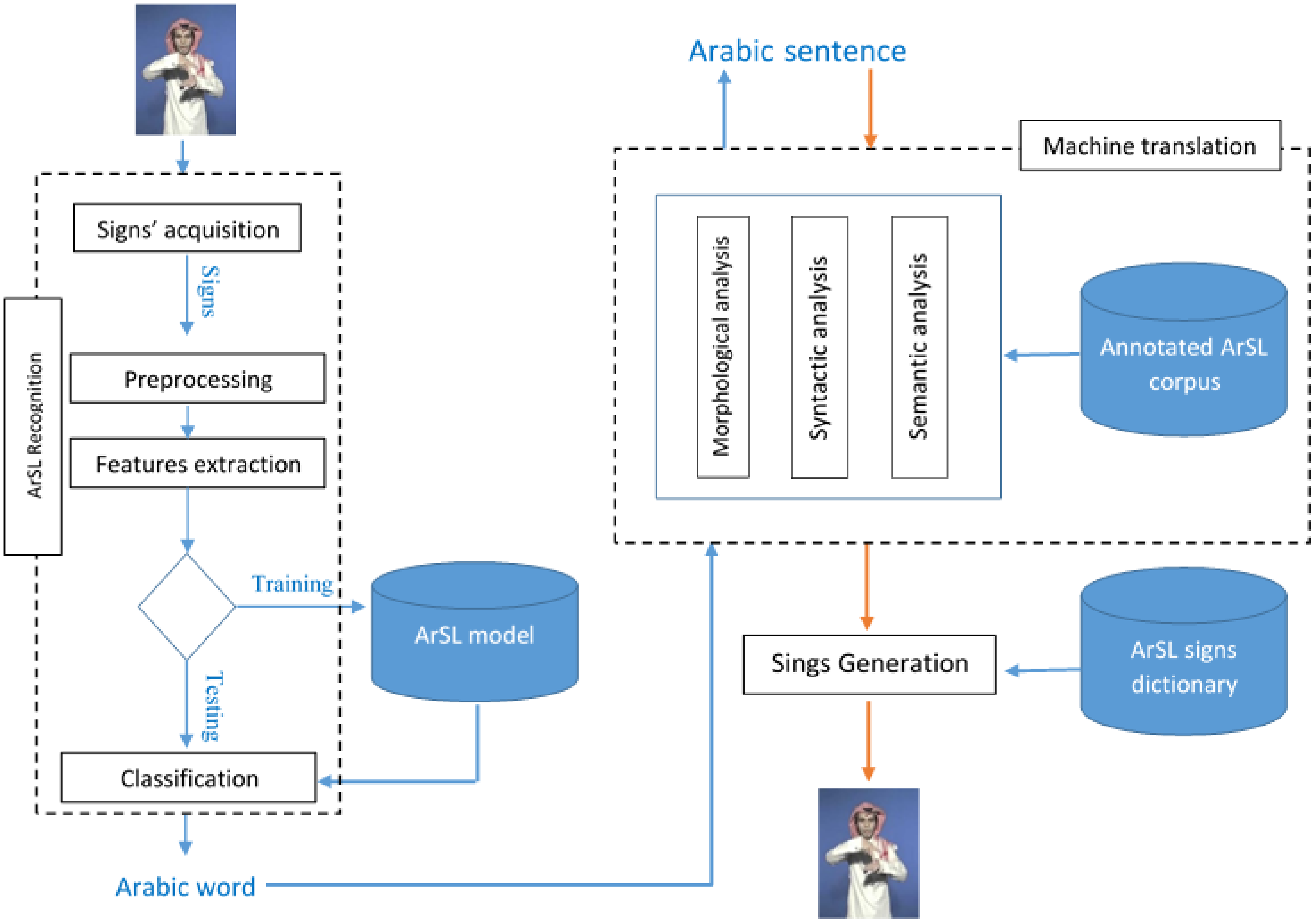


Figure 1.1: General pipeline for sign language recognition and translation system

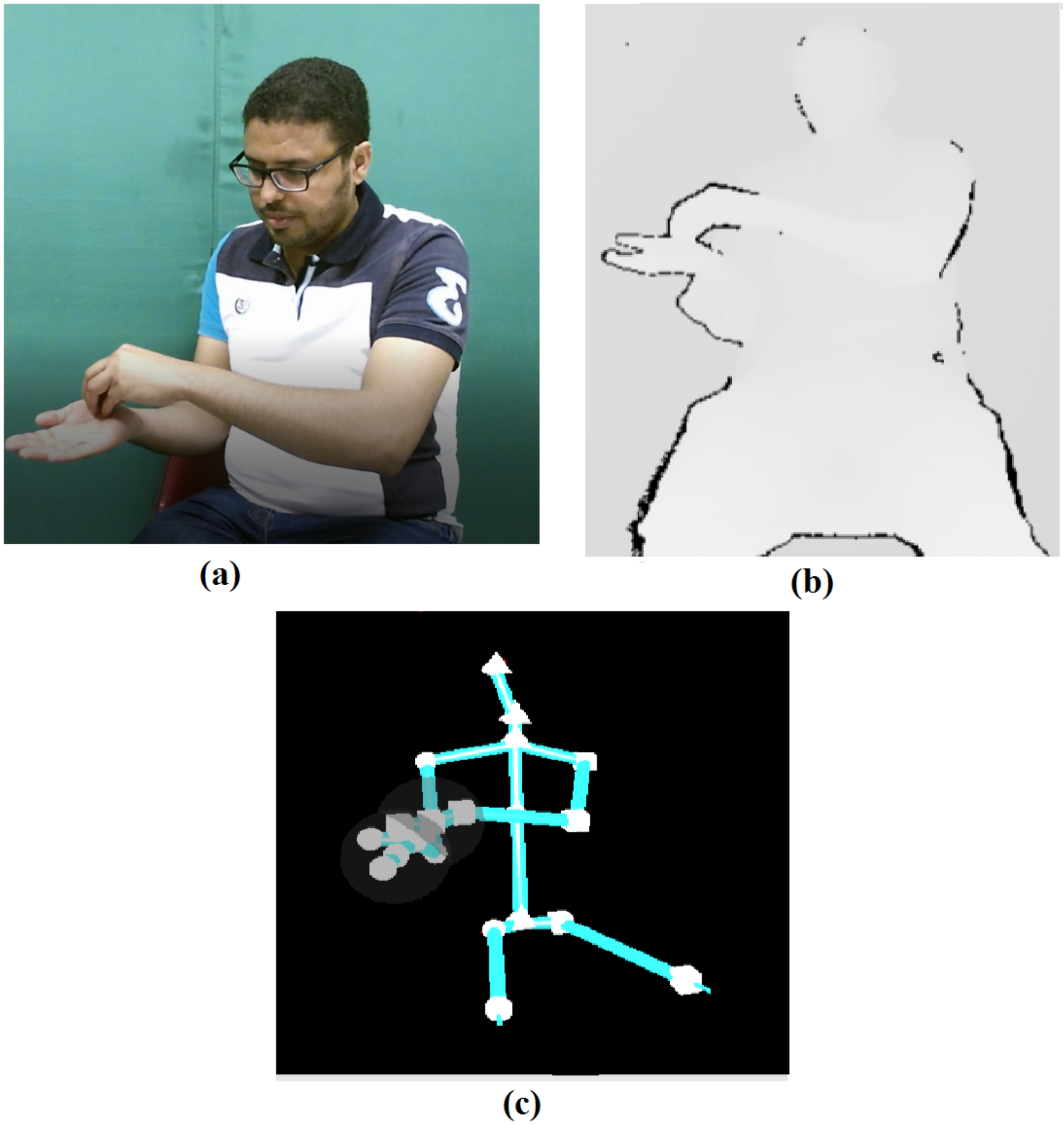


Figure 1.2: The recorded output of Kinect (a) Color image, (b) Depth image (c) Skeleton joints.

trajectory of hands motion during the sign.

Videos are usually recorded at 30 frames per second which results in many images if all frames are extracted. To segment the video into still images, some redundant images may be removed and only *key frames* should be extracted. The localization of key frames is a research problem that needs some techniques to solve. Similarly, the raw trajectory of hands‘ motion may include some wiggles that needs to be smoothed and filtered out. In this thesis, we propose an algorithm for smoothing the trajectories and locate the key frames of video as will be discussed in section 4.2. Some other recognition systems may require the localization of some parts of the signer‘s body while signing. This requirement is mostly needed by vision based systems. Different image processing techniques are usually used in the preprocessing stage to locate the required parts in the image. In this thesis, we propose an algorithm to locate the hands using optical flow as will be discussed in section 6.4. Figure 1.3 shows a sample preprocessing that locates the hands of the signer.

1.3.3 Features extraction and Classification

After preprocessing, the data is ready for features extraction. For vision based recognition, features are extracted from images or regions of the images. The features extraction methods can be classified into global and local features. In global features, the features vector represents the whole image, while in local feature, it represents patches of the images. Different global and local features

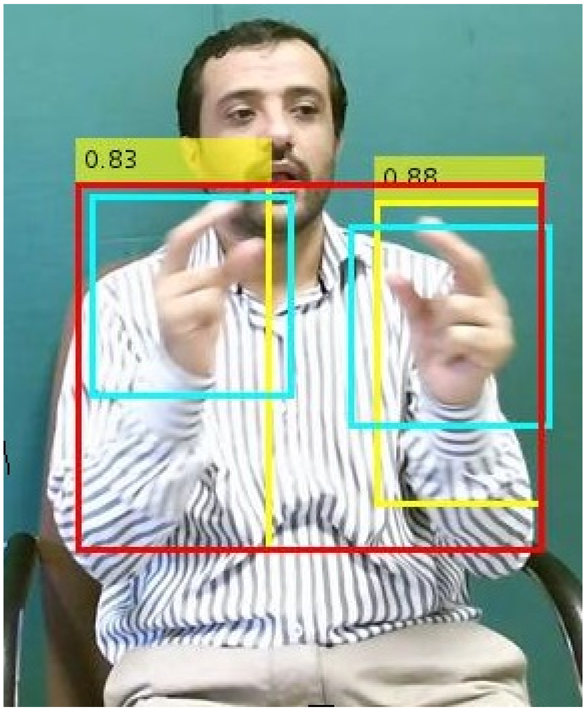


Figure 1.3: Sample preprocessing that locates the moving parts of the signer like the hands and head.

are used in this thesis as will be discussed in chapter 5. As an example the Harris local features extraction technique is pictured in figure 1.4. We also propose some features representation technique for the hands motion trajectories as will be discussed in section 4.3.

The extracted features are used to train a classifier when building a recognizer and test it to evaluate the accuracy of the recognizer. Different classification schemes are used in this thesis, including single stage , multistage, and ensemble of classifiers.

In *deep learning*, a deep convolutional neural network is used for both features extraction and classification. In this thesis we use deep learning for sign language recognition as will be discussed in section 7.1.

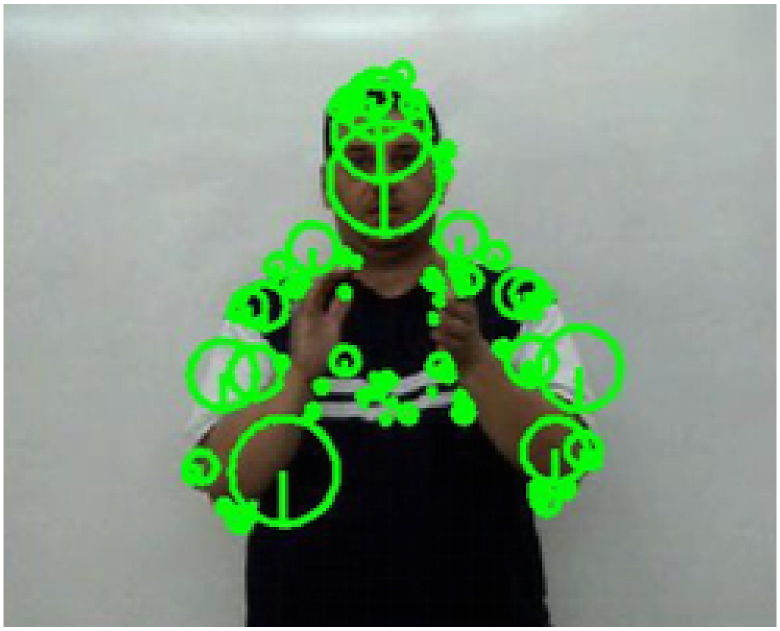


Figure 1.4: Example of a features extraction technique used in this thesis. The Harris features detector is used for extracting features around the signer‘s body.

The output of the classification stage is a sort of spoken language word. This can be vocal output or text. In this thesis, the output is a text. As shown in figure 1.5 the word ”thalathon” which means 30 is shown on the top of an image that shows the hand configuration for this sign.

1.4 Outcomes of this Thesis

The following summarizes the outcomes of this thesis.

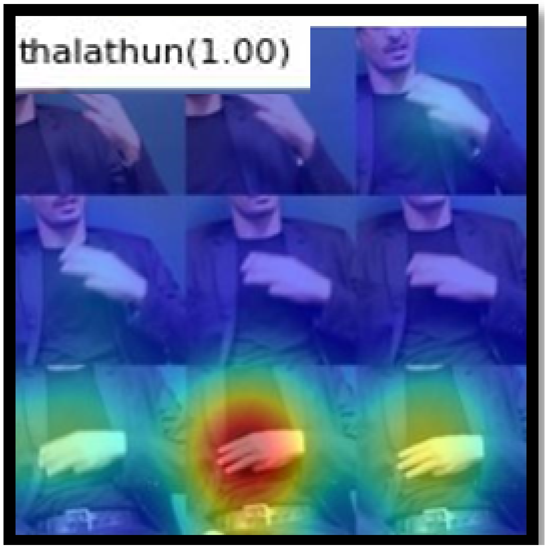


Figure 1.5: Sample output of the recognition system as text. The sign name 'thalathon' shown on the image.

1.4.1 KArSL database in cooperation with my colleague Hamza

- A recording tool for synchronized recording of color, depth and skeleton joints locations of the signer using Microsoft Kinect for Windows V2.
- Recording a database of 500 signs that includes alphabets, numbers, and words by four signers with 50 repetitions for each sign by each signer.

1.4.2 Algorithms for preprocessing

- An algorithm for hands trajectory noise removal and compression.
- An algorithm for video segmentation into sequence of still key frames. The algorithm also works if more than one sample are repeated in the same video.

- An algorithm for determining the key frames and locating hands of the signer in a video. The algorithm finds bounding boxes around the hands.
- An algorithm for converting the sign video into a single grid image of key frames.

1.4.3 Features Representations and Classification

- Two features representations of the hands motion trajectories.
- A feature representation that captures both the hand shape and motion based on optical flow and histogram of oriented gradients.
- Transformation of sign language recognition task into image captioning task.
- Employment and tuning of single-stage, multi-stage, ensemble of classifiers.

In summary, it can be seen that this thesis provides valuable contributions to the research in Arabic sign language recognition in all stages of the recognition pipeline.

1.5 Organization of this Thesis

This thesis is organized as follows. In chapter 2 the related works in the literature are described. In chapter 3 we describe the collected database, it‘s formats, organization, file naming, and statistics. Chapter 4 describes our proposed sensor based recognition system. Followed by vision based recognition systems described in chapter 5,6, and 7. Finally chapter 8 concludes this thesis.

CHAPTER 2

LITERATURE REVIEW

In this chapter, previous work on sign language recognition is presented. Focusing on but not limited to Arabic sign language recognition. With the general model for sign language recognition in mind, this chapter presents the techniques used in each stage of the recognition pipeline (see figure 1.1).

2.1 introduction

Deaf and hard hearing people form 17 million in the Arab world and 70 million worldwide[1]. In Arabic countries, there are several sign languages like Saudi, Yemeni, Jordanian, Egyptian ...etc. An effort has been made in 1999 to standardize the ArSL which is initiated by the League of Arab States (LAS) and the Arab League Educational, Cultural and Scientific Organization (ALECSO). This effort resulted in a dictionary of 1600 words published in two parts [14][15]. Automatic recognition of sign language systems are important for addressing the difficulties

[1] Deaf World Federation, http://www.wfdeaf.org/

faced by persons with speaking disabilities. These systems are the equivalents to speech-recognition systems used by people. A sign language recognition system acquires the signs and converts them into other forms such as text or speech. Sign language uses combination of hand shapes, orientation, and movement of head, arms, body poses, facial expressions, and lib patterns to convey meaning. Each of these components have special importance in modifying the meaning of signs. For example, moving the hand opened and facing the ground down may have different meanings. If the signer is looking down, it means a child, but, if the signer is looking front, it means relax. To successfully translate a sign, all these components need to be considered. Sign languages are not international and are not completely dependent on the spoken language of the region they coexist in. The research in sign language recognition has started around thirty years ago in Japanese [16, 17], American [18], Chinese [19], German [20], Korean [21], and others. Sign language consists of three main components:

1. Finger Spelling: used to spell words for telling names [22, 23, 24]

2. Word Signs: signs that correspond to the vocabulary of a language [25, 26, 27]

3. Non-manual Expressions: like eye blinks, eye gaze direction, eyebrows, nose, tongue, mouth, and body position.

The deaf population is increasing, many of them can't read and write spoken languages and few non deaf people are provisional sign language users. This results in isolation of deaf people from the society. Thus building a sign language recognition system helps in addressing this need.

ArSL is a full natural language that has its structure, word-order, and lexicon that is different from Arabic text. This imposes another challenge, in the translation between the ArSL and Arabic text. This problem is a complete machine translation problem since ArSL and Arabic languages are two different languages with different structures and grammars. To our knowledge, there is no comprehensive translation system that can recognize ArSL and translate it into meaningful Arabic sentences and vice versa. Most surveyed techniques for ArSL target either sign recognition or machine translation but not both. In addition, most of the available Arabic translation systems for ArSL work at the word level. In this type of translation, a direct mapping is made between the Arabic word and the equivalent Arabic sign. This type of translation is not sufficient for translation between languages since it ignores the structure and grammar of both languages which results in losing the meaning of the sentence. Figure 1.1 shows a proposed architecture of ArSL recognition and translation system. As shown in the figure, the ArSL recognition system identifies the equivalent Arabic word of each acquired sign. The recognition process starts by preprocessing the acquired signs to make them convenient for feature extraction. In the training stage, the extracted features are used in building the ArSL models that will be used for classifying the new acquired signs. The recognized signs are mapped to their equivalent Arabic words.

In section 2.2 we explore Arabic sign language data sets. Then in section 2.3 we review the techniques used for acquiring signs performed by signers, then in section 2.4 we address the techniques that are used to extract features from these signs for classification. Different classification schemes are addressed in section 2.5. The deep learning approaches for sign language recognition are reviewed in section 2.6. Finally, post processing techniques are shown in section 2.7.

2.2 Arabic Sign Language Data Sets

To our knowledge there is no comprehensive benchmarking database for Arabic sign language recognition that is freely available to researchers. Thus, different researchers use different data sets, and consequently the recognition accuracies of the different techniques may not be comparable. Although Arabic alphabet consists of only 28 letters, Arabic sign language uses 39 signs for alphabet. The 11 extra signs are combining two letters which are commonly used together. Therefore, most published work on ArSLR uses these 39 signs. The unified Arabic Sign Language Dictionary listed around 1600 isolated signs. However, most of the work in the literature examined below 50 signs as most of these works exclude most similar signs to ease the recognition task. Limited number of signs is acceptable for building domain specific applications. However, it is not adequate for general purpose applications.

Most researchers in the literature used datasets with low number of signs [28] [9] [2] [29] [30]. Shanableh et al in [1] built an ArSL database for 23 selected signs, each sign was repeated 50 times by 3 different signers, then the videos are segmented to keep each sign in an individual video. They reported that their setup doesn't restrict the background or the signer clothes. The database can be obtained by contacting the author. This database is extended by Sidig et al [31] by adding one more signer. This databases is used for some experiments here. Assaleh collected a continuous signs database of 40 sentences. Each sentence is repeated 19 times by one signer [32]. Alfonse et al [33] presents a database consisting of 1216 signs. The signs are captured using ordinary HD camera, Microsoft Kinect 2, and Leap motion tracking sensors. The database is performed by four signers in an unrestricted environment. However, this database can not be used for recognition systems using machine learning as each sign is repeated only thee times by two signers. In addition, the database is not available online. SignsWorld Atlas is another database for ArSL presented by Shohieb et al [34]. The database is captured using regular video camera. They reported a total of 500 manual and non manual sign elements were performed by different number of signers that ranges from two (for hand shapes in isolation and in single signs) to ten (for digits). Absence of signs repetition is a limitation of this database.

In contrast to ArSL, various databases are available for non-Arabic sign languages in different countries. RWTH-BOSTON-50, RWTH-BOSTON-104, and RWTH-BOSTON-400 are American Sign Language (ASL) video-based databases created by Boston University [35, 36, 37]. RWTH-BOSTON-400 is the largest and consists of 843 sentences performed by five signers. SIGNUM is a German

Sign Language database available online at a price of 1000 ER [38]. It consists of 780 sentences (450 signs) performed by 25 signers. ECHO is a sign language database available in three sign languages: Swedish (SSL), Dutch (NGT), and British (BSL) [39]. It consists of five stories and some interviews with signers.

2.3 Signs Acquisition

Sign acquisition techniques can be broadly classified into two categories: vision based and sensors based. In vision based, cameras are used to capture the signs while in sensors based other sensors are used to acquire the features that represent the signs.

2.3.1 Sensor Based

In the sensor-based approach, signers wear electronic sensor gloves like that in figure 2.1. The sensors detect the fingers' configurations and track the hands' and motion. The drawback of this approach is that the signer is not used to wear them naturally [40]. These gloves provide information on the position, rotation, movement, orientation of the hand, and more importantly, finger bending.

Ritchings et al. developed a computer-based system using Data Gloves for teaching sign language [41]. Bend sensors and push button switches were utilized to observe 17 measures. Two bend sensors are joined to every finger and thumb junctions to monitor the finger movements. Two bend sensors on the inner and on the outer side of the hand wrist to catch it's up-down movement. In addition, a bend sensor on the right side of the wrist to sense its yaw. The bend sensors reports different measures when the bent is in one direction, which is suitable for the fingers. Two sensors are mounted on opposite sides of the wrist to monitor pitch up/down movements. Lastly, to identify finger abduction and adduction, 4 push-down switches are attached to the side of every finger beginning with the thumb. The goal of the system was to evaluate the capability of learners in reproducing signs done by a professional signer. The used database covered 65 signs performed by four professional signers. All the signs were selected to reflect all the activities that could be done by a learner within the bounds of the 17 sensor configuration. The learners were able to reproduce the signs with 93% accuracy. They also reported that their system faced difficulties in recognizing some signs made by a trainee with small hand.

Mohandes and Deriche [42] used the Dempster-Shafer (DS) Theory of Evidence to combine decisions from the Cyber Glove with 22 sensors and the hand tracking system. Each glove provides 22 signals and the tracking device provides 6 signals for each hand. The authors reported that the fusion of classifiers decision outperformed the feature-based fusion. They tested using the glove-based and the electromagnetic-based tracking systems independently. The tracking system achieved a reported accuracy of 84.7% while the glove-based system achieved a reported accuracy of 91.3%. The authors claimed that the traditional feature-based combination provided an accuracy of 96.2% which was outperformed when using decision level fusion to reach an accuracy of 98.1%. They built a database of 100 signs performed by a volunteer from the deaf community.

Figure 2.1: Power Glove hand of sensors.

2.3.2 Vision Based

In vision based acquisition typically cameras are used to capture the signs and store them as sequence of images for processing. Most of the reviewed works use single video camera. Multi-camera systems may help in dealing with environmental difficulties. However, 3-D models add more computational overhead.

Tolba et al in [43] used 2 synchronized cameras positioned horizontally aligned with 90 degrees view angle difference. Signer hands should be located in a specific area with equal perpendicular distance from both cameras otherwise the sign would not be captured. The cameras capture 19 consecutive pair of images each pair is 5 degrees rotated from the previous pair. The ten odd ordered pairs are chosen for training and the other 9 for testing. These images are fed to Pulse-coupled Neural Networks (PCNN) to extract 2D signatures which are then weighted and combined linearly to produce 3D features.

Other devices exist that use more than one camera, depth cameras and Infra-

Red cameras. The Leap Motion Controller (LMC) [44] employs two infrared cameras and three infrared Light Emitting Diodes (LEDs) to capture information within its operating range. The device detects and tracks the hands' motion, fingers and finger-like objects positions, gestures, and motion. Figure 2.2 shows a schematic view of the LMC. LCM have been used by researchers to recognize Arabic signs in [45, 46] for alphabets recognition and in [47] for words recognition. In [46], two LCMs are used to capture the hand to avoid the occlusion of fingers.

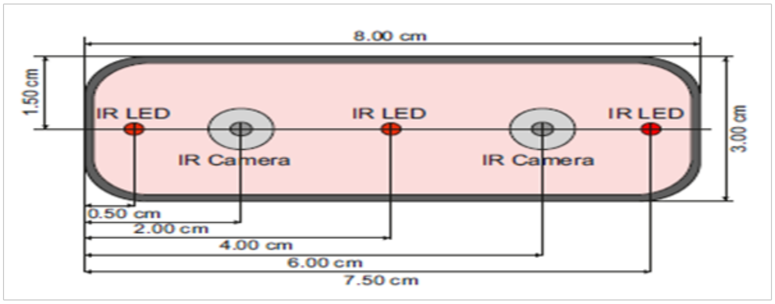


Figure 2.2: Schematic diagram of the Leap Motion Controller (LMC)

Microsoft's Kinect consists of depth sensor, color camera, and multi-array microphone running proprietary software. The Kinect sensor traces full-body motion and distinguishes speakers. Kinect sensor has four kinds of output: colour image, depth image, infrared image, and skeleton joints. The depth image is produced by CMOS IR sensor. This image can be used to create a mask image to remove the background by simple thresholding. Kinect can track skeletons with 25 joints (see Figure 2.4). Kinect has been used for recording signs by [48, 49, 31].

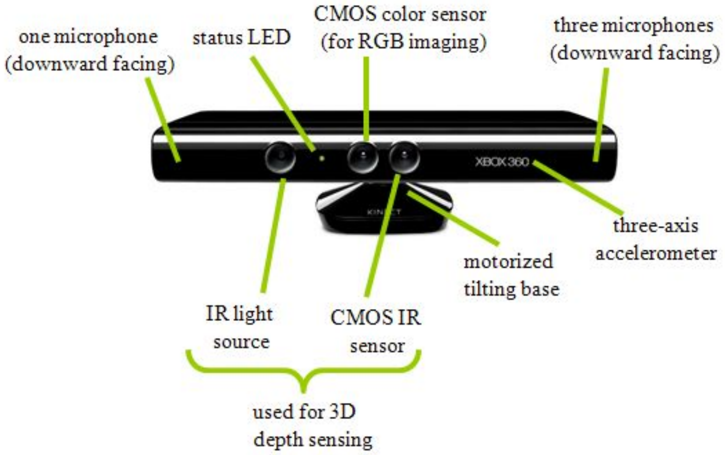
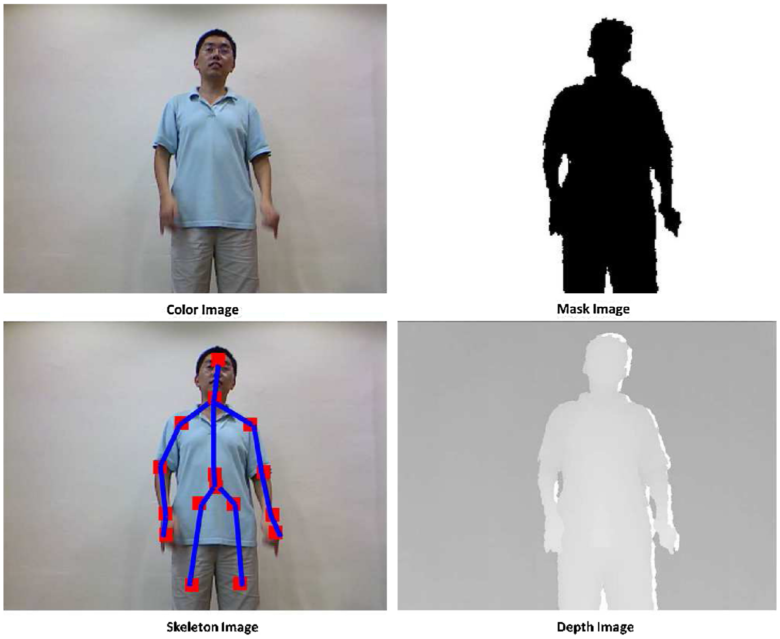


Figure 2.3: Schematic diagram of the Kinect


Color Image

Mask Image

Skeleton Image

Depth Image

Figure 2.4: Outputs from Kinect [2]

2.4 Features Extraction

After sign acquisition, features are extracted for use in training and testing the system. In this section a review of the used features and their extraction techniques is presented.

Mohandes et al addressed isolated words using skin color model in chromatic color space to detect the face [28]. *Region growing* is used to search for orange and yellow gloves (worn by signer to ease the segmentation) to locate hands. These are then used to collect the following features: Hands' Centroids with respect to face centroid, eccentericity of bounding ellipse of hands, angle of the first principal components, and the area of hands. They extended the work in [50] for bigger data set of 300 signs with 15 samples per sign. AL-Rousan et al in [9] converted images from a segmented video into frequency domain using Discrete Cosine Transform (DCT) and used zigzag zonal coding for descriptors encoding. For background removal, the DCT of consecutive images are subtracted. Then 50 DCT descriptors of these differences are used as features.

Shanableh et al in [3] built a database of 23 selected signs, each sign was repeated 50 times by 3 signers, then the videos are segmented to keep each sign in an individual video. They reported that their setup doesn't restrict the background or the signer cloth. They extracted temporal and spacial features. For temporal features the motion is accumulated into one image that describes the activity during the whole sign. This is done using forward accumulated difference of successive frames as follows[3]: Let $I_{g,i}^{j}$ denotes image index j of the i^{th} repetition of sign g the forward prediction is:

$$P_{g,i} = \sum_{j=1}^{n-1} \delta(I_{g,i}^{j} - I_{g,i}^{j+1}) \tag{2.1}$$

where n is the total number of images and δ is a binary threshold empirically determined as the mean of non zero difference pixels. Then the resulting image is filtered by median filter which removes the isolated prediction errors as they are assumed to be noise. However these errors can be due to facial expressions but this was out of their scope. They proposed two approaches for spacial features extraction: the first uses 2D DCT transformation and zonal coding, while the second applied Radon transform and then low pass filtering. In the first method they applied 2D DCT transformation on the accumulated temporal differences image, the Zigzag zonal coding is used and 50 descriptors are used as features. In the second method, the temporal difference image is projected at a given angle using Radon transform and then represented by 50 descriptors of 1D DCT and then low-pass filtered. The 50 DCT cut-off was determined empirically by examining different cut-off points. In the goal of signer independence they argued in [12] that different signs can have very similar accumulated difference. They proposed a weighted directional accumulated difference by assigning higher weights to the first half of the temporal sequence differences. They also accumulated two difference images, positive and negative (AD_+ and AD_-) computed as given by equations (2.2) and (2.3) , respectively.

A bounding box that encapsulates movement information in both ADs is determined as a region of interest. This region is then transferred to the frequency domain using DCT and the remaining parts are removed from the AD image. After zonal coding the resultant DCT coefficients of the AD images are interleaved

$$AD_{+}(x,y) = \begin{cases} AD_{+} + w_K & if(f(x,y,t_k) - f(x,y,t_{k-1})) \geq Th_{(k,k-1)} \\ AD_{+} & \text{otherwise .} \end{cases} \tag{2.2}$$

$$AD_{-}(x,y) = \begin{cases} AD_{-} + w_K & if(f(x,y,t_k) - f(x,y,t_{k-1})) \leq -Th_{(k,k-1)} \\ AD_{-} & \text{otherwise .} \end{cases} \tag{2.3}$$

where (x,y) are the pixel coordinates of the AD image and w_k is the accumulation weight at the k$^{\text{th}}$ image difference

to form feature vector. In this work they used two colored gloves to ease the segmentation of hands. This work has been extended by Sidig et. al by employing different transforms to the accumulated difference image. They tested Fourier Transform , Log-Gabor Transform, and Hartley Transform applied on the whole and slices of the accumulated image [51]. One limitation of this method is that it requires the sign video to be segmented in key frames which was done manually.

Zaki et al proposed combination of three vision based features [30]: principal component analysis (PCA), kurtosis position, and motion chain code (MCC). PCA is used as a representation of the hand shape and orientation. To find the place of articulation, Kurtosis position is used. MCC is used to represent the hand movement. Skin color detection is used to locate the head and hands, followed by connected components labelling to detect and track the dominant hand. PCA is then applied on the detected hand. Position of the highest kurtosis in the hand image is used as representation of the hand articulation. MCC is a sequence of numbers from 1 to 4 and 0, to encode the hand movement direction (1: up, 2: left, 3: down, 4: right, and 0: no motion). The chain code is built by subtracting the centres of the hand in 2 neighbouring frames. They chose 30 words from the American Sign Language database RWTHBOSTON-50 excluding signs with occlusion of hand and face.

Chao Sun in [2] proposed discriminative exemplar coding for sign language recognition using Kinect. They proposed two types of features: Histogram of Gradients (HOG) features to describe the appearance, and Kinect features that describe position, shape and motion of hands. For HOG features the image is resized to 256x128 and then divided into 8x8 pixels cells, from which they compute a 9 bin histogram of gradients orientations. This histogram is then normalized by gradient energy in the neighbourhood around this cell. Kinect body pose features are the combination of three parts.

1. 3D vectors: from elbows to the shoulders, from the wrists to the elbows, from the hands to the wrists, and from the left hand to the right hand.

2. The angles at the shoulders, elbows, and wrists joints.

3. The distance between the right and the left hands, normalized by double the shoulder width.

The hand shape feature is generated by computing HOG on 48x48 patches around the hand position. For generating the hand motion feature optical flow is calculated on two patches from two successive images. Inclusion of temporal information in features leads to improvement in the recognition rate as reported by Sun in [2].

Tharwat et al in [52] used SIFT for extracting local features from gray level images of Arabic sign language alphabets. After features are extracted they applied Linear Discriminant Analysis (LDA) to reduce the dimensionality of feature vector. Each image is represented by one feature vector. The described method can't handle dynamic signs which are present in some alphabets. Ahmed and Aly in [53] proposed using Local Binary Pattern (LBP) to capture the appearance of hands and head in a sign frame. Before applying LBP skin colour detection is used to detect hands and head regions. Then frames are cropped to the boundary of detected head and hands. In LBP each pixel gray level is compared to its 8 neighbours, if a pixel of the neighbourhood is greater than or equal to the current pixel (center pixel), one is assigned to that pixel, otherwise 0 is assigned. After this process each pixel will have a code of 8-bits, and these codes collectively form a feature vector. To reduce the dimensionality they applied PCA and reported that reduction to a vector of 30 results in an accuracy of 99%. In a similar approach Aly and Mohammed [54] used LBP to summarize the sign video into one spatio-temporal LBP feature vector. They proposed LBP on three orthogonal planes (LBP-TOP) in which LBP is performed in three planes: X-Y, X-T, and Y-T. where X and Y are the width and height of image and T is the successive images that form the sign video. They used uniform LBP codes in which the 8-bit binary patterns with at most 2 transitions from 0 to 1 or from 1 to 0 are labeled in 58 labels and other patterns that are not uniform are assigned a single label. A histogram of labels is generated for a block of 12x12 pixels in each plane and concatenated to form a feature vector. The method requires segmented images sequence which was done manually. An experiment to reproduce the results of this method has shown that it is not scalable for large databases.

Summary of the used features and classifiers is shown in Table 2.1.

2.5 Classification

Features are used for training and testing of the classifiers. Most of the reviewed work employed single classifier, however multi-classifiers are also used [2][55].

Mohandes et al. used HMM for isolated words' recognition. The suitable number of states that gives the maximum accuracy are evaluated and found that 6 states gave the best accuracy [10]. They likewise examined the impact of the number of Gaussian mixtures in HMM and concluded that 3 gives the best accuracy. Shanableh et al [3] examined linear classifiers like: linear discriminant function, Fisher's Linear Discriminant, Bayesian, K-NN, and HMM classifiers. They used 1-NN to determine the suitable cut-off for DCT, and the suitable value of threshold when forming the temporal difference image. They used left-to-right HMM with 2-4 states while preserving the temporal information of the image sequence by keeping the absolute differences images without accumulation in a single image. For signer independent recognition they used K-NN with correlation factor as a similarity measure, and polynomial classifier [12]. They used combination of signs performed by two signers for training and test by those performed by the third. The reported accuracy is 87%.

Tolba treated each image as a posture which is classified by multi-layer perceptron (MLP) neural network [43]. They represented the gesture as a sequence of posture classes called Non-deterministic Finite Automation. The Best Match algorithm is applied to decide the most probable match from a database of 50 signs. The accuracy drops dramatically when the object percentage is below 40% of the frontal view. Zaki used three HMMs for each sign, one for each feature[30]. The average number of frames in each sign was used to determine the number of states. For each of the three types of features (PCA, MCC, and Kurtosis position) a separate HMM classifier is employed. Majority voting is used for the final selection. The decision with the maximum log likelihood is used if all the three HMMs gave different decisions.

The work done by Sun [2] is an example of multi-classification stages employed for sign language recognition. The proposed technique can be described in three steps. First, a quantity of class-specific candidate exemplars are learned from sign language videos in each sign class looking for the most distinctive ones. Then, each video of all signs is represented as a collection of similarities between images within it and the candidate exemplars. The similarities are chosen by exemplar-based classifiers via Multiple Instance Learning (MIL). Lastly, they embed the choice of the most discriminative exemplars into a framework and concurrently produce a sign classifier to recognize the sign. Tharwat et al in [52] tested three classifiers to decide on SIFT features of 30 alphabets. Namely SVM, NN, and KNN and reported that SVM gave the best accuracy. Ahmed and Aly in [53] trained 23 left-to-right five-state HMMs. One HMM for each sign with single Gaussian component is trained using sequence of LBP features of each frame.

To test the spatio-temporal LBP-TOP, Aly and Mohammed [54] used SVM and KNN classifiers and reported that SVM outperformed KNN. For SVM they selected the kernel type experimentally and reported that linear kernel out performs polynomial and radial basis function (RBF) nonlinear kernels. El-Bendary et al used Nearest Neighbour classifier to decide on 30 alphabets[56]. Each sample is described by a vector of 50 normalized distances between the hand contour and a reference point that depend on the hand wrest location.

2.5.1 Phoneme Based

Inspired by speech recognition, in this approach signs are viewed as a collection of sub-signs. The task of sign recognition is based on recognition of a collection of sub-units of which the sign word is composed. This approach offers some advantages:

1. It requires less number of samples for training compared to word level classification.
2. Can support large vocabulary
3. Generalizes well for signer independence

On the other hand, multi-level classification is needed to recognise the sign.

Different procedures were used by researchers to extract and classify phonemes. Some used fixed number of phonemes [57, 58, 59, 43], others used data driven approach to detect phonemes [60, 61, 62]. Tolba et al represented the sign as a sequence of postures, they found that there is a total of 158 postures of which 88 are single hand and 70 uses two hands[43]. They built a data set of postures and trained multi layer perseptron to classify postures. Paulraj et al in [58] listed 44 phonemes of spoken English, and drove 11 signs which can be combined to form the 44 phonemes. They listed 11 categories of phonemes and used two hands, one to choose the phoneme's category, and the other performs the gesture of that phoneme. Neural network classifiers are used to classify the gestures performed by each hand. The output of these networks is applied to another network to decide on the phoneme. This work relies on spoken language to build a sign language vocabulary which is not applicable for sign language recognition.

Cooper et al in [57] followed linguistic studies of sign language and focused on 4 phonemes based on the Dictionary of British Sign Language. The 4 phonemes are HandShape, Location, Motion and Hand-Arrangement. They considered three scenarios: the first is based on appearance only, the second is based on 2D trajectories, and the third is based on 3D tracking. In the first scenario the Location, Motion and Hand-Arrangement are classified using AdaBoost from a labeled data set of phonemes. The main drawback in this scenario is the requirement of phoneme level labeled data set. In all scenarios two sign-level classification methods were used to combine the phonemes into full sign word. In the first method, a look up table of Markov chain, that are constructed for each sign, is used to find the most probable sign chain. In the second method Sequential Patten Boosting is used to select discriminative features that are found in the positive examples and not in the negative examples.

Li et al in [63] proposed a sensors based system for recognition of three sign language phonemes. Namely hand shape, orientation and movement. They designed custom accelerometers (ACC) and surface electromyographic (sEMG) sensors that can be worn as arm- and wrist-bands. ACC is used to measure hand orientations and movement trajectories, and sEMG capture the hand-shape. One ACC is mounted on the wrist of each hand and four sEMG sensors on the forearm. Measures from sEMGs are clustered for each arm to generate 8 hand-shape classes for the right hand and 7 classes for the left. A linear discriminant classifier is trained to detect different hand shapes. Features from ACC measures are used to train another linear discriminant classifier to distinguish between 5 hand orientations. Features from both ACC and sEMG sensors are jointly used to describe the hand motion. These features are used to train a multistream Hidden Markov model (MSHMM) to decide on the hand motion. To decide on a sign a two stage classifier is built. The first stage combines the output likelihood of hand-shape and hand-orientation classifiers as a weighted sum using Gaussian mixture model (GMM). The second stage combines the GMM and MsHMM as a weighted sum of the log likelihood. In testing for a single handed sign the maximum log likelihood indicates the sign's class. For two handed sign the class which maximizes the sum of likelihood for left and right hand is chosen. They tested the system on a database of 120 signs and 200 sentences and reported 96.5% and 86.7% accuracies respectively. Flasinski et al. built a system to recognize hand postures of Polish sign language [64]. In their system the hand posture is encoded as indexed edge-unambiguous (IE) graph which is then parsed by a syntactic graph language to determine it's class. Their system involves three phases namely image processing, graph generation, and graph parsing. In image processing phase skin-colour is used to binarize the image followed by morphological operations to fully extract the hand's contour. The contour is the approximated by a polygon. In the graph generation phase the polygon is transformed into a graph with labelled and directed edges and indexed nodes that preserves the image structure and is suitable for the graph parsing language. Nodes are labelled based on the internal angle made by edges that meet at it and indexed such that a breadth first spanning tree BFS can be made on the graph starting from the centroid. Edges are directed from the lower indexed node to the higher indexed node. These graphs are assumed to represent a grammar of language and a parser is used to classify postures[64].

Sutherland et al in [60] proposed data driven approach to detect phonemes. Phonemes are modeled as a continuous hand action in time and space. It is extracted by jointly analyzing the hand motion speed and its trajectory, assuming that the trajectory forms smooth curve in a phoneme. Speed discontinuity detector compares the motion of hands in two successive frames with a threshold, when exceeded a phoneme boundary point is marked. Similarly, when the bending angle of the hand trajectory curve exceeds a threshold, a candidate phoneme boundary point is marked. The phoneme candidates are described by a vector of features extracted from each frame and concatenated. These candidates are then clustered using agglomerative algorithm with dynamic time warping (DTW) as similarity distance to generate a code-book. The clusters' centers are then used to train weak classifiers which are combined to decide on the sign class by boosting. Two techniques were used for weak classifiers, HMM based and DTW based. One limitation of this work is that it ignores signs with local motion as the trajectory and speed will not vary much.

2.5.2 Continuous sign recognition

The natural use of sign language is to continuously perform signs one after another without pausing between signs. As in spoken language, speaker doesn't pronounce each word in isolation. The natural continuous signing suffers from the problem of signs' boundaries identification. When signing continuously, hand shape and position at the starting and ending of a sign is influenced by the shape and position of hand in the sequence of signs. It may be hard to identify the start and end of a sign. To tackle this problem different techniques were used to segment a continuously signed sentence into isolated words.

Yu et al in [65] proposed a technique to segment continuous signing video into sign and non-sign regions. The non-sign region is the transient state of hand when moving from one sign to the following sign. They assumed that in a word sign, homogeneous hand shape variation and trajectory is found and that doesn't exist in non-sign region. They proposed two stage segmentation. First coarse segmentation based on the hand position, followed by fine segmentation based on the hand shape. The coarse segmentation depends on dividing the signer's body into head, chest, and bottom regions. Segmentation occurs when the hand crosses the boundary between two regions. The fine segmentation depends on finding a large variation in hand shape which is assumed to characterise the non-sign. The variability is measured by a distance function that finds the distance between two successive hand-shape features vectors. Segmentation occurs if the variability is grater than a threshold. Assuming that muscles activities can differentiate between signs and motion epenthesis Li et al proposed to exploit measures of muscles activities provided by sEMG sensors[63]. They used measures of 4 sEMG sensors on the right arm to decide on the boundaries. The average energy of the 4 channels is calculated for consecutive chunks to form a time series. This time series is compared to a threshold to find the starting and ending of a sign. The staring boundary is the index in the time series for which the energy is grater than the threshold and the 2 succeeding are below it. The ending boundary is where the energy falls below the threshold provided that the two succeeding levels are above it. A third condition is that the duration of sign is not less than 5 chunks to avoid false segmentation. The threshold is calculated based on the background noise level and the maximum muscle tension energy.

The first attempt for continuous Arabic sign language recognition, to our knowledge, was by Assaleh et al [32]. However, they used a manually segmented database. Tolba et al in [29] modeled the signs sequence as a directed graph for each word in the sentence.The graph nodes are the classes of the hand-postures. To remove the transient frames they proposed two approaches. First calculates an energy difference between two consecutive frames based on their pixels values. If the energy is below a defined threshold the frame is discarded. In the second approach, each frame is divided into four regions (two halves horizontally and two halves vertically). Two successive frames are accepted to build the graph if they meet either of two conditions. The first condition, both have the same hand-posture but the hand in each one is in different region. The second condition, the two frames are from different hand-posture classes and the hand is in the same region in both frames.

2.6 Deep Learning Approaches For Sign Language Recognition

Deep learning refers to type of neural networks with large number of hidden layers. In deep learning no isolation between the features extraction stage and the classification stage. The network learns the appropriate features representation directly from the input. Deep learning has shown surge in computer vision applications with high accuracy in objects recognition. It has been adapted to the sign language recognition by many researchers such as [66, 67, 68, 69, 70, 71, 72]. The main challenge in using deep learning for sign language recognition is the temporal component of the dynamic signs. Researchers proposed different techniques for tackling this problem. Some used 3D convolution instead of 2D [67], Some used recurrent neural networks [73, 74], other used two streams one for spatial and one for temporal and fed the temporal one optical flow [75], others summarize the video in one image that captures the motion [76, 77, 78].

Pigou et al in [66] proposed a convolutional neural network for recognition of 20 signs from Italian sign language. The network is composed of two streams: one is fed by cropped upper body images, and the other is fed by cropped hand images. Each stream is fed by depth and grayscale images. Each stream is 3 layers deep composed of 3D/2D convolution and 3D pooling. The outputs of the two streams are concatenated and fed to an artificial neural network with one hidden layer. They reported a signer-independent accuracy of 91.7%. On a similar approach 3D convolution is used with five streamed network[70]. The first four streams are fed by cropped depth/intensity hand images of the right/left hand. The fifth is fed by a representation of upper body skeletal joints. Each of the first four streams is composed of three layers: Conv-Pool-Conv. The depth and intensity streams of each hand are early fused right after the third layer. The resulting streams are fed along with the skeleton features to MLP to generate the outputs. This five-stream network is repeated 5 times by varying the number of skipped frames to be 2,3,4, or 5 to capture the difference in signing speed. The output of these five parallel networks are lately fused to generate the final output.

In [77] Wang et. al. summarized the depth video into single image using rank pooling. Out of each depth video, six images are generated: two Dynamic Depth

Images (DDIs), two Dynamic Depth Normal Images (DDNIs) and two Dynamic Depth Motion Normal Images (DDMNIs) by applying forward and backward rank pooling. Each of the resulting images is fed to convolutional neural network. The resulting scores are element wise multiplied and the maximum score is used to classify the sign. They reported a recognition rate of 55.57% on a database of 249 signs.

Simonyan in [79] used a two stream architecture for spacial and temporal representation of the video. The spacial stream is fed by frames from the video while the temporal is fed by a volume the optical flow computed from these frames. The scores of the two streams are lately fused from the softmax layer by linear SVM.

For finger spelling recognition Li et al in [71] used a sparse auto encoder with CNN to learn features from RGB and depth images. The features learned are fed to a multi-stage PCA for features reduction and selection. Then a softmax classifier is used and reported an accuracy of 99.1% on 24 alphabets.

Non manual component of sign language also has been studied by deep learning to classify mouth shapes. Koller et al in [69] proposed a weakly supervised framework for recognition of mouth shapes incorporating HMM and CNN using Expectation Maximization algorithm. They reported an accuracy of 55.7% in classification of 40 mouth shapes.

2.7 Post Processing

After recognition of signs using sign recognition systems, some semantic errors might exist. In post processing phase these errors may be detected and corrected.

Samir and Aboul-Ela in [80] proposed a natural language processing based approach to detect and correct errors of the classification stage. The authors claim that the proposed approach was able to boost the recognition accuracy of Arabic sign language recognition by around 20%. To our knowledge this is the only published work applying post processing for ArSLR.

Table 2.1: Features and Classifiers used

Ref	Features	Classifier	Voc. size	Accur.
Mohandes 2005 [28]	Hands' centroids, Eccentericity of bounding ellipse of hands, Angle of the first principal components, Area of hands.	HMM	50	98%
Mohandes 2007 [50]	Hands' centroids, Eccentericity of bounding ellipse of hands, Angle of the first principal components, Area of hands.	HMM	300	93%
AL-Rousan 2009 [9]	vector of 50 DCT descriptors of segmented video images	HMM	30 offl	97%
				94 ind
			30 onl	93%
				90 ind
Shanableh 2007[3]	50 DCT descriptors of accumulated temporal difference	LDA	23	93.84%
		Fisher		93.18%
	Radon transform of accumulated temporal difference	LDA		88%
		Fisher		87%
Sun 2013 [2]	HOG and Kinect	mi-SVM and AdaBoost	37	85.5% spaci 86.8% temp+
Tolba 2012 [43]	weighted combination of 2D PCNN	MLP, Best match	50	96%
Shanableh 2011 [12]	DCT descriptors of weighted accumulated temporal difference	KNN and Polinomial	23	87%
Zaki 2011 [30]	PCA + Kurtosis Position + Motion Chain Code	HMM	30	89%

offl= offline, onl= online, ind= signer independent spaci= spacial feat.s only temp+ = spacial + temporal information

CHAPTER 3

KARSL : KFUPM ARABIC SIGN LANGUAGE DATABASE

To successfully build a robust system for sign language recognition based on machine learning, a database is needed. The database plays a significant rule in the success of such systems. Both the quantity and quality of the database effects the robustness of the recognition system. A High quality database that helps to build a robust recognition system should capture the variability in signing styles and the variability in signing environments. The quantity of the database samples determines the range of machine learning algorithms that it can train. Some machine learning algorithms requires a large number of samples to converge. This Chapter describes the collection of KArSL sign language recognition database as well as the formatting and naming styles.

3.1 Background

Over the last years, a number of attempts have been made to develop recognition systems for ArSL. However, one of the main challenges in automatic recognition of ArSL is the lack of benchmarking database [31]. To our knowledge there is no comprehensive benchmarking database for ArSL that is freely available for researchers. Thus, different researchers use different data sets, and consequently the recognition accuracy of the different techniques may not be comparable.

In this work, we are presenting ArSL database (KArSL) that was collected using state-of-the-art depth sensor: Microsoft Kinect V2 . Kinect V2 traces full-body motion and provides an accurate information about signers pose using 25 joints of the skeleton. In addition, it provides color image, depth map, infrared image, and user mask. To streamline the acquisition process, we have developed a recording software to capture the modalities provided by Kinect v2 sensor. The database consists of 500 signs from the ArSL dictionary. Each sign is repeated 50 times by each of four professional signers. This makes the database useful for machine learning.

Sign language corpora are scarce [81]. Even if there is enough video data available online, this data is not annotated and the signs are not segmented which makes it unsuitable for training sign language recognition (SLR) systems. For ArSL, the availability of a comprehensive benchmarking database is one of the challenges of ArSL recognition systems [82]. This is partially a result of the difficulties associated with sign language data collection such as time and cost

[83] in addition to the limited number of experts in the field of ArSL. As a result, some researchers developed their own data that is usually limited or unavailable for other researchers and consequently the recognition accuracies of the different techniques may not be comparable. In addition, most of these datasets are camera-based which do not provide any depth information.

3.2 KArSL database

In this section we present the KArSL database. KArSL (**K**FUPM **Ar**abic **S**ign **L**anguage) database consist of collected signs for ArSL using Microsoft Kinect v2. The collected database consists of eleven chapters of ArSL dictionary. Each sign of the database is performed by four professional signers and each signer repeated each sign fifty times. This resulted in 200 samples of each sign. Table 3.1 shows statistics of the collected signs while Table A.1 in the appendix lists the recorded signs. The database collection is part of the project number INF158001. "Recognition of Arabic Sign Language and Two-Way Translation between Arabic Text and Arabic Sign language using Natural Language Processing". KFUPM Internal Research Grand.

3.2.1 Setup and recording software

All signs of KArSL are recorded in an unconstrained environment. We didn't use dedicated lights in the recording room as the room lights were adequate where no shadow is shown in the records. We used fixed background (green) to facilitate

SN	Category	Number of signs
1	Letters	39
2	Numbers	30
3	Health	89
4	Common verbs	32
5	Family	32
6	Characteristics	49
7	Directions and places	16
8	Social relationships	10
9	In house	57
10	Religion	103
11	Jobs and professions	44
	Total	**502**

Table 3.1: Statistics of KArSL database

background removal for researchers who prefer using color video recording. In addition, the signers were not restricted to wear specific clothes or remove eye classes or watches. Each sign is recorded by each signer in two sessions where the signer wearing different clothes in each session. This helps in adding variety to the samples of each sign. To add more variety to the database, some signs, letters, are performed alternately between the left and right hands of the signer.

The database is recorded using multi-modal device, namely, the Microsoft Kinect V2. Microsoft Kinect was launched in 2010 and it consists of depth sensor, color camera, and multi-array microphone running proprietary software. The depth image is produced by CMOS IR sensor. This image can be used to create a mask image to remove the background by simple thresholding. The Kinect sensor traces full-body motion and provides an accurate information about signers pose. It provides color and depth images along with 25 human skeleton joints locations. For each sample, synchronized recording of color, depth, and skeleton joints is saved as shown in Figure 3.1. The color images are saved into MP4 video while depth frames and skeleton joints are saved into binary files.

The database is recorded at a rate of 30 frames per second with 1920x1080 pixels color video resolution.

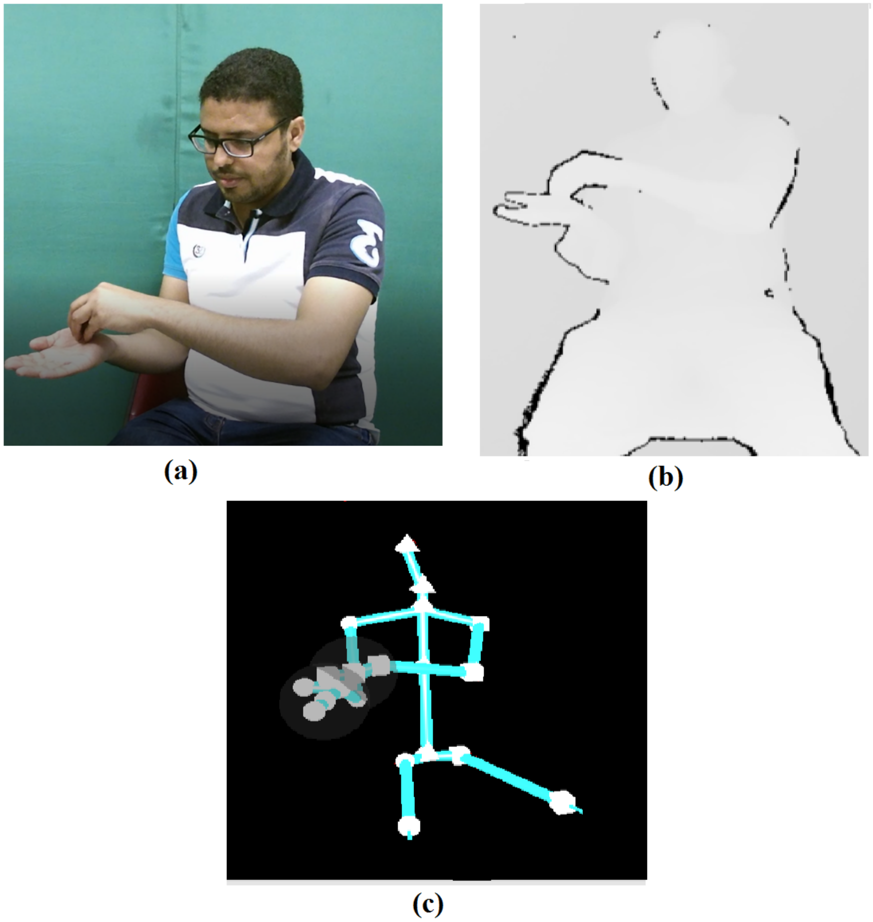
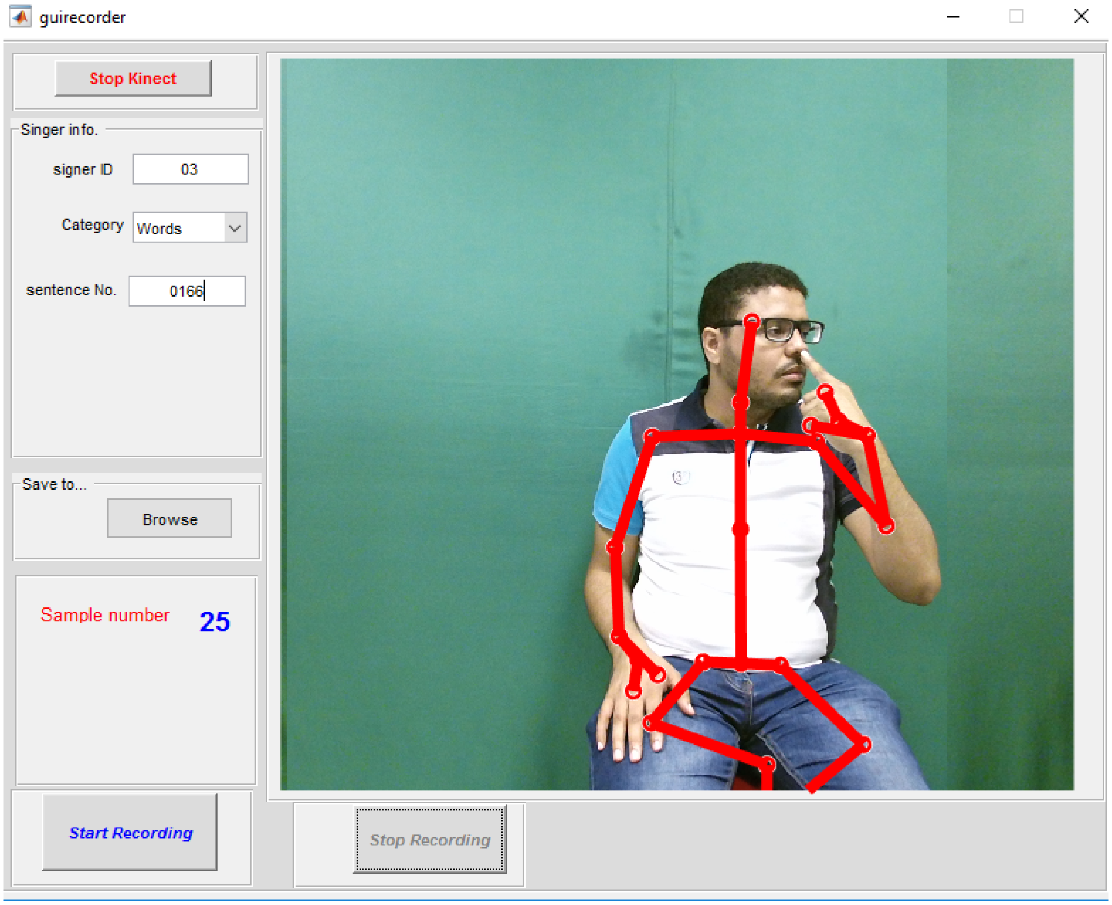


Figure 3.1: Output images of Kinect sensor. (a) color (b) depth (c) skeleton.

To facilitate the recording procedure, we have developed a data collection tool for capturing all the modalities provided by Kinect V2 sensor. Figure 3.2 shows the interface of the software for data collection. The software allows the user to select the category of ArSL, like letter, digit, word, sentence...etc, and gives a unique number for each signer. The signer unique number along with the sign serial number and category are used for sign file naming.



Singer info.

signer ID 03

Category Words

sentence No. 0166

Sample number 25

Figure 3.2: ArSL recording tool.

3.2.2 Structure of KArSL database

The hierarchical structure of KArSL database is shown in Figure 3.3. The signs are categorized into three classes: numbers (01), letters (02), and words (03). Each category folder contains a set of subfolders, one for each sign. Each sign is given a unique number of four digits. Inside each sign folder, there are subfolders, one for each signer. For each sign, there are 50 samples performed by each signer that are saved in: MP4 file for color video, and two binary files for depth and skeleton joint points. Each sign sample is saved with the name format: *Category_Sign_Signer_ddddd_Type.ext.* Where *Category* can be numbers, letters, or word; *Sign* is four digits sign' number; *Signer* is two digits signer number; *ddddd* is unique sample name that includes the recording time-stamp of the sample; *Type* is the type of Kinect output: color, depth, or Skeleton; and *ext* is the file type: color video (.MP4), depth and skeleton (.mat).

The file named xxxx_d.mat is the recording from the Kinect depth channel. It contains a matrix of 424 × 512 number of recorded frames. In some samples - early recorded- the depth data is saved to mp4 video. This is done by normalizing the depth data to the range [0 - 255] and replicating it in three channels to save RGB video.

The file named xxxx_c_s.mat is a MATLAB struct recording the skeleton joints data with the following fields:

- *body.Position:* The 3D position of each of the 25 joints listed in Table 3.2.
- *body.Orientation* The orientation of each of the 25 joints.
- *body.TackingState:* The tracking state of each of the 25 joints. 2: tracked, 1: inferred, 0: not tracked.
- *body.LeftHandState:* The left hand state: 2: opened, 3: closed, 4: lasso, 0: unknown.
- *body.RightHandState:* The right hand state: opened, closed, lasso, unknown.
- *body.Cpos2D:* The 2D position of each of the 25 joints on the color frame.

- *body.Dpos2D:* The 2D position of each of the 25 joints on the depth frame.

The joints locations are shown in Figure 3.4.

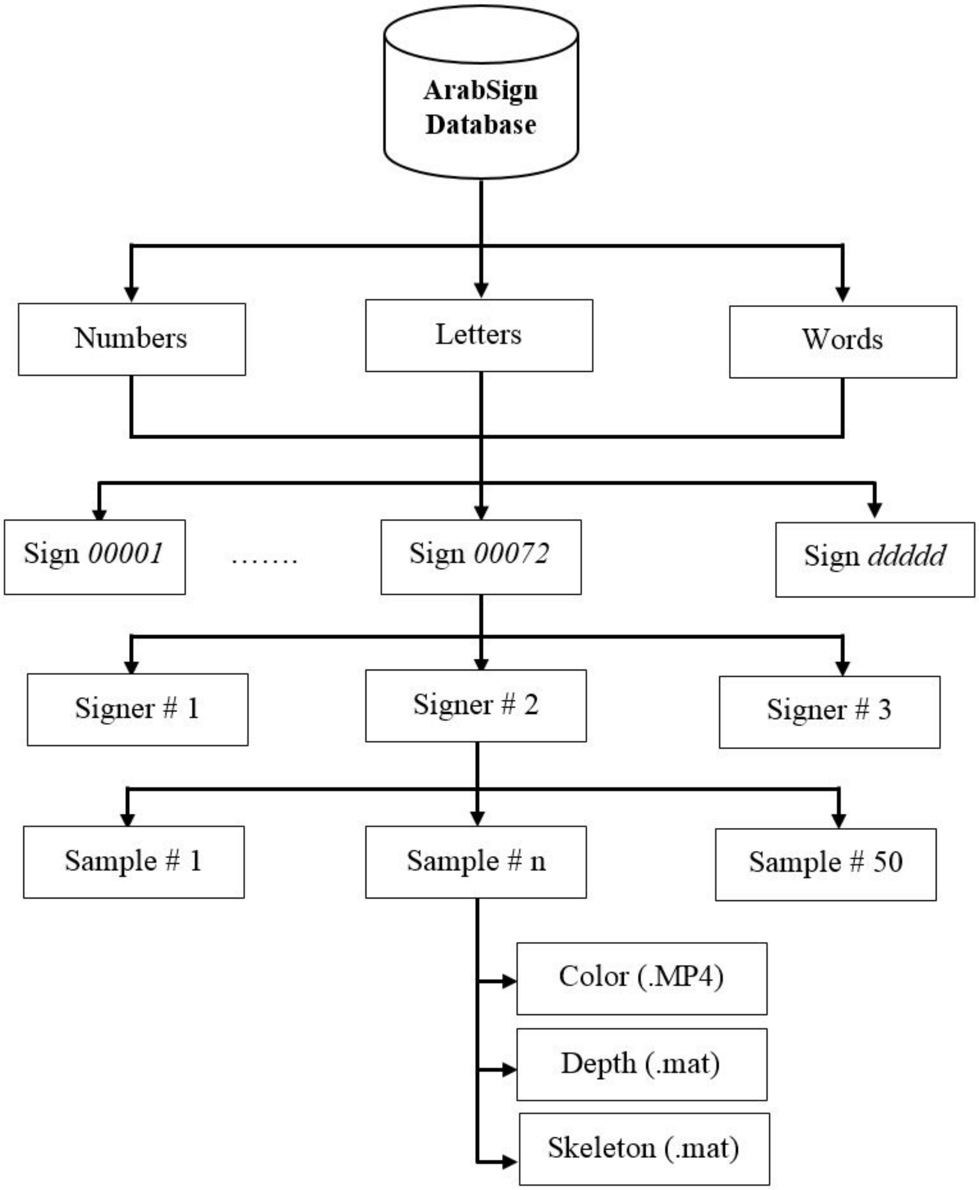


Figure 3.3: KArSL file structure.

Using this database several recognition systems are proposed and tested as will be shown in the next chapters. The features of this data base include:

- Synchronized recording of color, depth, and skeleton.

Table 3.2: List of the joints and their index in the body struct

Index	Joint Name	Description
1	SpineBase	Base of the spine
2	SpineMid	Middle of the spine
3	Neck	Neck
4	Head	Head
5	ShoulderLeft	Left shoulder
6	ElbowLeft	Left elbow
7	WristLeft	Left wrist
8	HandLeft	Left hand
9	ShoulderRight	Right shoulder
10	ElbowRight	Right elbow
11	WristRight	Right wrist
12	HandRight	Right hand
13	HipLeft	Left hip
14	KneeLeft	Left knee
15	AnkleLeft	Left ankle
16	FootLeft	Left foot
17	HipRight	Right hip
18	KneeRight	Right knee
19	AnkleRight	Right ankle
20	FootRight	Right foot
21	SpineShoulder	Spine
22	HandTipLeft	Tip of the left hand
23	ThumbLeft	Left thumb
24	HandTipRight	Tip of the right hand
25	ThumbRight	Right thumb

- Large vocabulary of signs.

- Large number of samples for each sign.

- Suitable for sensor based and vision based recognition.

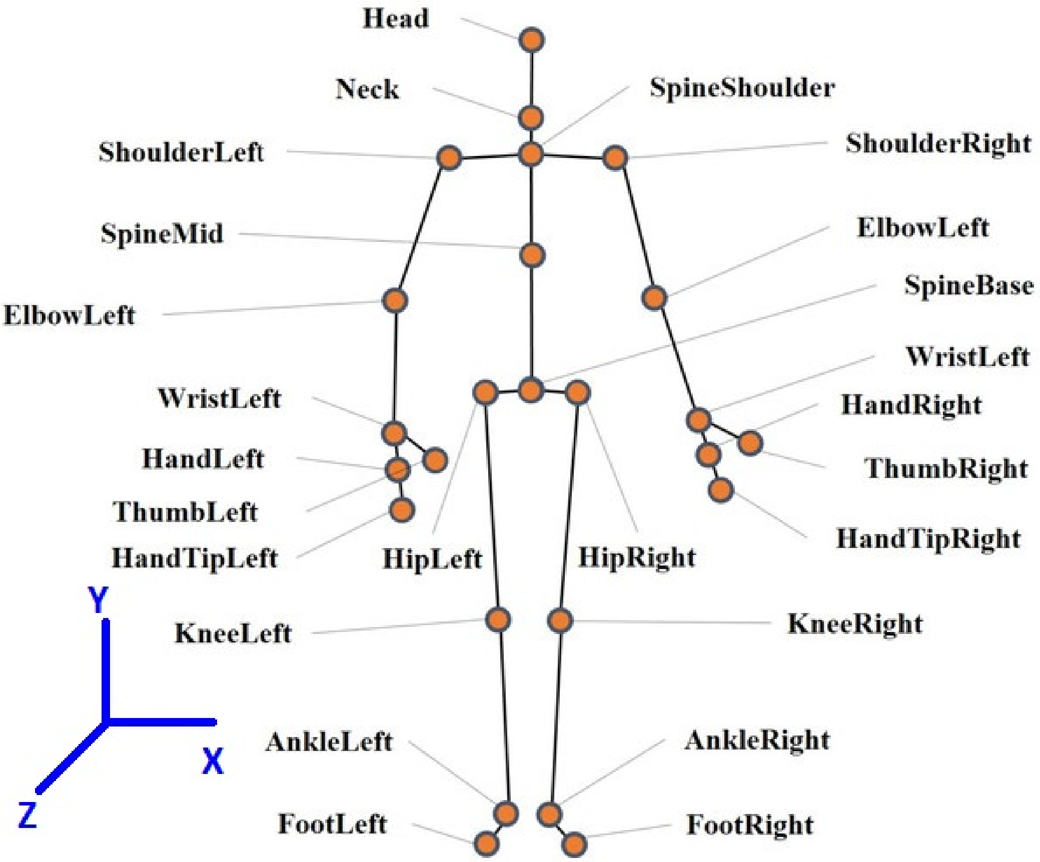


Figure 3.4: The 25 Skeleton joints recorded in the body struct. Note the Z axis is pointing towards the Kinect.

CHAPTER 4

SENSORS BASED SIGN LANGUAGE RECOGNITION

Sign language recognition systems can be broadly classified as vision-based or sensors-based systems. The vision based systems employ cameras to capture videos of the signs and then recognize signs based on these videos. On the other hand, sensors based systems uses different sensors to record some measures that can be used to recognize the signs. Different types of sensors has been used by researchers to model the signs and recognize them. Some sensors are used to measure the status of the fingers, and others to find the location of hands.

In this chapter, we propose an Arabic sign language recognition system based on hands motion trajectory obtained by the hand joint locations from Kinect.

4.1 Overview

In sensor based recognition, researchers tend to mount some sensors on a glove and the signer wear that glove while signing. The use of gloves can be counted as a limitation of these systems because natural signers are not used to wear them. In contrast there exist other sensors that can provide measures without being attached to the signer‘s body. Samples of these sensors are the Leap Motion Controller and Kinect. The Leap Motion Controller is limited: in range of coverage, and fails to track the hands on occlusion, and can only provide information about the hands. The Kinect can provide measures of 3D location for 25 body joints. These joints include the head, shoulders, elbows, wrists, hands, and others.

Sign languages are full featured languages with their own vocabularies and grammar. They make use of hands-motion, fingers-configurations, facial-expressions, and body lane in parallel to express different signs. To effectively translate a signed language all its components need to be considered. Of these components, the hands-motion is one of the most important modalities of signed language.

This chapter discusses the usage of the 3D/2D trajectory of hands to recognize signs.

Trajectory processing exists in a wide range of applications. Therefore a lot of work is done on trajectory processing in on-line character recognition [84, 85], action recognition [86, 87], gesture recognition [88] and more.

Lin and Hsieh in [89] proposed a kernel based trajectory representation using

Kernel Principal Component Analysis (KPCA) and Nonparametric Discriminant Analysis (NDA). In their method a 2D/3D trajectory is first min-max normalized then projected to higher dimensional space using KPCA. The dimensionality is reduced using NDA with the hope of maximizing the inter-class variability and minimizing the within-class variability. The resulting representation is hoped to be more discriminative and is classified using the nearest neighbor rule. The approach is tested on a limited set of 38 words from the Australian sign language and reported accuracy of 69% for 2D trajectory and 78% for 3D.

Naftel and Khalid in [90] encoded the 2D trajectory along x and y dimensions by Discrete Fourier Transform (DFT) separately. Then the first 4 coefficients are used as feature vector that represent the trajectory. The coefficients are then clustered using Self Organized Map (SOM). They tested the approach on 24 words from Australian sign language and reported an accuracy of 70.1%

Pu et al. [91] modeled the trajectory as a sequence of M sub-motions and used HMM to model the transition between these sub-motions. For each point on the sub-motion trajectory, they find the *shape context* as a histogram of relative coordinates of other points on the sub-motion trajectory. Then a codebook is generated from these shape contexts. The features vector of each sub motion curve is composed as a weighted histogram of the code book centers. The weights are found by soft clustering the shape context of each point. Finally, the sign curve feature is a sequence of M sub-motion features. They tested the system on a database of 100 signs from the Chinese sign language and reported an accuracy of 67.3% for signer dependent and 54.4% for signer independent.

Boulares in [92] extracted signatures from 3D hands trajectories and used SVM to classify different signs. To extract trajectory signature, they use non linear regression to fit the trajectory points to a conic section. The trajectory signature along with hand shape and other features is used to train and test SVM classifier. Curve fitting does not accurately represent complex trajectories that include cycles.

Geng et al. in [93] used a combination of trajectory modeling and hand shape representation as a feature to train an Extreme Learning Machine (ELM) classifier. A combination of 3D trajectories of hand, wrist, and elbow are used. They normalized the values of trajectory points to [0 , 1] range and smoothed the trajectory by average convolution. To form a feature vector from the smoothed trajectory, they subtracted the starting point of the trajectory from all following points. The difference between the hand trajectory and wrist trajectory is represented by spherical coordinates system and similarly for the hand-elbow trajectory difference. The final features vector is concatenation of hand trajectory, hand-wrist spherical difference, hand-elbow spherical difference, and hand shape features from depth image. These features are used to train ELM and 82.8% accuracy is reported on a limited database of 8 words from the Chinese sign language. Normalization of trajectory points to the range of [0 , 1], results in loss of information about where was the hand motion with respect to body when signing the word.

Wang et al. in [94] formed the trajectory of hands as a combination of hands location and orientation. The hand location is defined as the hand location with respect to the face centroid and with respect to the non dominant hand location. Similarly, the orientation is defined as the direction between successive hand locations. For single handed signs the trajectory of non dominant hand is set to zeros. All trajectories are normalized to have the same length. Similarities between trajectories are measured by dynamic time wrapping (DTW). Based on the trajectory matching the reported top 10 accuracy of the sign search results is about 74% and improved to 78% when incorporating additional hand shape feature. They slightly modified the trajectory feature in [95] by including the hand velocity and defining separate feature for single handed signs doesn't include the hand location with respect to non dominant hand location. However the information of single or two handed sign need to be given by the user.

Bhuyan et al. in [96] modeled the trajectory as a combination of shape and motion features. The shape features include, the trajectory length, and the number of curves in the trajectory. The motion features include, the average speed, standard deviation of the speed, and the number of minima in the velocity. The classification of gestures is done in two stages. First candidate signs are included based on the trajectory shape similarities using maximum boundary deviation as similarity measure. In the second stage trajectories are aligned using DTW then the trajectory features are classified based on the nearest candidate template.

Mohandes and Deriche proposed a system for Arabic sign language recognition [42]. The trajectory is composed of 3D position and orientation with 12 dimensional vector for both hands. For each dimension the acquired readings are partitioned into 5 equal partitions. From each partition the mean and standard deviation is calculated. That results in 120 dimensional features vector. LDA is used to reduce the dimensionality to 20. The nearest neighbor classifier is used to find the class of a sign. They reported an accuracy of 84.7% on a dataset of 100 words.

4.2 Preprocessing

In this work, we used Kinect to record signs. We recorded synchronized color image, depth image, and 25 body joints locations. We recorded the 3D locations of joints and the 2D mapping to both color and depth images. For this work we use the sequence of hands locations in 3D to recognize signs.

Trajectory preprocessing includes: Noise removal, and Compression. The joints‘ locations obtained by Kinect are noisy and include some outliers. The noise removal stage smooths out these outliers by using median filter. Since the frame-rate for recording is at 30 frames per second, fine details of part of second trajectory is not very useful and results in redundant information. Trajectory compression stage compresses the trajectory into few key points. To find such key points we treat the trajectory as a polygon formed by connecting the locations of the hand while signing. The key points are obtained by reducing the number of vertices of this polygon to a specific number. The reduction is done by recursively calculating the importance of each vertex based on angle and segment length and then removing the least important. The process is repeated until the desired number of vertices is reached. Figure 4.1 shows the calculation of vertex importance. The algorithm for trajectory compression is shown in Algorithm 1 . Figure 4.2 shows the effect of 3D trajectory preprocessing. The preprocessing of a 2D trajectory is shown in figure 4.3.

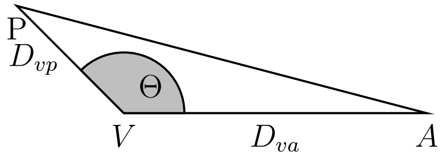


Figure 4.1: The importance calculation for vertex V is found by multiplying the distances from v to adjacent vertices P (previous), A (after) and the angle Θ as $IMP_v = D_{vp} \times D_{va} \times \Theta$.

Algorithm 1 TrajectoryCompression

```
procedure COMPRESS(Traj,NumVertices)
    $TrajLength \leftarrow$ LENGTH($Traj$)
    for all points v in Traj do
        $IMP(v) \leftarrow$ CALCIMPOTRANCE($Traj, v, NumVertices$)
    end for
    while $TrajLength > NumVertices$ do
        $I \leftarrow IndexOfMin(IMP)$
        $Traj \leftarrow Traj - Traj(I)$                ▷ The - sign is set difference
        $IMP \leftarrow IMP - IMP(I)$
        $TrajLength \leftarrow TrajLength - 1$
        update $IMP$ by recomputing the importance of the removed vertex's neighbors.
    end while
    function CALCIMPOTRANCE(Traj,v,NumVertices)
        $imp \leftarrow D_{vp} \times D_{va} \times \Theta$
        return $imp$
    end function
end procedure
```

Some of the previous works include another stage in preprocessing called min-

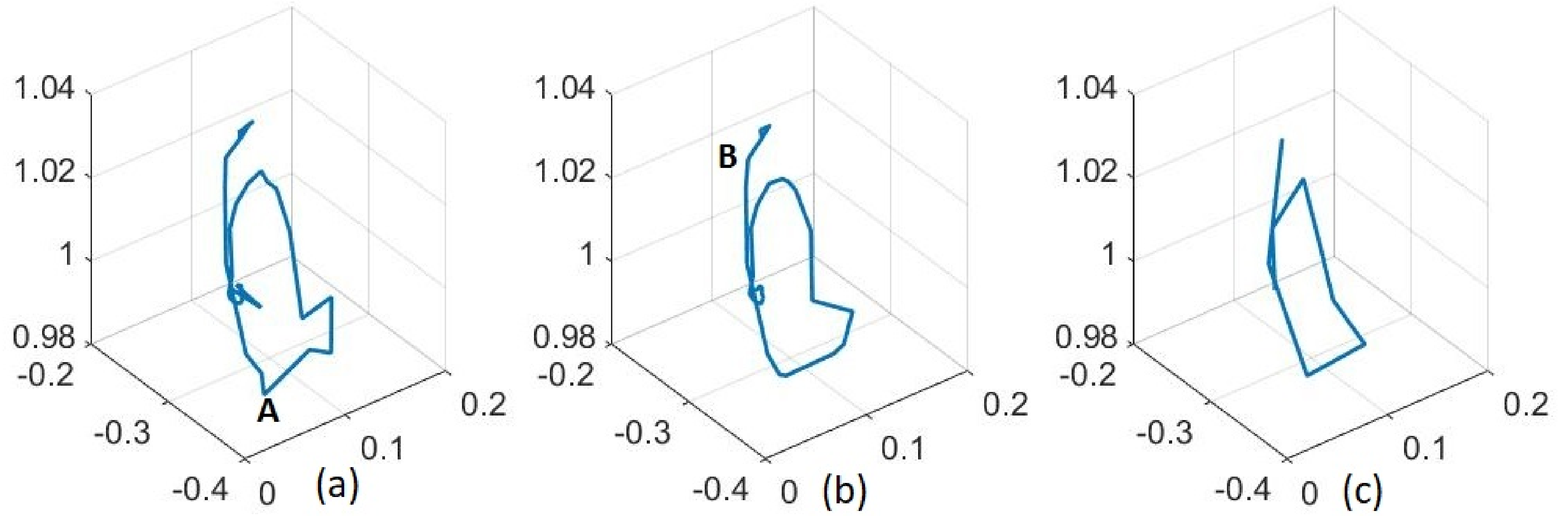


Figure 4.2: The preprocessing stage of the 3D trajectory.(a) raw trajectory 'A' is a noisy point smoothed out by the median filter. (b) Median filtered 'B' is a less important point removed by compression stage. (c) compressed.

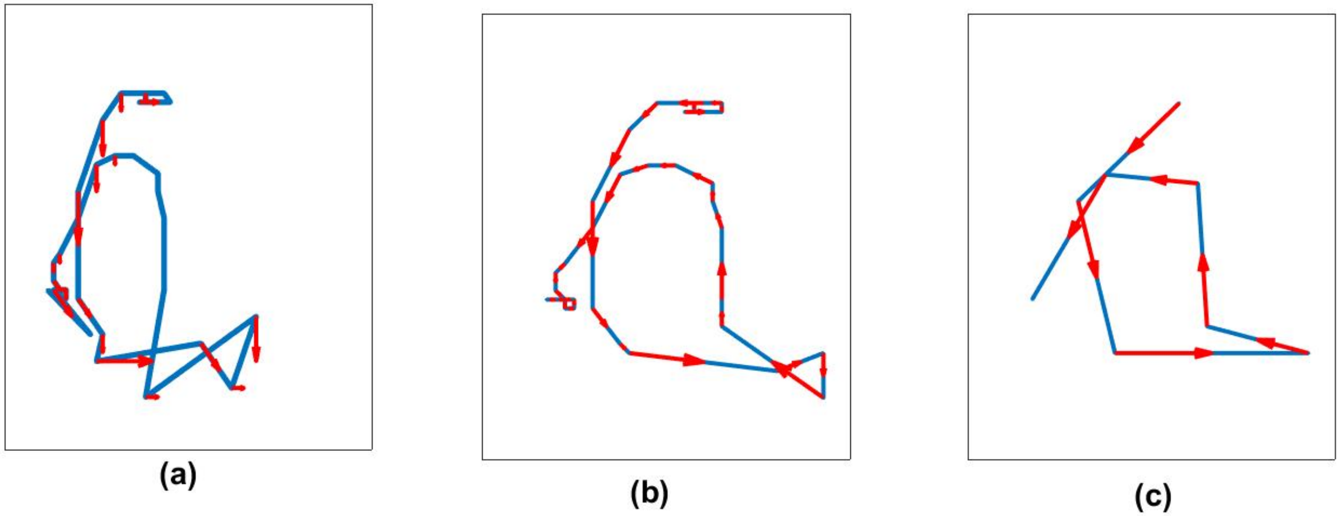


Figure 4.3: The preprocessing stage of a 2D trajectory. (a) Raw trajectory. (b) The median filter reduce the noise of the trajectory resulting in smoother one. (c) The compression stage finds the most important 8 points in the trajectory. The arrows indicate the direction of motion.

max normalization. In this stage the trajectory is normalized to be in [0-1] range. In this work, we exclude such stage arguing that it leads to loss of discriminative information. Signs can have similar trajectory pattern but at different locations and min-max normalization leads to loss of the localization features of the trajectory. For example face sign and colic sign share the same circular trajectory, but face is signed by circulating around the face and colic is signed by circulating at the abdomen.

4.3 Features Representation

After noise removal and compression, features are extracted from each sign trajectory. Here we describe two types of features.

4.3.1 Polygon-based Features

In this method the 3D hand trajectory of hand is represented as a polygon. The description of this polygon is represented by: it‘s center of gravity and the distances from the perimetric points to the center of gravity point. The center of gravity point is approximated by the mean of perimetric points calculated as $G = (\bar{x}, \bar{y}, \bar{z})$ where $\bar{r} = \frac{1}{N}\sum_{i=1}^{N} r_i$ and N is the number of perimetric points. The distance from G to permetric points is calculated using the Euclidean distance formula $d_i = ||G - P_i||$, $i = 1, 2, 3, ..., N$. Figure 4.4 illustrates the polygon-based features extraction procedure.

Then the polygonal features vector is formed by concatenating G and d_i as

$$F = [\bar{x}, \bar{y}, \bar{z}, d_1, d_2, d_3, ..., d_n]$$

This feature representation captures both of the trajectory shape and more importantly the position of hand motion. The position of hand motion is important as it distinguishes between signs with similar trajectories but different body positions.

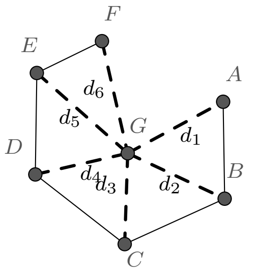


Figure 4.4: The polygonal features are found by the center of gravity G and distances $[d_1, d_2, d_3, d_4, d_5, d_6]$ form G to perimetric points [A,B,C,D,E,F] respectively

4.3.2 Positional Trajectory Features

In this feature representation only perimetric points of the trajectory polygon are included. The feature vector is a concatenation of perimetric points formed as

$$F2 = [x_1, y_1, z_1, x_2, y_2, z_2, ..., x_N, y_N, z_N]$$

This feature representation although it is simple, but shown very good discrimination and generalization as will be shown in the experimental results section.

4.4 Classification techniques

After preprocessing and features extraction, in the classification stage features are used to train and test classifiers. We tested several classifiers and found that the best accuracy is obtained when using ensemble of classifiers. Specifically, the best performing classifier is Ensemble Subspace KNN.The tested classification algorithms are listed in Table 4.1. We use 5 folds cross validation.

Table 4.1: List of classifiers used in the experiments along with the accuracy obtained by each.

Tree (90)	Linear Discriminant (92)	Ensemble Boosted Trees (86)
SVM Linear (97)	SVM Quadratic (97)	Ensemble Subspace Discriminant (96)
SVM Gaussian (97)	KNN Euclidean (97)	Ensemble Subspace KNN (**99.5**)
SVM Cubic (97)	Quadratic Discriminant (97)	Ensemble Bagged Trees (99)
KNN Cubic (97)	KNN Cosine (96)	Ensemble RUSBoosted Trees (30)

The basic random subspace algorithm uses these parameters:

- N is the number of learners in the ensemble.
- D is the number of dimensions in the features vector.
- M is the number of dimensions to sample in each learner.

In subspace ensemble algorithm, a set of N weak learners each is trained on a randomly chosen partition of the features vector of M dimensions less than the D dimensions of the original feature vector. On prediction, the average score from weak learners is calculated and the class with the highest average score is chosen as the true class [97]. In this work, we used KNN as a weak learner to build the ensemble subspace classifier. It is clear that N,M and K (of the KNN) are hyper parameters that need to be chosen for best performance of the classifier.

To find the best values for these parameters we use cross validation as shown by Algorithm 2.

Algorithm 2 Fine Tune parameters of Ensemble Subspace KNN

```
function FineTune(Feats)
    $S \leftarrow NumberOfSamples$
    $D \leftarrow NumberOfDimentions$
    $KCanidates \leftarrow$ set of 10 values between 1 and $\log S$
    for all k in KCanidates do
        $Loss(k) \leftarrow$ XValidateKNN$(Feats, KCanidates[k])$
    end for
    $BestK \leftarrow KCanidates[MinimalLoss]$                ▷ Find the best K
    $MCanidates \leftarrow$ set of 10 values between 1 and $D$
    $N \leftarrow 100$                ▷ Fixed Number of weak classifiers
    for all m in MCanidates do
        $Loss(m) \leftarrow$ XValEnsemKNN$(Feats, MCanidates[m], BestK, N)$
    end for
    $BestM \leftarrow MCanidates[MinimalLoss]$                ▷ Find the best M
    $NCanidates \leftarrow$ set of 100 values between 1 and 100
    for all n in NCanidates do
        $Loss(m) \leftarrow$ XValEnsemKNN$(Feats, BestM, BestK, NCanidates[n])$
    end for
    $BestN \leftarrow NCanidates[MinimalLoss]$                ▷ Find the best N
    return $BestK, BestM, BestN$
end function
```

The algorithm first run KNN with different values of K to find the best performing one (BestK). Then fix the number of weak classifiers to 100 and K to BestK and search for the best number of partitions BestM. With BestK and BestM the algorithm then searches for best number of weak learners BestN.

4.5 Experimental Results

A set of experiments are carried out to evaluate each stage of the proposed system. Starting by preprocessing stage to the classification stage to fine tune the hyper parameters and then test the sign language recognition. In these experiments, portion of KArSL consisting of 100 signs starting from sign 0071 up to 0170 (Table A.1) recorded by 3 signers is used. For this work, only the hands joints trajectories are employed to recognize signs.

4.5.1 Effect of Trajectory Compression

Here we investigate the effect of the number of vertices used to represent the trajectory as a polygon on the accuracy. For this experiment, we use the trajectories of all signs performed by one signer (signer 3) and apply the preprocessing stage by varying the number of vertices from 4 to 18. We call this data-set the validation set in this chapter. Figure 4.5 shows the classification error rates for different representations of trajectory features. In this figure, F1 represents the *polygon description* feature representation of trajectory (see section 4.3.1) while F2 stands for the *positional trajectory* feature representation. The 1H and 2H encodes the usage of only one hand trajectory of both hands respectively in building the feature vector. In 1H the features encode only the trajectory of the dominant hand while in 2H a concatenation of features that encode both hands trajectories is used. The 2D and 3D for which trajectory points representation being used, X-Y or X-Y-Z respectively. From this figure, many properties can be inferred. First, the best average accuracy can be obtained when using a polygon with 12 vertices. Using small number of vertices will not capture complex trajectories well, and using very high number of vertices will include noisy details that mix up distinct classes. Second, the usage of 3D trajectory always performs better than the 2D. This can be attributed to the fact that the Z dimension captures front-back motion of hands, and there are some signs in the database with only front-back motion pattern. Third, the inclusion of non-dominant hand in the feature representation increases it‘s discrimination power. The state of non-dominant hand in sign language can either be static, mirrors the motion of dominant hand, moving in different way than the dominant hand. In all cases of non-dominant state, it‘s motion pattern helps in distinguishing similar signs that are of similar dominant hand trajectory. Forth, as a comparison between the two features representation the *positional trajectory* feature representation outperforms the *polygon description* feature representation of the trajectory.

4.5.2 Fine tuning EnsembleSupspaceKNN classifier

We run Algorithm 2 on a the same validation set used in section 4.5.1 to find the best parameters for each feature representation. Table 4.2 lists the best parameters settings for each feature representation. In this table the best value for K is 1 for all features, the best value for M for feature F1 is roughly half D which is similar to the findings in [97]. The values in BestN column are for the value of N after which no significant drop in loss is seen. Based on this table, the parameters settings for following experiments will be: K=1, N=40, M= BestM from the table.

Table 4.2: Best parameters for EnsembleSupspaceKNN classifier

Feature	D	BestK	BestM	BestN
F1-1H-2D	14	1	8	40
F1-1H-3D	15	1	9	25
F1-2H-2D	28	1	13	40
F1-2H-3D	30	1	14	25
F2-1H-2D	24	1	6	40
F2-1H-3D	36	1	9	40
F2-2H-2D	48	1	6	40
F2-2H-3D	72	1	9	40

4.5.3 Evaluation of the Proposed Features

After choosing the best trajectory compression ratio and the best parameters settings for the classifier, the system is tested on the 100 words database. Table 4.3 lists the recognition rates obtained when using each feature representation for each signer in the database. The results reflects that the 3D trajectory is more informative and discriminative than the 2D one. In addition, the inclusion of non dominant hand status improves the accuracy for both types of trajectories. The third signer shows better accuracies than the other two which can be attributed to the less variability in his performance of signs, and the samples used for fine tuning the hyper parameters (validation set) are performed by him. The fifth column lists the accuracies when using mixed samples from all signers for both training and testing. This shows the scalability of the system to larger number of samples.

Although the number of signers is not big enough to evaluate the system for signer independent recognition, we run experiments to get initial intuition about the generalization of the system to unseen signer. Table 4.4 lists the accuracies of

Table 4.3: Signer dependent classification recognition rate

Feature	Signer1	Signer2	Signer3	All Signers	Average
F1-1H-2D	89.80	88.80	91.50	84.40	88.63
F1-1H-3D	96.20	95.00	97.60	94.70	95.88
F1-2H-2D	97.60	96.80	97.90	96.40	97.18
F1-2H-3D	99.30	98.80	99.40	99.50	99.25
F2-1H-2D	97.70	96.00	98.20	95.60	96.88
F2-1H-3D	99.20	99.00	99.80	99.20	99.30
F2-2H-2D	99.50	98.90	99.70	99.10	99.30
F2-2H-3D	99.70	99.60	100.00	99.70	99.75
Average	97.38	96.61	98.01	96.08	

the different type of features in signer independent mode. Each column is named by the test signer when the training is done by samples performed by the other two signers. The lower results of the second signer are due to the different signing style, some signs are repeated more than once in the same sample. Overall average performance is around 53% for all features 48%, and 57% for F1, and F2 features respectively.

Table 4.4: Signer independent classification recognition rate

Feature	Signer1	Signer2	Signer3	Average
F1-1H-2D	40.10	27.20	43.10	36.80
F1-1H-3D	44.60	30.70	50.90	42.07
F1-2H-2D	57.50	48.60	60.10	55.40
F1-2H-3D	60.00	51.00	64.80	58.60
F2-1H-2D	56.80	41.90	58.70	52.47
F2-1H-3D	60.20	43.20	65.70	56.37
F2-2H-2D	58.90	47.80	63.90	56.87
F2-2H-3D	61.30	49.70	64.40	58.47
Average	54.93	42.51	58.95	

4.5.4 Comparison with published work

In this experiment we test the proposed features representation and classification algorithm on a publicly available dataset and compare the results of our method with published work on the same dataset. The used dataset is composed of 95 Australian sign language words. Each word is performed by 1 signer 27 times. For each sample a vector of 22 measures is recorded per frame. These measures include the 3D position of hands (X,Y,Z), the orientation of hands (Roll, Pitch, Yaw), and the status of fingers. Some of the previous works used only the (x,y) points to form 2D trajectory while others used 3D. In this work, we use the 2D/3D trajectory as well as the hand orientation. We follow the same steps of trajectory preprocessing, features representation, and classification. In this dataset, the signer starts with his hands on the rest position and return them back to the rest position after signing. This makes the center of gravity of some signs to be the same. To avoid that, we run the compression stage twice. First with 14 vertices which includes the starting and ending rest position. Then we find the 12 vertices after excluding the first and last points which results in removing the rest position from the calculation of the center of gravity. Table 4.5 shows the accuracy reported by previous works along with our work (the last 4 lines). The first row shows the number of classes out of 95 used. In this table, F1 stands for the polygonal description feature representation and F2 for the positional feature. 3D stands for the only use of 3D hand position to form the feature while 3DO for inclusion of the hand orientation too.

Table 4.5: Comparison with published work on AUSLAN.

Reference	2 Words	4 Words	8 Words	16 Words	29 Words	38 Words	All Words
[89] 2D	98%	93%	86%	78%	72%	69%	-
[90] 2D	98%	92%	88%	83%	-	-	-
[89] 2D	98%	93%	86%	78%	72%	69%	-
[89] 3D	99%	96%	92%	89%	82%	78%	-
[98]	-	-	-	-	-	-	86.7%
F1-2D	100%	100%	95.4%	76.7%	63.7%	58.8%	46.8%
F1-3D	100%	100%	98.1%	90.4%	76.4%	70%	58.3%
F1-3DO	100%	100%	99.1%	95.4%	89.5%	86.5%	82.8%
F2-2D	100%	100%	96.3%	85.2%	74.3%	68.8%	61.7%
F2-3D	100%	100%	99.1%	94.8%	86.2%	79.7%	74.5%
F2-3DO	100%	100%	98.1%	95.9%	92.8%	88.7%	88.4%

Note that the work in [98] uses the 22 features while ours use only 3 (in case of 3D feature) and only 6 (in case of 3DO) of them.

4.6 Conclusions

In this chapter, we proposed a system for Arabic sign language recognition based on the trajectories of hands. We modeled the trajectory as a polygon and proposed two polygonal description features. The system shown good performance for both signer dependent and signer independent recognition. The accuracy of the system reached 99% for signer dependent and 64% for signer independent recognition. The proposed system is tested on two different datasets and was compared with published works that use the same dataset and shown better performance than most of them. The proposed system features simplicity, scalability, and generalization to unseen signer.

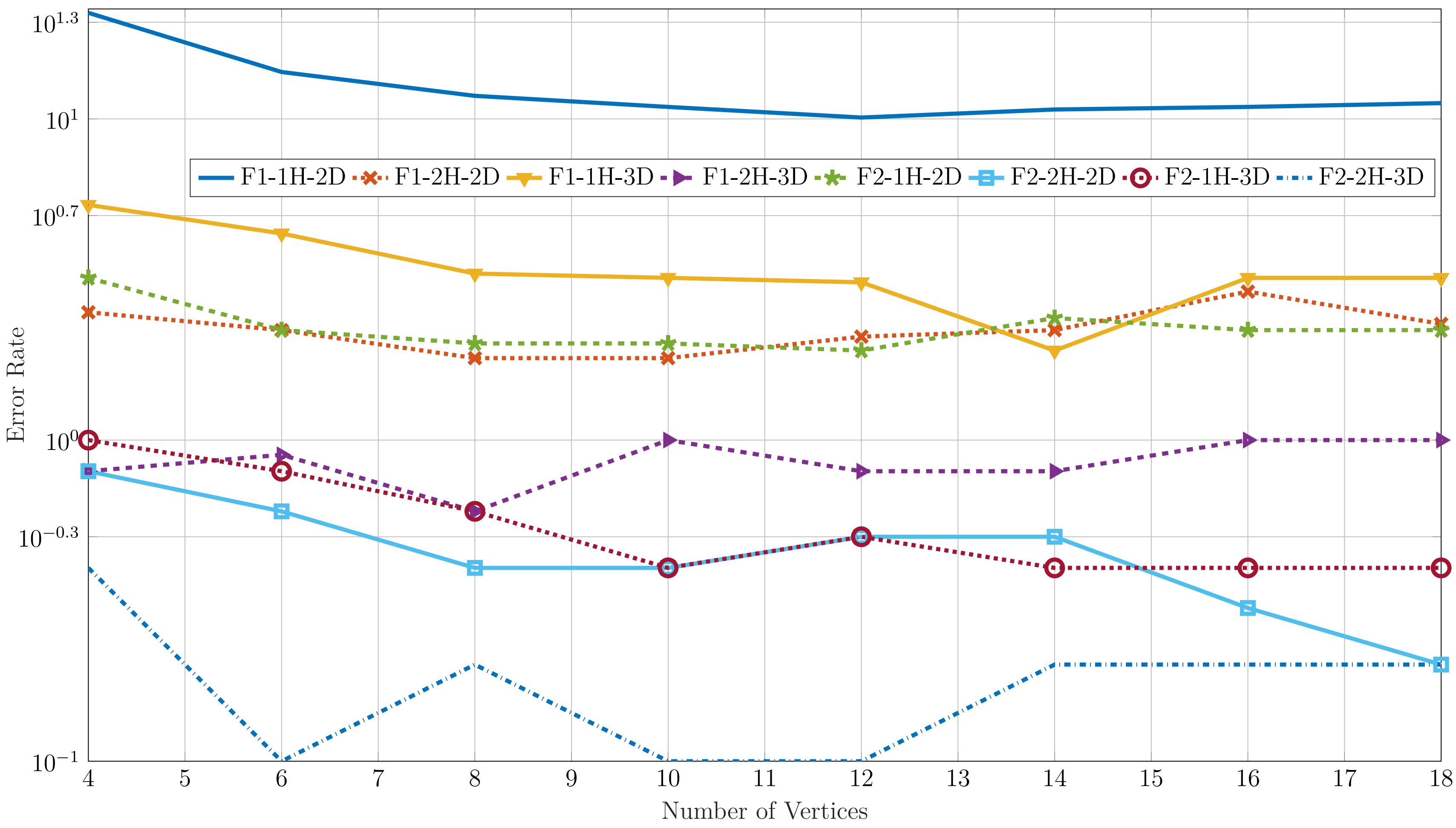


Figure 4.5: The compression effect on the classification error rate for different versions on the proposed features representation. The Y axis is log scaled for better visualization

CHAPTER 5

VISION BASED RECOGNITION USING BOF AND BOP

Vision based sign language recognition systems are more natural than sensors based ones as they do not require sensors to be worn by the signer. Generally, in vision based recognition, the system pipeline starts by finding a suitable representations of signs, and then encoding this representation as set of features. These features are then used to train a classifier.

In this chapter, we represent the sign as a sequence of body postures. This representation accounts for both body lane and hands gesturing. To this end, the body posture is described using *bag of visual features* (BoF).

5.1 Introduction

In BoF, features are extracted from the training images of the different classes which are clustered to generate a *code book*. Each class is characterized by a histogram of the code book entries. These histograms are used to train a classifier to model the classes. One drawback of BoF is that it doesn't preserve the context.

The first step in BoF is features extraction in which the interest points are detected and described by discriminative descriptors. Different techniques are commonly employed to detect interest points. Interest points are localized in scale space of an image which is generated by convolving the image with different Gaussian kernels σ. Then a filter is applied to the scale and the extremal responses are marked as interest points. Different filters localize different types of interest points. The Difference of Gaussian (DoG) filter finds blob like shapes in the image as interest points. DoG is used by Scale Invariant Features Transform (SIFT) which is one of the most popular interest-points localization techniques. Another filter that detects corners is Harris detector [99]. It searches for high gradients in two perpendicular directions to locate corners. After detecting interest points, they are described by a function of the neighboring pixels. The most used descriptor in BoF is SIFT descriptor [100]. SIFT descriptor is a 128 dimensional histogram of gradients' orientations around the interest points. Another approach for features extraction omits the interest points detection step and directly finds descriptors at regular grid points. Dens SIFT (DSIFT) is an example of such approach [101].

The second step after features‘ extraction is the generation of the code book. Clustering is used to group the features in k representatives based on the similarity between them. This code book forms the vocabulary of the visual words found in the dataset. Then, each image is represented by a histogram of visual words (BoF), regardless of their order of occurrence. These BoFs is then used to train and test a classifier based on the images‘ labels. In this work we use BoF to detect the body posture in each frame of the sign video. Then bag of postures (BoP) is used to describe the sign. In our approach, we argue that local features would lead to better recognition rates than global features as it is more robust to occlusion and geometrical transformation. Inspired by speech recognition techniques, in which speech is assumed to be composed of primitive phonemes, we reflect that on sign language to be composed of primitive poses. Thus building a system that employs robust local features to recognize primitive poses is expected to generalize well for unseen signs. Keeping these goals in mind we utilize bag of features (BoF) and Bag of Poses (BoP) in our system in a two stage implementation.

The system can be summarized as follows:

- Firstly, local features are extracted from the training frames of all postures. We tested three techniques for local features extraction, namely SIFT detector and descriptor, Harris Laplace detector with SIFT descriptor, and Dense Scale Invariant Features Transform (DSIFT). The first two techniques detect and then describe interest points while the third describe points in a grid of scale and space. The features, in all cases, are described using SIFT descriptor.

- Secondly, these local features are clustered into K visual words to generate a code book using k-means algorithm.
- Third, an SVM is trained to classify each frame into one of 33 primitive postures.
- Fourth, a signs table is constructed of histogram of primitive poses found in each sign, called Bag of Poses (BoP). The BoP of the sign is then used as a feature vector.

Figure 5.1 illustrates the stages of the features' extraction employed in our approach.

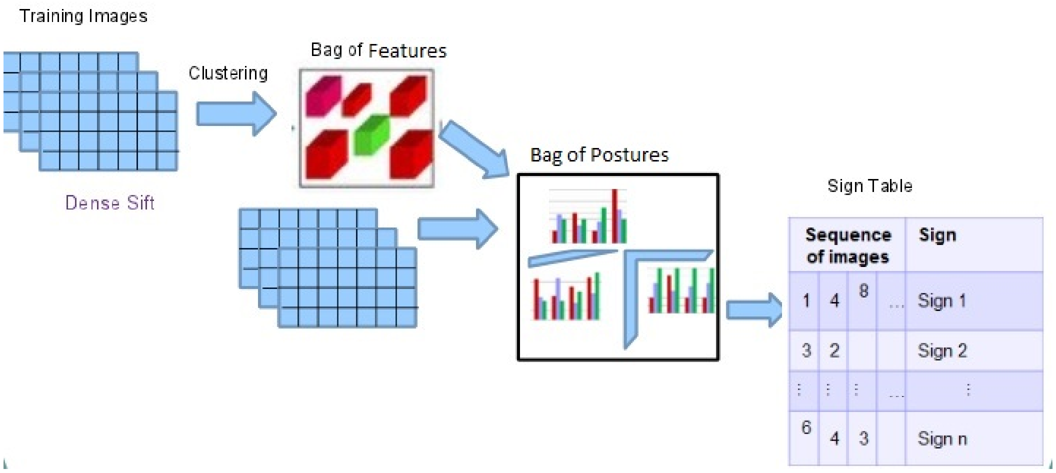


Figure 5.1: BoF and BoP system

For comparison with published work on the same dataset, we reproduced the work done in [3]. They extracted temporal and spacial features. For temporal features the motion is accumulated into one image that describes the activity during the whole sign. This is done using forward accumulated difference of successive frames as follows [3]:

Let $I_{g,i}^{j}$ denotes image index j of the i^{th} repetition of sign g, the forward prediction is given by equation (5.1)

$$P_{g,i} = \sum_{j=1}^{n-1} \delta(I_{g,i}^{j} - I_{g,i}^{j+1}) \tag{5.1}$$

where n is the total number of images and δ is a binary threshold empirically determined as the mean of non zero difference pixels.

Then the resulting image is filtered by median filter which removes the isolated prediction errors as they are assumed to be noise. However this can be due to facial expression but this was out of their scope.

For spacial features extraction they applied 2D DCT transformation on the accumulated temporal differences image, the Zigzag zonal coding is used and 50 descriptors are used as feature. The 50 DCT cutoff was determined empirically by examining different cutoff points. In the experimental results section we will show the effect of the DCT cutoff on the recognition rate. Figure 5.2 shows the accumulated difference image for a sample sign from the used dataset.

5.2 Classification

In the classification stage the extracted features are fed to a classifier. In our approach a two-stage classifier is employed. In the first stage we extracted the

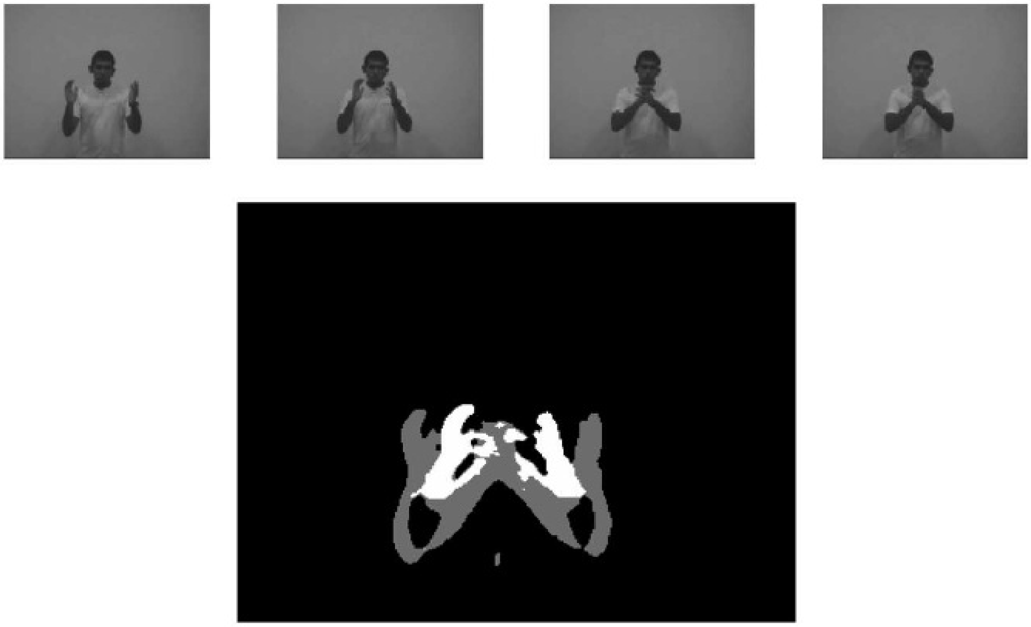


Figure 5.2: Accumulated difference image [3]

BoF of the frames of the test sign. Then SVM is used to classify the posture in each frame. In the second stage we extracted the BoP which are used to classify the signs using K-nearest neighbours (K-NN) classifier. Several experiments were carried out to tune the choice of the suitable number of clusters in generating bag of features (BoF) and the bag of primitive poses (BoP) using the validation data.

5.3 Experimental Results

This section describes the experiments for evaluating the first vision-based recognition system described in section 5.1. Experiments here are done on a data set consisting of 23 signs from commonly used words collected from thee signers. Each signer repeated each sign 50 times [1]. The list of words in this database is shown int Table 5.1. Firstly the results of the accumulated difference method is shown

Table 5.1: List of words in the database [1].

#	Arabic word	English meaning	#	Arabic word	English meaning
1.	صديق	Friend	2.	جار	Neighbor
3.	ضيف	Guest	4.	هدية	Gift
5.	عدو	Enemy	6.	السلام عليكم	Peace upon you
7.	أهلا وسهلا	Welcome	8.	شكرا	Thank you
9.	تفضل	Come in	10.	يشم	Shame
11.	بيت	House	12.	أنا	I / me
13.	يأكل	To eat	14.	ينام	To sleep
15.	يشرب	To drink	16.	يستيقظ	To wake up
17.	يسمع	To listen	18.	يسكت	To stop talking
19.	يشم	To smell	20.	يساعد	To help
21.	أمس	Yesterday	22.	يذهب	To go
23.	يأتي	To come			

followed by those of BoF and BoP.

5.3.1 Accumulated difference approach

To compare our results with the results of [3], the accumulated difference image for each sign is produced using equation (5.1) and the image is filtered then transformed using DCT as illustrated in Section 5.1. The 50 DCT feature vector for 70% of the signs are used as training features and the remaining 30% are used as testing features. The most confused signs in the confusion matrix are 12 with 21 as shown in figure 5.3. In sign 12 the user points to his chest by rotating his hand towards his body while in 21 the signer's hand starts pointing to his chest and then rotated outwards in the opposite direction of the motion in 12. As the system accumulates the differences these two different motions are equivalent and thus confusing. This can be true for any sign with the same motion trajectory and opposite directions. The recognition accuracy is 98.8% for the first signer, 96.8% for the second signer, and 99% for the third signer. These results are comparable to those reported by [3] (2.14% error rate). Figure 5.4 shows the effect of the DCT cutoff on the accuracy. Interestingly, this is found to be signer dependent and cutoff at 50 is found to be at the peak for most of signers. However, a value of 40 to 45 is suitable for all.

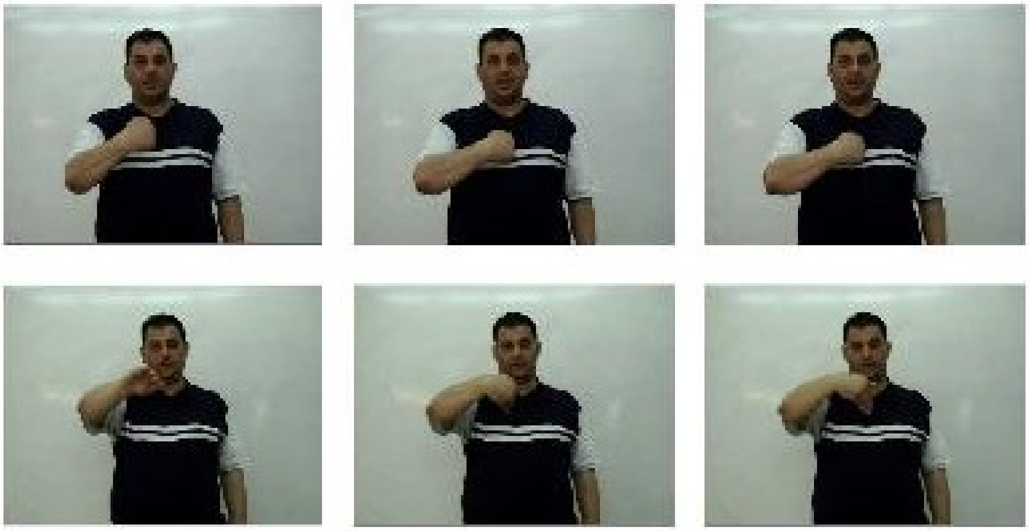


Figure 5.3: The most confusing two signs for DCT approach ; top sign (12) and bottom sign (21).

Testing the DCT approach for signer-independence have shown very poor accuracy 17.67% as the DCT is a global feature.This accuracy is obtained when training using the first signer and testing using the third. It is clear that this method is not suitable for signer independent recognition. In the goal of signer-independent recognition this method is used with some modifications but signers were asked to wear colored gloves [11].

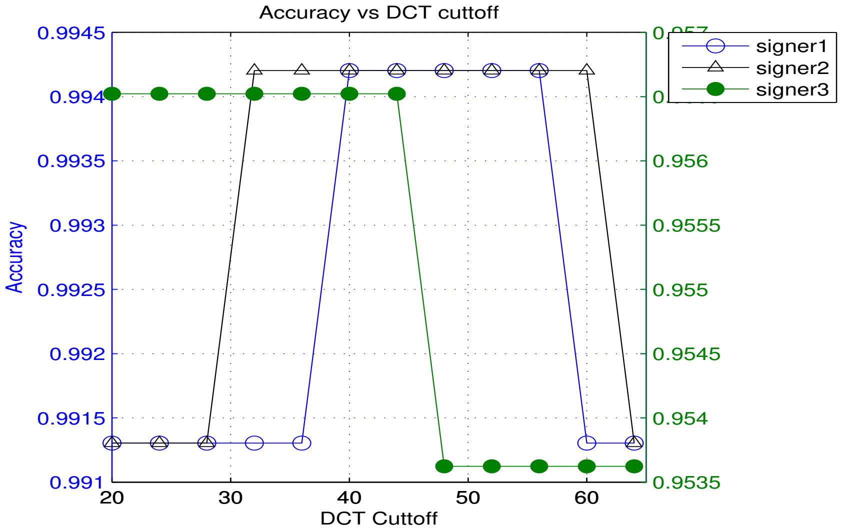


Figure 5.4: The effect of DCT cutoff on the recognition accuracy.

5.3.2 BoF and BoP approach

The second approach involves building a code book using bag of visual words (BoF) and then the bag of poses (BoP) and finally recognize the sign . Firstly local features are collected. In this stage we tested three techniques DSIFT, SIFT, Harris Laplace detector with SIFT descriptor. The difference between interest points detected by each technique is shown in figure 5.5. Interest points detected by SIFT detector shown to be distributed on different locations of the image and many of them are on the background. Harris detector tends to fit well on the signer's body but there are fewer points to catch the details of the body parts. However, DSIFT is able to provide good details as it samples the image on a uniform grid. These differences directly influence the classification accuracy. Accuracies of 99.39%, 97.7%, and 91.5%, are obtained using DSIFT, Harris, and SIFT, respectively.

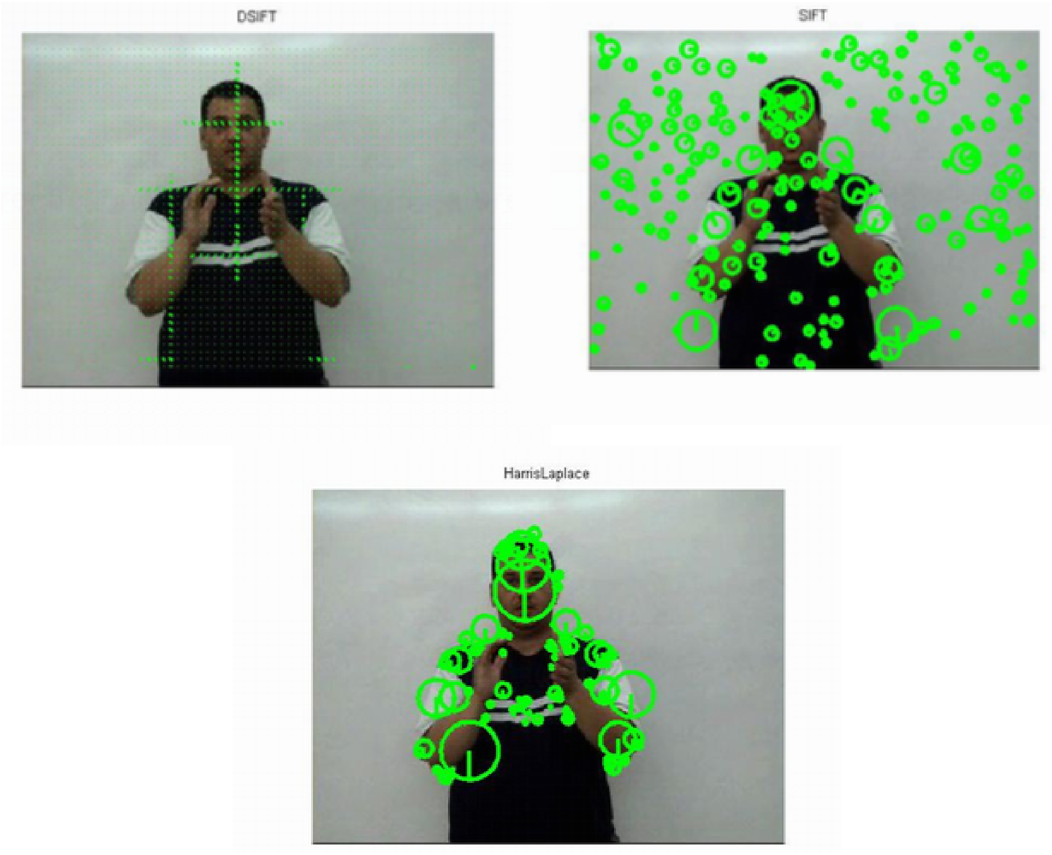


Figure 5.5: The interesting points by each detector top left DSIFT, top right SIFT, bottom Harris.

Then clustering is applied to build a vocabulary of visual words. We experimented with the number of clusters for the code book of visual words using validation data, we tested 100, 200, ..., 2000 and finally settled on 900 visual words which gave the best accuracy as shown in figure 5.6. K-means algorithm is used for clustering using Elkan's algorithm [102] to speed up the clustering process. The code book is then used to describe each posture as a BoF. We trained an SVM to classify 33 primitive postures that we identified in the data set using clustering. These 33 postures are the distinctive postures from which the 23 signs in the data set can be generated. The accuracy of this classifier is 94%. The most

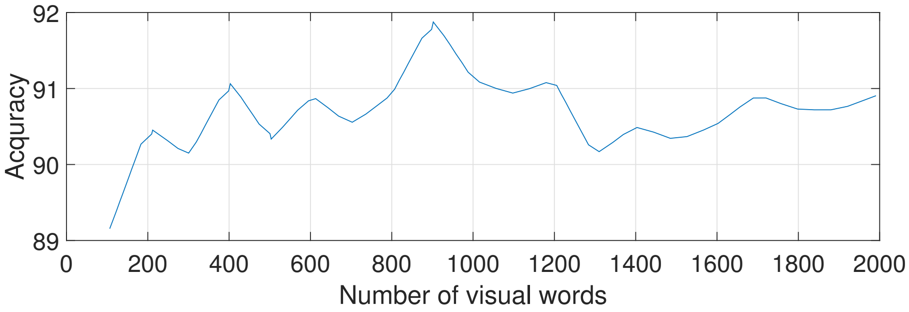


Figure 5.6: The effect of the number visual words on the postures classifier accuracy.

confusing postures are 4 and 6 which are highly similar with a difference of only one hand shape as shown in figure 5.7.

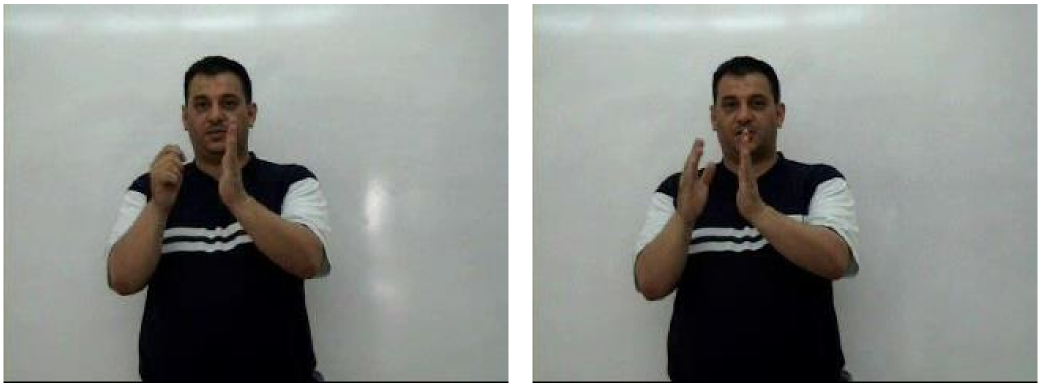


Figure 5.7: The two most confusing postures, left is posture 4, right is posture 6.

The signs‘ table is built of Bag of postures (BoP). We experimented with three types of features to describe the sign, namely, BoP, a concatenation of BoP for each half of the sign sequence, and the normalized concatenation of BoP for each half of the sign sequence. The first builds BoP of the sign which ignores the order of postures. To preserve the order we split the sign sequence into two parts and build BoP for each part, then concatenate these BoPs to form the second type of features. Two parts are enough as most of the signs in the data set are composed of 4 frames or less. We realized that the number of frames in each sample for the same signs is not fixed as it depends on how fast the sign was performed. To form a consistent feature vector we normalized the BoP by the number of frames in the sign sample. An accuracy of 90.47% was obtained when using non-split and non-normalized BoPs. The most confusing signs are 23 and 21. The two signs are composed of the same postures but in revers order. This is confusing as the BoP of both sequences will be the same. Splitting the sign into two parts and extracting the BoP of each part and concatenating the result resolved this problem.

5.3.3 Signer-independent recognition

To test for signer independence several experiments were conducted. First: we build the sign table using signs performed by the first signer and tested by the signs performed by the second signer or vice versa while training the posture classifier by both. In the second scenario we build the sign table using signs performed by the first or second signer and tested using the third which was not used for training of the posture classifier. For the first scenario high accuracy is obtained 91.3% when testing using the second signer (the first is used in training) and when testing using the first (the second is used in training). KNN is used to classify signs, we tested different distance measures: Cosine, Correlation and Euclidean and the best results are obtained when using the Euclidean.

An interesting advantage of our approach over the reproduced one is that it outperforms the accumulated difference when tested in user independent mode.

An accuracy of 45.17% is obtained by our approach when testing the second scenario compared to 17.67% by the accumulated difference. These results support our generalization claim that we stated earlier.

We like to note that the valuable information about the signs is in the head and arms. Hence, in additional experiments we used skin-colour thresholding to keep only face and arms before extracting features. We transformed the RGB image to YCbCr colour space and simple thresholding is applied on the values of Cb and Cr images for each pixel. This affected both of the classification stages. The posture classifier accuracy slightly improved to reach 94.98%. This improved the results for signer independent case. The accuracy jumped to 66.96%. Table 5.2 summarizes the results of the reproduced work in the first row and the proposed one in the second row. The %C and %E columns list the percentage of the correct and error rates respectively.

Table 5.2: Summary of Results

	Signer dependent		Signer independent					
			Scenario 1		Scenario 2		Scenario 2 Skin	
	%C	%E	%C	%E	%C	%E	%C	%E
Acc Diff	98.84	1.16			17.67	82.33		
BoF&BoP	99.39	0.61	91.3	8.7	45.17	54.83	66.96	33.04

Scalability of the system is tested by varying the number of signs included from the database. As shown in figure 5.8 stable and steady accuracy is reported for different vocabulary sizes. This reflects the stability and scalability of our proposed system.

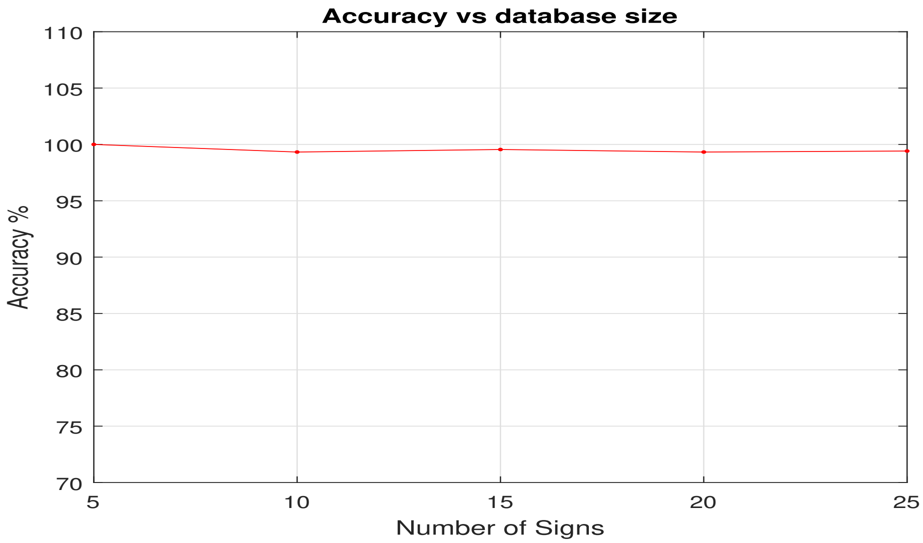


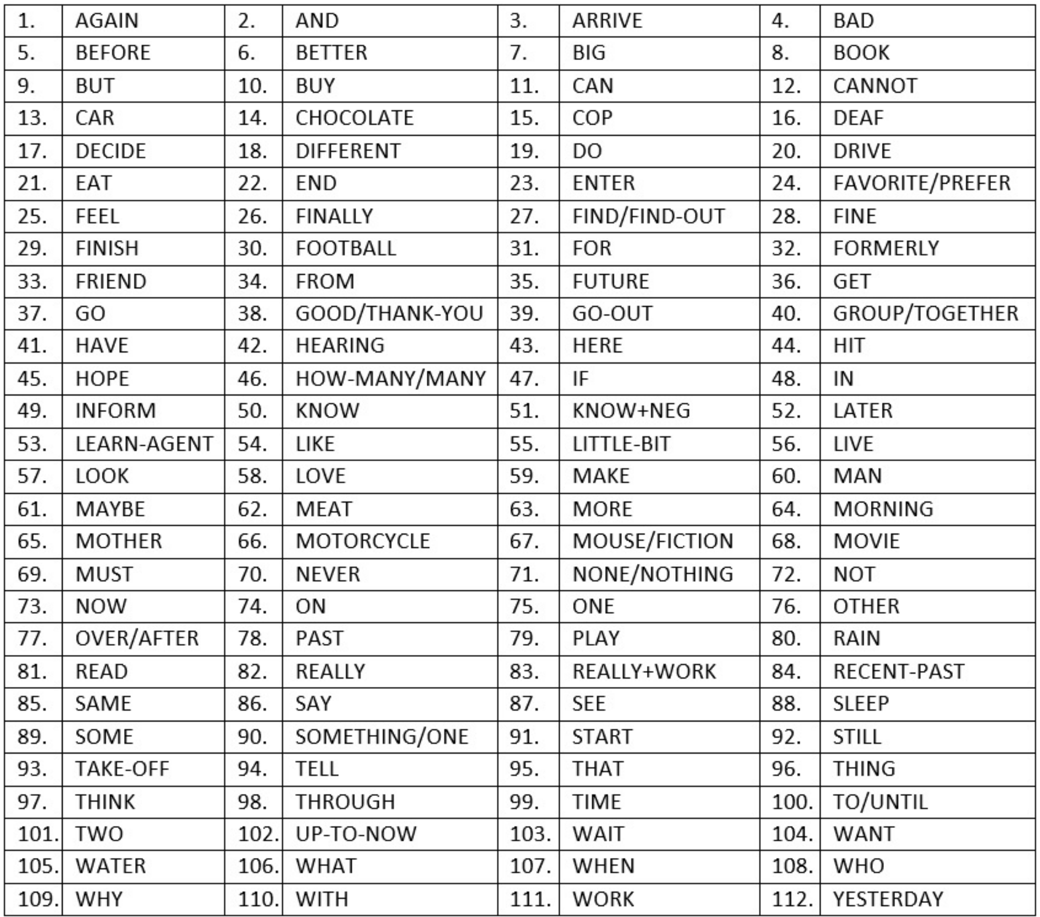
Figure 5.8: Testing the scalability of the system with different vocabulary sizes.

To test the proposed system on a larger database, we downloaded a database of 112 ASL signs from National Center for Sign Language and Gesture Resources site [103, 104]. This database was collected for continuous sign language and linguistic research. We segmented the videos of signs into key still images. The 112 signs were chosen such that each sign has at least 10 samples. This database includes most frequently used hand postures in ASL. Table 5.3 lists the words used in this database. Then we extracted 86 postures found in this database and applied the BoF technique on it. The accuracy of postures classifier on this database is 93.0%.

5.4 Conclusions

In this work, we presented our proposed system for isolated words Arabic sign language recognition. Seeking for features that can generalize well for large vo-

Table 5.3: List of the 112 words in ASL database



1.	AGAIN	2.	AND	3.	ARRIVE	4.	BAD
5.	BEFORE	6.	BETTER	7.	BIG	8.	BOOK
9.	BUT	10.	BUY	11.	CAN	12.	CANNOT
13.	CAR	14.	CHOCOLATE	15.	COP	16.	DEAF
17.	DECIDE	18.	DIFFERENT	19.	DO	20.	DRIVE
21.	EAT	22.	END	23.	ENTER	24.	FAVORITE/PREFER
25.	FEEL	26.	FINALLY	27.	FIND/FIND-OUT	28.	FINE
29.	FINISH	30.	FOOTBALL	31.	FOR	32.	FORMERLY
33.	FRIEND	34.	FROM	35.	FUTURE	36.	GET
37.	GO	38.	GOOD/THANK-YOU	39.	GO-OUT	40.	GROUP/TOGETHER
41.	HAVE	42.	HEARING	43.	HERE	44.	HIT
45.	HOPE	46.	HOW-MANY/MANY	47.	IF	48.	IN
49.	INFORM	50.	KNOW	51.	KNOW+NEG	52.	LATER
53.	LEARN-AGENT	54.	LIKE	55.	LITTLE-BIT	56.	LIVE
57.	LOOK	58.	LOVE	59.	MAKE	60.	MAN
61.	MAYBE	62.	MEAT	63.	MORE	64.	MORNING
65.	MOTHER	66.	MOTORCYCLE	67.	MOUSE/FICTION	68.	MOVIE
69.	MUST	70.	NEVER	71.	NONE/NOTHING	72.	NOT
73.	NOW	74.	ON	75.	ONE	76.	OTHER
77.	OVER/AFTER	78.	PAST	79.	PLAY	80.	RAIN
81.	READ	82.	REALLY	83.	REALLY+WORK	84.	RECENT-PAST
85.	SAME	86.	SAY	87.	SEE	88.	SLEEP
89.	SOME	90.	SOMETHING/ONE	91.	START	92.	STILL
93.	TAKE-OFF	94.	TELL	95.	THAT	96.	THING
97.	THINK	98.	THROUGH	99.	TIME	100.	TO/UNTIL
101.	TWO	102.	UP-TO-NOW	103.	WAIT	104.	WANT
105.	WATER	106.	WHAT	107.	WHEN	108.	WHO
109.	WHY	110.	WITH	111.	WORK	112.	YESTERDAY

cabulary and signer independence, we proposed a two stage classification system which firstly classify each frame into one of the primitive postures that accounts for both hands-shapes and body-lane using BoF. Then in the second stage the sequence of postures is classified into signs using BoP. For postures classification BoF is used with three different local features extraction techniques, namely, SIFT, Harris, and DSIFT. K-means is used for building the vocabulary, and SVM for the classification of the primitive postures using BoF. The second stage encodes the signs as BoP composed of primitive postures and use KNN classifier to classify the signs. We reproduced the results obtained by Shanableh et. al. on a database of 23 Arabic sign language words and proposed application of computer-vision techniques.

The proposed system shows the strength of the local features compared to the global features used in the reproduced work. Our results have shown better accuracy for signer-dependent than the reproduced results. We also showed the advantages of using BoP compared with using the whole sign. It has better generalization towards signer-independence. The proposed approach have shown excellent properties when tested on an ASL database almost 5 times bigger than that of the Arabic database. First, the ratio of number of postures required to generate signs to the number of signs is reduced (33 postures for 23 signs reduced to 86 postures for 112 signs). This helps in simplifying the problem of sign recognition and allow for large vocabulary to be recognized. Second, the extension does not hurt the recognition accuracy.

CHAPTER 6

VISION BASED RECOGNITION USING HOG-HOF FEATURES

The previous chapter shown the strength of local features extraction techniques over the global techniques. Here, we carry on using the local features and propose a recognition system for Arabic sign language using four types of features, namely Modified Fourier Transform, Local Binary Pattern, Histogram of Oriented Gradients, and combination of histogram of oriented gradients and Histogram of Optical Flow. These features are evaluated using Hidden Markov Model on two databases. In addition, we propose an algorithm for segmentation of video streams acquired by Microsoft Kinect V2 into signs. We also propose an algorithm for hand detection in video streams and its detection accuracy is evaluated by measuring the overlap ratio between bounding box generated by the proposed algorithm and one based on the hand joint location obtained by Microsoft Kinect V2. The obtained results show that our algorithm is efficient for hand detection in video streams

6.1 Overview

This technique works for both raw videos and segmented ones. The work being described here is done in two stages. First stage on a segmented images database [1] and it‘s extension [31] by another signer with raw videos. The second stage on a pilot set of KArSL dataset.

The proposed system which consists of two phases: signs’ segmentation and recognition. Signs’ segmentation is performed by monitoring the Optical flow magnitude to mark the starting and ending frames of the sign in each video sample. These videos are then segmented to still images that represent the signs’ samples. For ArSL recognition, we propose using four types of features, namely MFT, LBP, HOG, and HOG-HOF with HMM for classification. We propose combination of motion and appearance modeling in one feature vector using HOF-HOG. This feature encodes both appearance and motion components of the sign without the need to explicitly track the hands.

We used Hidden Markov Model (HMM) for classfication. We evaluated our system on a database consisting of 23 signs and the obtained results show that the MFT and HOG have the highest recognition rates. To test the scalability of our system on larger database, we run experiments on a pilot set from ArSLR. In addition, we propose two algorithms one for segmentation of video streams acquired by Microsoft Kinect V2 into signs and the second for hands' detection in video streams. The detection accurancy of this algorithm is evaluated by measuring the overlap ratio between bounding boxes generated by the proposed algorithm and the hand joint location obtained by Kinect. The obtained results show that our algorithm is efficient for hand detection in video streams.

6.2 Video segmentation

This Stage is needed for the extension of [1] done in [31]. We developed an application for recording color, depth, and skeleton joints using Microsoft‘s Kinect V2. For video segmentation, only color images are employed. The signer repeated each sign 50 times (15 samples per video). These videos are then segmented to still images that represent the signs' samples. The signer makes a sign starting from neutral pose and then returns his hands to the same neutral position before doing another sample. The segmentation algorithm exploits the pauses between the samples to segment them. This is done by monitoring the Optical flow magnitude to mark the starting and ending frames of each video sample. The pseudo code of the segmentation algorithm is shown in Algorithm 3. The first step finds the sum of the significant optical flow magnitudes ignoring small motions in the frame. This sum quantifies how much motion is found in this frame. Based on this quantity, a frame can be either a motion frame or a pause frame. The second and third steps are used to smooth the motion trajectory throughout the video and make it suitable for segmentation using a single threshold value. The forth step calculates this threshold to mark a frame as motion by setting a mask at its index to one or as pause by setting the mask to zero. The transition of mask value from zero to one indicates the starting point of the sign, and from one to zero marks the end of the sign. Figure 6.1 illustrates the segmentation algorithm. In this figure, the x axis is the frame number. The solid curve represents the raw sum of maximal optical flow magnitudes. It is clear that it strongly swings within the single sign. In contrast, the smoothed one (the dashed curve) is more stable and suitable for single thresholding.

Algorithm 3 VideoSegmentation

```
procedure VidSeg(vid)
    for all frames in vid do
        a. Extract the quartile of pixels with high Optical flow magnitude
        b. Trajectory is the sum of the magnitudes of pixels of (a)
    end for
    Convolve the ”sum trajectory” with smoothing filter of length 5
    Remove outliers from the smoothed sum of magnitudes by excluding values greater than the 95th percentile
    The Threshold (T) is equal to the mean of smoothed sums
    Threshold the trajectory (TrajectoryT),
    if $Trajectory(i) > T$ then TrajectoryT(i) = 1
    elseTrajectoryT(i) = 0; where i = 1 to N-frames
    end if
    for all  frame in vid do
        if $TrajectoryT(i) = 0\&TrajectoryT(i+1) = 1$ then Mark the next frame as a starting frame;
        end if
        if $TrajectoryT(i) = 1$ then Save the frame to the sample‘s folder
        end if
    end for
end procedure
```

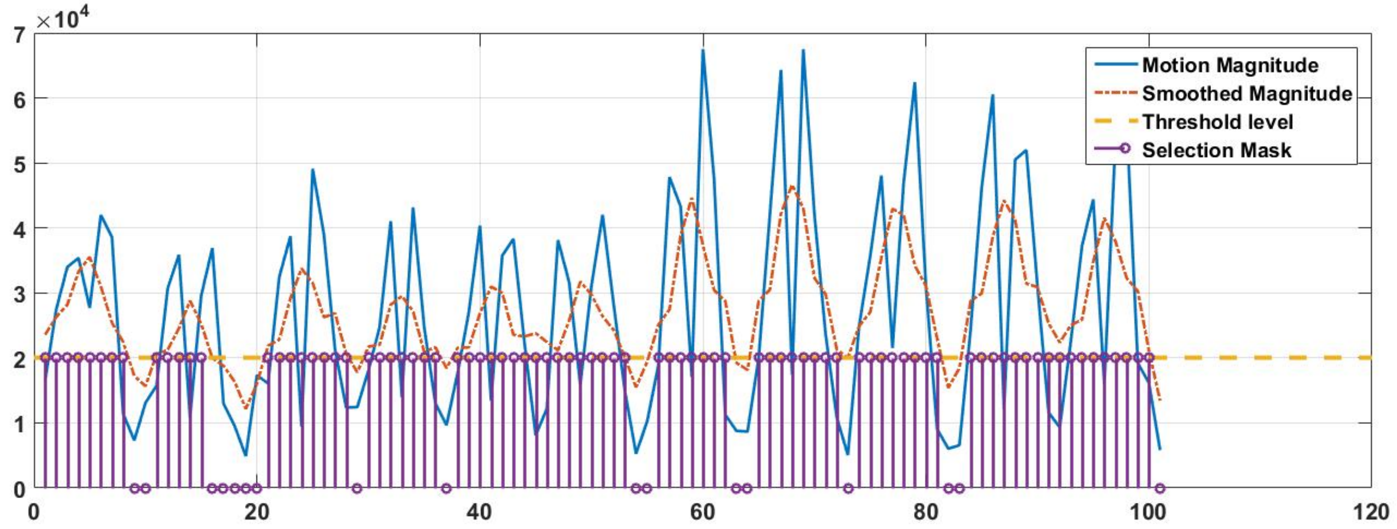


Figure 6.1: segmentation algorithm with boundary of samples indicated by selection mask. Frame index on the x-axis and the total magnitude on the y-axis.

6.3 Arabic Sign Language recognition

The segmented signs are fed into the feature extraction phase where MFT, LBP, HOG, and HOG-HOF features are extracted.

6.3.1 Modified Fourier transform

The Modified Fourier transform (MFT) was first introduced by Mahmoud [105], in motion analysis. It was proved to be more efficient in terms of speed than using the amplitude of the Fourier spectrum. MFT descriptors are computed from the real and imaginary components of the Fourier spectrum by subtracting the imaginary component from the real component ($F^*(k) = F_r(k) - F_i(k)$). MFT is used to extract 63 descriptors of each image that the sign is composed of. The sequence of these vectors is used as a feature vector.

6.3.2 Local Binary Pattern

The Local Binary Pattern (LBP) operator, introduced in 1995 by Ojala et al. [106], is a powerful feature for texture classification. Each pixel in the image is assigned a binary number with LBP operator by thresholding the 3x3 neighbourhood of that pixel with the center value. Then a histogram of the labels (binary numbers) is used as a texture description. In our experiments, each image is segmented into 100 blocks (of 24 * 32 pixels) and LBP is applied on each block and then concatenated to form a feature vector of size 5900.

6.3.3 Histogram of Oriented Gradients

The histogram of oriented gradients (HOG) technique counts occurrences of gradient orientation in connected regions of an image. To compute HOG descriptor, the image is divided into cells, and for the pixels within each cell, a histogram of gradient directions is built. The descriptor is the concatenation of the image cells' histograms. To extract the HOG features, the sign image is divided into 100 cells and a 9 bin histogram of the gradients' orientations in each cell is computed. This histogram is normalized by the intensity of a block of 4 X 4 cells. This normalization results in better invariance to changes in illumination and shadowing. The concatenation of these histograms forms a feature vector of 2946.

6.3.4 Histogram of Optical Flow

Histogram of optical flow (HOF) features are more stable than using raw optical flow magnitudes as features. The raw magnitude is sensitive to the distance between the moving object and the camera. HOF encodes the motion as function of optical flow magnitude and orientation. This is done by accumulating the magnitudes into binned orientation. The histogram is then normalized to make it scale invariant. HOF was successfully applied in human actions recognition [107], but it is not suitable for sign language recognition. That is because hand motion patterns can be common between different signs. This was confirmed by empirical experiments that the most confused signs (15 and 17) share the same motion but with different hand shapes. This is why we use it in addition to other feature.

For each of the features extraction techniques, the sequence of feature vectors extracted from frames that form the sign are used to train HMM classifier. We use HMM implementation of Gesture Recognition Toolkit (GRT) [108] with three states left to right HMM. We tested with more number of states but three-states was enough. GRT provides discrete HMM that expects the features to be of one dimensional and integer valued. So quantization using K-means is applied to shrink the dimensionality to 1. In other words, each frame is having a label. We experimented using different number of clusters and found that 64 clusters works well for both accuracy and training time.

Signs can be viewed as combination of hand-shapes and hand-motion. In another feature (HOG-HOF), we generate a vector that encode both. The hand-motion is represented by a vector of 72 bins weighted histogram of optical flow orientations $(-\pi/2 \backsim \pi/2)$. This captures the general motion direction of hands. The Optical flow velocity is used to weight the 72 directional bins. This histogram is then normalized to sum to 1. For hand-shape we compute HOG in a bounding box around the hand. As the hands' pixels are the most moving parts of the image, we select the region where the velocity is greater than the 95th quantile to be the location of a hand. These regions are variable in size according to the locations of hands. To generate a fixed length feature vector, this region is divided into 4 cells and a 9 bin histogram of the gradients' orientations in each cell is computed. Figure 6.2 shows the region selected when two consecutive frames with two hands moving horizontally towards each other. Computing HOG only in a bounding box

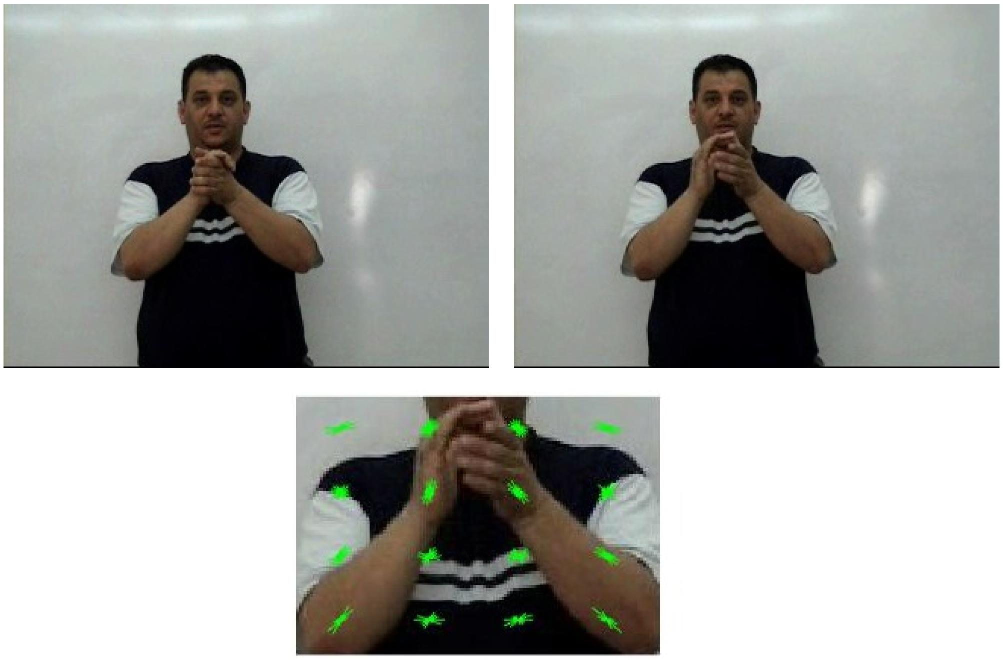


Figure 6.2: Bounding box around the hand: top two consecutive frames, bottom the selected region for HOG features.

around the hands, captures the shape of hands and ignores other parts that are having low contribution to the sign. The concatenation of these two histograms forms a vector of 396 used as a feature with the same HMM settings.

6.4 Experimental Evaluation

In this section we evaluate the proposed vision based recognition system. In which features are extracted from each frame and then HMM is used to classify different signs. Firstly, we evaluate the proposed features extraction techniques on the extended database of [1], follwed by evaluation on a pilot set from KArSL .

6.4.1 Evaluation on Shanableh‘s database

The accuracy of different features is tabulated in Table 6.1. The best performance obtained on a dataset consisting of 23 signs performed by 4 signers is with MFT features. One reason for the high accuracy of MFT may be attributed to its lowest dimensionality which is more suitable for clustering than higher dimensionality. The second best technique is HOG which has been indicated as a good feature descriptor for humans' detection. This indicates that the system is encoding each sign as a sequence of human postures.

Table 6.1: Accuracies for each signer and used features, K is the number of clusters

	MFT K = 33	MFT K = 64	LBP K = 64	HOG K = 64	HOG-HOF K = 64
Signer 1	97.0	99.1	98.5	100	98.5
Signer 2	90.1	99.1	99.1	99.1	96.5
Signer 3	95.9	100	100	100	97.4
Signer 4	95.1	98.25	94.75	95	97.7
Average	94.53	99.11	98.09	98.53	97.53

As it is clear from the table that the number of clusters (K) chosen to quantize the features is a key parameter that affects the overall accuracy. These values were chosen based on our previous research analysis of section 5.3.2 - the BoP approach consists of 33 postures- to provide baseline for tuning the value of K and found that 64 clusters works well in terms of accuracy and training time. Small value of K results in combining discriminative features in one cluster while they should appear in different clusters to help the classification of different signs. A large value of K results in a larger training time for the classifier to merge the over clustered features. The lower accuracy obtained by HOG-HOF can be attributed to 1) the high dimensionality of the vector which requires more training data, 2) the errors resulting in finding the bounding box due to noise that results in higher velocity in regions that does not include the hand. The value of K=33 is reported in Table 6.1 as an example of the tested values of K with the best performing features extraction method (MFT).

6.4.2 Evaluation on KArSL Database

Using another database, we run the proposed system on a subset of 50 signs from the health chapter of KArSL. This dataset is used to test the scalability of the system with larger vocabulary and to evaluate the proposed hand detection algorithm. Algorithm 4 is used to extract the proposed features. The input to this algorithm is a sign video, which means that there is no need to extract key frames from the video as it was needed for the previously used dataset. The algorithm does the extraction of key frames automatically. The algorithm iterates over all frames of the input video and calculates the optical flow on that frame. If the magnitude of optical flow velocity at a pixel exceeds a threshold T_m, the pixel of the binary image at this location is set to one. Otherwise, it is set to zero. The best value for T_m is found empirically to be 4. From this binary image, the algorithm calculates the areas of connected components, and finds the pixels included in an area greater than a threshold T_a if any. The best value of T_a is 3000 pixels. The largest two areas are the locations of hands. Figure 6.3 shows these intermediate images. A frame that pass this test is a key frame and features are extracted from it. The HOG, LBP, and MFT features are extracted as discussed earlier. For HOG-HOF feature two versions are evaluated here. Namely, split (HOG-HOF-S) and merged (HOG-HOF-M). The split one extract the features from two bounding boxes and concatenate to form a features vector. The merge one extracts the features from one bounding box that covers the two hands. If there is only one moving hand and hence one bounding box, the split features are extracted after splitting the bounding box along the longer direction either horizontally or vertically. Since the hands locations are available by Kinect, a HOG-HOF-S feature is extracted from the bounding boxes around the two hands based on this information.

Algorithm 4 Process video and extract features

```
function E(x)tactFeatures(InputVideo)
    for all frames in InputVideo do
        Run optical flow
        $BImage = \begin{cases} 1 & if\ opflowMag > T_m \\ 0, & otherwise \end{cases}$
        Find areas of connected components in BImage
        if exist area $> T_a$ then
            Find Bbox of the largest 2 areas
            Extract LBP, MFT, HOG from this frame
            Extract HOG-HOF from Bbox
            Extract HOG-HOF from Kinect Bbox
        end if
    end for
end function
```

This algorithm is run for each sample in the dataset and the extracted features are used to train and test HMM classifier with the previous settings. Table 6.2 lists

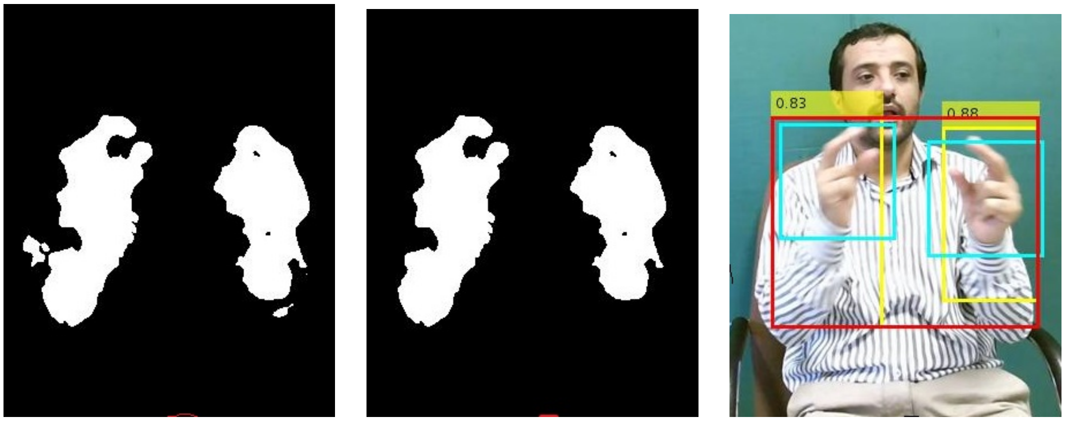


Figure 6.3: The images resulting from different stages of the hand localization algorithm. left binary image of optical flow $> T_m$, middle image of areas $> T_a$, right the bounding boxes: cyan for Kinect bbox, yellow bboxes by the algorithm, red the merged bbox.

the recognition rates for different features extracted by this algorithm. Several important conclusions can be inferred from this table.

First, MFT feature failed to stay on the top of performance. The reason behind that can be detected by checking the most confused signs when using this feature viz signs with ID 119 and 120. In these two signs, the hand shape is the same throughout the sign with only difference in the motion trajectory of the hand. In addition, the place of motion is close to each other. Since MFT is a global feature that cover the whole image, these fine differences cannot be captured by MFT. One solution is to extract MFT features from blocks as described on LBP feature extraction in section 6.3.

Secondly, HOG feature preserve its performance however, PCA is used to reduce the dimensionality by keeping 95% of the variance. In this dataset, the frame size is bigger - which results in longer features vector - and the number of signs is more than the previous dataset. These two factors lead to larger memory

Table 6.2: Accuracies of sign recognition of different features on KArSL database

	Signer 1	Signer 2	Signer 3	Average
MFT	93.10%	97.10%	98.20%	96.13%
HOG	99.30%	99.20%	99.50%	99.33%
LBP	93.00%	96.60%	98.20%	95.93%
HOG-HOF-M	98.30%	99.10%	99.60%	99.00%
HOG-HOF-K	95.60%	96.50%	99.80%	97.30%
HOG-HOF-S	95.80%	98.20%	99.60%	97.87%

requirement for the quantization step. Applying PCA on HOG features reduced the dimensionality from 2916 to 165.

A third observation from this table is that the performance of the split HOG-HOF-S and the Kinect based HOG-HOF-K is relatively similar. This tells that the algorithm is implicitly locating the hands. The slight performance boost obtained by HOG-HOF-S over HOG-HOF-K does not indicate a better hands detection performed by the algorithm than Kinect. Figure 6.4 shows the bounding boxes generated by the algorithm and based on Kinect joints in two frames of a sample video. In this sign, the signer moves his hand towards his hart and leans his head to the right. The number shown on the bounding box is the overlap ratio between the box generated by the algorithm and the box that is based on the Kinect readings. Although the algorithm is not capturing the left hand (because it is not moving), it is capturing a more discriminating features from the head. Note that there is another sign in the database (hart sign) with identical hand shape and motion but with static head.

A forth observation from Table 6.2, the second best performing feature is HOG-HOF-M. In this feature, HOG-HOF features are extracted from a single bounding

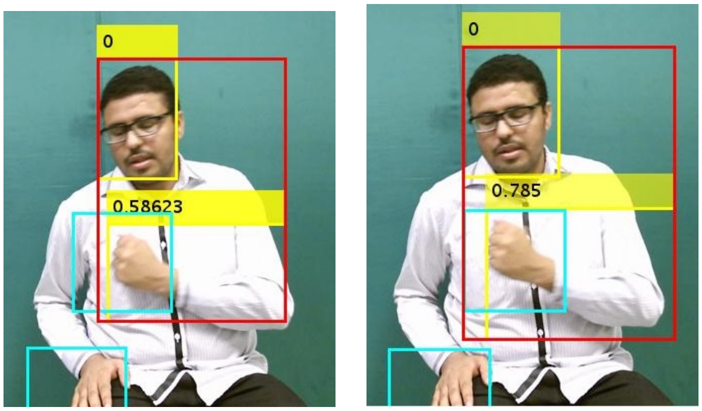


Figure 6.4: Two frames from the "Hart frailer" sign, cyan Kinect based bbox, yellow bboxes by the algorithm, the number shows the overlap ratio with the Kinect bbox, and red is the merged bbox.

box. If the algorithm results in two boxes, the union of these two boxes is used as shown by the red box in Figure 6.4. The better performance of HOG-HOF-M is due to the inclusion of more context when extracting the features, and the lower dimensionality. To evaluate the hand detection performed by the algorithm we exploit the hands' joints locations provided by Kinect. Based on the hand joint location a box of 200x200 pixels centered on the joint location roughly covers the hand. The overlap ratio between this box and the one generated by the algorithm is a good metric to measure the goodness of the hand detection algorithm. The overlap ratio is defined as: -

$$ovr = \frac{area(K \cap B)}{min\ (area(K), area(B))}$$

where K is the Kinect box, B is the box generated by the algorithm. The value of this metric is a fraction between zero and one. The larger the value, the more intersection between the two boxes. As shown in Figure 6.4, the overlap ratio between the two boxes around the right hand is 0.785 while it is 0 for the left hand as the box generated by the algorithm is around the head. To evaluate the hand detector, we find the overlap ratio from all sign samples in the database.

Table 6.3 shows the average overlap ratio for each sign and per signer. The last row shows the average overlap ratio over all signs, it shows that the overlap ratio is more than 80%. However, this is not a precise value because the Kinect based box is not guaranteed to cover the hand and errors in Kinect measures was observed when monitoring several samples. Besides, the precise localization of hand is not necessary for recognition of signs. A live video (slowed down) of the algorithm is publically available online [1] .

[1] https:www.youtube.com/watch?v=Au9hO4SuZeU&feature=youtu.be

Table 6.3: the average overlap ratio for each sign in the database

Overlap ratio			Sign	ID	Overlap ratio			Sign	ID
Signer 3	Signer 2	Signer 1			Signer 3	Signer 2	Signer 1		
0.822	0.776	0.850	pharmacy صيدلية	26	0.944	0.913	0.938	Skeleton هيكل عظمي	1
0.892	0.884	0.921	analysis blood تحليل دم	27	0.857	0.831	0.841	Skull جمجمة	2
0.827	0.858	0.868	examination physical فحص سريري	28	0.777	0.788	0.851	bone Back عمود فقري	3
0.845	0.813	0.740	examination sight فحص النظر	29	0.825	0.851	0.935	Chest قفص صدري	4
0.793	0.861	0.776	thermometer ميزان حرارة	30	0.813	0.793	0.742	Lungs رئتان	5
0.819	0.830	0.779	analysis تحليل طبي	31	0.918	0.861	0.957	exhalation Inspiration شهيق زفير	6
0.815	0.790	0.797	laboratory مختبر	32	0.869	0.818	0.865	system digestive جهاز هضمي	7
0.833	0.831	0.805	inflammation التهاب	33	0.855	0.884	0.852	Face وجه	8
0.825	0.738	0.730	swelling تورم	34	0.888	0.814	0.859	pharynx بلعوم	9
0.740	0.730	0.515	infection عدوى	35	0.741	0.669	0.729	liver كبد	10
0.779	0.799	0.776	fever حمى	36	0.872	0.869	0.901	intestine Small الأمعاء الدقيقة	11
0.744	0.657	0.720	diarrhea إسهال	37	0.836	0.758	0.818	intestine Large الأمعاء الغليظة	12
0.826	0.871	0.886	constipation إمساك	38	0.861	0.796	0.858	appendix Vermfiorm الزائدة الدودية	13
0.873	0.805	0.806	colic مغص	39	0.835	0.882	0.962	heart قلب	14
0.701	0.795	0.765	diabetes مرض السكر سكري	40	0.707	0.568	0.656	senses Five حواس خمس	15
0.829	0.816	0.798	attack heart أزمة قلبية	41	0.793	0.733	0.696	muscle عضلة	16
0.830	0.862	0.842	cancer سرطان	42	0.934	0.804	0.836	tissue أنسجة	17
0.677	0.640	0.719	aids الإيدز	43	0.836	0.765	0.813	hospital مستشفى	18
0.919	0.940	0.856	hair of loss تساقط الشعر	44	0.872	0.887	0.876	aid first إسعافات أولية	19
0.825	0.838	0.832	failure heart سكتة قلبية	45	0.872	0.843	0.885	wound جرح نازف	20
0.663	0.659	0.518	hemiplegia شلل نصفي	46	0.811	0.765	0.838	burning حروق	21
0.836	0.860	0.816	paralysis cerebral شلل دماغي	47	0.926	0.896	0.926	anesthetist مخدر بنج	22
0.853	0.789	0.669	pressure blood ضغط الدم	48	0.836	0.790	0.764	surgery عملية جراحية	23
0.917	0.885	0.853	allergy حساسية	49	0.887	0.847	0.847	dressing شاش ضمادة	24
0.703	0.704	0.679	infection عدوى	50	0.850	0.677	0.868	tape adhesive شريط لاصق بلاستر	25
0.828	0.803	0.808							Average

CHAPTER 7

DEEP LEARNING APPROACH FOR ARSLR

In the previous chapters we've proposed different systems for ArSLR. In these systems, several feature extraction techniques are proposed for representing the signs. In this chapter, we use deep learning to recognize signs. In deep learning, features are learned by the system rather than manually extracted and fed to the system. Deep learning based systems automatically finds the best representation of signs to discriminate different ones. Also, in the previous chapters only one modality is used for recognition, while in this chapter we utilize the skeleton joints and the color video modalities to recognize signs.

As stated in Chapter 2 the main challenge in adopting deep learning to the sign language recognition is the way to handle the time component involved. Opposite to Action recognition where the time component can be less informative than in sign language due to the existence of other context that helps in distinguishing different actions. For example, the playground helps in distinguishing playing tennis from cooking with out the need for temporal information. While in sign language, two signs may have identical fingers configuration but only differ in the motion trajectory. Thus, the temporal information is needed. In this chapter, the proposed system encodes the temporal information of the entire video in a single image that includes the hand shape information and the hand motion information. This compact representation of video makes it easy to adopt the state of the art in deep learning based image classification to sign language recognition.

7.1 Deep Learning approach for ArSLR

In this section we propose machine learning systems to recognize signs using the collected KArSL database. The design goals of such systems includes:

1. A general system that can be used for both static gestures like finger spelling and dynamic gestures that involve hands motion.

2. A system that makes use of the available measures provided by Kinect.

3. A scalable system in terms of vocabulary size and difference of signing styles.

A general pipeline of the proposed systems includes: Preprocessing, Features Extraction, and Classification. One of the most important modalities of sign language is the hand motion pattern. We utilize this feature in the preprocessing stage for all proposed recognitions systems. In the next section we describe the preprocessing stage followed by the proposed recognition systems.

7.1.1 Preprocessing

The database videos were recorded at the rate of 30 frames per second. For a system that processes frames of sign videos to recognize signs, this frame rate results in a large number of frames. This large number of frames need to be reduced to minimize the recognition time. In addition, most of these frames are redundant. For these reasons, the preprocessing stage extracts only key frames from the sign video and discards the remaining frames. To find key frames, we use the hand trajectory recorded by tracking the hand joint of the skeleton data obtained by Kinect. Two steps are used: noise removal followed by key frames' extraction. In noise removal, the noisy measures of hand location is smoothed using median filter to remove outliers. In the second step we treat the hand trajectory as a polygon, and apply a polygonal approximation algorithm to reduce the number of vertices, keeping the most important ones N. The importance of a vertex is quantized by the product of the lengths and angle between the edges at this vertex. This process starts by removing the least important vertix and iteratively recompute the importance of all remaining vertices until N vertices remain. More details are in section 4.2. Since the color, depth, and skeleton frames are recorded in sync, the important N vertices corresponds to the N key frames of the color and depth videos.

7.1.2 Modeling the signs as a sequence of body postures

A sign can be viewed as a sequence of human body postures. Modeling the transition between different postures can be used to recognize different signs. In this technique, we utilize the joints locations of the head, shoulders, elbows, and hands obtained from Kinect to describe the human postures. To obtain the key postures, we follow the same steps as in the preprocessing stage. Then a features vector is built by stretching the X-Y positions of these joints for each key posture as $F_p = \{x_j, y_j\}$ for $j = head, sholder, elbow, hand$ (both right and left). A sequence of N feature vectors is then used to train and test HMM classifier. In another formulation, the features vector is formed by a sequence of hand-shape extracted from the key frames of the color video. The hand shape is represented by Histogram of Oriented Gradients (HOG) features extracted from a bounding box centered at the hand joint location of the frame. Sample bounding boxes are shown in figure 7.1. Then the features sequence is formed by $F_{sh} = \{HOG_i\} for i = 1, 2, 3...N$ and used to train and test HMM classifier. In a third formulation, the features vector is composed by concatenating the postures' features and the hand-shape features as $F_{psh} = \{F_{pk}, HOG_k\} for\, k = 1, 2, 3, ...N$. These features arethen used to train and test the HMM classifier.

7.1.3 Transforming SLR into Image Classification

Here we transform the video of the sign into a single image. By this transformation, the SLR can be viewed as an image classification problem. Then, advanced image classification techniques such as deep learning can be used for SLR. The transformation is achieved by placing frames of the video into a grid as follows. First, the key frames of the sign in hand are extracted as described in the preprocessing stage. Second, a bounding box is centered at the dominant hand joint location on each key frame. The bounding box is not tightly bounding the hand rather, it loosely includes the hand along with some of the surrounding parts of the frame. The extra parts provide useful context about the position of the hand with respect to the signer‘s body. Third, the images inside these boxes are cropped and placed in a grid that forms the final image. Figure 7.1 shows a sample image generated using this procedure. After creating this image, it is labeled using the sign label to be used for training the image classifier.

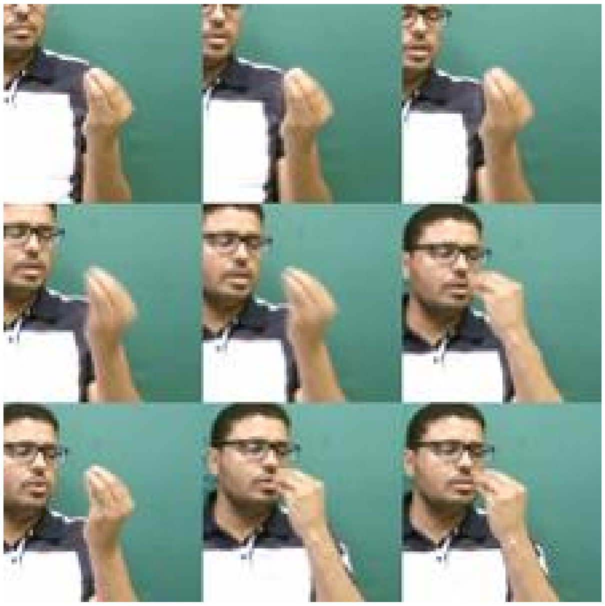


Figure 7.1: A sample Image composed from video of eat sign

7.1.4 Transforming SLR into Image Captioning

In this modeling, the sign video is transformed into a single image following the same procedure of section 7.1.3. The name of each sign is used as a caption for this image. Then automatic captioning system can be used to translate the sign language words. One advantage of this modeling is that a language model can be built over the captioning to generate effective continuous translation system.

7.2 Experimental Work

The following experiments are done to evaluate the performance of the HMM postures sequence. Then the transformation SLR into image classification using deep learning. And then the transformation of SLR into image captioning.

7.2.1 Evaluating the postures sequence modeling of SLR

In this scenario, the sign is preprocessed to extract the key frames. Here we extract 9 key frames from each sign and form the features vectors as described in section 7.1.2. The features are divided into 70% for training and 30% for testing. Using the training data we train an HMM classifier with 3 states left to right. The recognition rate per signer dependent system is shown in table 7.1. From this table,

- The performance of the posture only feature F_p is the worst, however, it is faster to train and test due to its low dimensionality. The low accuracy can be attributed to the fact that some signs are of the same sequence of postures but of different hand fingers configuration.

- The hand shape based features performed the best as it captures the change of fingers‘ configuration along with semantics from the surrounding body parts.

- Combining the two features into one improved the results of postures based features slightly but still less than the hand shape based. This motivates us to rely only on the hand shape and the context around the hand and ignore the postures as the case in the following two systems.

Table 7.1: The recognition accuracies obtained using the postures sequence formulation of SLR

Features	Signer 1	Signer 2	Signer 3
F_p	85.4%	82.6%	85.0%
F_{sh}	89.2%	87.6%	87.4%
F_{psh}	85.9%	83.8%	86.2%

7.2.2 Evaluating the Image classification modeling of SLR

Here, 9 key frames are extracted in the preprocessing stage then a bounding box of 300x300 pixels around the hand joint is used to crop the hand and surrounding context from each key frame. The cropped image is then resized to 75x75 pixels. Then, the 9 cropped images are placed in a grid of 3x3 to form an image of 225x225 pixels as shown in figure 7.1. These images are composed of samples performed by all three signers. We train from scratch a VGG-S network [109] using 80%

of the images from all signers. The training schedule starts by 0.01 learning rate multiplied by 0.1 every 10 training epochs and using batch size of 128 images. The accuracy of this network reached 71.9%. We also fine tuned a VGG-19 pretrained on ImageNet [110]. We fine tuned the network assigning higher learning rate to the last fully connected layer, set global learning rate to 0.0001 for 10 epochs. The recognition rate of this network is 76.1%.

At the time of this writing, only one signer has finished recording the full 502 signs of the KArSL. To get intuition about how good the grid image approach, we run an experiment on the 502 signs using the grid image approach. Since the number of samples per sign is not enough to train a deep network, we adopt a Bag of Features approach. In this experiment, each video is converted into grid image as described in the preprocessing stage. This resulted in 50 images per sign class. Then dense SURF features are extracted from each image. The features are extracted from points on the image separated by 8 pixels horizontally and vertically. And from multi-scale boxes of 32,64,96,128 pixels size around each extraction point. Then, features are clustered into 1000 visual words. Then each class is encoded based on the visual words codebook. The resulting features are then used to train a multiclass linear SVM classifier. 50% of the images are used for training and the remaining are used for testing. The accuracy of this classifier on the entire 502 signs is 98%. Figure 7.2 shows the confusion matrix of this classifier. The least per class recognition rate is 50% for only two classes out of the 502 classes.

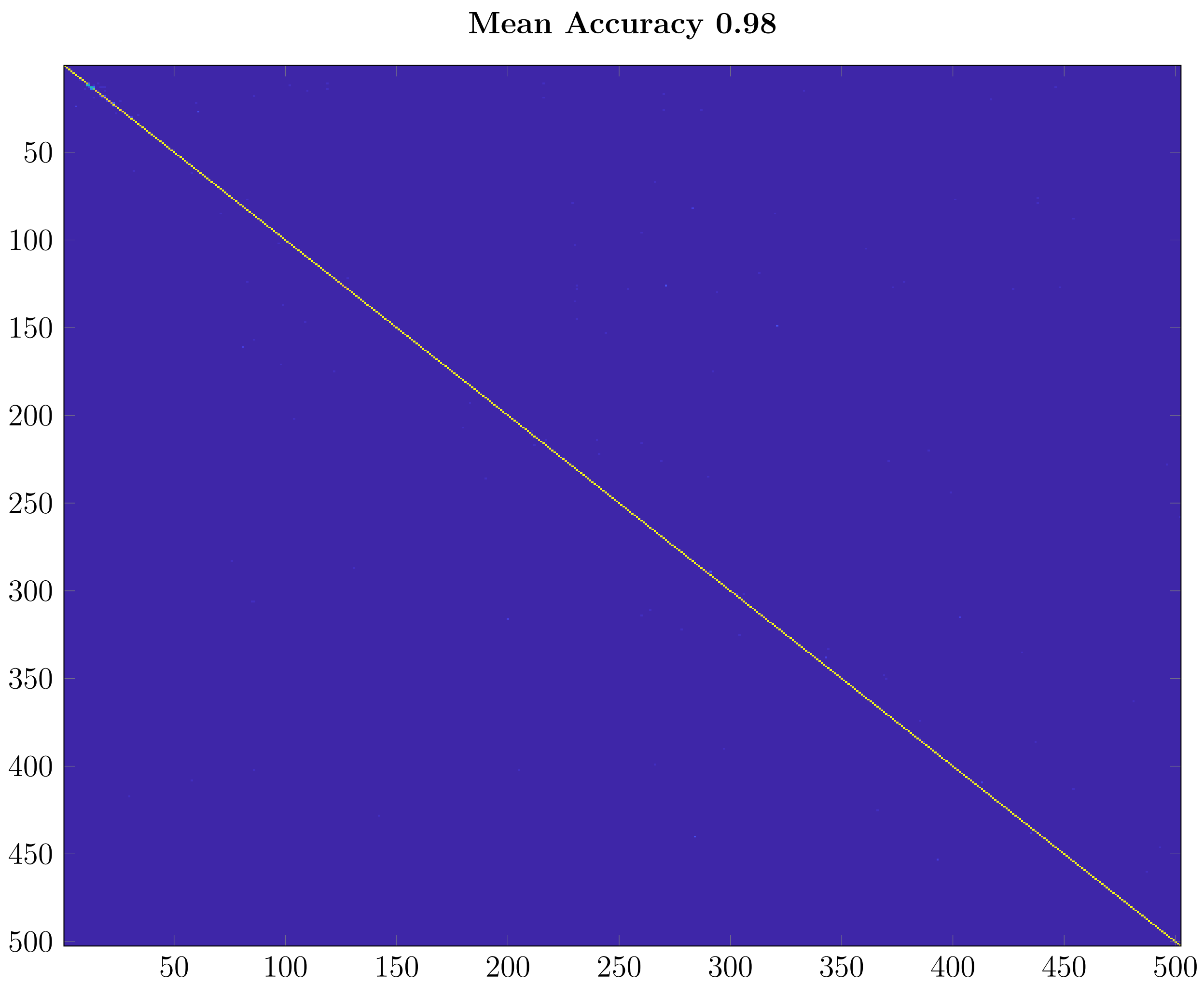


Figure 7.2: Confusion matrix of the 502 classes in KArSL when using BoF to classify the grid image representation

7.2.3 Evaluating the Image captioning modeling of SLR

Here the same images used for image classification modeling are used. The captions are the sign name in words. For example, digit 9 is pronounced as *tise'* in Arabic the caption for the grid image is *tise*. Some captions are composed of two words like *tise meiah* (900). After preparing the captions, we train *Show Attend and Tell* image captioning model [111]. We chose this captioning model because it employs visual attention which is more suitable for the nature of our grid images. We adapted a Tensorflow [112] based implementation [1] to suite our dataset. We use 70% of the dataset for training, 15% for validation, and 15% for testing. Table 7.2 shows the evaluation of the captioning system. Since the captions are maximally two words in length, the higher n-gram (Blue-3, and Blue-4) are not included.

Table 7.2: Measures for evaluating the captioning

Metric	Validation	Test
Bule-1	0.80	0.80
Bule-2:	0.70	0.72
METEOR	0.41	0.40
ROUGE-L	0.82	0.81
CIDEr	3.28	3.73

Sample test images with attention visualized are shown in figure 7.3. The signs shown are very similar with slight difference in hand shapes. The first sign is *thalath* which is the digit 3 with three fingers in static sign. The second sign is *thalathon* (30). It shares the finger configuration with the previous sign with only

[1] available on https://github.com/yunjey/show-attend-and-tell

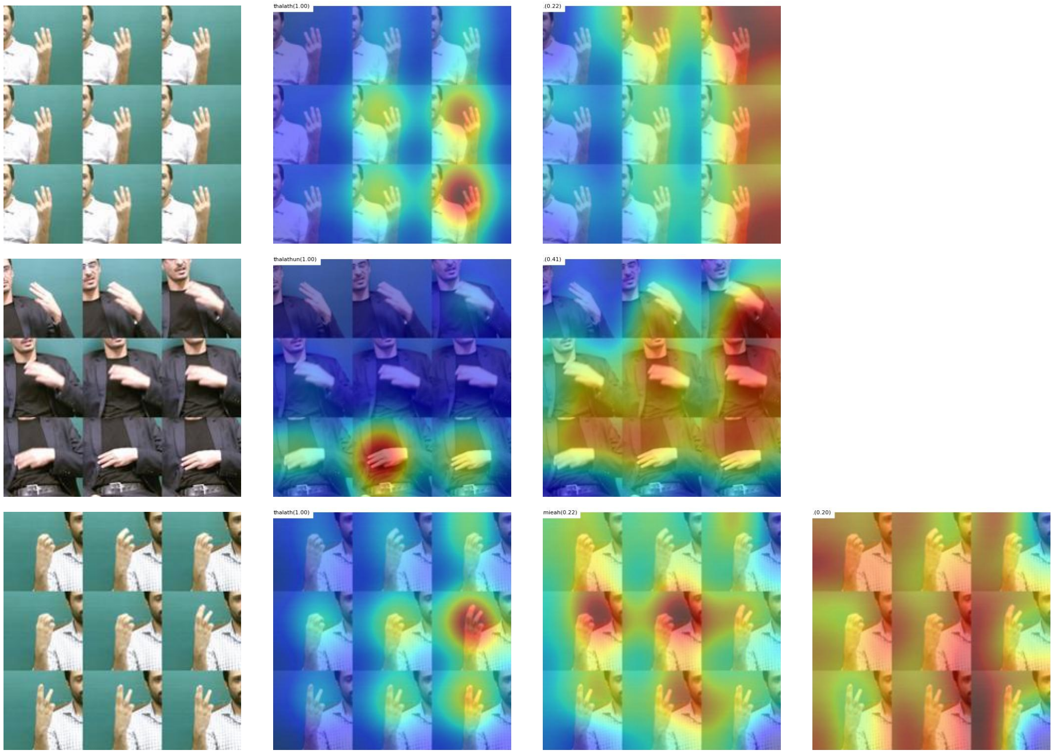


Figure 7.3: Attention sample for signs: three on the first row, thirty on the second, three hundreds on the third.

the hand moves from vertical orientation to horizontal. This is captured by the system as attention at the horizontal orientation. The third sign is *thalth meiah* (300) which shares the fingers configuration with both of the previous signs with only fingers being bent and released. This is also captured by attention on the bending and releasing portions of the grid image. Also as shown in the figure, the system is able to recognize the sign regardless if it is being performed by the right hand or the left.

7.2.4 Signer independent Recognition

In the previous sections the training and testing signs are either performed by one signer or by all signers. In this section, the training samples are performed by two signers and the test signs are performed by the third. Despite the number of signers is not large enough to evaluate signer independent system, we run this test to have intuition about the generalization of the systems. Table 7.3 shows the results of signer independent test. The columns are labeled by the test signer and the other two are the training signers. The last column labeled Gn is a generalization metric computed as $Gn = \frac{Average\, accuracy\, of\, signer\, independent}{Average\, accuracy\, of\, signer\, dependent}$. The value of Gn ranges from 0 to 1 with higher represents better generalization. It can be observed from the table that the generalization of the deep learning approach is better than the traditional HMM based system.

We also run our proposed system on a publicly available database (Chalearn 2011) [113]. The dataset consists of 20 Italian signs performed by 27 signers. We use the training set for training and the validation set for testing. The signers in the validation set never appeared in the training set. Figure 7.4 pictures the application of the preprocessing stage on one video from this database. The accuracies of the proposed systems on this database are shown in the last column in table 7.3. It can be seen from this column how the deep learning approach is able to generalize much better than hand-crafted features. It also shows the effectiveness of the grid image approach although the network is only fine-tuned on one modality (RGB).

Table 7.3: Testing generalization of features for signer independence

	Signer 1	Signer2	Signer 3	Gn	on [113]
F_p	16.0%	08.0%	11.7%	0.14	04.3%
F_{sh}	19.0%	15.0%	17.7%	0.20	25.3%
F_{psh}	15.6%	08.0%	11.6%	0.14	24.0%
VGG-19	28.0%	26.7%	22.2%	0.36	60.0%

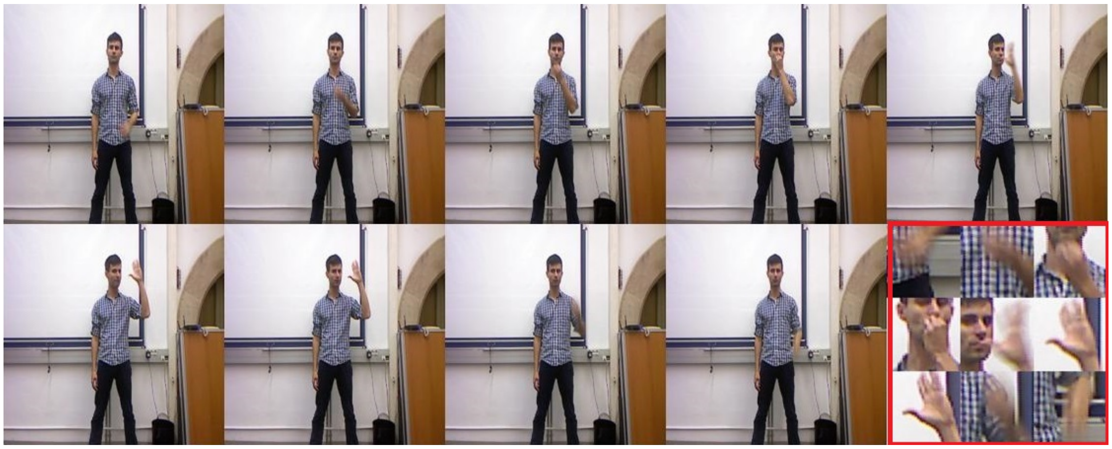


Figure 7.4: Sample Shows the grid image (bottom right) generated from nine key frames of *perfetto* sign

7.3 Conclusions

In This chapter, we've shown how to utilize the available measures provided by the Kinect to model the ArSLR as a sequence of body-postures or a sequence of hand-shapes and use HMM to recognize the sign. We also transform the sign video into a single grid image and train a deep convolutional neural network to classify the grid images and hence recognize the sign. Using the sign name as a caption for the grid image and applying attention based captioning system has shown good discrimination between similar signs. The usefulness of the proposed grid image was shown by the high accuracy obtained on the whole 502 signs in KArSL . We also show the generalization capability of the proposed systems towards signer independent recognition. The proposed recognition systems are general and database independent and can be used for other similar tasks like action recognition and gesture recognition.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

In this chapter, conclusions are drawn on the results obtained in previous chapters. Some future directions are outlined.

8.1 Conclusions

In this thesis we have conducted research on the recognition of Arabic sign language. Arabic sign language recognition enables the automatic translation of Arabic signs into a form of spoken language such as text or speech. The automation of the translation requires building a computerized system that is capable of acquiring signs performed by a signer and then process these signs to produce the corresponding spoken words. To build such system a database of signs and their translation is needed to teach the computer system to translate.

In this thesis, a database of 500 signs is collected using Microsoft Kinect for

Windows V2. The database consists of signs from eleven chapters of ArSL dictionary: numbers, letters, and words from different domains. The database is performed by four professional signers and each sign is repeated fifty times by each signer. For each sample in the database, three channels are synchronously recorded. The channels are color videos, depth map, and skeletal joints locations in 3D and their projection on 2D for both color and depth. These different modalities can be utilized to build recognition system while keeping the natural motion freedom for the signer. The number of samples per sign available in the database makes it suitable for most of computer vision and machine learning algorithms.

The raw recordings from the database requires some processing before being suitable for machine learning algorithms. In this thesis, several preprocessing techniques are proposed.

The first deals with the hand trajectory in 2D/3D smoothing , removes noise, and compress it.

The second segments the video of a sign into a sequence of key frames.

The third locates the moving hands in a video and generates a bounding box around them.

After preprocessing the sign some features are extracted. We model the hand trajectory as a polygon and propose two polygonal description features. The first finds distances from the center of gravity of the polygon to all points on it‘s perimeter. The second uses the location of the points on the perimeter to build a feature vector. These features are then used to train an ensemble classifier. The system shown good performance for both signer dependent and signer independent recognition. The proposed system is tested on two different datasets and compared with published works that use the same dataset and shown better performance than most of them. The proposed system features simplicity, scalability, and generalization to unseen signer.

In a vision based recognition system, we proposed a two stage classification system which firstly classify each frame into one of the primitive postures that accounts for both hands-shapes and body-lane using BoF. Then in the second stage the sequence of postures is classified into signs using BoP. For postures classification BoF is used with three different local features extraction techniques, namely, SIFT, Harris, and DSIFT. K-means is used for building the vocabulary, and SVM for the classification of the primitive postures using BoF. The second stage encodes the signs as BoP composed of primitive postures and use KNN classifier to classify the signs. We have shown the advantages of using local features over global features by comparing the system with published work on the same database. We also showed the advantages of using BoP compared with using the whole sign. It has better generalization towards signer-independence.

Deep learning have made a surge in the recognition capabilities than traditional computer vision algorithms. In this thesis deep learning techniques have been used to recognize signs in two ways. First by converting the sign language recognition task into image classification task. Second by transforming SLR problem into image captioning problem. Both techniques are applied after converting the sign video into a grid image composed of key hand shape frames. We have shown how to train a convolutional neural network from scratch as well as fine-tuning a pretrained network.

The recognition techniques presented in this thesis are general and not limited to Arabic sign language recognition. They can be applied to any other language and any other similar task such as Action Recognition, Gesture Recognition, Robot Control, Human Computer Interaction, and more.

8.2 Future Research Directions

This thesis has proposed techniques for Arabic sign language recognition and contributed to all stages of recognition pipeline. Still some other parts needs more research to reach the satisfying level. These parts includes:

- **Continuous Sentences Recognition:** The continuous recognition is more natural than recognizing words in isolation. The main challenge in continuous recognition is the determination of the boundaries of each word in the signed sentence. Also there is no comprehensive benchmark for continuous Arabic sign language.

- **Online recognition:** Online recognition requires real time recognition of words. It can be easier for isolated words recognition provided that the boundaries of each word is determined by the user. It is more harder when used for continuous signing as it inherits the segmentation difficulties.

- **Non Manual Features:** These are used to modify the meaning of signs and reflect the emotion of signer about the signed sentence. It includes facial expressions like eye gaze, eyebrow rising, mouth deformation, tong movements, air exhausting, and more. The difficulties here includes the lack of annotated database with non-manual features. The number of classes in each component is an open research question. Also, to capture these features the camera needs to be focusing on the face to capture them clearly. Kinct can capture facial landmarks when the face is close to the camera, however the other parts of the body might not be captured. One solution can be to use two Kinects with one focused on the face and the second capturing the other body parts. This solution, requires more powerful machine capable of capturing from two Kinects. Also the synchronization between the two Kinects is challenging. Another solution may employ two machines one for capturing the face landmarks and the other for capturing other body parts. This also requires synchronization between the two machines.

APPENDIX

LIST OF WORDS IN THE KARSL DATABASE

This appendix lists all signs in KArSL database. In Table A.1, The column labeled Arabic lists the Arabic words, the column labeled English contains the transliteration and translation of the sign in English. The transliteration is based on Habash‘s transliteration system [114].

Table A.1: List of signs in KArSL database: the column labeled English contains transliteration and translation of the Arabic word.

Domain	SignID	Arabic	English
Numbers	0001	0	0
	0002	1	1

Continued on next page

Table A.1 – continued from previous page

Domain	SignID	Arabic	English
	0003	2	2
	0004	3	3
	0005	4	4
	0006	5	5
	0007	6	6
	0008	7	7
	0009	8	8
	0010	9	9
	0011	10	10
	0012	20	20
	0013	30	30
	0014	40	40
	0015	50	50
	0016	60	60
	0017	70	70
	0018	80	80
	0019	90	90
	0020	100	100
	0021	200	200

Continued on next page

Table A.1 – continued from previous page

Domain	SignID	Arabic	English
	0022	300	300
	0023	400	400
	0024	500	500
	0025	600	600
	0026	700	700
	0027	800	800
	0028	900	900
	0029	1000	1000
	0030	100000	1000000
	0031	1000000	10000000
Letters	0032	ا	Alif
	0033	ب	Ba
	0034	ت	Taa
	0035	ث	Tha
	0036	ج	Jim
	0037	ح	Ha
	0038	خ	Kha
	0039	د	Dal
	0040	ذ	Dhal

Continued on next page

Table A.1 – continued from previous page

Domain	SignID	Arabic	English
	0041	ر	Ra
	0042	ز	Zayn
	0043	س	Sin
	0044	ش	Shin
	0045	ص	Sad
	0046	ض	Dad
	0047	ط	Ta
	0048	ظ	Za
	0049	ع	Ayn
	0050	غ	Ghayn
	0051	ف	Fa
	0052	ق	Qaf
	0053	ك	Kaf
	0054	ل	Lam
	0055	م	Mim
	0056	ن	Nun
	0057	ه	Ha
	0058	و	Waw
	0059	ي	Ya

Continued on next page

Table A.1 – continued from previous page

Domain	SignID	Arabic	English
	0060	ة	Taa marbutah
	0061	أ	Alif with hamza
	0062	ؤ	Waw with hamza
	0063	ئ	Alif maqsurah with hamza
	0064	ئـ	Hamza on line
	0065	ء	Hamza
	0066	إ	Alif with hamza
	0067	آ	Alif maddah
	0068	ى	Alif maqsurah
	0069	لا	Lam alif
	0070	ال	Al
Health	0071	عظمي هيكل	hykl EZmy Skeleton
	0072	جمجمة	jumjumah Skull
	0073	عمود فقري	Emwd fqry Back bone
	0074	قفص صدري	qfS Sdry Chest
	0075	جهاز تنفسي	jhAz tnfsy Respiratory system
	0076	قصبة هوائية	qSbp hwA}yp Trachea
	0077	رئتان	r}tAn Lungs

Continued on next page

Table A.1 – continued from previous page

Domain	SignID	Arabic	English
	0078	زفير شهيق	$hyq zfyr Inhalation and exhalation
	0079	جهاز هضمي	jhAz hDmy system Digestive
	0080	وجه	wjh Face
	0081	بلعوم	blEwm Pharynx
	0082	كبد	kbd Liver
	0083	البنكرياس	AlbnkryAs Pancreas
	0084	الأمعاء الدقيقة	Al>mEA' Aldqyqp Small intestine
	0085	الأمعاء الغليظة	Al>mEA' AlglyZp Large intestine
	0086	الزائدة الدودية	AlzA}dp Aldwdyp Appendix
	0087	جهاز عصبي	jhAz ESby Nervous system
	0088	قلب	qlb Heart
	0089	حواس خمس	HwAs xms Five senses
	0090	عضلة	EDlp Muscle
	0091	أنسجة	>nsjp Tissue
	0092	مستشفى	mst$fY Hospital
	0093	إسعافات أولية	<sEAfAt >wlyp First aid
	0094	جرح نازف	jrH nAzf Wound
	0095	حروق	Hrwq Burns

Continued on next page

Table A.1 – continued from previous page

Domain	SignID	Arabic	English
	0096	مخدر / بنج	mxdr / bnj Anaesthetic
	0097	عملية جراحية	Emlyp jrAHyp Surgery
	0098	شاش / ضمادة	A / DmAdp Bandage
	0099	شريط لاصق	$ryT lASq Plaster
	0100	صيدلية	Sydlyp Pharmacy
	0101	تحليل دم	tHlyl dm Blood test
	0102	فحص سريري	fHS sryry Clinical examination
	0103	فحص النظر	fHS AlnZr Sight examination
	0104	ميزان حرارة	myzAn HrArp Thermometer
	0105	سماعة أذن	smAEp >*n Stethoscope
	0106	جهاز قياس الضغط	jhAz qyAs AlDgT Sphygmomanometer
	0107	نبض القلب	nbD Alqlb Heart pulse
	0108	تحليل طبي	tHlyl Tby Medical analysis
	0109	مختبر / معمل التحاليل	mEml AltHAlyl / mxtbr Laboratory
	0110	صورة اشعة	Swrp A$Ep Ray image
	0111	التهاب	AlthAb Inflammation
	0112	تورم	twrm Swelling
	0113	زكام	zkAm Cold

Continued on next page

Table A.1 – continued from previous page

Domain	SignID	Arabic	English
	0114	عدوى	EdwY Infection
	0115	صداع	SdAE Headache
	0116	ألم	>lm Pain
	0117	حمى	HmY Fever
	0118	إسهال	<shAl Diarrhea
	0119	إمساك	<msAk Constipation
	0120	مغص	mgS Colic
	0121	مرض سكر / سكري	mrD Alskr / skry Diabetes
	0122	أزمة قلبية	>zmp qlbyp Heart attack
	0123	سرطان	srTAn Cancer
	0124	الإيدز	Al<ydz AIDS
	0125	تساقط الشعر	tsAqT Al$Er Hair loss
	0126	سكتة قلبية	sktp qlbyp Heart failure
	0127	شلل نصفي	$ll nSfy Hemiplegia
	0128	شلل دماغي	$ll dmAgy Brain paralysis
	0129	ضغط الدم	DgT Aldm Blood pressure
	0130	حساسية	HsAsyp Allergy
	0131	هرش / حكة	Hkp / hr$ Itch
	0132	دواء	dwA’ Medicine

Continued on next page

Table A.1 – continued from previous page

Domain	SignID	Arabic	English
	0133	دورة شهرية	dwrp $hryp Menstrual period
	0134	مرض / مريض	mryD / mrD Patient / Disease
	0135	كبسولة	kbswlp Capsule
	0136	دواء شراب	dwA' $rAb Liquid medicine
	0137	مرهم	mrhm Ointment
	0138	قطارة	qTArp Dropper
	0139	أخذ إبرة	>x* <brp Take a needle
	0140	تلقيح	tlqyH Vaccination
	0141	تطعيم	tTEym Inoculation
	0142	أشعة ليزر	>$Ep lyzr X-ray
	0143	مخدرات	mxdrAt Drugs
	0144	إدمان	<dmAn Addiction
	10045	توحد	twHd Autism
	0146	منغولي	mngwly Down's syndrome
	0147	بكتريا	bktryA Bacteria
	0148	جرثومة	jrvwmp Germ
	0149	فيروس	fyrws Virus
	0150	إنتشار	<nt$Ar Spread
	0151	إعاقة	<EAqp Disability

Continued on next page

Table A.1 – continued from previous page

Domain	SignID	Arabic	English
	0152	إعاقة ذهنية	<EAqp *hnyp Intellectual disability
	0153	اعاقة جسدية	AEAqp jsdyp Physical disability
	0154	اعاقة بصرية	AEAqp bSryp Visual impairment
	0155	إعاقة سمعية	<EAqp smEyp Hearing impairment
	0156	وباء	wbA' Epidemic
	0157	مناعة	mnAEp Immunity
	0158	عصب	ESb Nerve
	0159	معافى	mEAfY Healthy
Common verbs	0160	يأكل	y>kl Eat
	0161	يشرب	y$rb Drink
	0162	ينام	ynAm Sleep
	0163	يستيقظ	ystyqZ Wake up
	0164	يسمع	ysmE Hear
	0165	يسكت	yskt Silence
	0166	يشم	y$m Smell
	0167	يصعد	ySEd Go up
	0168	ينزل	ynzl Go down

Continued on next page

Table A.1 – continued from previous page

Domain	SignID	Arabic	English
	0169	يفتح	yftH Open
	0170	يغلق	yglq Close
	0171	يبني	ybny Build
	0172	يكسر	yksr Break
	0173	يمشي	ym$y Walk
	0174	يحب	yHb Love
	0175	يكره	ykrh Hate
	0176	يشوي	y$wy Grill
	0177	يحرث	yHrv Plow
	0178	يزرع	yzrE Plant
	0179	يسقي	ysqy Irrigate
	0180	يحصد	yHSd Harvest
	0181	يفكر	yfkr Think
	0182	يساعد	ysAEd Help
	0183	يدخن	ydxn Smoke
	0184	يدعم	ydEm Support
	0185	يختار	yxtAr Choose
	0186	ينادي	ynAdy Call
	0187	يتنامى	ytnAmY Grow

Continued on next page

Table A.1 – continued from previous page

Domain	SignID	Arabic	English
	0188	يصبغ	ySbg Dye
	0189	يقف	yqf Stop
	0190	يستحم	ystHm Takes a shower
	0191	يدخل	ydxl Enter
Family	0192	أسرة	>srp family
	0193	جدة	jdp Grandmother
	0194	جد	jd Grandfather
	0195	أب	>b Father
	0196	أم	>m Mother
	0197	أخت	>xt Sister
	0198	أخ	>x Brother
	0199	بنت	bnt Girl
	0200	رضيع	rDyE Baby
	0201	توأم	tw>m Twin
	0202	رجل	rjl Man
	0203	شاب	$Ab Young man
	0204	شابة	$Abp Young woman
	0205	حفيد	Hfyd Grandchild
	0206	زواج	zwAj Wedding

Continued on next page

Table A.1 – continued from previous page

Domain	SignID	Arabic	English
	0207	حمل	Hml Pregnant
	0208	ولادة	wlAdp Childbirth
	0209	عم	Em Uncle
	0210	عمة	Emp Aunt
	0211	خال	xAl Maternal uncle
	0212	خالة	xAlp Maternal aunt
	0213	ابن الأخ	Abn Al>x Nephew
	0214	ابن الأخت	Abn Al>xt Nephew
	0215	ابن العم	Abn AlEm Cousin
	0216	ابن	Abn Son
	0217	ابنة	Abnp Daughter
	0218	ناس	nAs People
	0219	طلاق	TlAq Divorce
	0220	خطوبة	xTwbp Engagement
	0221	حفلة	Hflp Party
	0222	وفاة	wfAp Death
	0223	طفل	Tfl Child
Characteristics	0224	جميل	jmyl Beautiful
	0225	قبيح	qbyH Ugly

Continued on next page

Table A.1 – continued from previous page

Domain	SignID	Arabic	English
	0226	طويل	Twyl Long
	0227	قصير	qSyr Short
	0228	نحيف	nHyf Thin
	0229	سمين	smyn Fat
	0230	غني	gny Rich
	0231	فقير	fqyr Poor
	0232	محبط	mHbT Offended
	0233	مشمئز	m$m}z Disgusted
	0234	مرتبك	mrtbk Confused
	0235	قلق	qlq Worried
	0236	مشوش	m$w$ Deranged
	0237	خائف	xA}f Afraid
	0238	سعيد	sEyd Happy
	0239	حزين	Hzyn Sad
	0240	شجاع	$jAE Courageous
	0241	جبان	jbAn Coward
	0242	طموح	TmwH Ambitious
	0243	معجب	mEjb Admirable
	0244	غائر من	gA}r mn To be jealous of

Continued on next page

Table A.1 – continued from previous page

Domain	SignID	Arabic	English
	0245	غائر على	gA}r ElY To protect jealously
	0246	ودود	wdwd Affectionate
	0247	كريم	krym Generous
	0248	بخيل	bxyl Miserly
	0249	طماع	TmAE Greedy
	0250	كذاب	k*Ab Liar
	0251	أناني	>nAny Selfish
	0252	متكبر	mtkbr Proud
	0253	متواضع	mtwADE Humble
	0254	شعور	$Ewr Feeling
	0255	تعب	tEb Tired
	0256	بكاء	bkA' Crying
	0257	احتقار	AHtqAr Contempt
	0258	إعتماد على الذات	AEtmAd ElY Al*At Self dependence
	0259	خفيف	xfyf Light
	0260	ثقيل	vqyl Heavy
	0261	قديم	qdym Old
	0262	حسد	Hsd Envy

Continued on next page

Table A.1 – continued from previous page

Domain	SignID	Arabic	English
	0263	صدق	Sdq Believe
	0264	غدر	gdr Betrayal
	0265	صبر	Sbr Patient
	0266	لوم	lwm Blame
	0267	الحق	AlHq Truth
	0268	خيانة	xyAnp Treason
	0269	إيثار	<yvAr Preference
	0270	تضحية	tDHyp Self sacrifice
	0271	شفقة	$fqp Pity
	0272	ذكي	*ky Intelligent
Directions	0273	أمام	>mAm In front of
	0274	بجانب	bjAnb Side
	0275	بعيد	bEyd Far
	0276	بين	byn Between
	0277	تحت	tHt Under
	0278	حول	Hwl Around
	0279	خارج	xArj Outside
	0280	خلف	xlf Back
	0281	داخل	dAxl Inside

Continued on next page

Table A.1 – continued from previous page

Domain	SignID	Arabic	English
	0282	فوق	fwq Up
	0283	قريب	qryb Near
	0284	من خلال	mn xlAl Through
	0285	هنا	hnA Here
	0286	هناك	hnAk There
	0287	يسار	ysAr Left
	0288	يمين	ymyn Right
Social relationships	0289	أهلا وسهلاً	>hlA wshlA Welcome
	0290	السلام عليكم	AlslAm Elykm Greeting
	0291	تفضل	tfDl Here you are
	0292	جار	jAr Neighbor
	0293	شكراً	$krA Thanks
	0294	صديق	Sdyq Friend
	0295	ضيف	Dyf Guest
	0296	عدو	Edw Enemy
	0297	عيب	Eyb Fault
	0298	هدية	hdyp Gift
Home	0299	بيت	byt House
	0300	جدار	jdAr Wall

Continued on next page

Table A.1 – continued from previous page

Domain	SignID	Arabic	English
	0301	سقف	sqf Roof
	0302	باب	bAb Door
	0303	شباك	$bAk Window
	0304	غرفة	grfp Room
	0305	غرفة نوم	grfp nwm Bedroom
	0306	سرير	sryr Bed
	0307	مرتبة سرير	mrtbp sryr Mattress
	0308	شرشف	rf Bedsheet
	0309	وسادة	wsAdp Pillow
	0310	شماعة	$mAEp Clothes rail
	0311	خزانة ملابس	xzAnp mlAbs Wardrobe
	0312	مطبخ	mTbx Kitchen
	0313	فرن غازي	frn gAzy Gas Oven
	0314	بوتاغاز	bwtAgAz Cooker
	0315	طبق	Tbq Dish
	0316	سكين	skyn Knife
	0317	شوكة	$wkp Fork
	0318	ملعقة	mlEqp Spoon
	0319	فنجان	fnjAn Cupful

Continued on next page

Table A.1 – continued from previous page

Domain	SignID	Arabic	English
	0320	مجمد	mjmd Freezer
	0321	كأس	k>s Cup
	0322	ترمس	trms Thermos
	0323	غرفة طعام	grfp TEAm Food room
	0324	إبريق	<bryq Pitcher
	0325	طاولة	TAwlp Table
	0326	دلة	dlp Dalah
	0327	حمام	HmAm Bathroom
	0328	كرسي	krsy Chair
	0329	غسالة	gsAlp Washing machine
	0330	منشفة	mn$fp Towel
	0331	سجادة	sjAdp Carpet
	0332	موكيت	mwkyt Moquette
	0333	ثريا	vryA Chandelier
	0334	مسجل	msjl Cassette
	0335	شريط كاست	$ryT kAst Cassette tape
	0336	تلفزيون	tlfzywn Television
	0337	دش	d$ Satellite
	0338	شريط فيديو	$ryT fydyw Video tape

Continued on next page

Table A.1 – continued from previous page

Domain	SignID	Arabic	English
	0339	كاميرا فيديو	kAmyrA fydyw Video camera
	0340	كاميرا فوتوغرافية	kAmyrA fwtwgrAfyp Photo camera
	0341	تلفون	tlfwn Telephone
	0342	غرفة ضيوف	grfp Dywf Guests room
	0343	فيديو	fydyw Video
	0344	مبخرة	mbxrp Censer
	0345	مدفأة	mdf>p Heater
	0346	مفتاح	mftAH Key
	0347	مروحة	mrwHp Fan
	0348	تدفئة مركزية	tdf}p mrkzyp Central heater
	0349	كهرباء	khrbA' Electrics
	0350	مكيف	mkyf Air conditioner
	0351	سلك كهربائي	slk khrbA}y Wire
	0352	فيش	fy$ Plug
	0353	حقيبة سفر	Hqybp sfr Bag
	0354	مكواة	mkwAp Iron
	0355	ميزان حرارة جوي	myzAn HrArp jwy Thermometer
Religion	0356	الله تعالى	Allh tEAlY God

Continued on next page

Table A.1 – continued from previous page

Domain	SignID	Arabic	English
	0357	محمد رسول الله	mHmd rswl Allh Prophet Mohammed
	0358	عيسى المسيح	EysY AlmsyH Jesus christ
	0359	القرآن الكريم	Alqr|n Alkrym The holy Quran
	0360	الخلفاء الراشدون	AlxlfA' AlrA$dwn The caliphs
	0361	السنة النبوية	Alsnp Alnbwyp Sunnah
	0362	أركان الإسلام	>rkAn Al<slAm Pillars of Islam
	0363	أركان الإيمان	>rkAn Al<ymAn Pillars of faith
	0364	الشهادتين	Al$hAdtyn Two declarations of faith
	0365	الصلاة	AlSlAp prayer
	0366	الزكاة	AlzkAp Alms
	0367	الصوم	AlSwm Fasting
	0368	العمرة	AlEmrp Umrah
	0369	الحج	AlHj Pilgrimage
	0370	ملائكة	mlA}kp Angels
	0371	رسول	rswl Prophet
	0372	يوم القيامة	ywm AlqyAmp Doomsday
	0373	القضاء والقدر	AlqDA' wAlqdr Fate and destiny

Continued on next page

Table A.1 – continued from previous page

Domain	SignID	Arabic	English
	0374	أركان الصلاة	>rkAn AlSlAp The pillars of prayer
	0375	خير	xyr Good
	0376	النية	Alnyp Intention
	0377	شر	$r Evil
	0378	طهارة	ThArp Purity
	0379	مبطلات الصلاة	mbTlAt AlSlAp Nullifications of prayer
	0380	تيمم	tymm Tayammum
	0381	وضوء / يتوضأ	ytwD> / wDw' Ablution
	0382	يؤذن	y&*n Call to prayer
	0383	مسح على الخفين	msH ElY Alxfyn Wiping over shoes
	0384	صلاة الفجر	SlAp Alfjr Al-fajr prayer
	0385	صلاة الظهر	SlAp AlZhr Dhuher prayer
	0386	صلاة العصر	SlAp AlESr Asr prayer
	0387	صلاة المغرب	SlAp Almgrb Maghreb prayer
	0388	صلاة العشاء	SlAp AlE$A' Isha prayer
	0389	خطبة	xTbp sermon
	0390	خطبة الجمعة	xTbp AljmEp Friday sermon

Continued on next page

Table A.1 – continued from previous page

Domain	SignID	Arabic	English
	0391	خطيب	xTyb Fiance
	0392	خطبة العيد	xTbp AlEyd Eid sermon
	0393	مسجد	msjd Mosque
	0394	سورة	swrp Quran chapter
	0395	إمام	<mAm Leader
	0396	سورة الفاتحة	swrp AlfAtHp Al-Fatihahsurah
	0397	آية	|yp Verse
	0398	ليلة القدر	lylp Alqdr Alqadir night
	0399	عيد	Eyd Feast
	0400	عيد الفطر	Eyd AlfTr Eid Al Fitr
	0401	عيد الأضحى	Eyd Al>DHY Eid al-Adha
	0402	إحرام	<HrAm Ihram
	0403	مكة	mkp Makkah
	0404	الكعبة	AlkEbp Kaaba
	0405	المدينة المنورة	Almdynp Almnwrp Al-madina Al-monawara
	0406	منى	mnY Mona
	0407	جبل عرفات	jbl ErfAt Arafat Mountain
	0408	الصفا والمروة	AlSfA wAlmrwp Safa and Marwa

Continued on next page

Table A.1 – continued from previous page

Domain	SignID	Arabic	English
	0409	مزدلفة	mzdlfp Muzdalifah
	0410	رمي الجمرات	rmy AljmrAt Throwing pebbles
	0411	طواف	TwAf Circling
	0412	ماء زمزم	mA' zmzm Zamzam water
	0413	الأضحية	Al>DHyp Sacrifice
	0414	فك الإحرام	fk Al<HrAm Removing Ihraam
	0415	طواف الوداع	TwAf AlwdAE Leaving circling
	0416	المسجد الأقصى	Almsjd Al>qSY Al-Aqsa Mosque
	0417	يسبح	ysbH Glorifies
	0418	الحمد لله	AlHmd llh Praise be to Allaah
	0419	سبحان الله	sbHAn Allh Glory be to Allah
	0420	إيمان	<ymAn Faith
	0421	الشرك بالله	Al$rk bAllh Polytheism
	0422	أعوذ بالله	>Ew* bAllh I seek refuge in God
	0423	شيطان	$yTAn Demon
	0424	جن	jn Demons
	0425	الضالون	AlDAlwn The lost
	0426	المغضوب عليهم	AlmgDwb Elyhm Angry at them
	0427	صنم	Snm Fetish

Continued on next page

Table A.1 – continued from previous page

Domain	SignID	Arabic	English
	0428	زنى	znY Adultery
	0429	الجنة	Aljnp Paradise
	0430	النار	AlnAr Fire
	0431	روح	rwH Soul
	0432	حقوق	Hqwq Rights
	0433	واجبات	wAjbAt Duties
	0434	حسنات	HsnAt Good deads
	0435	سيئات	sy}At Sins
	0436	حلال	HlAl Permitted
	0437	حرام	HrAm Forbidden
	0438	مغفرة	mgfrp Forgiveness
	0439	دين	dyn Religion
	0440	مسيحي	msyHy Christian
	0441	يهودي	yhwdy Jewish
	0442	كنيسة	knysp Church
	0443	إسراء ومعراج	<srA' wmErAj Isra and Maraj
	0444	حواء	HwA' Eve
	0445	آدم	|dm Adam
	0446	صدقة	Sdqp Charity

Continued on next page

Table A.1 – continued from previous page

Domain	SignID	Arabic	English
	0447	نبي	nby Prophet
	0448	هداية	hdAyp Guidance
	0449	أمانة	>mAnp Honesty
	0450	شهيد	$hyd Martyr
	0451	حقوق الوالدين	Hqwq AlwAldyn Parental rights
	0452	عقوق الوالدين	Eqwq AlwAldyn Disobedience to parents
	0453	مسلم	mslm Muslim
	0454	رحمة	rHmp Mercy
	0455	عيد رأس السنة	Eyd r>s Alsnp New Year's Holiday
	0456	فروض	frwD Assignments
	0457	قبة الصخرة	qbp AlSxrp Dome of the Rock
	0458	خلق	xlq Create
Jobs	0459	مهندس	mhnds Engineer
	0460	مصور فوتوغرافي	mSwr fwtwgrAfy Photographer
	0461	جزار	jzAr Meat man
	0462	سائق	sA}q Driver
	0463	صائغ	SA}g Jeweler
	0464	خادم	xAdm Servant

Continued on next page

Table A.1 – continued from previous page

Domain	SignID	Arabic	English
	0465	رئيس قسم	r}ys qsm Chairman
	0466	حداد	HdAd Blacksmith
	0467	لحام كهربائي	lHAm khrbA}y Welding
	0468	مترجم لغة الإشارة	mtrjm lgp Al<$Arp Sings language translator
	0469	مذيع	m*yE Announcer
	0470	مترجم لغات	mtrjm lgAt Translator
	0471	مدير	mdyr Manager
	0472	سفير	sfyr Ambassador
	0473	وزير	wzyr Minister
	0474	سلطان / ملك	mlk / slTAn King
	0475	رئيس الجمهورية	r}ys Aljmhwryp Premier minister
	0476	شيخ	$yx Sheikh
	0477	محافظ	mHAfZ Governor
	0478	ولي عهد / وكيل وزارة	wly Ehd / wkyl wzArp Under secretary of state
	0479	رئيس مجلس النواب	r}ys mjls AlnwAb Speaker
	0480	أمين عام	>myn EAm General secretary
	0481	صحفي	SHfy Journalist

Continued on next page

Table A.1 – continued from previous page

Domain	SignID	Arabic	English
	0482	رسام	rsAm Draftsman
	0483	خياط	xyAT Tailor
	0484	ضابط	DAbT Policeman
	0485	طيار	TyAr Pilot
	0486	جندي	jndy Soldier
	0487	حلاق	HlAq Barber
	0488	صباغ	SbAg Dyer
	0489	رجل إطفاء	rjl <TfA' Fireman
	0490	نجار	njAr Carpenter
	0491	معلم	mElm / mdrs Teacher
	0492	طباخ	TbAx Chef
	0493	فلاح	flAH Farmer
	0494	موظف	mwZf Agent
	0495	أمين صندوق	>myn Sndwq Treasurer
	0496	صيدلي	Sydly Pharmacist
	0497	طبيب	Tbyb Doctor
	0498	ممرضة	mmrDp Nurse
	0499	ممرض	mmrD Orderly
	0500	محام	mHAm Lawyer

Continued on next page

Table A.1 – continued from previous page

Domain	SignID	Arabic	English
	0501	انتقال	AntqAl Traveling
	0502	تعيين	tEyyn Appointment

REFERENCES

[1] T. Shanableh and K. Assaleh, "Telescopic vector composition and polar accumulated motion residuals for feature extraction in arabic sign language recognition," *Journal on Image and Video Processing*, vol. 2007, no. 2, pp. 9–9, 2007.

[2] C. Sun, T. Zhang, B.-K. Bao, C. Xu, and T. Mei, "Discriminative exemplar coding for sign language recognition with Kinect." *IEEE transactions on cybernetics*, vol. 43, no. 5, pp. 1418–28, Oct. 2013.

[3] T. Shanableh, K. Assaleh, and M. Al-Rousan, "Spatio-temporal feature-extraction techniques for isolated gesture recognition in Arabic sign language." *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, vol. 37, no. 3, pp. 641–50, Jun. 2007.

[4] M. F. Tolba and a. S. Elons, "Recent developments in sign language recognition systems," *2013 8th International Conference on Computer Engineering & Systems (ICCES)*, pp. xxxvi–xlii, Nov. 2013.

[5] K. Al-Fityani and C. Padden, “Sign language geography in the arab world,” *Sign languages: A Cambridge survey*, pp. 433–450, 2010.

[6] L. A. S. A. L. E. C. S. Organ, “Las: First part of the unified arabic sign dictionary,” 2000.

[7] M. Mohandes, M. Deriche, and J. Liu, “Image-Based and Sensor-Based Approaches to Arabic Sign Language Recognition,” *IEEE Transactions on Human-Machine Systems*, vol. 44, no. 4, pp. 551–557, Aug. 2014.

[8] 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. 873–91, Jun. 2005.

[9] M. AL-Rousan, K. Assaleh, and a. Tala’a, “Video-based signer-independent Arabic sign language recognition using hidden Markov models,” *Applied Soft Computing*, vol. 9, no. 3, pp. 990–999, Jun. 2009.

[10] M. Mohandes, M. Deriche, U. Johar, and S. Ilyas, “A signer-independent Arabic Sign Language recognition system using face detection, geometric features, and a Hidden Markov Model,” *Computers & Electrical Engineering*, vol. 38, no. 2, pp. 422–433, Mar. 2012.

[11] T. Shanableh and K. Assaleh, “Arabic sign language recognition in user-independent mode,” *2007 International Conference on Intelligent and Advanced Systems*, pp. 597–600, Nov. 2007.

[12] ——, “User-independent recognition of Arabic sign language for facilitating communication with the deaf community,” *Digital Signal Processing*, vol. 21, no. 4, pp. 535–542, Jul. 2011.

[13] H. Cooper, B. Holt, and R. Bowden, “Sign Language Recognition,” in *Visual Analysis of Humans.* Springer London, 2011, pp. 539–562.

[14] “LAS: First part of the Unified Arabic Sign Dictionary,” *The League of Arab States & the Arab League Educational, Cultural and Scientific Organization, Tunisia*, 2000.

[15] *LAS: Second part of the Unified Arabic Sign Dictionary*, 2006.

[16] S. Tamura and S. Kawasaki, “Recognition of sign language motion images,” *Pattern recognition*, vol. 21, no. 4, pp. 343–353, 1988.

[17] K. Murakami and H. Taguchi, “Gesture recognition using recurrent neural networks,” in *Proceedings of the SIGCHI conference on Human factors in computing systems.* ACM, 1991, pp. 237–242.

[18] T. E. Starner, “Visual recognition of american sign language using hidden markov models.” Massachusetts Inst Of Tech Cambridge Dept Of Brain And Cognitive Sciences, Tech. Rep., 1995.

[19] C.-L. Huang, W.-Y. Huang, and C.-C. Lien, “Sign language recognition using 3-d hopfield neural network,” in *Proceedings., International Conference on Image Processing*, vol. 2, Oct 1995, pp. 611–614 vol.2.

[20] H. Hienz, B. Bauer, and K.-F. Kraiss, "Hmm-based continuous sign language recognition using stochastic grammars," in *Gesture Workshop.* Springer, 1999, pp. 185–196.

[21] C.-S. Lee, Z. Bien, G.-T. Park, W. Jang, J.-S. Kim, and S.-K. Kim, "Real-time recognition system of korean sign language based on elementary components," in *Proceedings of 6th International Fuzzy Systems Conference*, vol. 3, Jul 1997, pp. 1463–1468 vol.3.

[22] M. ZakiAbdo, A. Mahmoud Hamdy, S. Abd El-Rahman Salem, and E.-S. Mostafa Saad, "Arabic Sign Language Recognition," *International Journal of Computer Applications*, vol. 89, no. 20, pp. 19–26, Mar. 2014.

[23] E. E. Hemayed and A. S. Hassanien, "Edge-based recognizer for arabic sign language alphabet (ars2v-arabic sign to voice)," in *Computer Engineering Conference (ICENCO), 2010 International.* IEEE, 2010, pp. 121–127.

[24] M. Al-Rousan and M. Hussain, "Automatic recognition of arabic sign language finger spelling," *International Journal of Computers and Their Applications*, vol. 8, pp. 80–88, 2001.

[25] M. Mohandes, S. A-Buraiky, T. Halawani, and S. Al-Baiyat, "Automation of the arabic sign language recognition," in *Information and Communication Technologies: From Theory to Applications, 2004. Proceedings. 2004 International Conference on.* IEEE, 2004, pp. 479–480.

[26] M. F. Tolba, M. Abdellwahab, M. Aboul-Ela, and A. Samir, “Image signature improving by pcnn for arabic sign language recognition,” *Canadian Journal on Artificial Intelligence, Machine Learning & Pattern Recognition*, vol. 1, no. 1, pp. 1–6, 2010.

[27] M. A. Mohandes, “Recognition of two-handed arabic signs using the cyberglove,” *Arabian Journal for Science and Engineering*, pp. 1–9, 2013.

[28] M. Mohandes and M. Deriche, “Image based arabic sign language recognition,” in *Signal Processing and Its Applications, 2005. Proceedings of the Eighth International Symposium on*, vol. 1. IEEE, 2005, pp. 86–89.

[29] M. F. Tolba, A. Samir, and M. Aboul-Ela, “Arabic sign language continuous sentences recognition using PCNN and graph matching,” *Neural Computing and Applications*, vol. 23, no. 3-4, pp. 999–1010, Aug. 2012.

[30] M. M. Zaki and S. I. Shaheen, “Sign language recognition using a combination of new vision based features,” *Pattern Recognition Letters*, vol. 32, no. 4, pp. 572–577, 2011.

[31] A. a. I. Sidig, H. Luqman, and S. A. Mahmoud, *Arabic Sign Language Recognition Using Optical Flow-Based Features and HMM.* Cham: Springer International Publishing, 2018, pp. 297–305.

[32] K. Assaleh, T. Shanableh, M. Fanaswala, H. Bajaj, and F. Amin, “Vision-based system for continuous arabic sign language recognition in user depen-

dent mode," in *Proceeding of the 5th International Symposium on Mechatronics and its Applications, ISMA 2008*, Amman, 2008, pp. 25–29.

[33] M. Alfonse, A. Ali, A. S. Elons, N. L. Badr, and M. Aboul-Ela, "Arabic sign language benchmark database for different heterogeneous sensors," in *Information & Communication Technology and Accessibility (ICTA), 2015 5th International Conference on.* IEEE, 2015, pp. 1–9.

[34] S. M. Shohieb, H. K. Elminir, and A. Riad, "Signsworld atlas; a benchmark arabic sign language database," *Journal of King Saud University-Computer and Information Sciences*, vol. 27, no. 1, pp. 68–76, 2015.

[35] M. Zahedi, D. Keysers, and H. Ney, "Pronunciation clustering and modeling of variability for appearance-based sign language recognition," in *International Gesture Workshop.* Springer, 2005, pp. 68–79.

[36] M. Zahedi, P. Dreuw, D. Rybach, T. Deselaers, and H. Ney, "Continuous sign language recognition-approaches from speech recognition and available data resources," in *Second Workshop on the Representation and Processing of Sign Languages: Lexicographic Matters and Didactic Scenarios*, 2006, pp. 21–24.

[37] P. Dreuw, D. Rybach, T. Deselaers, M. Zahedi, and H. Ney, "Speech recognition techniques for a sign language recognition system," *hand*, vol. 60, p. 80, 2007.

[38] U. Von Agris and K.-F. Kraiss, “Towards a video corpus for signer-independent continuous sign language recognition,” *Gesture in Human-Computer Interaction and Simulation, Lisbon, Portugal, May*, 2007.

[39] O. A. Crasborn, J. Mesch, D. Waters, A. Nonhebel, E. Van der Kooij, B. Woll, and B. Bergman, “Sharing sign language data online: Experiences from the echo project,” *International journal of corpus linguistics*, vol. 12, no. 4, pp. 535–562, 2007.

[40] A. V. Nair and V. Bindu, “A Review on Indian Sign Language Recognition,” *International Journal of Computer Applications*, vol. 73, no. 22, pp. 33–38, 2013.

[41] T. Ritchings, A. Khadragi, and M. Saeb, “An Intelligent Computer-Based System for Sign Language Tutoring,” *Assistive Technology*, vol. 24, no. 4, pp. 299–308, Dec. 2012.

[42] M. Mohandes and M. Deriche, “Arabic sign language recognition by decisions fusion using dempster-shafer theory of evidence,” in *Computing, Communications and IT Applications Conference (ComComAp), 2013*. IEEE, 2013, pp. 90–94.

[43] M. Tolba, A. Samir, and M. Abull-ela, “3D Arabic Sign Language Recognition Using Liner Combination of Multible 2D Views,” in *Informatics and Systems (INFOS), 2012 8th International Conference on*. cairo: IEEE, 2012.

[44] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, "Analysis of the accuracy and robustness of the Leap Motion Controller," *Sensors (Switzerland)*, vol. 13, no. 5, pp. 6380–6393, 2013.

[45] M. Mohandes, S. Aliyu, and M. Deriche, "Arabic sign language recognition using the leap motion controller," in *Industrial Electronics (ISIE), 2014 IEEE 23rd International Symposium on.* IEEE, 2014, pp. 960–965.

[46] ——, "Prototype arabic sign language recognition using multi-sensor data fusion of two leap motion controllers," in *Systems, Signals & Devices (SSD), 2015 12th International Multi-Conference on.* IEEE, 2015, pp. 1–6.

[47] A. Elons, M. Ahmed, H. Shedid, and M. Tolba, "Arabic sign language recognition using leap motion sensor," in *Computer Engineering & Systems (ICCES), 2014 9th International Conference on.* IEEE, 2014, pp. 368–373.

[48] M. Fraiwan, N. Khasawneh, H. Ershedat, I. Al-Alali, and H. Al-Kofahi, "A kinect-based system for arabic sign language to speech translation," *International Journal of Computer Applications in Technology*, vol. 52, no. 2-3, pp. 117–126, 2015.

[49] S. Aliyu, M. Mohandes, M. Deriche, and S. Badran, "Arabie sign language recognition using the microsoft kinect," in *Systems, Signals & Devices (SSD), 2016 13th International Multi-Conference on.* IEEE, 2016, pp. 301–306.

[50] M. Mohandes, S. Quadri, and M. Deriche, “Arabic Sign Language Recognition an Image-Based Approach,” *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW’07)*, pp. 272–276, 2007.

[51] A. addin I. Sidig, H. Luqman, and S. A. Mahmoud, “Transform-based arabic sign language recognition,” vol. 117, 2017, pp. 2 – 9, arabic Computational Linguistics.

[52] A. Tharwat, T. Gaber, M. Shahin, B. Refaat, and A. E. H. Ali, “Sift-based arabic sign language recognition system,” in *The 1st Afro-European Conference for Industrial Advancement,*, Addis Ababa, Ethiopia, November 17-19, 2014.

[53] A. Ahmed and S. Aly, “Appearance-based arabic sign language recognition using hidden markov models,” in *Engineering and Technology (ICET), 2014 International Conference on*, April 2014, pp. 1–6.

[54] S. Aly and S. Mohammed, “Arabic sign language recognition using Spatio-Temporal Local Binary Patterns and Support Vector Machine,” in *Advanced Machine Learning Technologies and Applications*, ser. Communications in Computer and Information Science, A. Hassanien, M. Tolba, and A. Taher Azar, Eds. Springer International Publishing, 2014, vol. 488, pp. 36–45.

[55] O. Aran and L. Akarun, “A multi-class classification strategy for fisher scores: Application to signer independent sign language recognition,” *Pattern Recognition*, vol. 43, no. 5, pp. 1776–1788, 2010.

[56] N. El-Bendary, H. M. Zawbaa, M. S. Daoud, A. E. Hassanien, and K. Nakamatsu, “ArSLAT: Arabic Sign Language Alphabets Translator,” *2010 International Conference on Computer Information Systems and Industrial Management Applications (CISIM)*, pp. 590–595, Oct. 2010.

[57] 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.

[58] M. P. Paulraj, S. Yaacob, M. S. Zanar Azalan, and R. Palaniappan, “A phoneme based sign language recognition system using 2D moment invariant interleaving feature and neural network,” *Proceedings - 2011 IEEE Student Conference on Research and Development, SCOReD 2011*, pp. 111–116, 2011.

[59] a. S. Elons, M. Abull-ela, and M. Tolba, “A proposed PCNN features quality optimization technique for pose-invariant 3D Arabic sign language recognition,” *Applied Soft Computing*, vol. 13, no. 4, pp. 1646–1660, Apr. 2013.

[60] A. Sutherland, G. Awad, and J. Han, “Boosted subunits: a framework for recognising sign language from videos,” *IET Image Processing*, vol. 7, no. September 2012, pp. 70–80, 2013.

[61] G. Awad, J. Han, and A. Sutherland, “A unified system for segmentation and tracking of face and hands in sign language recognition,” *Proceedings - International Conference on Pattern Recognition*, vol. 1, pp. 239–242, 2006.

[62] ——, “Novel boosting framework for subunit-based sign language recognition,” *Proceedings - International Conference on Image Processing, ICIP*, pp. 2729–2732, 2009.

[63] Y. Li, X. Chen, X. Zhang, K. Wang, and Z. J. Wang, “A sign-component-based framework for Chinese sign language recognition using accelerometer and sEMG data,” *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 10, pp. 2695–2704, 2012.

[64] M. Flasiński and S. Myśliński, “On the use of graph parsing for recognition of isolated hand postures of polish sign language,” *Pattern Recognition*, vol. 43, no. 6, pp. 2249 – 2264, 2010.

[65] S.-h. Yu, C.-l. Huang, and S.-c. Hsu, “Vision-based continuous sign language recognition using product HMM,” *The First Asian Conference on Pattern Recognition*, pp. 510–514, Nov. 2011.

[66] L. Pigou, S. Dieleman, P.-J. Kindermans, and B. Schrauwen, “Sign language recognition using convolutional neural networks,” in *Workshop at the European Conference on Computer Vision.* Springer, 2014, pp. 572–578.

[67] J. Huang, W. Zhou, H. Li, and W. Li, "Sign language recognition using 3d convolutional neural networks," in *Multimedia and Expo (ICME), 2015 IEEE International Conference on.* IEEE, 2015, pp. 1–6.

[68] L. Pigou, A. van den Oord, S. Dieleman, M. Van Herreweghe, and J. Dambre, "Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video," *International Journal of Computer Vision*, pp. 1–10, 2015.

[69] O. Koller, H. Ney, and R. Bowden, "Deep learning of mouth shapes for sign language," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 85–91.

[70] N. Neverova, C. Wolf, G. W. Taylor, and F. Nebout, "Multi-scale deep learning for gesture detection and localization," in *Workshop at the European conference on computer vision.* Springer, 2014, pp. 474–490.

[71] S.-Z. Li, B. Yu, W. Wu, S.-Z. Su, and R.-R. Ji, "Feature learning based on sae–pca network for human gesture recognition in rgbd images," *Neurocomputing*, vol. 151, pp. 565–573, 2015.

[72] S. Gattupalli, A. Ghaderi, and V. Athitsos, "Evaluation of deep learning based pose estimation for sign language recognition," in *Proceedings of the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments.* ACM, 2016, p. 12.

[73] N. Neverova, C. Wolf, G. Paci, G. Sommavilla, G. Taylor, and F. Nebout, “A multi-scale approach to gesture detection and recognition,” in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2013, pp. 484–491.

[74] E. Tsironi, P. Barros, and S. Wermter, “Gesture recognition with a convolutional long short-term memory recurrent neural network,” in *Proceedings of the European symposium on artificial neural networks computational intelligence and machine learning (ESANN)*, 2016, pp. 213–218.

[75] J. Wu, P. Ishwar, and J. Konrad, “Two-stream cnns for gesture-based verification and identification: Learning user style,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 42–50.

[76] P. Wang, W. Li, S. Liu, Y. Zhang, Z. Gao, and P. Ogunbona, “Large-scale continuous gesture recognition using convolutional neural networks,” in *Pattern Recognition (ICPR), 2016 23rd International Conference on.* IEEE, 2016, pp. 13–18.

[77] P. Wang, W. Li, S. Liu, Z. Gao, C. Tang, and P. Ogunbona, “Large-scale isolated gesture recognition using convolutional neural networks,” in *Pattern Recognition (ICPR), 2016 23rd International Conference on.* IEEE, 2016, pp. 7–12.

[78] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, and S. Gould, “Dynamic image networks for action recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3034–3042.

[79] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” in *Advances in neural information processing systems*, 2014, pp. 568–576.

[80] A. Samir and M. Aboul-Ela, “Error detection and correction approach for Arabic sign language recognition,” *2012 Seventh International Conference on Computer Engineering & Systems (ICCES)*, pp. 117–123, Nov. 2012.

[81] M. Filhol, M. N. Hadjadj, and B. Testu, “A rule triggering system for automatic text-to-sign translation,” *Universal Access in the Information Society*, vol. 15, no. 4, pp. 487–498, 2016.

[82] A. Almohimeed, M. Wald, and R. Damper, “Arabic text to arabic sign language translation system for the deaf and hearing-impaired community,” in *Proceedings of the Second Workshop on Speech and Language Processing for Assistive Technologies.* Association for Computational Linguistics, 2011, pp. 101–109.

[83] S. Morrissey, “Data-driven machine translation for sign languages,” Ph.D. dissertation, Dublin City University, 2008.

[84] C.-L. Liu, I.-J. Kim, and J. H. Kim, “Model-based stroke extraction and matching for handwritten chinese character recognition,” *Pattern Recognition*, vol. 34, no. 12, pp. 2339–2352, 2001.

[85] M. Kherallah, L. Haddad, A. M. Alimi, and A. Mitiche, “On-line handwritten digit recognition based on trajectory and velocity modeling,” *Pattern Recognition Letters*, vol. 29, no. 5, pp. 580–594, 2008.

[86] H. Wang and C. Schmid, “Action recognition with improved trajectories,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3551–3558.

[87] L. Wang, Y. Qiao, and X. Tang, “Action recognition with trajectory-pooled deep-convolutional descriptors,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4305–4314.

[88] M. Elmezain, A. Al-Hamadi, J. Appenrodt, and B. Michaelis, “A hidden markov model-based continuous gesture recognition system for hand motion trajectory,” in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on.* IEEE, 2008, pp. 1–4.

[89] W.-Y. Lin and C.-Y. Hsieh, “Kernel-based representation for 2d/3d motion trajectory retrieval and classification,” *Pattern Recognition*, vol. 46, no. 3, pp. 662 – 670, 2013.

[90] A. Naftel and S. Khalid, "Motion trajectory learning in the dft-coefficient feature space," in *Fourth IEEE International Conference on Computer Vision Systems (ICVS'06)*, Jan 2006, pp. 47–47.

[91] J. Pu, W. Zhou, J. Zhang, and H. Li, "Sign language recognition based on trajectory modeling with hmms," in *International Conference on Multimedia Modeling.* Springer, 2016, pp. 686–697.

[92] M. Boulares and M. Jemni, "3d motion trajectory analysis approach to improve sign language 3d-based content recognition," *Procedia Computer Science*, vol. 13, pp. 133–143, 2012.

[93] L. Geng, X. Ma, H. Wang, J. Gu, and Y. Li, "Chinese sign language recognition with 3d hand motion trajectories and depth images," in *Intelligent Control and Automation (WCICA), 2014 11th World Congress on.* IEEE, 2014, pp. 1457–1461.

[94] H. Wang, A. Stefan, S. Moradi, V. Athitsos, C. Neidle, and F. Kamangar, "A system for large vocabulary sign search," in *European Conference on Computer Vision.* Springer, 2010, pp. 342–353.

[95] P. Jangyodsuk, C. Conly, and V. Athitsos, "Sign language recognition using dynamic time warping and hand shape distance based on histogram of oriented gradient features," in *Proceedings of the 7th International Conference on PErvasive Technologies Related to Assistive Environments.* ACM, 2014, p. 50.

[96] M. Bhuyan, P. Bora, and D. Ghosh, “Trajectory guided recognition of hand gestures having only global motions,” *World Academy of Science, Engineering and Technology*, vol. 21, pp. 753–764, 2008.

[97] T. K. Ho, “The random subspace method for constructing decision forests,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 20, no. 8, pp. 832–844, 1998.

[98] M. Simão, P. Neto, and O. Gibaru, “Using data dimensionality reduction for recognition of incomplete dynamic gestures,” *Pattern Recognition Letters*, 2017.

[99] C. Harris and M. Stephens, “A combined corner and edge detector.” in *Alvey vision conference*, vol. 15. Citeseer, 1988, p. 50.

[100] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[101] A. Bosch, A. Zisserman, and X. Munoz, “Image classification using random forests and ferns,” in *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.

[102] C. Elkan, “Using the triangle inequality to accelerate k-means,” in *ICML*, vol. 3, 2003, pp. 147–153.

[103] C. Neidle and C. Vogler, “A new web interface to facilitate access to corpora: Development of the asllrp data access interface (dai),” in *Proc. 5th Work-*

shop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon, LREC, 2012.

[104] NCSLGR, "The national center for sign language and gesture resources (ncslgr) corpus."

[105] S. A. Mahmoud, "Motion estimation based on modified Fourier spectrum," *Information Processing Letters*, vol. 37, no. 6, pp. 311–313, mar 1991.

[106] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[107] K. Lertniphonphan, S. Aramvith, and T. H. Chalidabhongse, "Human action recognition using direction histograms of optical flow," in *Communications and Information Technologies (ISCIT), 2011 11th International Symposium on.* IEEE, 2011, pp. 574–579.

[108] N. Gillian and J. A. Paradiso, "The Gesture Recognition Toolkit," *Journal of Machine Learning Research*, vol. 15, pp. 3483–3487, 2014.

[109] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *British Machine Vision Conference*, 2014.

[110] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern*

Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009, pp. 248–255.

[111] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” in *International Conference on Machine Learning*, 2015, pp. 2048–2057.

[112] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, “Tensorflow: Large-scale machine learning on heterogeneous distributed systems,” *arXiv preprint arXiv:1603.04467*, 2016.

[113] S. Escalera, J. Gonzàlez, X. Baró, M. Reyes, O. Lopes, I. Guyon, V. Athitsos, and H. Escalante, “Multi-modal gesture recognition challenge 2013: Dataset and results,” in *Proceedings of the 15th ACM on International conference on multimodal interaction.* ACM, 2013, pp. 445–452.

[114] N. Habash, A. Soudi, and T. Buckwalter, “On Arabic transliteration,” in *Arabic computational morphology.* Springer, 2007, pp. 15–22.

Vitae

- Name: Ala addin Ismaeil Sidig Sabour
- Nationality: Sudanese
- Date of Birth: 12 Oct 1981
- Email: *ala.sidig@gmail.com*
- Permanent Address: Khartoum 11111 - Sudan

- **Education**
  - Msc in Electronics Engineering (Computer Engineering)
  - Bsc in Electronics Engineering (Computer Engineering)

Conference Papers:

- **Sidig, Ala addin I.**, Hamzah Luqman, and Sabri A. Mahmoud (2018b). “Arabic Sign Language Recognition using Optical flow-based features and HMM.”. In: Recent Trends in Information and Communication Technology: Proceedings of the 2nd International Conference of Reliable Information and Communication Technology (IRICT 2017). Cham: Springer International Publishing, pp. 297–305.

- **Sidig, Ala addin I.**, Hamzah Luqman, and Sabri A. Mahmoud (2017). “Transform-based Arabic sign language recognition”. In: vol. 117. Arabic Computational Linguistics, pp. 2–9. Presented by Hamzah and won the first prize in CCSE participation in KFUPM 8th students forum.

Journal Papers:

- **Sidig, Ala addin**, and Sabri Mahmoud (2017). “Trajectory based Arabic Sign Language Recognition”. In: Pattern Analysis and Applications. (submitted).

- **Sidig, Ala addin**, Hamzah Luqman, and Sabri Mahmoud (2017). “Efficiency of Optical-based Features with HMM for Arabic Sign Language Recognition”. In: Jurnal Teknologi. (submitted).

- **Sidig, Ala addin** et al. (2017). “KArSL: Arabic Sign Language Database.” In: Language Resources and Evaluation. (Submitted).

- **Sidig, Ala addin**, Hamzah Luqman, and Sabri Mahmoud (2017). “State of the art of Arabic Sign Language Recognition and Translation”. (in preparation).

Patent

- Sabri Mahmoud, **Sidig, Ala addin**, ” Arabic Sign Language Recognition using Bag of Features and Bag of Postures”, (submitted).