Scientific Visualization, 2018, volume 10, number 5, pages 140 - 159, DOI: 10.26583/sv.10.5.09

Arabic Dynamic Gestures Recognition Using Microsoft Kinect

B. Hisham[1], A. Hamouda[2]
Computer engineering department, Al-Azhar University, Cairo, Egypt

[1] ORCID: 0000-0002-1468-0145, basmahisham.2015@gmail.com
[2] ORCID: 0000-0001-9041-1978

Abstract

Sign language is an expressive way for deaf persons and hearing impaired to communicate with their societies, it is the basic alternative communication method between them and others. There are several studies have been done on sign language recognition systems, however, practically deployable system for real-time use is still a challenge also the researches in Arabic Sign Language Recognition (ArSLR) is very limited. This paper proposes Arabic Sign Language (ArSL) recognition system using Microsoft Kinect. The proposed system normalizes user's position and size captured by Microsoft Kinect then applies machine learning algorithms such as Support Vector Machine (SVM), K- Nearest Neighbors (KNN) and Artificial Neural Network (ANN) in order to provide a comparison on recognition accuracy. Also, we used Dynamic Time Wrapping (DTW) in order to match the sequence that represents the captured sign with the stored reference sequences, this is based on that all signs are dynamic. Recognized continuous signs are segmented using motion speed that segment a sequence of words with an accurate manner. We use a dataset for ArSL words from collected signs; it is composed of 42 Arabic signs in medical field to aid communication between a deaf or hard-of-hearing patient with the doctor. The experimental results showed that the proposed system recognition rate reached 89 % for KNN classifier with majority voting and the segmentation accuracy reached 91%. The system was trained on 840 samples and tested on 420 samples.

Keywords: Sign Language, Static Gestures, Microsoft Kinect, KNN, ANN, SVM, DTW.

1. Introduction

Sign language is the most essential way for deaf people to communicate and interact with others. There are two main sign language recognition systems: image-based and sensor-based. The major advantage of image-based system is that user does not need to use complicated devices, but this technique needs extra computations in the preprocessing stage, image processing, and artificial intelligence to recognize and interpret signs. Sensor-based systems use some equipment with sensors like gloves equipped with sensors. Sensor based systems require the users to wear sensor-based gloves or any sensor-based instruments. Microsoft Kinect is a motion sensing input device by Microsoft. It relies on depth technology, which allows users to deal with any system via a camera in which the user uses hand gestures or verbal commands to recognize objects without the need to touch the controller. It used to track standing skeleton with high-depth fidelity. It has an RGB camera, voice recognition capability, face-tracking capabilities, and access to the raw sensor records. Once the data has been collected from the user, the recognition system, whether it is sensor-based or image-based, must use this data for processing to recognize the signs. Several approaches have been proposed for sign recognition including fuzzy logic, neural networks, support vector machines, k-nearest neighbors, hidden markov models...etc. Single sign classifier assumes that signs are pre-segmented, it recognizes sign by sign not continuous sentences. It supposed to automate the process of splitting a sentence into words, this process is called segmentation. Segmentation is one of the major issues of in-

formation processing in sign languages. Motion speed during capturing of continuous sentences is used as a segmenter. It is noticeable that the motion speed is changed during performing the signs, when the transition from one sign to another occurred the motion speed is slowed down.

The main aim of this paper is to develop Arabic Sign Language (ArSL) recognition system that identify the user signs captured by Microsoft Kinect using one of two techniques: model-based (i.e.) machine learning classifiers or direct machine (i.e.) DTW (Dynamic Time Wrapping) and segment Arabic continuous sentences using speed motion detection method. The structure of this paper is organized as follows. Section 2 presents the related work of the sign language recognition. The proposed system is presented in Section 3. The experiments setup and results are presented in section 4. Section 5 contains conclusion & future work.

2. Related Work

Al-Jarrah and Al-Omari developed automated translation system of alphabets gestures in the Arabic Sign Language (ASL) that does not use visual markings. The images of not covered hands are processed to get a set of features that are translation, rotation, and scaling invariant. A system accuracy of 97.5% was achieved on a database of 30 Arabic alphabet signs [1]. Maraqa and Abu-Zaiter introduced the use of neural networks in human hand gesture recognition for static images and dynamic gestures. They presented the use of feedforward and recurrent neural networks and recurrent neural networks along with its different architectures. Then, they tested the proposed system, a database of 900 samples, containing 30 gestures performed by two persons with colored gloves that used in their experiments. The accuracy rate for the recognition of the static gestures reaches 95% [2]. El-Bendary et al. developed a sign language recognition system for the Arabic alphabets with an accuracy of 91.3%. The proposed Alphabets Translator (ArSLAT) does not rely on using any gloves or visual markings to complete the recognition task. ArSLAT deals with images of bare hands, which allows the signer to interact with the system in a natural way. Extracted features from a video of signs are the input to the system and the output is recognized sign as a text. The proposed ArSLAT system composed of five steps; pre-processing, detect the best-frame, detect the category, Extract the features, and finally classification. The used extracted features are translation, scale, and rotation invariant to make the system flexible [3]. Hemayed and Hassanien introduced a recognition technique for the hand gestures which represent the Arabic sign language alphabet and perform sign to voice conversion in order to enable Arabian deaf people to communicate with their societies. The proposed system focused on static and simple moving gestures. Principal Component Analysis algorithm is applied to the extracted edges that form the predefined feature vectors for signs library. The Euclidean distance is used to measure the similarity between the signs' feature, the nearest vectors sign is selected, and the corresponding sound clip is played. They applied the system to more than 150 signs and gestures with accuracy near to 97% at real time test for three different users [4]. A survey on sign language processing is proposed by Kausar & Javed. They categorized algorithms into two categories static and dynamic methods. They identified important topics in sign language recognition as segmentation, size of dictionary, invariance, unrestricted environment, gestures variety, generality, feature extraction, start/end identification of gesture sequences and feature selection. They showed some challenges and recommendations for future research in the field of sign language processing [5]. Azad et al proposed a novel and real-time approach for sign language recognition system using hand gestures. In the proposed model, the main step is extracting the hand gestures from the image by using three main steps: segmentation, morphological processing and finally features extraction. They applied the cross-correlation coefficient in order to recognize the gestures and they

used the data base of the American Sign Language, the accuracy reached 98.34 and they suggested the using of video sequence for virtual reality to recognize the dynamic hand gestures [6]. Naoum et al. developed an image-based system for sign language alphabets with an accuracy of 50%, 75%, 65% and 80 % for bare hand, hand with a red glove, hand with a black glove, and hand with a white glove respectively. The proposed system starts by finding histograms of the images. Extracted profiles from histograms are then used as input to the K-nearest Neighbor classifier. The algorithm is designed to work as a first level of recognition upon a series of steps to convert the captured character images into actual spelling [7]. Elons et al. proposed an ArSLR system based on neural networks that able to compensate for lighting non-homogeneity and background brightness. The proposed system showed stability under geometrical transforms, lighting conditions and bright background, achieving a recognition accuracy of 90%. The "Continuity Factor" is defined and considered as a weight factor of the current pulse in signature generation process. This factor measures the simultaneous firing strength for connected pixels [8]. SamirElons et al, proposed a method to enhance the feature quality based on neural network. The model defines continuity factor is proposed as a weight factor of the current pulse in signature generation process. The proposed method has been employed in a feature extraction model that is followed by classification process that training and testing for Arabic sign language static hand recognition [9].

Researches on Arabic sign language recognition (ArSLR) has a great rise recently. Mohandas et al, at 2014, presented a study that focused on both image-based and sensor-based approaches. They showed the most popular types of ArSLR algorithms, mentioned the main features of the different approaches. They categorized their work to three types alphabet, isolated word, and continuous recognition [10]. Almasre et al, developed a supervised machine learning model for hand gesture recognition to recognize Arabic Sign Language (ArSL), using two sensors: Microsoft's Kinect and a Leap Motion Controller. The proposed model relies on the concept of supervised learning to predict a hand pose from the two sensors depending on depth images and defines a classifier to transform gestures based on 3D positions of a hand-joints direction into their letters. Recognized letters are compared and displayed in real time. They used the 28 letters of the Arabic alphabet many times from different volunteers to create a dataset gestures for each letter of an ArSL built by the depth images retrieved both devices. The results indicated that using the two devices for the ArSL model were essential in detecting and recognizing 22 of the 28 Arabic alphabets correctly 100 % [11]. ElBadawy et al, proposed an integrated system used a hybrid types of sensors to capture all sign features. They customized Leap motion to capture hands with fingers movements. Two digital cameras are used to capture face features and body movement. The system performed 95% recognition accuracy for a dataset of 20 dynamic signs due to the additional modules for facial expressions recognition and body movement recognition [12]. Aliyu et al developed a Kinect based system for Arabic sign language recognition system. The developed system was tested with 20 signs from the Arabic language dictionary. Video samples of both true color images and depth images were collected from volunteer user. Linear Discriminant analysis was used for features reduction and sign classification. Furthermore, fusion from RGB and depth sensor was carried at feature and decision level performed an overall accuracy of 99.8% [13]. Jma et al, proposed a new approach based on hand gesture analysis for Arabic sign language (ArSL) alphabet recognition by extracting a histogram of oriented gradient (HOG) features from a hand image and then using them to train an SVM models. Their approach involves three steps: (i) Hand detection and localization using a Microsoft Kinect camera, (ii) hand segmentation and (iii) feature extraction using Arabic alphabet recognition. The results showed accuracy about 90% [14]. Almasre et al, introduced a sign language

recognition model that interpreted a set of Arabic Sign Language alphabets using Microsoft's Kinect and a supervised learning algorithm was used with the Candescent Library. The dataset of the model was collected by allowing users to make certain letters of the Arabic alphabets. The proposed model filtered each sign based on the number of the used fingers and then calculated the Euclidean distance between the contour points of a captured sign and a stored sign, and then comparing the results with a certain threshold [15].

Mohandes et al, proposed a new method for Arabic sign language to track one hand or both and one finger or more at different locations using two different sensors; They generated a 3-dimensional (3D) interaction space and extracted 3D features related to the detected hand(s). They analyzed the metrics related to the processed features. The applied classifier integrated with two different sensors, Leap Motion (LMC) and Kinect then all Arabic alphabets signs are performed in the interaction space [16].

Almasre et al, proposed a model to detect the hand gestures of Arabic Sign Language (ArSL) using two depth sensors applied on Arabic words. They examined 143 signs gestured by set of users (10 users) for five Arabic words. These sensors worked with depth images of the upper part of humans, from which 235 angles (features) were used for all joints and every pair of bones. The used dataset was divided into a training dataset about more than one hundred observations and a testing set is about 34 observations. They used support vector machine (SVM) classifier with different parameters in order to obtain four SVM models using both linear and radial kernel function. The accuracy of the model for the training set for the SVMLD, SVMLT, SVMRD, and SVMRT models was 88.92%, 88.92%, 90.88%, and 90.884%, respectively. The accuracy of the testing set for SVMLD, SVMLT, SVMRD, and SVMRT was 97.059%, 97.059%, 94.118%, and 97.059%, respectively [17].

Most of the previous systems concerned with static gestures which has no motion ,there for the aim of this research is to present a very accurate model for recognizing the Arabic dynamic gestures using Microsoft Kinect in order to overcome the constraints of the sensor based approach and vision based approach also as presented in the literature there are few studies presented continuous recognition systems for Arabic sign Language especially using Kinect device ,so that we introduces accurate and robust model for sentence segmentation based on simple method (The minimal velocity detection) with acceptable accuracy for the segmentation and also segmented words recognition .

3. The proposed System

In this section we will discuss the elements of our proposed system for Arabic sign language recognition using Microsoft Kinect. We describe the various phases of our system from the capturing the data using Kinect until the gesture is recognized. We concentrate on the feature extraction and selection phase because we believe it is a very crucial phase in the recognition system. The structure of the recognition system is shown in Fig.1. First of all, (1) data acquisition phase is performed, in this step Kinect depth camera is used to infer human skeleton positions, then (2) skeleton tracking & joint selection phase coming after it, in this step we were trying to access each skeleton data such as joint coordinates, joint types, position and bone orientation individually and choose the joints of interest, then (3) the information about these joints are received by our system as frames which are updated with the skeleton of the user , then (4) normalization phase which is applied on the collected frames should overcome mainly on two problems ,firstly the variation of the user positions ,secondly the variation of the users' sizes , then (5) applying some classifiers as SVM, KNN and ANN to recognize the Arabic signs then, (6) applying majority voting to select the most frequent classified class and output the sign. Finally (7) applying the DTW (Dynamic Time Warping) instead of the classifiers in order to compare the results of it with the results of the machine learning approach.

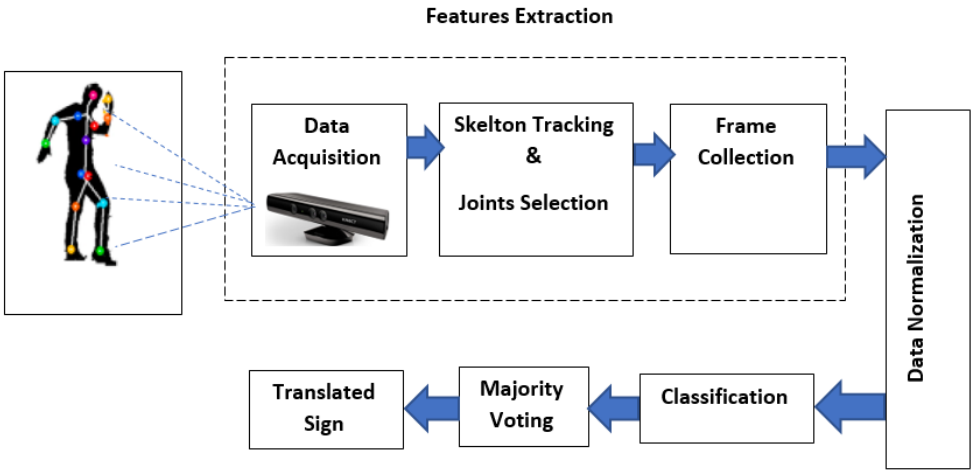


Fig.1 System Block Diagram

The following diagram in Fig.2 represents the flow chart of our proposed model.

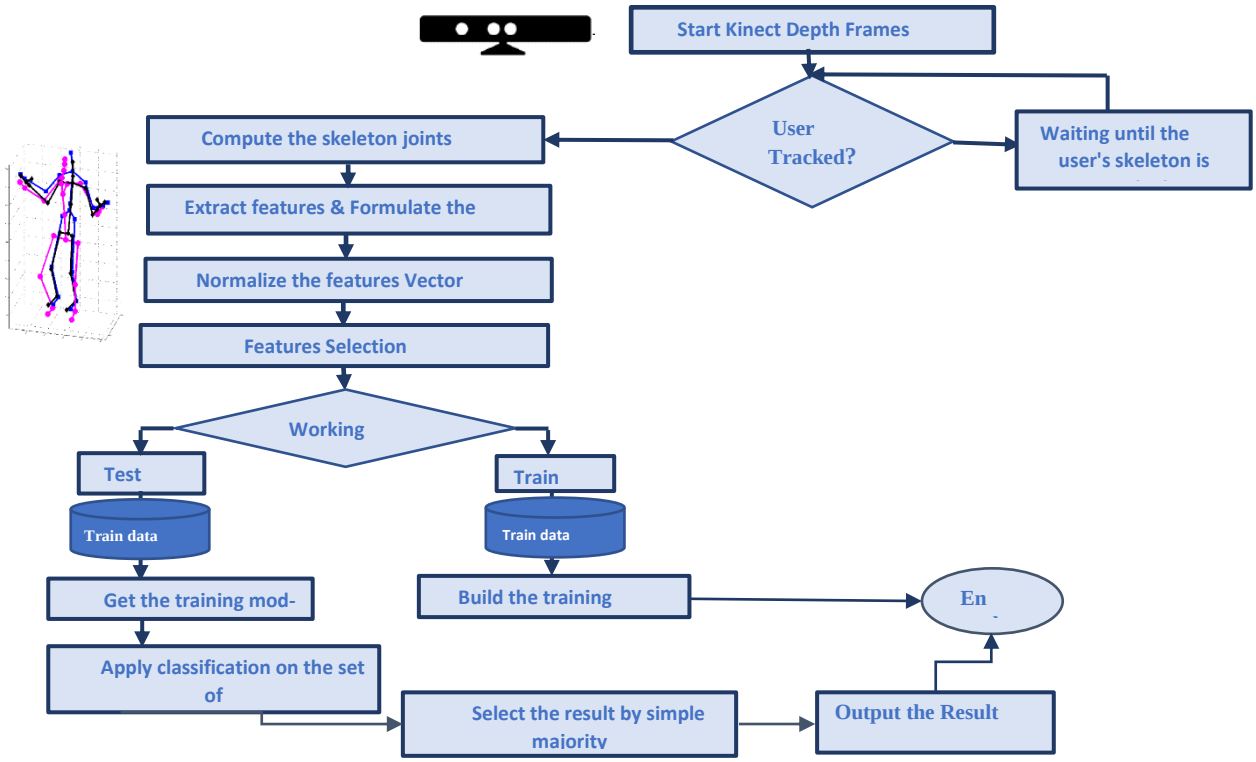


Fig. 2 Proposed Model Flow Chart

3.1 Data Acquisition

Kinect windows SDK contains a set of APIs which access easily to the skeleton joints. So, in this phase, information about the movements of the user are collected. When starting the kinect's depth camera, it could capture 20 skeleton joints as shown in Fig. 3 with rate of 30 frames per second using its SDK.

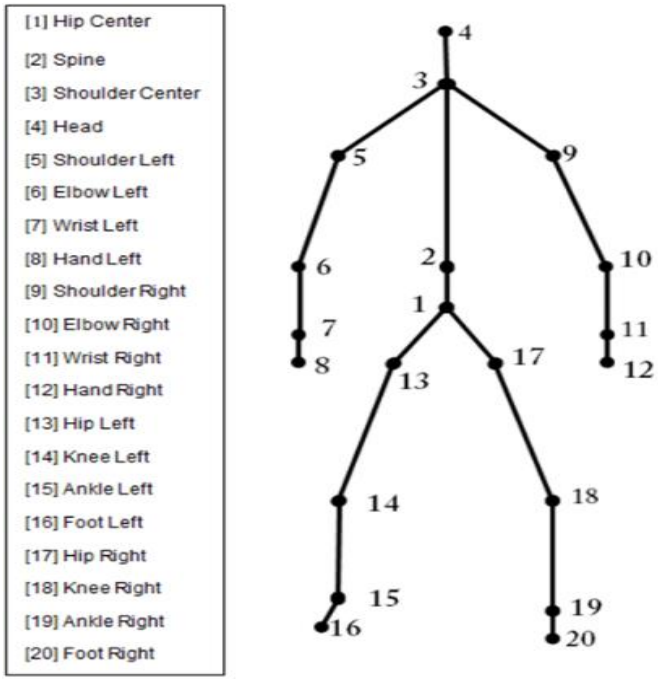


Fig. 3 Skeleton Joints

3.2 Skelton Tracking & Joints selection

Skeleton tracking allows Kinect to recognize people and follow their actions. In this phase, some algorithms and mathematical operations in kinect's SDK are used to interpret the 3D data from Kinect and infer the positions of detected objects. For joint selection, after carefully studying the signs of our dictionary for the system, only 10 joints out of the 20 resulted to be significant for the description of a sign: Hand (Left & Right), Shoulder (Left & Right), Elbow (Left & Right), Wrist (Left & Right), Spin Mid and Head Center. There is no point in tracking others joints such as the knees, the feet, etc. because they remain almost static during the execution of the sign.

3.3 Normalization

While performing the gestures user can be at any position and can be of any size i.e. the variation in height and overall body built up. Any variation in depth can cause a considerable variation of the X and Y values.

a) Normalize User Position

First for the user's position as shown in Fig. , the distances obtained for joints in X, Y, Z coordinates are scaled by subtraction of the spine-mid joint coordinates from the required joint skeleton points. Now if the coordinates are taken at any position whether it is totally right or left corner, it will be scaled, and no conflict will be happened.

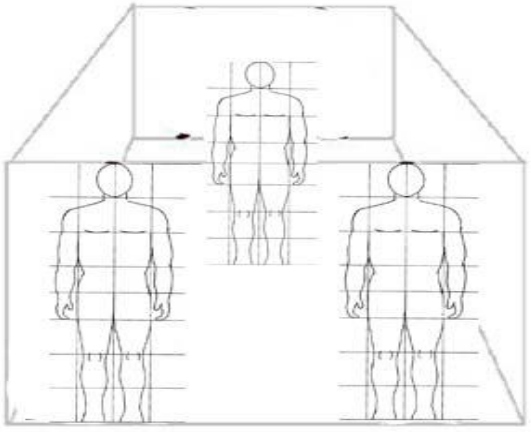


Fig. 4 Normalization of Position

For our features we get the spherical coordinates instead of using the Cartesian coordinates X, Y, and Z, the spherical coordinates considering SpinMid Joint as the origin. In mathematics, the spherical coordinate system is represented by three-dimensional space where the position of a point is specified by three numbers: the radial distance r for any point from a fixed , the (ϕ) which is the angle between the positive x-axis and the line denoted by r as in Fig. , (θ) which is the angle that is measured between the positive z-axis and the line from the origin to the point . After the evaluation of the system, the results showed that the (θ) feature does not has any affect and no meaningful, so we will depend on r and ϕ in our calculations and we compute them through Eq.1 and Eq.2 respectively.

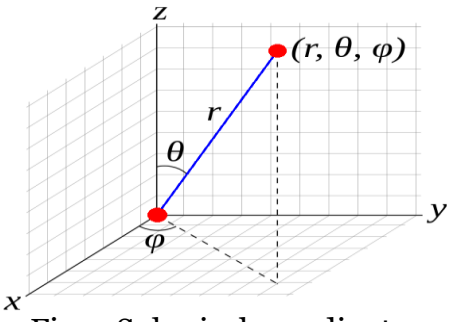


Fig. 5 Spherical coordinates.

$$\sum_{i=1}^{n} r(i) = \sqrt{(J(i)_x - S_{M_x})^2 + (J(i)_y - S_{M_y})^2 + (J(i)_z - S\_M_z)^2} \quad (1)$$

$$\sum_{i=1}^{n} \emptyset(i) = \tan 2\left((J(i)_y - S\_M_y), (J(i)_x - S\_M_x)\right) \quad (2)$$

Where,
n is the number of joints from J,
r is a radial distance,
S_M_x is x coordinate of spin-mid joint,
S_M_y is y coordinate of spin-mid joint,
S_M_z is z coordinate of spin-mid joint.

b. Normalize User Size

For the user's size, it may differ from one user to another as in Fig. and this may cause a conflict to the system so in our system we normalize all relative distances by the factor (r_{H,S_M}) the distance between the head and spin-mid as in Eq.3, this value refers to the size of the user and all distances can be normalized according to its value .

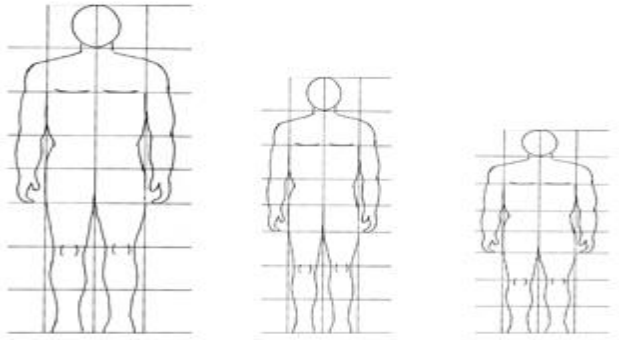


Fig. 6 Normalization of User's size

$$\sum_{i=1}^{n} r_{norm}(i) = \frac{r(i)}{r_{H,S\_M}} \quad (3)$$

Where,
n is the number of joints from J,
r_{norm} is a normalized radial distance of the joint,
r_{H,S_M} is a radial distance from head center to spin-mid.

3.4 Features Selection

As it is explained in Skelton Tracking & Joints selection that the used joints are Hand (Left & Right), Shoulder (Left & Right), Elbow (Left & Right), Wrist (Left & Right) , also Spin Mid and Head Center which are used in normalization .To identify the important features we selected a subset of features from this set to enhance the recognition process by running an automated feature selection process using filter feature selection , these features as the difference in distance between Hand (Left & Right) and Shoulder (Left & Right) .The total number of features are about 32 features in spherical coordinates, they are listed as a following:

{r, Ø} of right, left {hand, wrist, elbow, shoulder} position.

{r} of separation between right and left {hand, wrist, ...}

3.5 Classification

The sign recognition goal can only be achieved when the collected frames are coupled with an effective features extraction followed by highly efficient classification. In the proposed system the classifier takes the sequence of frames that formed the single sign which is pre-segmented and classify each frame separately to predicts the class that frame belongs to it. To achieve that we trained our classifier on the individual frames of the training set (e.g. " وفاة" means "death") .There are two ways of solving the classification problems: linear and nonlinear classifier as in Fig. 7. The frames collection with defined features help to identify the classification problem as linear or nonlinear classification [18], so we used machine learning algorithms to build classifiers using SVM, ANN and KNN that are efficient to deal with multiclass nonlinear classification problems and find the best matching class given a list of classes.

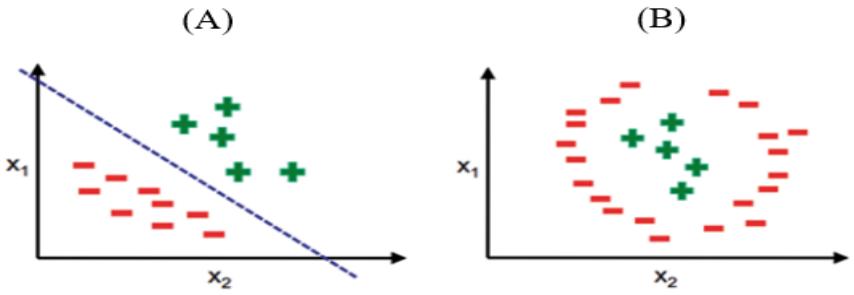


Fig. 7 Classification Types

3.6 Majority voting

After the classification phase, when the signer performs a sign, it enters to the system as a test sequence then the classifier starts to classify each frame separately. The result finally is selected by majority voting i.e. the classification with the most frequent frames assigned to it. For example, if the classification results of the test sequence were as in Fig.8, then the major index will be index "1" so the translated sign will be sign with index=1 (اسهال/dysentery).

Fig.8 Majority Voting

3.7 DTW (Dynamic Time Warping)

There are two types of gestures, static gestures which do not require to any movement and dynamic gestures which is in contrast to the static, they are composed of a series of static gestures with some movements transition from one gesture to another, so they are composed of a set of frames. In this case, we can apply direct matching technique for recognition. So that, the captured sign can be directly compared with the stored signs even the difference in sequences length. There are two most widely used techniques for dynamic gesture recognition: (1) DTW (Dynamic Time Warping) and (2) HMM (Hidden Markov Model). They are used for recognition of similarities between two temporal sequences that do not need to be synchronized in time according to their features [19]. We used DTW (Dynamic Time Warping) in order to compare the captured gesture in real time with the recorded dynamic gestures. The role of DTW is to find the optimal alignment between two time-dependent sequences [20], in this research the sequence is composed from the vector of the extracted features from the captured sign. The sequence is warped automatically in real time with nonlinear form in order to match the one of the stored sequences that represents the gestures data set [21].

Let we have two-time dependent temporal sequences X: = (x1, x2, x3.....xi) of length I and Y: = (y1, y2, x3...yj) of length J and let Ƒ is a feature space where $X_I, Y_J \in Ƒ$. The main objective of DTW is to compare these sequences and analyze the similarities between them and finally find the optimal alignment, so to compare x and y sequences we need to find a measure called local cost or local distance as in Eq.4.

$$C: Ƒ \times Ƒ \quad (4)$$

The value of C (x, y) must be very small when x and y represents the same gesture else it must be large. In this step we generate the local cost matrix with dimension of (I X J) as in Fig.9. The cost of any position at the local cost matrix M (i, j) can be determined as in Eq.5

$$M(i, j) = d(i, j) + \min \{M(i-1, j-1), M(i-1, j), M(i, j-1)\}. \quad (5)$$

The equation composed of two parts the first part is the Euclidean distance d (i, j) between the feature vectors of the sequences X and Y, the second part is the minimum cost of the adjacent elements of the cost matrix up to that point [22].

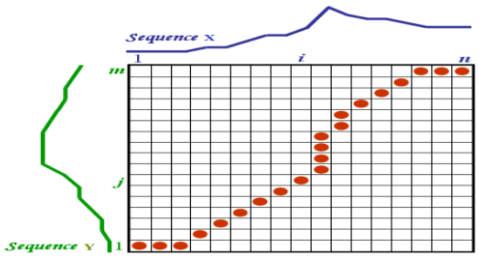
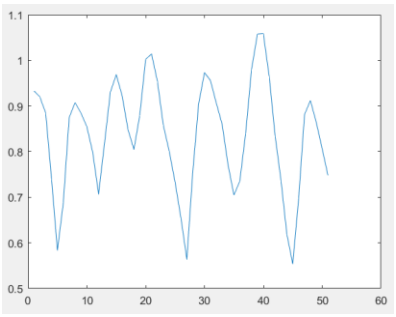
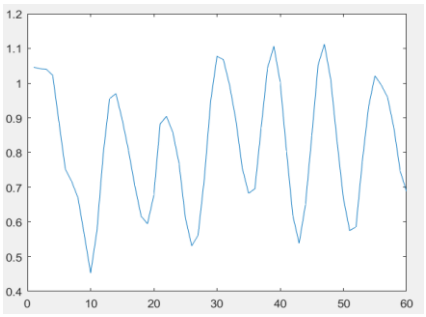


Fig.9 DTW (Dynamic Time Wrapping) cost matrix

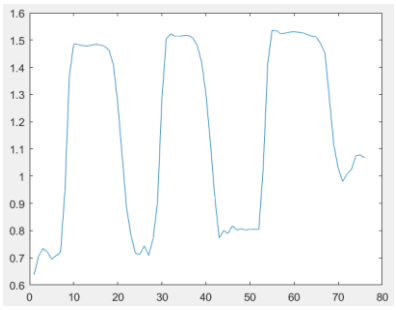
Fig.10 represents the difference between the two signs " نزيف" or "Bleeding" with ID=39 and "اشعة " or "X-Ray" with ID= 2, the graphs represents the orientation of left-hand distance (LH. D), X-axis represents the number of frames and Y-axis represents the coded values corresponded to each frame. The first two graph represent the similarities between two samples for the same sign which performed from two different users at different time and the third graph represents a different sign and it is clear that it is completely different from them.



a) LH. D feature for a sample of sign "نزيف" or "Bleeding" stored in training set



b) LH. D feature for a sample of sign "نزيف" or "Bleeding" stored in testing set



c) LH. D feature for a sample of sign "اشعة" or "X-Ray" stored in testing set

Fig.10 LH. D (Left hand -distance) feature over samples of two different signs

Fig.11 represents the original signal and warped signal for the left hand -distance feature for the sign " نزيف" or "Bleeding".

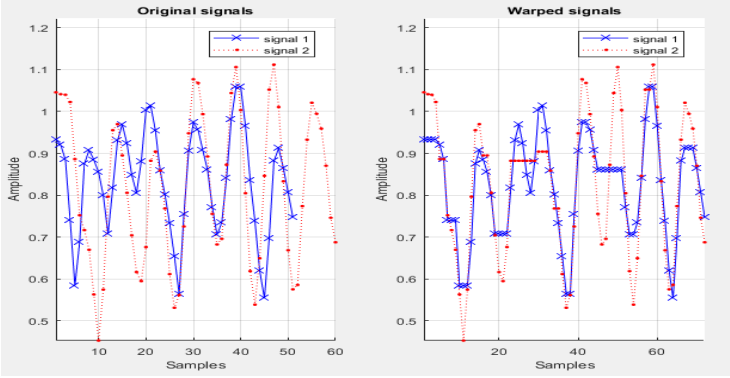


Fig.11 Original signal and warped signal for left-hand distance (LH. D) over a sample for Sign_ID (39) " نزيف" or "Bleeding ".

Fig.12 represents the DTW graph between the original signal and aligned sign for the sign "نزيف" or "Bleeding".

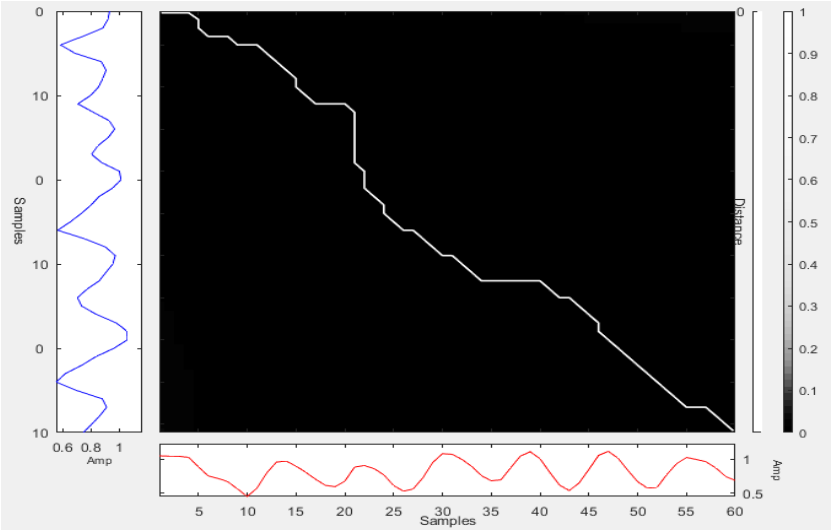


Fig.12 DTW graph between the original signal and aligned sign for the sign "نزيف" or "Bleeding".

4. Experimental Results

For the purpose of studying the performance of the proposed system, we developed a software application based on the C# Language to interface with Kinect sensor and we used Microsoft Kinect SDK.

4.1 Arabic signs Data Set

For the purpose of studying the performance of the proposed system, we recorded 40 different gestures in medical field which are listed in Table 1 using Microsoft Kinect V2. The data collected from two different volunteers in different position and with different sizes.

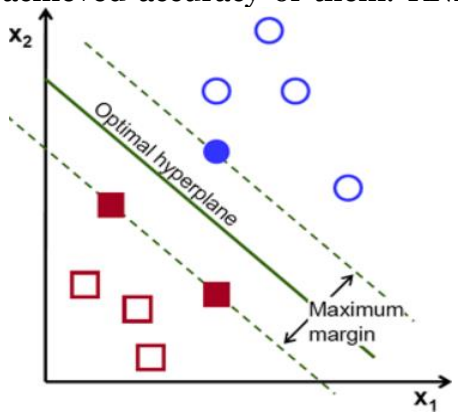
Table 1 Medical Dataset

Index	Arabic Sign	Meaning in English	Index	Arabic Sign	Meaning in English
1	اسهال	Dysentery	22	نزيف	Bleeding
2	اشعة	X-Ray	23	وفاة	Death
3	استقبال	Reception	24	طبيب عظام	Orthopedic doctor
4	رئتان	Two lungs	25	علاج طبيعى	Physical therapy
5	كبد	liver	26	حقن	Injection
6	كلى	kidneys	27	زغللة	Blurred vision
7	معدة	stomach	28	سرطان	Cancer
8	امساك	Constipation	29	جهاز قياس ضغط	Pressure measuring device
9	تحليل	analysis	30	صداع	A headache
10	تطعيم	Vaccination	31	صمم	Deafness
11	شلل	Paralysis	32	طبيب اطفال	Pediatrician
12	طبيب توليد	Obstetrician	33	طبيب انف واذن	Doctor of nose and ear
13	تقيوء	Vomiting	34	طبيب باطنة	Internist
14	تورم	Swelling	35	طبيب عام	General Doctor
15	جرح	Wound	36	كسر عظام	Broken bones
16	حامل	Pregnant	37	فيتامينات	Vitamins
17	حرارة	Fever	38	فشل كلوى	Kidney failure
18	اوردة	Veins	39	قرحة	ulcer
19	حساسية	Allergic	40	قولون	The colon
20	مغص	Colic	41	مختبر	laboratory
21	انا	I	42	اشعر	Feel

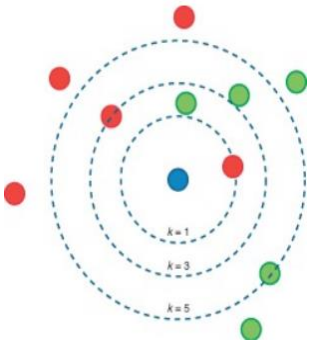
4.2 Training and Testing

For training phase, we collected 20 samples from different 5 signers for each sign as a training set and 10 samples from different 2 signers as a testing set. Thus, making it a total of 1260 samples which we divided into 840 samples for train set and 420 samples for test set. All collected signs are dynamic (i.e.) multiple joints are moving like hand, wrist, elbow and shoulder. For each sign we recorded a sequence of skeleton data consisted of 20 joint positions per frame, which are formed from x, y, and depth coordinates). Each incoming frame are preprocessed, the joints of interest are selected and normalized relative to SpinMid joint as in sec.3.3. Each sign's sequence contains on average (100 or 170) frames, making for around 36,456 frames in total for training set and 33, 995 frames for testing set. Then the feature vector is formed by applying our mentioned tech-

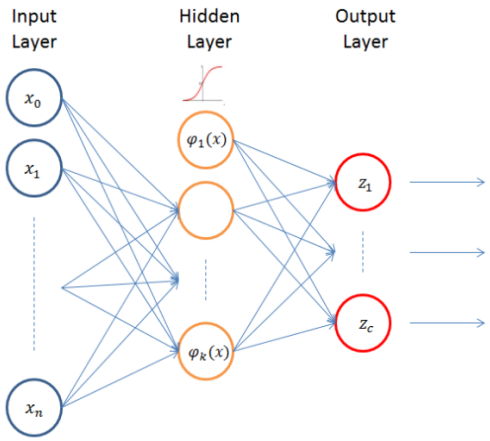
nique and appending the extracted features which are recorded for each frame. This is done for both the training as well as the test dataset. We applied three different classifiers in the classification phase (SVM, KNN and ANN) for comparing the achieved accuracy of them. KNN classifier achieved the best results in classification of separated frames. Fig.13 represents the classifiers with the selected parameters which gave the best performance for each one.



RBF Kernel with gamma = 0.48 and C = 0.5.



K = 1 with LinearNN search algorithm



Input layer contains 32 nodes equal to the frame features

Single hidden layer with 6 neurons and sigmoid activation function

Output layer with 40 neurons equal to the number of recognized signs

a) SVM Classifier b) KNN Classifier c) Neural network

Fig. 13 Classifiers Parameters

4.3 Accuracy Measures

There are two types of recognition: (1) Model-Based matching such as machine learning techniques (Classifiers as ANN, KNN, SVM, Decision Tree ...etc.) and (2) Direct matching techniques such as DTW (Dynamic Time Warping), HMM (Hidden Markov Model), maximum correlation coefficient. In our research, we tested the model by both techniques firstly with a set of classifiers and compared the results of them, then the system was tested using DTW. The results of our system are divided into two parts: the accuracy of recognition and the time of response.

4.3.1 Classifiers Accuracy

We have empirically calculated the accuracy which is defined as the ratio of correctly recognized frames to the total number of frames as in Eq.6

$$Accuracy = \frac{Total\ number\ of\ correctly\ recognized\ instances}{Total\ number\ of\ instances} * 100\% \quad (6)$$

For example, if we have a signIndex_3 with 150 frames to be classified that 30 frames classified as signIndex_1, 20 frames classified as signIndex_2 and 100 frames classified as signIndex_3 so after applying majority voting, the final predicted class will be signIndex_3. In previous example, accuracy of frames classification equals $\frac{100}{150}$ = 67 % and after applying majority voting, the accuracy equals 100 %. We used KNN, SVM and ANN classifiers in the classification phase to compare the accuracy of classification for the separated frames, we got the accuracy of 79 %, 66 % and 69 % respectively for frames classification (without majority voting) in addition to 89 %, 79 % and 87 % (with majority voting) as it is shown in Fig. 14. It is noticed that applying

the majority voting on the set of frames to get the sign meaning had enhanced the accuracy. Fig. represents the classifiers accuracy for each sign.

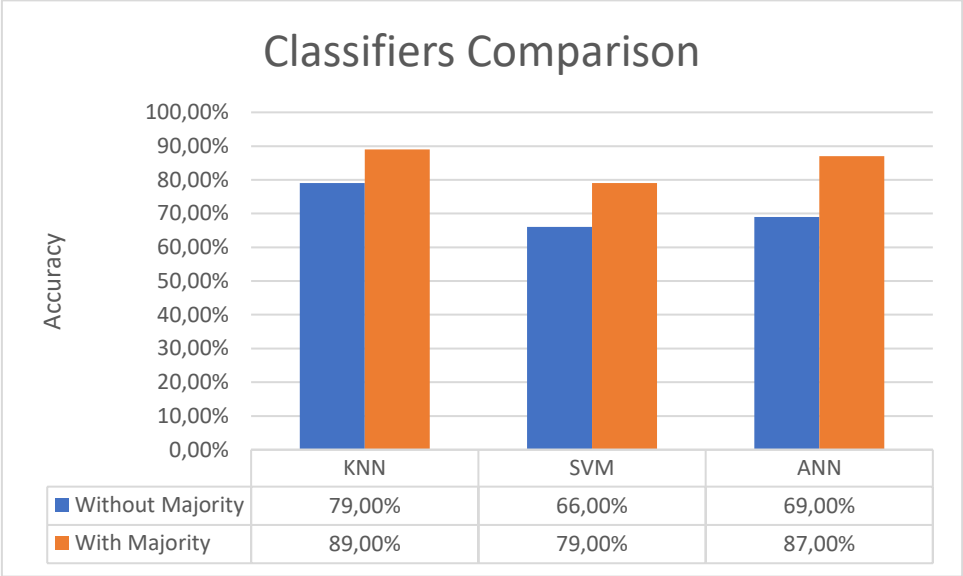


Fig. 14 Classifiers Accuracy Comparison

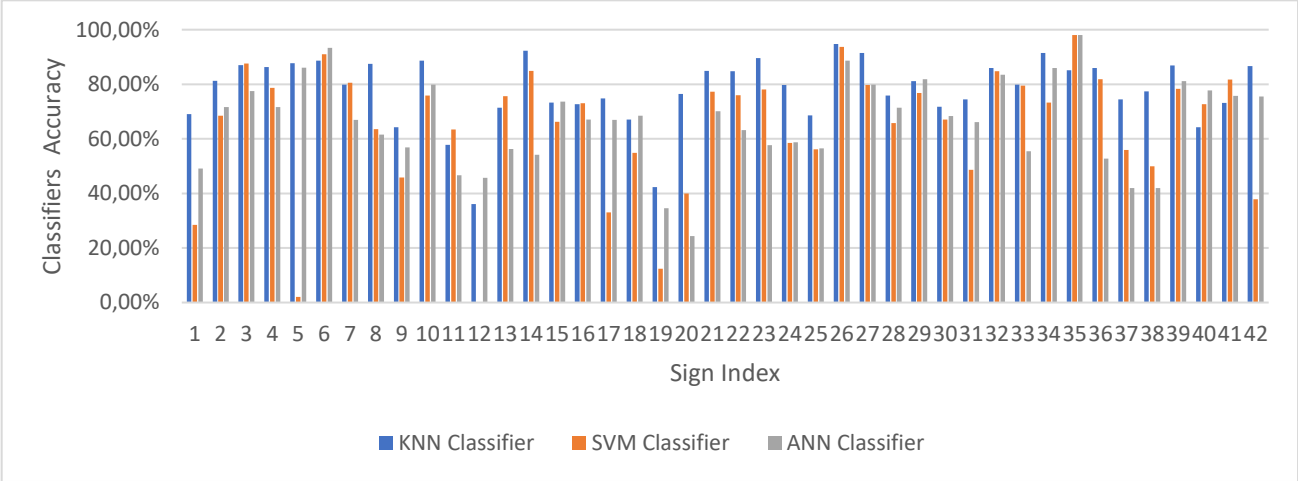


Fig. 15 Classifiers Accuracy for Each Sign

4.3.2 Classifiers Response Time

The response time means the time which is required to recognize the captured sign in real time from the time that Kinect camera captures the signer who is standing in from of its camera till the process of gesture recognition is done.

To calculate the average time which is taken by the proposed system to recognize the performed signs, we calculate the time of response for each samples of the testing data set and finally get the average of the calculated response time. Fig.16 represents the response time over the 10 samples of the test-data set for sign "الرئتان " or "Two-Lungs". It is clear that the recognition process with SVM classifier has the least response time. Also Fig.17 represents the response time for the 42 signs using the different classifiers over the test data set (10 sample per sign) .It is clear that the system takes the least response time when we use SVM in classification step (7.2 Sec.) as in average, then KNN (15.6 Sec.) finally ANN (23.2 Sec.)

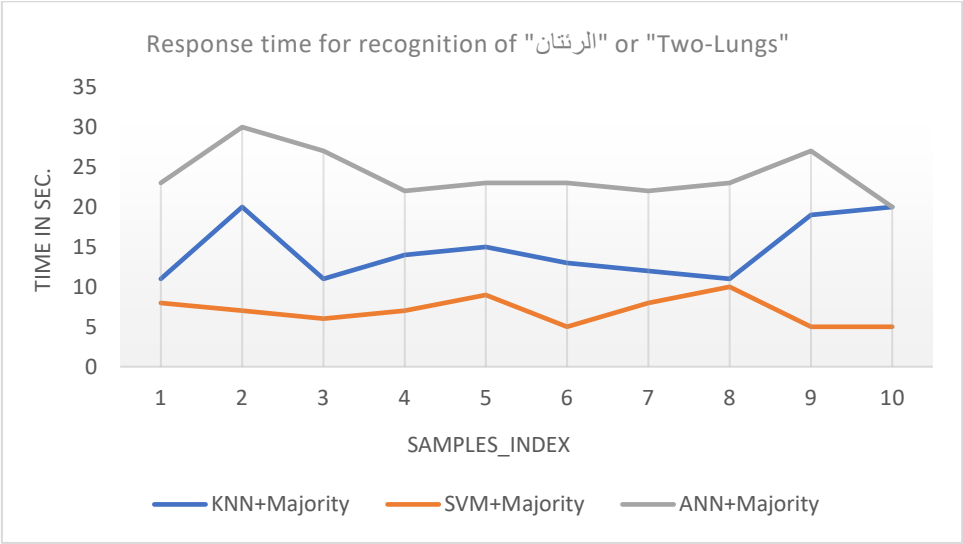


Fig.16 Response time for recognition of "الرئتان" or "Two-Lungs"

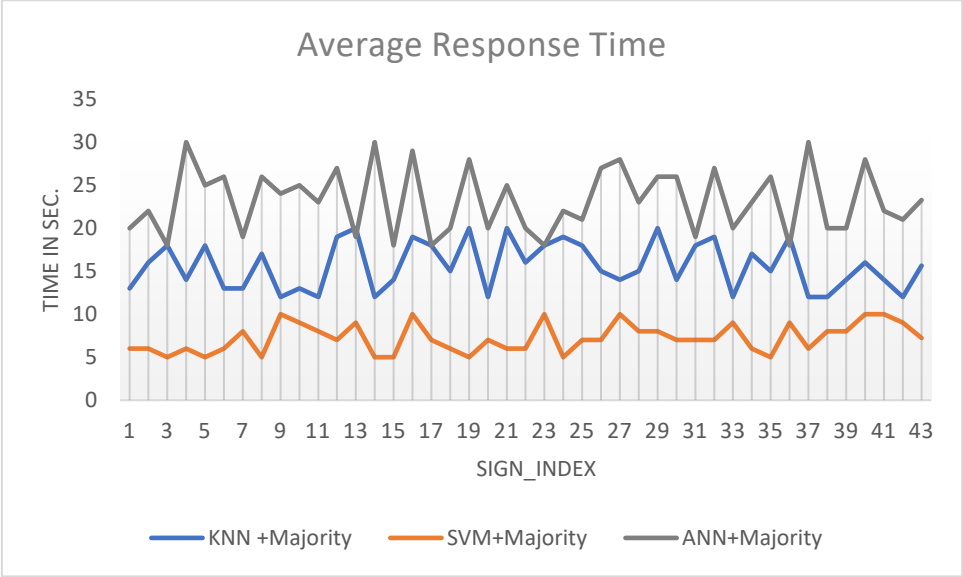


Fig.17 Average Response Time

4.3.3 DTW (Dynamic Time Warping) Accuracy

We also tested the proposed system using DTW algorithm and applied it on the same test data set to compare them with the results of applying the different classifiers. The overall accuracy that was achieved using DTW equals 77.85%. Fig.18 shows the results of applying DTW on the test samples.

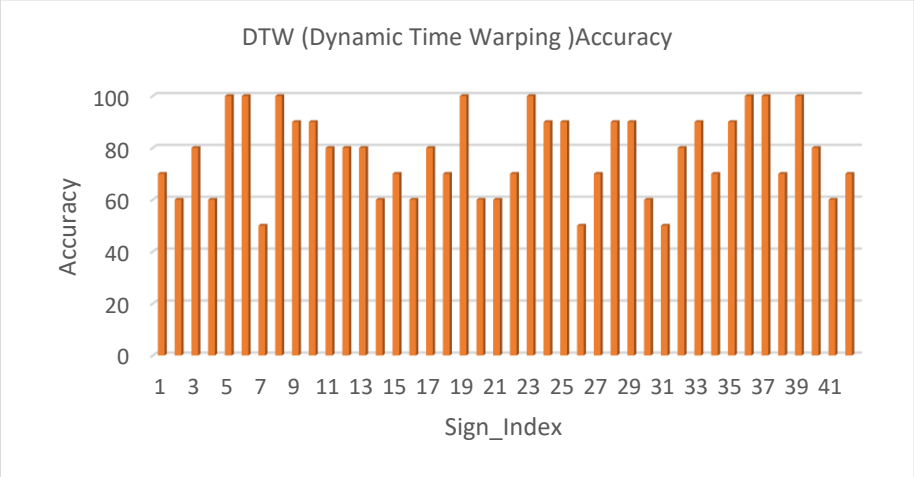


Fig. 18 DTW (Dynamic Time Warping) Accuracy

4.3.4 DTW (Dynamic Time Warping) Response Time

In this section, we calculated the response time when the proposed system operates based on DTW (Dynamic Time Warping). The test set was 10 samples for each sign (i.e.) 420 test samples, the average timing over this set was (16.38 Sec.) Fig.19 represents the average response time of DTW (Dynamic Time Warping) over the test set.

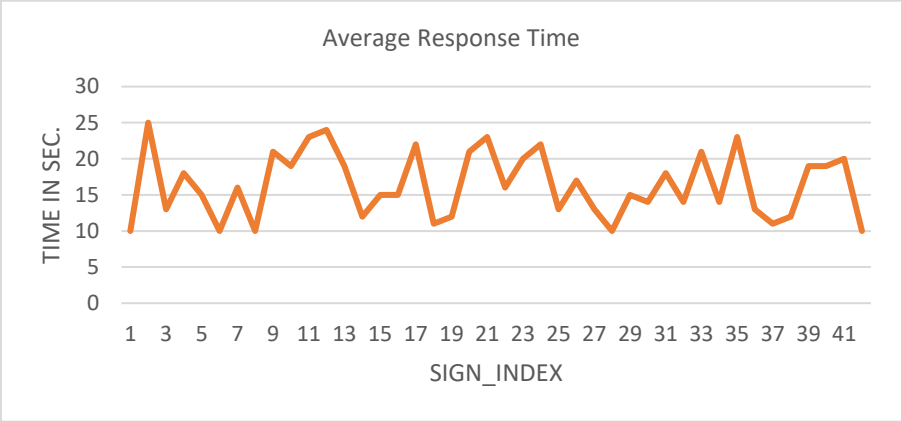


Fig. 19 DTW (Dynamic Time Warping) Response Time

4.4 Segmentation

Sign language segmentation means dividing the continuous stream of signs into the basic units(words/signs) by detecting the boundary of each sign i.e. the sign's start and end, major of the direct segmentation methods depended on the detecting the minimal speed or sign pauses ,it can be defined by the holding which occurs during performing the sign, this holding occurs for a specific length of time in the transition from one sign to another. To get the sign pauses the spatial parameters such as x, y and z coordinates should be tracked in order to compute the pause length between the signs.

The work mentioned above assumes the signs to be pre-segmented that use single sign classifier. Single sign classifier recognizes sign by sign not continuous sentences. If we deal with real-time recognition, we should use automatic segmentation method to separate between consequent

signs. Here, we introduce a simple method for segmentation depends on the motion speed during performing the continuous signs. It is noticeable that the motion speed is changed during performing the signs, when the transition from one sign to another occurred the motion speed is slowed down, so we can segment the continuous signs if we can detect the slowing down periods, the following steps represents the segmentation process:

1. Capture the data of the joints as a (3-D) three-dimensional position in (x, y, z) using Kinect and expresses these coordinated in meters.
2. Compute the selected features and record them as frames, the dynamic of each joint is computed through the variation of position over the time during performing the signs.
3. Compute the distance by calculating the difference between each frame as in Eq.7

$$D= \sqrt{(x[n]-x\,[n\,-\,1]\,)^2\,+\,(y[n]-y\,[n\,-\,1]\,)^2\,+\,(z[n]-z\,[n\,-\,1]\,)^2}, \quad (7)$$

4. Calculate the motion speed by multiplying the sampling rate in calculated distance as in Eq.8.

$$V_{motion} = D * S_f \quad (8)$$

Where,

V_{motion} is the motion speed,

D is calculated distance and

S_f is sampling rate.

5. If we detect a sequence of 20 or more observations with a speed less than or equal 5 m/s, then we will consider these observations as segmenter between two words. The start of the detected observations is considered as the end of the previous sign and the end of detected observation is considered as start of next sign.

The joints that are tracked and used in the minimal speed detection were left and right (Hand, Shoulder, Wrist, Elbow), because they are the main parts used in performing the signs, there is no point in tracking other joints because they are stationary during the sign execution.

We explained here two continuous sentences and their segmentation process: the first sentence is "اشعة الرئتان " or "Lungs X-Ray" and the second sentence. For the sentence "اشعة الرئتان" or "Lungs X-Ray" which consists of two words, Fig.20 represents the sentence performance over the time, Fig.21 represents the sentence segmentation based on the motion speed. As it is showed in the Fig. 25 there is slowing down period occurred between the two signs during the transition from the first word " اشعة/ X-Ray" and the second word "الرئتان/ Lungs" this period occurred from frame number 80 to 140 (i.e.) 60 frames this it means that the slowing down period was (2 sec.) because of the Kinect frame rate is 30 frame /sec. So, the frame number 79 is the end of the word " اشعة/ X-Ray" and the frame number 141 is the start of the word "الرئتان/ Lungs".

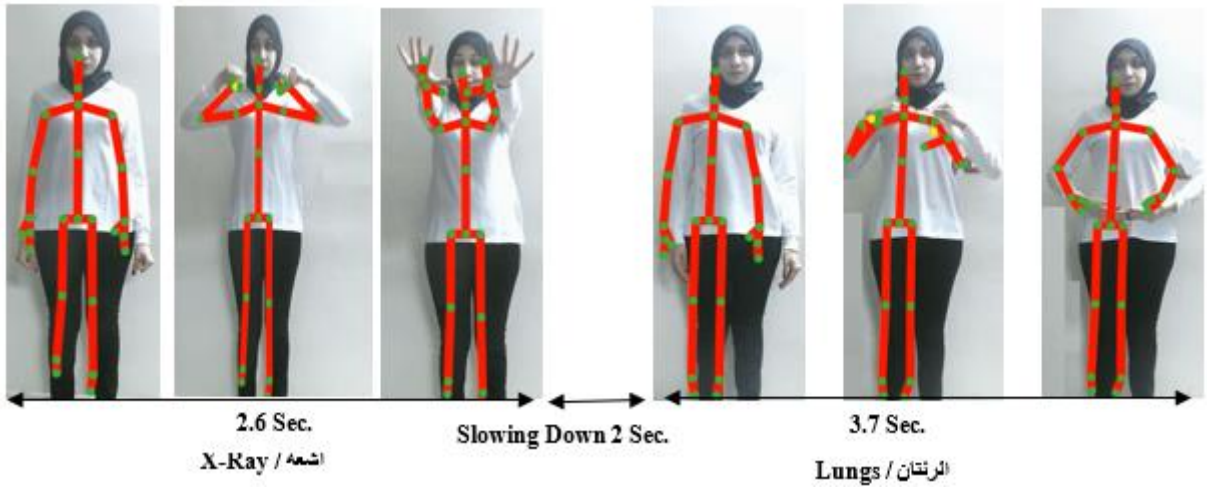


Fig. 20 Performance of Two Words Sentence "اشعة الرئتان " or "Lungs X-Ray"

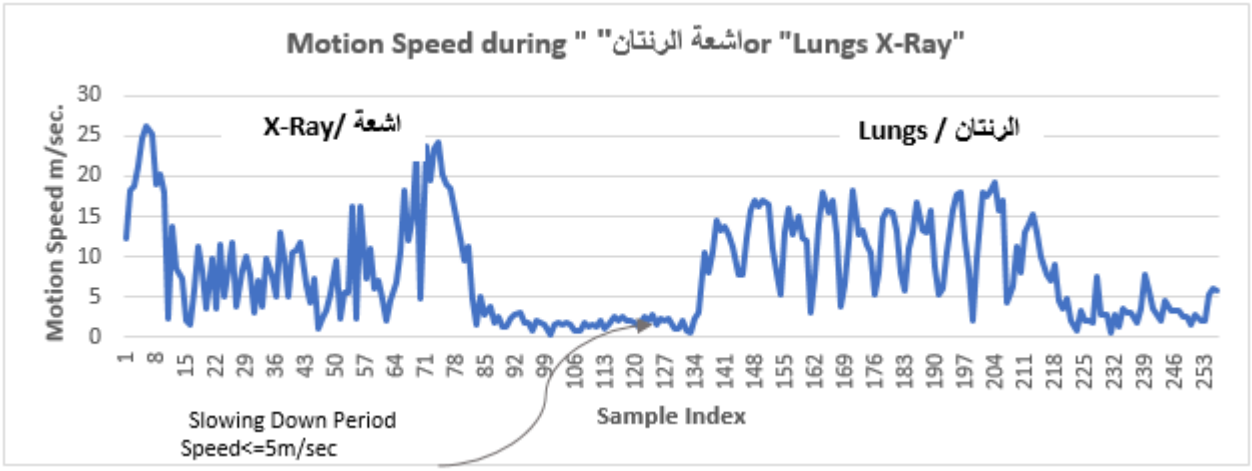


Fig. 21 Segmentation for Two words Sentence "الرئتان" or "Lungs X-Ray"

For the sentence "انا اشعر بمغص " or "I am feeling colic " which consists of three words, Fig.22 represents the sentence performance over the time, Also, Fig.23 represents the sentence segmentation based on the motion speed. As it is showed in Fig.23 there are two slowing down periods between the three words. The first period starts from 30 to 69 (i.e.) 39 frames (1.3 sec), the second period starts from 91to 123 (i.e.) 32 frames (1.06 sec.).

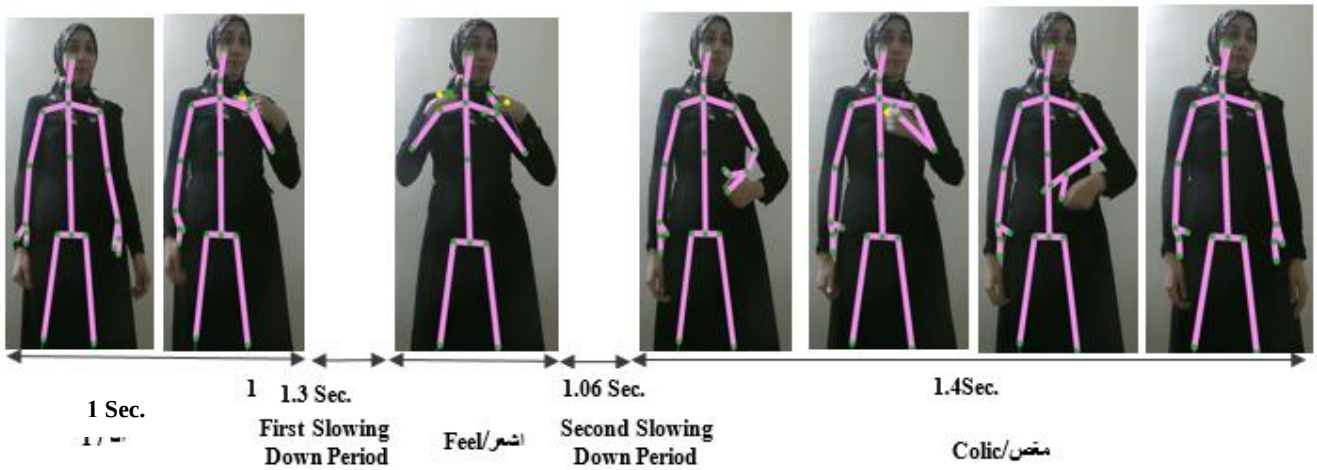


Fig. 22 Performance of Third Words Sentence"انا اشعر بمغص " or "I am feeling colic"

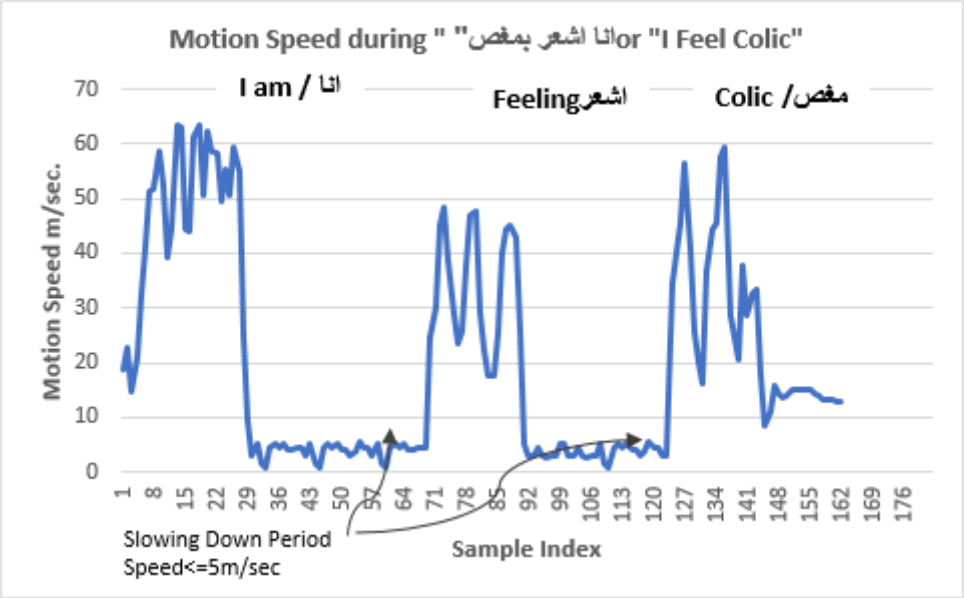


Fig. 23 Segmentation of Three Words Sentence "انا اشعر بمغص " or "I feel colic "

We tested the segmentation on over 40 sentences with different lengths composed from the Arabic signs data set which are listed in Table 5, the test focused on the ability of the segmentation method to segment the sentences which were composed from several signs correctly in real time. The result of the segmentation was satisfied and reach over 91 %, also we got the results of the recognition for the signs that were composed the tested sentences and the recognition reached 86% using the model based on KNN.

4.5. Comparative Study

The absence of comparative studies and a benchmark dataset on ArSLR makes it hard to make a comparative study using existing ArSL research. However, visible comparison was compared with CHALEARN dataset which is available on [23]. The ChaLearn 2013 dataset contains 20 Italian gestures which were recorded using Kinect, it contains (Both depth and RGB images for face and body, skeleton data, Joint orientation and position also audio sources), it includes 15,000 samples from several signers. The result of our comparison is listed in Table 2.

Table 2 Comparative Study Results

Dataset	Number of Signers	Number of Gestures	Accuracy [54]	Accuracy [55]	Accuracy [26]	Proposed Model
CHALEARN 2013	27	20	76% 7 joints were used as features set	85% Skeleton data	63.34% (4 joints)	87%

5. Conclusion

In this paper, an Arabic sign recognition system using Microsoft Kinect is proposed. The proposed system is developed and applied on medical words. Captured user's position and size are normalized to solve the variation problem in collected frames. Feature selection method is applied on collected frames and selected 32 features which are the most effective features. We applied classification algorithms like KNN, ANN and SVM to recognize captured signs. Segmentation method using motion speed is applied to segment a sequence of words with accurate manner. System perfor-

mance is analyzed using 40 Arabic words from medical field. The system was trained on 1260 samples and tested on 840 samples, the experimental results showed that the proposed system recognition rate reached 79 % for KNN classifier and enhanced using majority voting to reach 89 %. The segmentation accuracy reached 91 %. The proposed system is dynamic and robust, it can deal with any word in Arabic language depends on body motion.

For further research, the proposed system can be enhanced by applying other techniques in the recognition phase like dynamic time wrapping (DTW) that compare the sequence of each frames in training dataset with frames in testing data set. Segmentation method can be enhanced to be more accurate. Also, conducting more experiments with other words from different fields will be enhance the proposed system and make it widely used.

Acknowledgment

This research was supported and revised by my supervisors Dr. Alaa Hamouda and Prof.Dr. Ali Rashed. We thank the reviewers for their effort in reviewing and for their insights.

References

1. Al-Jarrah, Omar, and Faruq A. Al-Omari. "Improving gesture recognition in the Arabic sign language using texture analysis." Applied Artificial Intelligence 21.1 (2007): 11-33.
2. Maraqa, Manar, and Raed Abu-Zaiter. "Recognition of Arabic Sign Language (ArSL) using recurrent neural networks." Applications of Digital Information and Web Technologies, 2008. ICADIWT 2008. First International Conference on the. IEEE, 2008.
3. El-Bendary, Nashwa, et al. "ArSLAT: Arabic sign language alphabets translator." Computer Information Systems and Industrial Management Applications (CISIM), 2010 International Conference on. IEEE, 2010.
4. Hemayed, Elsayed E., and Allam S. Hassanien. "Edge-based recognizer for Arabic sign language alphabet (ArS2V-Arabic sign to voice)." Computer Engineering Conference (ICENCO), 2010 International. IEEE, 2010.
5. 2011 A survey on sign language recognition.
6. Azad, Reza, Babak Azad, and Iman Tavakoli Kazerooni. "Real-time and robust method for hand gesture recognition system based on cross-correlation coefficient." arXiv preprint arXiv:1408.1759 (2014).
7. Naoum, Reyadh, Hussein H. Owaied, and Shaimaa Joudeh. "Development of a new arabic sign language recognition using k-nearest neighbor algorithm." (2012).
8. Elons, A. Samir, Magdy Abull-ela, and Mohamed Fahmy Tolba. "Neutralizing lighting non-homogeneity and background size in PCNN image signature for Arabic Sign Language recognition." Neural Computing and Applications 22.1 (2013): 47-53.
9. SamirElons, Ahmed, Magdy Abull-ela, and Mohamed F. Tolba. "Pulse-coupled neural network feature generation model for Arabic sign language recognition." IET Image Processing 7.9 (2013): 829-836.
10. Mohandes, Mohamed, Mohamed Deriche, and Junzhao Liu. "Image-based and sensor-based approaches to Arabic sign language recognition." IEEE transactions on human-machine systems 44.4 (2014): 551-557.
11. Almasre, Miada A., and Hana Al-Nuaim. "A Real-Time Letter Recognition Model for Arabic Sign Language Using Kinect and Leap Motion Controller v2." International Journal of Advanced Engineering, Management and Science (IJAEMS) 2.5 (2016).
12. ElBadawy, Menna, et al. "A proposed hybrid sensor architecture for arabic sign language recognition." Intelligent Systems' 2014. Springer, Cham, 2015. 721-730.

13. Aliyu, S., et al. "Arabie sign language recognition using the Microsoft Kinect." Systems, Signals & Devices (SSD), 2016 13th International Multi-Conference on. IEEE, 2016.
14. Jmaa, Ahmed Ben, et al. "Arabic sign language recognition based on HOG descriptor." Eighth International Conference on Graphic and Image Processing (ICGIP 2016). Vol. 10225. International Society for Optics and Photonics, 2017.
15. Almasre, Miada Ahmmed, and Hana A. Al-Nuaim. "Using the Hausdorff Algorithm to Enhance Kinect's Recognition of Arabic Sign Language Gestures." International Journal of Computer Science and Security (IJCSS) 7.1 (2017): 2.
16. Mohandes, Mohamed, Mohamed Abdelouaheb Deriche, and Salihu Oladimeji Aliyu. "Arabic sign language recognition using multi-sensor data fusion." U.S. Patent No. 9,672,418. 6 Jun. 2017.
17. Almasre, Miada A., and Hana Al-Nuaim. "Comparison of Four SVM Classifiers Used with Depth Sensors to Recognize Arabic Sign Language Words." Computers 6.2 (2017): 20.
18. Premaratne, Prashan. "Effective hand gesture classification approaches." Human Computer Interaction Using Hand Gestures. Springer, Singapore, 2014. 105-143.
19. Gani, Eriglen, and Alda Kika. "Albanian Dynamic Dactyls Recognition using Kinect Technology and DTW." Proceedings of the 8th Balkan Conference in Informatics. ACM, 2017.
20. Lekova, Anna, D. Ryan, and Reggie Davidrajuh. "Fingers and Gesture Recognition with Kinect v2 Sensor." Information Technologies and Control 14.3 (2016): 24-30
21. Ahmed, Washef, Kunal Chanda, and Soma Mitra. "Vision based hand gesture recognition using dynamic time warping for Indian sign language." Information Science (ICIS), International Conference on. IEEE, 2016.
22. Bautista, Miguel Angel, et al. "Probability-based dynamic time warping for gesture recognition on RGB-D data." Advances in Depth Image Analysis and Applications. Springer, Berlin, Heidelberg, 2013. 126-135
23. https://www.kaggle.com/c/multi-modal-gesture-recognition
24. Escobedo-Cardenas E, Camara-Chavez G (2015) A robust gesture recognition using hand local data and skeleton trajectory. In: International Conference on Image Processing, pp 1240–1244
25. Chen, Xi, and Markus Koskela. "Using appearance-based hand features for dynamic rgb-d gesture recognition." Pattern Recognition (ICPR), 2014 22nd International Conference on. IEEE, 2014.
26. Kumar, Pradeep, et al. "A position and rotation invariant framework for sign language recognition (SLR) using Kinect." Multimedia Tools and Applications 77.7 (2018): 8823-8846.