Arabic Dynamic Gestures Recognition using Data Fusion

1Basma Hisham and 2Alaa Hamouda

1Teaching Assistant at Computers & Systems Department, Faculty of Engineering, Al-Azhar University, Cairo, Egypt
2Assistant Professor Dr. at Computers & Systems Department, Faculty of Engineering, Al-Azhar University, Cairo, Egypt

Abstract: Sign language is a visual language which is the primary way used by deaf people in order to connect and communicate with each other and with their societies. There are some studies have been done on Arabic sign language recognition (ArSL) systems and practically deployable system for real-time use is still a challenge. The main objective of this paper is to develop a novel model which is able to recognize the Arabic Sign Language using Microsoft's Kinect V2. This paper works on the dynamic gestures which are performed by both hands and body parts, it introduces an effective way of capturing and detecting the hand and skeleton joints from the depth image which is provided by Kinect. The model used two supervised machine learning algorithms: Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) then applied DSmT (Dezert-Smarandache Theory) as a fusion technique in order to fuse their results, finally we applied two most widely methods which are used with dynamic gestures recognition: a) DTW (Dynamic Time Warping) and b) HMM (Hidden Markov Model) to compare their results with the previous classifiers fusion. We applied the model on ArSL dataset which is composed of 40 Arabic medical signs to ease the communication between hearing impaired patients and their doctor. The accuracy of the model is improved when applied the classifiers fusion as compared to the use each classifier separately. The overall accuracies for SVM, KNN and DSmT fusion are 79%, 89% and 91% respectively and both DTW and HMM achieved 82.6% and 79.5 respectively.

Key words: Sign language • Kinect • Dynamic Gestures • KNN • SVM • DSmT

INTRODUCTION

Sign language is the most basic way for deaf people to connect and interact with hearing impaired persons and integrate with their societies. The main problem is that most of the normal people do not understand sign language [1]. Therefore, the need to develop an automatic system which is capable of translating sign languages into understandable words and sentences is becoming very necessary. There are two main approaches for sign language recognition systems: vision-based and sensor-based. The main advantage of vision-based system is that there is no need to use complex devices so it is low cost and no need to pre-setup, but this approach requires extra calculations in the preprocessing stage, image processing and artificial intelligence to recognize and interpret signs. Also, it suffers from the background problems because it needs subtraction techniques to subtract the signer from the background and it may fail if the background changes. Sensor based systems provide robust, reliable and more accurate data but it is not user-friendly like vision-based systems because it requires extra equipment like data-gloves, the user is required to wear the gloves in order to collect the data so it is not practical [2]. Microsoft Kinect is a motion sensing input device which is developed by Microsoft. It provides live streams of depth information about the skeleton joints and body motion. This information is essential to create the 3D view of the tracked objects. It used to track standing skeleton with high-depth fidelity so that by comparing Kinect sensor with other depth sensors, Kinect is the best choice in short range environment [3]. Kinect also 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 [4]. Several approaches have been proposed for sign recognition, the most important and effective approach is using machine learning algorithms. It can handle the complexity and the differentiation of sign language gestures [5], also it can handle the different manner in which the people repeat different signs [6]. Machine learning algorithms such as neural networks, support vector machines, k-nearest neighbors, decision tree models...etc. have been focused on the classification stage of recognizing a gesture after captured from the signer. The 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 information processing in sign languages. Motion speed during capturing of continuous sentences may be 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 identifies the Arabic signs which are captured by Microsoft Kinect based on the data that represents of body and hand motion. These data will be excluded from the depth image information got from Kinect sensor. However, Kinect cannot accurately detect the hand movement and also the details of fingers but we overcome such limitations and introduced a very effective and simple method for hand detection. Also, we used two machine learning algorithms: K- Nearest Neighbors (KNN) and Support Vector Machine (SVM) and introduced a very effective fusion method based on DSmT (Dezert-Smarandache Theory) to fuse the classifiers results and enhance the accuracy, also we applied two direct matching algorithms: DTW (Dynamic Time Wrapping) and HMM (Hidden Markov Model) to compare the results of fusion with other dynamic gestures recognition techniques. The structure of this paper is organized as follows. Section 2 presents the related work of the sign language recognition. The methodology is presented in Section 3. The experiments setup and results are presented in section 4. Section 5 contains conclusion & future work.

Related Work: Sign language is a combination of words that are represented by using movements of different body parts such as head, shoulders, elbow, wrist...etc. and finally added to the hand signs to create a meaning [7-11]. Many researchers aimed to build an automated system in order to translate ArSL to Arabic text or voice. Arabic sign language researches faced several difficulties such as it is not defined well, the works in it started at the last decade. However, in Arabic Sign Language there are more than 9000 signs and it uses 26 handpostures and 5 dynamic gestures in order to represent the Arabic alphabet. There is a variation in ArSL between the different Arabic countries. Some Arabic countries have their own sign language such as Tunis, Gulf countries, Egypt...etc. We are concerned with Arabic Sign Language in Egypt. The organization of Arabic Sign Language in Egypt has started in 1983, there are 7 million HI (Hearing impaired) persons upon to the last studies performed by United nations, this is a large number so that they are need to be merged with their societies as any normal person. Several studies depend on different machine learning approaches to develop auto-recognition system for Arabic Sign Language. In 2001, Mohandes developed an Arabic sign recognition model for Arabic alphabets, after feature extraction he applied Hu’s moments on features vector followed by support vector machines (SVMs) as a classifier with accuracy 87% [12]. Ahmed et al. proposed a model for sign-language recognition. Several statistical analyses were performed on the data extracted from the collected images to create the features vector which is the input to an SVM. The model was tested on 10 letters and the accuracy was 83%. They suggested to build a real time system which is able to work on both dynamic and static signs [13]. Maraca and Abu-Zaiter developed a static and dynamic ArSL recognition system. The gestures were performed by the human hand. They used feed-forward and recurrent neural networks on the features extracted from the captured images. They tested the proposed system on 30 gestured performed by two persons that wear color gloves, the database contains 900 samples, the accuracy rate reached 95% [14]. Assaleh and Al-Rousan developed Arabic sign language recognition for alphabet signs. They used polynomial classifiers which has several advantages over other classifiers such as it does not need to iterative training also it is very accurate and scalable. They compared the results of the system with the previous published results using ANFIS-based classification on the same dataset and feature extraction method. The comparison showed significant improvement, the misclassified patterns was reduced by 36% on the training set and by 57% on the test set [15]. Al-Jarrah and Al-Omari developed an automatic system for translating the Arabic
al phabets gestures. The bare hands images are processed in order to extract the features such as translation, rotation and scaling invariant. The system was tested on 30 Arabic alphabet signs and achieved 97.5% accuracy [16]. El-Bendary et al. proposed a sign language recognition system for the Arabic alphabets, it dealt with the images of bare hands which allow the user interacting naturally with the system and achieved an accuracy of 91.3%. The proposed Alphabets Translator (ArSLAT) didn’t 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 [17].

Hemayed and Hassanien proposed hand gestures recognition system for Arabic alphabets and convert it to a voice, the proposed system focused on static gestures which they do not need any movements. PCA (Principal Component Analysis) algorithm is applied to the extracted edges that form the predefined feature vectors for signs library. They used Euclidean distance to measure the similarity between the signs’ features, the nearest sign is selected and the corresponding sound clip is played. The system is applied on more than 150 signs and gestures with accuracy near to 97% at real time test for three different users [18]. Almasre et al., proposed hand gesture recognition systems in order to recognize the Arabic sign language, they used a supervised machine learning to predict the hand pose using two sensors: Microsoft Kinect and Leap Motion Controller depending on the depth images. They define a classifier to transform gestures based on 3D positions of a hand-joints direction into their letters. They collected data about 28 letters from different signers and the result reached about 100% detection rate in recognizing 22 letters from 28 Arabic letters [19]. El-Badawy et al., proposed a system that integrates a set of different types of sensors to capture all sign features. They used Leap Motion in order to capture hands with fingers movements, also they used two digital cameras to capture face features and body movement. They applied their system on 20 dynamic signs and the system achieved accuracy of 95% [20]. Aliyu et al. proposed a Kinect based system for Arabic sign language recognition, the system was applied on 20 signs, they collected video samples of both true color images and depth images from different volunteers. They used Linear Discriminant Analysis (LDA) for features reduction and sign classification. Furthermore, fusion from RGB and depth sensor was carried out at features and the decision level performed an overall accuracy of 99.8% [21]. 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 involved 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% [22]. Mohandes et al., developed a new model for Arabic sign language recognition in order to detect and track at least on hand and one finger, two different sensors in two different location in a room that generate 3-dimensional (3D) interaction space. They used a classifier integrated with two different sensors - Leap Motion Controllers (LMC) and Microsoft Kinect - and 28 Arabic alphabet signs are performed in the interaction space [23]. Almasre et al. proposed a model to recognize the hand gestures of Arabic Sign Language (ArSL) words using two depth sensors. They examined 143 signs gestured by 10 users for 5 ArSL words. The sensors captured depth images of the upper human body, from which 235 angles (features) were extracted for each joint and between each pair of bones. The dataset was divided into a training set (109 observations) and a testing set (34 observations). They used support vector machine (SVM) classifier with different parameters in order to proceed four SVM models, with linear kernel (SVMLD and SVMLT) and radial kernel (SVMRD and SVMRT) functions. The accuracy of the words in 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 [24].

Several sign language recognition works were performed based on data fusion, Rashid et al., (2009) developed a multimodal in order to combining both of the gestures and postures for recognizing alphabets and number, the fusion was done on the decision level. The gestures recognition system is trained and learned using HMM and it concerned with the dynamic motion and the posture recognition system is trained and learned using SVM and it concerned with the static hand at the same time, they applied Gaussian distribution on the captured 3-D depth
information in order to detect and segment the gestures and postures. Then, feature vectors are constructed from extracted from spatial and temporal hand properties. Finally, they used the rule of AND / OR combination to state the decision, the model achieved 98% for alphabets and numbers gestures and achieved 98.65% for ASL alphabets and 98.6% for both for ASL numbers [201]. Song et al., (2013) introduced a model of gestures recognition based on using Microsoft Kinect, the 3-D position data about all body skeleton joints are captured using Kinect, then the features of interest for each gesture were extracted, then they segmented the features from the captured video using two algorithms, firstly elliptical Fourier descriptions and secondly, Principle component analysis (PCA) and finally performed fusion on the level of features and applied a Sugeno type fuzzy inference system, the system was applied on 80 common Indian signs and achieved accuracy of 96% [203]. Penelle et al., (2014) proposed data fusion system using Leap Motion and Microsoft Kinect sensors to improve hand recognition accuracy [204]. ElBadawy et al., (2015) proposed a hybrid system using Leap motion and two digital cameras. They used leap motion for fingers tracking and the digital cameras used for body movements tracking with facial emotions. The proposed system applied by Neural Network (ANN) on 20 Arabic signs with an accuracy of 95% [205]. Marin et al., (2016) proposed a framework to recognize static American signs. They used leap for fingers and capturing features based on distance while Kinect is used for capturing features based on body and correlation. The proposed system applied SVM with an accuracy of 91% [206]. Fok et al., (2015) proposed data fusion system based on two devices. Kalman filter is used for fusion and HMM is used for sign recognition. They applied the system on 10 American digits [207]. Yang et al., (2017) proposed an optimized framework based on a tree-structure classification model using three sensors sEMG, ACC and GYRO to get the best performance as single sensor, two-sensors fusion and three-sensors fusion. The final recognition rate was 94.31% and 87.02% were obtained for 150 Chinese Sign Language for two test scenarios [208]. Sadek et al., (2017) proposed a hand gesture recognition using a smart glove which was designed from a set of sensors, the recognition was based on a statistical analysis of the hand shape of during performing the 1300 words of the Arabic sign language (ArSL) [209]. Kumar et al., (2017) proposed multi-sensor fusion framework for Sign Language recognition using Coupled Hidden Markov Model (CHMM). They used Microsoft Kinect and Leap Motion [210]. Sun, Ying, et al., (2018) proposed a weighted fusion method based on D-S evidence theory. The proposed recognition method depends on Kinect and sEMG signal. The average recognition rate was about 87 % [211]. Mohandes et al proposed Arabic sign language recognition for the Arabic alphabets using two Leap Motion controller and applied DST (Dempster–Shafer theory), they tested the system using 10 -cross validation, the first LMC achieved 93.077 % accuracy, the second LMC achieved 89.907% then they applied the DST on the features level and on the decision level and the achieved accuracy reached 97.686 % and 97.053% respectively [212]. The main contribution of our proposed model is applying the data fusion on the decision level by combining the results of the two classifiers K- Nearest Neighbors (KNN) and Support Vector Machine (SVM) using an effective fusion technique (Dezert-Smarandache Theory), this combination enhances the accuracy of the system and make it more accurate and robust rather than depending on single classifier , we have to mention that Arabic sign language is not a unified language and is varied from one country to another so we focused on the Egyptian Arabic sign language which is most generally comprehended by Arabs. The works of data fusion in sign Language recognition especially the Arabic sign language is very rare, so our research introduces a way for improving the existing sign language recognition systems by applying the concept of data fusion techniques which make the system robust and more reliable.

**MATERIALS AND METHODS**

In this section, we will discuss the structure of our proposed system for Arabic sign language recognition usingMicrosoft Kinect. We describe the various phases of our system from capturing the gestures using Kinect till the gesture recognition. The structure of the proposed model is shown in Fig. 1. The first step in the proposed model is (1) Data acquisition phase which is occurred when the Kinect depth camera started to capture the skeletonof standing singer in front of the Kinect camera and infer his/her skeleton positions, the system receives joints information such as type and coordinates, bone
orientation and motion velocity as a stream of frames, (2) Preprocessing phase includes: a) Extracting the features of interest for both signer skeleton joints and signer hands, b) Normalization is applied on the collected frames to overcome mainly two problems, firstly the variation of user position and secondly the variation of users' sizes and c) Features integration for fusing the hand features with the skeleton features in order to form the final features vector, (3) Applying two classifiers SVM (Support Vector Machine) and KNN (K-Nearest Neighbors) each one works separately on the features vector, these classifiers work as two sources of information and the results of each classifier can be considered as the BBA (Basic Belief Assignment), (4) Applying the late fusion using DSmT (Dezert-Smarandache Theory) to fuse the BBA of both classifiers SVM and KNN, this includes applying the BBA fusion, applying PCR 5 rule, calculating the pignistic probability and finally recognize the performed sign according to the pignistic probability, (5) Applying the direct matching techniques DTW (Dynamic Time Wrapping) and HMM (Hidden Markov Model) instead of the fusion model as an alternative method for recognition in order to compare between the fusion model and other dynamic gestures recognition techniques.

**Data Acquisition:** In this step, we used Microsoft’s Kinect Version 2.0 to track the skeleton joints of the standing signer. Kinect provides the color, depth and joint coordinates information using its open-source SDK. The depth information is captured frame by frame. So, when Kinect depth camera starts, we capture the coordinates of 20 skeleton joints with a rate of 30 frames per second. In our system, we interested in the upper human joint points as in Fig. 2.

Fig. 2: Points of the upper human body joint
1—Spin, 2—Shoulder Center, 3—Head, 4—Right Shoulder, 5—Right Elbow, 6—Right Wrist, 7—Right Hand, 8—Left Shoulder, 9—Left Elbow, 10—Left Wrist, 11—Hand Left
Fig. 3: Position Normalization

Fig. 4: Spherical coordinates

Preprocessing: In this step, we concerned with the features extraction, preparation and normalization for both skeleton and hands of the signer.

Features Extraction: The features extraction step has a very important role in distinguishing between the captured signs. The features are extracted from the sequences of depth information. The extracted features from Kinect frames are divided into two parts: a) Skeleton joints features, b) Hand features.

Skelton Joint Features: Kinect has ability to infer the positions of the detected objects, after studying the selected signs carefully, we found that only 10 joints of the skeleton are required to represent and describe the sign. These joints are: Hand (Left & Right), Shoulder (Left & Right), Elbow (Left & Right), Wrist (Left & Right), Spin Mid and Head Center. All signs are represented and performed with the upper part of body and the lower part will remain static during performing the sign. The captured frames are required to be normalized in order to overcome the variation in signer's position and signer's size.

Position Normalization: The signer can be in any position during performing the sign as in Fig. 3 and this variation can make a conflict to the model, so we performed the position normalization. The captured coordinates (X, Y, Z) for any joint are scaled by subtracting them from the spin-mid coordinates.

The coordinates of the selected joints will be converted from Cartesian coordinates X, Y and Z, into spherical coordinates which are represented by (r, θ, φ) as in Fig. 4.

The computation of the spherical coordinates is illustrated in the following equations:

\[
\sum_{i=1}^{n} (l) = \sqrt{(J(i)_{x} + S_{M_{x}})^2 + (J(i)_{y} - S_{M_{y}})^2 + (J(i)_{z} - S_{M_{z}})^2}
\]

(1)

\[
\sum_{i=1}^{n} \theta(i) = \tan^{-1}\left(\frac{\sqrt{(J(i)_{x} - S_{M_{x}})^2 + (J(i)_{y} - S_{M_{y}})^2}}{J(i)_{z} - S_{M_{z}}}\right)
\]

(2)

\[
\sum_{i=1}^{n} \varphi(i) = \tan^{-1}\left(\frac{J(i)_{y} - S_{M_{y}}}{J(i)_{z} - S_{M_{z}}}\right)
\]

(3)

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.

Size Normalization: To overcome the problem raising from the variation of user's size, we normalized all the distances that are resulted from the position normalization step by the factor, in our model we chose this factor as \( r_{HS_{M}} \) which is the distance between the head and spin-mid as in Eq. 4.

\[
\sum_{i=1}^{n} r_{\text{norm}}(i) = \frac{r(l)}{r_{HS_{M}}}
\]

(4)

Where,

n is the number of joints from J,

r_{norm} is a normalized radial distance of the joint,

r_{HS_{M}} is a radial distance from head center to spin-mid.
Finally, we selected another subset features added to the spherical coordinates of the selected joints in order to enhance the recognition process as the difference in distance between hand (left & right) and shoulder (left & right). The total number of Kinect features ($f$) are about 32 features in spherical coordinates. These features are denoted by $(f_s, f_h, f_e, ..., f_b)$ where the feature-vector is comprised from:

- $\{r, \emptyset\}$ of right, left $\{\text{hand, wrist, elbow, shoulder}\}$ position.
- $\{r\}$ of separation between right and left $\{\text{hand, wrist, ...}\}$

**Hand Features:** Adding the hand features to the skeleton features will give a complete view and accurate description of the performed sign. The extracting of the hand features is based on the algorithm in [38]. The methodology of hand features extraction starts by detecting hand joints of the tracked human body, the detected coordinates $(x, y, z)$ for the hand represent the palm center. The next step is to specify the search area of the hand which the hand is lied in, this 3D area can be limited by the captured hand and tip position as in Fig. 6, after specifying the search area all depth values that does not belong to the hand area can be excluded. The fingers can be detected by applying the algorithm of the convex hull on the search area, the edges of the convex hull above the wrist represents the fingertips as in Fig. 7.

Finally, the total number of hand features ($f_h$) are about 30 features in Cartesian coordinates. These features are denoted by $(f_{h1}, f_{h2}, f_{h3}, ..., f_{h30})$ where the feature-vector is comprised from: fingers tip positions which is composed of 3D data $<x_i, y_i, z_i>$. After fingertips positions are detected, the fingers direction vectors can be easy calculated by subtracting the tip position of each finger from the palm center $P_c(p_x, p_y, p_z)$. The vectors which are pointing from the palm center to fingers can be calculated using Eq. 5.

$$V_{Direction} = (f_{x} - p_{x}, f_{y} - p_{y}, f_{z} - p_{z}) \quad (5)$$

**Features Integration:** Features integration is the process of integrating the features vector of both skeleton joints features ($f_s$) and hand features ($f_h$) in order to produce the fused vector $f = \{f_s, f_h\}$. The resultant fused features vector has a dimension of 62. It should be mentioned that data sequence is synchronized perfectly because they are coming from the same device.

**Classification:** Once the gesture features have been extracted, the descriptor of gestures that the system must classify will be formed. The goal of our system is to recognize the gestures, so after extracting the features, we applied two classifiers KNN (K-Nearest Neighbour) with K=1 and SVM (Support Vector Machine) with RBF kernel function (gamma = 0.48 and Cost = 0.5). We chose these classifiers after applying different classifiers on the test set, they gave us the best accuracy, also they are used widely in many pattern recognition applications as the handwritten digit recognition [37] and they are efficient in dealing with multiclass nonlinear classification problems. These two classifiers work as two sources of information, it is better to depend on two classifiers in order to improve the overall accuracy than using one classifier. The combination of information from different sources is critical especially when developing a system that depends on conflicting, imprecise and uncertain data. In the proposed model each classifier takes the sequence of frames that formed the single pre-segmented gesture and classify each frame separately to predict the class that frame belongs to. However, there are similarities between some gestures so that for example if frames of gestures enter to the classifier, the output may be
classified by 70% of frames as Sign_ID= “1”, 10% of frames as Sign_ID =4 and 20% of frames as Sign_ID =8. These values were considered as BBA (Basic Belief Assignment) which will enter to the fusion phase.

To define (BBA) let X be the universe that represents all possible states of a system under consideration. In the evidence theory, the basic belief assignment (BBA) assigns belief mass to each element of the power set \( 2^X \) formed from the underlying universe X. We can consider the function \( m: 2^X \rightarrow [0,1] \) as a basic belief assignment, when two conditions occurred:

- The mass of the empty set is (0) (i.e.) \( m(\emptyset) = 0 \) and
- The masses of the remaining members of the power set add up to a total of 1 (\( \sum_{A \in 2^X} m(A) = 1 \)).

**Late Fusion:** The late fusion occurred byfusing the results of the two classifiers SVM and KNNand applying the rulesof Dezert-Smarandache Theory (DSmT). The fusion of these classifiers was done on the measurements level which is more confident. The evidence (results of the classifier) is considering as BBA.

**Dezert-Smarandache Theory (DSmT):** Dezert-Smarandache Theory (DSmT) is a very effective fusion method, it can deal with the uncertainty and the data coming from highly conflict sources. It allows the combination of information which is coming from different independent sources, this information is represented in terms of belief function. Dezert-Smarandache rules combine the conflict evidence accurately, so it is very successful in problems of object recognition [39].

DSmT is a theory of plausible and paradoxical reasoning, it overcame the limitations of DST (Dempster-Shafer theory) [40]. We can summarize the comparison between DST (Dempster-Shafer theory) and Dezert-Smarandache Theory (DSmT) as a following:

Let \( \vartheta = \{\vartheta_1, \ldots, \vartheta_n\} \) is a finite set of hypotheses;

- The DST considers a discrete and finite frame of discernment \( \Theta \) based on a set of exhaustive and exclusive elementary elements \( \Theta \).
- The bodies of evidence are assumed independent and provide their own belief function on the power set \( \Theta \) but with same interpretation for \( \Theta \) [40].

DSmT has two types of models: (1) Free model in which combine the evidence without taking the integrity constraint into consideration, (2) Hybrid model in which includes all operators such as union and intersection and the constraints that are required to build the class \( \Phi \), so it is used in a real application.

Based on that model, the hyper-power set is given by:

\[ \vartheta = \{\vartheta_1, \ldots, \vartheta_n\} \] as a finite set (called frame) of n exhaustive elements. The free Dedekind’s lattice denoted hyper-power set \( D^\emptyset \) is defined as:

- \( \emptyset, \vartheta_1, \ldots, \vartheta_n \in D^\emptyset \)
- If \( A, B \in D^\emptyset \), then \( A \cap B \) and \( A \cup B \) belong to \( D^\emptyset \).
- No other elements belong to \( D^\emptyset \), except those obtained by using rules 1 or 2. [39].

**Basic Belief Assignment:** For any finite discrete frame \( \vartheta \), we define a belief assignment as a mapping \( m: \Theta \rightarrow [0,1] \) associated to a given body of evidence, \( B \), that satisfies the following conditions which is represented in Eq. 6:

\[ m(\emptyset) = 0 \text{ and } \sum_{A \in G^\emptyset} m(A) = 1 \] (6)

In Eq. (6), \( m(A) \) is the generalization of basic belief assignment/mass (bba) where \( A \) and the belief function is defined as:

\[ Bel(A) \equiv \sum_{B \subseteq A} m(B) \] (7)

In DsmT theory there are a 2-level process: credal (for combination of evidences) and pignistic (for decision-making), i.e. when we need to take a decision, we should depend on a probability function. The Classical Pignistic Probability Transformation (CPT) is defined as [39]:

\[ BetP[A] = \sum_{X \in 2^\emptyset} \frac{|X \cap A|}{|X|} m(X) \] (8)

Where \( |x| \) denotes the cardinality of \( x \) (with convention \( |\emptyset|/|\emptyset| = 1 \), when defining \( BetP[\emptyset] \)). Decisions are achieved by computing the expected utilities of the acts using the subjective/pignistic \( BetP[.] \) as the probability function needed to compute expectations. It is easy to show that \( BetP[.] \) is a proper probability function [39].

**Fusion Frame Work:** As it was introduced we used the evidence theory of Dezert-Smarandachethe beliefs of each evidence and the second is applying the combination rule. We summarize the fusion framework in Fig. 8.
The belief calculation is computed using Eq. 6 and 7 then the conflict is redistributed using PCR5 (Proportional Conflict Redistribution) rule which is the mathematical form to redistribute the conflicting mass to non-empty sets as in Eq. 9, finally calculate the pignistic probability using Eq. 8 in order to decide the performed sign according to the highest probability.

\[ m_{PCR5}(X) = m_{12}(X) + \sum_{Y \in \Theta(X)} \left( m_2(X)^T m_2(Y) \right) \begin{pmatrix} m_2(X)^T m_1(Y) + m_2(Y) m_1(X) + m_1(Y) m_2(X) \end{pmatrix} \]

(9)

The data set contains 40 signs so that it is divided into four parts in order to simplify the calculation. Sign_ID = “1” is chosen as a common sign between the divided data sets in order to relate them with each other. When the test sign enters to the system, it will pass four stages of fusion with each divided data set. The goal is to calculate the ranked pignistic probability in order to recognize the performed sign.

The following calculation represents “tested sign with sign_ID=1” when entered to the fusion frame work. As it was mentioned previously, the first step of the model is applying the classification using the two classifiers SVM and KNN as two sources of information, Table 1 represents the results of the two classifiers which is considered as BBA (Basic Belief Assignment). The first stage of fusion is done with the first data set which contains the signs with ID = 1, 2, …, 10.

The second stage is applying the classical DSM combination rule, which states that, Table 2 represents the fusion of the beliefs after applying the fusion rules:

\[ m_{12}(A) = \sum_{X_1, X_2, \ldots, X_k \in D^k} \prod_{l=1}^{k} m_{l}(X_l) \]

(10)

Consequently, redistribute the conflict factor using PCR5 rule

Redistribute: “2 X 4 = \emptyset”

So, we will distribute this conflict proportionally,

\[ m_{PCR5}(1) = m_1(1), m_2(1) = 0.8 \times 0.75 = 0.6 \]

\[ m_{PCR5}(2) = m_1(2), m_2(2) = 0 \times 0.25 = 0 \]

\[ m_{PCR5}(4) = m_1(4), m_2(4) = 0.2 \times 0 = 0 \]

(1\, \cap \, 2) = m_1(1), m_2(1) = 0.2 \times 0.2 = 0.04

(1\, \cap \, 4) = m_1(1), m_2(4) + m_2(1) = 0.15

(2\, \cap \, 4) = m_1(2), m_2(4) + m_2(2) = 0.05

Table 3 represents the values of the beliefs after applying PCR5 rules.

The pignistic probability can be obtained from the above beliefs using Eq. 8., Table 4 represents the pignistic probability

\[ CM(1) = 3, CM(2) = 2, CM(4) = 2, CM(1 \, \cap \, 2) = 1 and CM(1 \, \cap \, 4) = 1. \]

Again, reprocess the sign in the second stage with the second data set where ID = 11, 12,…, 20. Table 5 represents the basic belief assignment for the second group.
Applying the classical DSM combination rule as in Eq. 10, which states that, Table 6 represents the fusion results of the second group:

\[
\begin{align*}
(1) &= m_1(1), m_2(1) = 0.64*0.7 = 0.469 \\
(14) &= m_1(14), m_2(14) = 0.2*0.2 = 0.04 \\
(15) &= m_1(15), m_2(15) = 0.16*0.1 = 0.016 \\
(1\cap14) &= m_1(1), m_1(14) + m_2(1), m_1(14) = 0.268 \\
(1\cap15) &= m_1(1), m_1(15) + m_2(1), m_1(15) = 0.176 \\
(14\cap15) &= m_2(2), m_2(4) + m_2(2), m_1(4) = 0.052
\end{align*}
\]

Consequently, redistribute the conflict factor using PCR5 rule, Table 7 represents the beliefs of the second group after redistribute the conflict using PCR5 rules.

\[\Phi = "14\cap15" \text{ Redistribute:}\]

So that we will distribute this conflict proportionally,

\[
m_{24} (14) = 0.04 + (0.72*0.052) = 0.07744
\]

\[
m_{24} (15) = 0.016 + (0.28*0.052) = 0.03056
\]

The pignistic probability can be obtained from the above beliefs using Eq. 8, Table 8 represents the pignistic probability

\[
\begin{align*}
\text{CM} (1) &= 3, \text{CM} (2) = 2, \text{CM} (4) = 2, \text{CM} (1\cap2) = 1 \text{ and CM} (1\cap4) = 1.
\end{align*}
\]

\[
P (1) = \frac{1}{2}*m_{24}(14) + \frac{1}{3}*m_{24}(1) + \frac{1}{2}*m_{24}(15) = 0.203
\]

\[
P (14) = \frac{1}{2}*m_{24}(14) + \frac{1}{3}*m_{24}(1) = 0.18853
\]

\[
P (15) = \frac{1}{2}*m_{24}(15) + \frac{1}{3}*m_{24}(1) = 0.16447
\]

\[
P (1\cap14) = \frac{1}{1}*m_{24}(1\cap14) = 0.268
\]

\[
P (1\cap15) = \frac{1}{1}*m_{24}(1\cap15) = 0.176
\]

Again, reprocess the sign in the third stage with the third data set where ID = 21, 22, ... 30. Table 9 represents the basic belief assignment of the third group.

Applying the classical DSM combination rule as in Eq. 10, which states that, Table 10 represents the fusion results of the third group:

\[
\begin{align*}
(1) &= m_1(1), m_2(1) = 0.8*0.7 = 0.56 \\
(24) &= m_1(24), m_2(24) = 0.2*0.1 = 0.02 \\
(26) &= m_1(26), m_2(26) = 0*0.2 = 0 \\
(1\cap24) &= m_1(1), m_1(24) + m_2(1), m_1(24) = 0.22 \\
(1\cap26) &= m_1(1), m_1(26) + m_2(1), m_1(26) = 0.16 \\
(24\cap26) &= m_2(2), m_2(4) + m_2(2), m_1(4) = 0.04
\end{align*}
\]

Consequently, redistribute the conflict factor using PCR5 rule, Table 11 represents the beliefs of the third group after redistribute the conflict using PCR5 rules.

\[\Phi = "24\cap26" \text{ Redistribute:}\]

So, we will distribute this conflict proportionally,

\[
m_{24} (24) = 0.02 + (0.04) = 0.06
\]

The pignistic probability can be obtained from the above beliefs using Eq. 8, Table 12 represents the pignistic probability.
<table>
<thead>
<tr>
<th>Table 8: Pignistic Probability Output</th>
<th>Stage 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>“1”</td>
<td>“14”</td>
</tr>
<tr>
<td>DSm</td>
<td>0.203</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 9: Basic belief assignment</th>
<th>Stage 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>“1”</td>
<td>“21”</td>
</tr>
<tr>
<td>BBA(S1/KNN)</td>
<td>0.8</td>
</tr>
<tr>
<td>BBA(S2/SVM)</td>
<td>0.7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 10: Fusion Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>“1”</td>
</tr>
<tr>
<td>m_{DSm}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 11: PCR5 output</th>
</tr>
</thead>
<tbody>
<tr>
<td>“1”</td>
</tr>
<tr>
<td>m_{PCR5}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 12: Pignistic Probability output</th>
</tr>
</thead>
<tbody>
<tr>
<td>“1”</td>
</tr>
<tr>
<td>DSm</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 13: Basic belief assignment</th>
<th>Stage 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>“1”</td>
<td>“31”</td>
</tr>
<tr>
<td>BBA(S1/KNN)</td>
<td>0.9</td>
</tr>
<tr>
<td>BBA(S2/SVM)</td>
<td>0.95</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 14: Fusion Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>“1”</td>
</tr>
<tr>
<td>m_{DSm}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 15: PCR5 output</th>
</tr>
</thead>
<tbody>
<tr>
<td>“1”</td>
</tr>
<tr>
<td>m_{PCR5}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 16: Pignistic Probability output</th>
</tr>
</thead>
<tbody>
<tr>
<td>“1”</td>
</tr>
<tr>
<td>DSm</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 17: Pignistic Probability output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sign Index</td>
</tr>
<tr>
<td>BetP(·)</td>
</tr>
<tr>
<td>BetP(·)</td>
</tr>
</tbody>
</table>

CM (1) =3, CM (2) =2, CM (4) =2, CM (1n2) =1 and CM (1n4) =1.

\[
P(1) = \frac{1}{2} * m_{12}(24) + \frac{1}{3} * m_{12}(1) + \frac{1}{2} * m_{12}(26) = 0.2167
\]

\[
P(24) = \frac{1}{2} * m_{12}(24) + \frac{1}{3} * m_{12}(1) = 0.217
\]

\[
P(26) = \frac{1}{2} * m_{12}(15) + \frac{1}{3} * m_{12}(1) = 0.1867
\]

\[
P(1 \cap 24) = \frac{1}{2} * m_{12}(1 \cap 14) = 0.22
\]

\[
P(1 \cap 26) = \frac{1}{1} * m_{12}(1 \cap 15) = 0.16
\]
Again, reprocess the sign in the fourth stage with the fourth data set where ID = 31, 32, ..., 40. Table 13 represents the basic belief assignment of the fourth group.

Applying the classical DSM combination rule as in Eq. 10, which states that, Table 14 represents the fusion results of the fourth group:

\[(1) = m_1(1), m_2(1) = 0.9 \times 0.95 = 0.855\]
\[(35) = m_1(35), m_2(35) = 0.05 \times 0.03 = 0.0015\]
\[(39) = m_1(39), m_2(39) = 0.05 \times 0.2 = 0.01\]
\[(1n39) = m_1(1), m_2(39) + m_2(1), m_2(39) = 0.0655\]
\[(1n35) = m_1(1), m_2(35) + m_2(1), m_2(35) = 0.0745\]
\[(39n35) = m_1(39), m_2(35) + m_2(39), m_2(35) = 0.0025\]

Consequently, redistribute the conflict factor using PCR5 rule, Table 15 represents the beliefs of the fourth group after redistribute the conflict using PCR5 rules.

\[\Phi = "35\&39"\]

Redistribute: to find the similarities between them and finally find the optimal alignment \(O(nm)\) [41], to compare \(x\) and \(y\) sequences we need to find the local cost matrix which represents the cost distribution between each two elements in the two sequences as in Eq. 11.

\[C: \begin{pmatrix} x_1 & x_2 & \cdots & x_n \end{pmatrix} \quad (11)\]

The value of \(C\) must be very small when \(x\) and \(y\) represents the same sign else it must be large, to generate the local cost matrix with dimension of \((nXm)\) as in Fig. 10. The cost of any position at the local cost matrix \(M(i, j)\) can be determined as in Eq. 12.

\[M(n, m) = d(n, m) + \min \{M(n-1, m-1), M(n-1, m), M(n, m-1)\}\]

The previous equation has two parts firstly 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 [42].

After getting the local cost matrix, we must find the wrapping path through it by applying the Eq. 13 to get the wrapping list.

\[wp_{n,m} = \min(c_{n-1,m-1}, C_{n-1,m}, C_{n,m-1})\]

Then finally apply the distance equation Eq. on the wrapping list in order to calculate the DTW distance.
Hidden Markov Model (HMM): It is a statistical model and time domain process, it represents the statistical behavior for the observed sequence using a set of hidden states called "hidden network" the model can make transition from one state to another with probability assignment [42], the expression of "hidden" comes from that the Markov model construct a sequence of hidden states from the observed sequence. HMM was successful and achieved a good accuracy with the applications of speech recognition and it is noted that there are similarities between the nature of speech and dynamic gestures [42].

\[ Q = q_1, q_2, q_3, \ldots q_n \] aset of \( n \) states.
\[ \pi = \pi_1, \pi_2, \pi_3, \ldots \pi_n \] the probability distribution over the states.
\[ A = a_{01}, a_{02}, \ldots a_{0n}, \ldots a_{mn} \] the matrix \( A \) of transition probability which contains the transition probability for the transition from one state to another.
\[ B = b_j(O), \] the observation probability from state \( j \) and the observing sequence \( O \).
\[ O = o_1, o_2, o_j \] a sequence \( T \) of observation
\[ q_0, q_n, \] Start state and End (final) state

There are two axioms in Hidden Markov Model, a) from the law of probability, the sum of all values on the directed arcs from a given state to other must equal 1 as in Eq. 15, b) the sum of all \( \delta \) probabilities must equal 1 as in Eq. 16.

Axiom #1 \( \sum_{j=1}^{n} a_{ij} = 1 \) \hspace{1cm} (15)
Axiom #2 \( \sum_{i=1}^{n} \pi_i \) \hspace{1cm} (16)

The Markov model assumed two important assumptions a) the probability of each state depends only on the previous state in the states sequence as in Eq. 17, b) the probability of any observation \( o_i \) depends only on the state \( q_i \) that produced the observation and not on any other states or any other observations as in Eq. 18.

Markov assumption #1 \( P(q_1|q_1, \ldots q_{t-1}) = P(q_t|q_{t+1}) \) \hspace{1cm} (17)
Markov assumption #2: \( P(o|q_1, \ldots q_t, \ldots o_{i+1}, \ldots o_{i-1}, \ldots o_j) = P(o|q_i) \) \hspace{1cm} (18)

For our model there are two phases:
Training Phase: In which we fed the model with all gestures sequences and their features vector to build the model for each sequence and then re-estimate the probability distribution using Baum-Welch algorithm also K-mean clustering is used to clusters all the 3-D sequence's points in the training set into $n$ clusters, this will reduce the data of the stored gestures to a set of discrete states and symbols. Now each point in the training set is converted to a specific symbol which is tightly related to the clustered $n$ states. Fig. 12 represents building the HMM states for one gestures "Injection / حقن" as an example.
Table 18: Medical Dataset

<table>
<thead>
<tr>
<th>Index</th>
<th>Arabic Sign</th>
<th>Meaning in English</th>
<th>Index</th>
<th>Arabic Sign</th>
<th>Meaning in English</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>استهلاك</td>
<td>Dysentery</td>
<td>21</td>
<td>تزيف</td>
<td>Bleeding</td>
</tr>
<tr>
<td>2</td>
<td>X-Ray</td>
<td></td>
<td>22</td>
<td>وفاة</td>
<td>Death</td>
</tr>
<tr>
<td>3</td>
<td>استقبال</td>
<td>Reception</td>
<td>23</td>
<td>طبيب عظام</td>
<td>Orthopedic doctor</td>
</tr>
<tr>
<td>4</td>
<td>رنينان</td>
<td>Two lungs</td>
<td>24</td>
<td>علاج طبيعي</td>
<td>Physical therapy</td>
</tr>
<tr>
<td>5</td>
<td>كبد</td>
<td>Liver</td>
<td>25</td>
<td>حقن</td>
<td>Injection</td>
</tr>
<tr>
<td>6</td>
<td>كلي</td>
<td>Kidneys</td>
<td>26</td>
<td>زغته</td>
<td>Blurred vision</td>
</tr>
<tr>
<td>7</td>
<td>معدة</td>
<td>Stomach</td>
<td>27</td>
<td>سرطان</td>
<td>Cancer</td>
</tr>
<tr>
<td>8</td>
<td>اضطراب</td>
<td>Constipation</td>
<td>28</td>
<td>جهاز قياس ضغط</td>
<td>Pressure measuring device</td>
</tr>
<tr>
<td>9</td>
<td>تحنيط</td>
<td>Analysis</td>
<td>29</td>
<td>صداع</td>
<td>A headache</td>
</tr>
<tr>
<td>10</td>
<td>تطعيم</td>
<td>Vaccination</td>
<td>30</td>
<td>عموم</td>
<td>Deafness</td>
</tr>
<tr>
<td>11</td>
<td>فشل</td>
<td>Paralysis</td>
<td>31</td>
<td>طبيب أطفال</td>
<td>Pediatrician</td>
</tr>
<tr>
<td>12</td>
<td>طبيب نفاذ</td>
<td>Obstetrician</td>
<td>32</td>
<td>طبيب نفاذ وذوى احتياجات خاصة</td>
<td>Doctor of nose and ear</td>
</tr>
<tr>
<td>13</td>
<td>قيصر</td>
<td>Vomiting</td>
<td>33</td>
<td>طبيب عظام</td>
<td>Internist</td>
</tr>
<tr>
<td>14</td>
<td>نورم</td>
<td>Swelling</td>
<td>34</td>
<td>طبيب عظام</td>
<td>General Doctor</td>
</tr>
<tr>
<td>15</td>
<td>جرح</td>
<td>Wound</td>
<td>35</td>
<td>اصابة</td>
<td>Broken bones</td>
</tr>
<tr>
<td>16</td>
<td>حامل</td>
<td>Pregnant</td>
<td>36</td>
<td>فيتامينات</td>
<td>Vitamins</td>
</tr>
<tr>
<td>17</td>
<td>حارة</td>
<td>Fever</td>
<td>37</td>
<td>حمى كلوه</td>
<td>Kidney Failure</td>
</tr>
<tr>
<td>18</td>
<td>اوردة</td>
<td>Veins</td>
<td>38</td>
<td>أوردة</td>
<td>Ulcers</td>
</tr>
<tr>
<td>19</td>
<td>حساسية</td>
<td>Allergic</td>
<td>39</td>
<td>أورام</td>
<td>The colon</td>
</tr>
<tr>
<td>20</td>
<td>مفصل</td>
<td>Colic</td>
<td>40</td>
<td>مخاطر</td>
<td>Laboratory</td>
</tr>
</tbody>
</table>

**Testing Phase:** In this phase we used Viterbi decoding algorithm to match the test sign sequence with the stored sign that has highest likelihood L which is computed using Eq. 19.

\[
L(S_1, \ldots, S_n | O_1, \ldots, O_n) = \prod_{i=1}^{n} P(S_i | O_i).
\]

\[
\prod_{i=1}^{n} P(S_i | O_{i-1})
\]

Fig. 13 represents the work flow of the system when using the HMM in dynamic gestures recognition.

**Experimental Results:** The experimental results have two aspects:

- Recognition accuracy,
- Latency (execution time).

Firstly, we applied the proposed model using Microsoft Kinect V2 which consists of an IR emitter, an RGB camera, an IR depth sensor and a microphone array [43], itis used to acquire signs and obtain the depth streams with a rate of 30 frames per second. We connected Kinect with a laptop which has a 64-bit architecture, Windows 8 operating system, 8 GB of physical memory also with Intel Core i7-5500U and 2.40GHz with x-64 based processor. The proposed model is developed using Microsoft C# program and Microsoft Kinect SDK library.

**Arabic Signs Dataset:** We chose 40 different gestures in the medical field, they are listed in Table. We collected the data from three different volunteers in different position and with different sizes.

**Proposed Model Accuracy:** For each sign, we collected 30 samples from 3 different signers and divided them as 20 for training and 10 for testing. The total samples for all signs were 1200 samples (800 for training set and 400 for testing set). The collected signs are dynamic i.e. the sign was performed through moving body joints such as: wrist, elbow, shoulder and hands. Each sign's stream contains on average (120 to 200) frames, so that the total number of
frames was around 40,000 frames for training set and 32,500 frames for testing set. The features’ vector is formed from the skeleton joints features and hand features which are combined to form 64 features. We applied two classifiers (SVM and KNN) in the classification phase, they were applied on the separated frames, the accuracies were 79% and 66% respectively. Because of the selected signs are dynamic nature, so that we can apply the majority voting on the classified frames for each sign in order to get the accuracy of recognizing each sign, the accuracy of KNN and SVM was improved after majority voting to 89% and 79% respectively. After applying the DsmT fusion of the two classifiers results, the accuracy reaches 91%. The accuracies of DTW and HMM reaches 82.6% and 79.5% respectively. Fig. 14 represents the classifiers accuracy before and after applying the majority voting for each classifier without applying the fusion. Fig. 15 represents the comparison between the accuracy of each classifier individually and after fusing the results of them.

It is notices in Fig. 15, the DsmT fusion of classifiers evidence improve the model recognition accuracy compared to the individual classifiers. The misclassified signs using SVM reaches 21% and in KNN reaches 11% while no misclassified signs after using DsmT, Table 19 represents the misclassified signs using only SVM, Table 20 represents the misclassified signs using only KNN.

For DTW and HMM, Fig. 16 represents the comparison between the achieved accuracy per sign using DTW, HMM and the classifiers fusion.

From Fig. 16 accuracy of the DsmT model is more accurate than both DTW and HMM and achieved higher recognition accuracy over the 40 signs.

The system performance is a very important metric, especially when the system works in real-time so the computation latency was computed. We took into consideration, the main processes in the system are performed sequentially and also the frames of Kinect are captured in the rate of 30 frame per second. Table 21 lists the time in seconds as average time for each process which was calculated over 30 experiments during performing the selected signs from data set, finally the total time will be the result of aggregating the time of all processes.
Also, for both DTW and HMM, from the experiments we found that DTW is faster than HMM, over 30 experiments DTW take 5 sec. in average for recognition and HMM take 7 sec. in average, Fig. 17 represent the processing time for both DTW and HMM for X-Ray /"ناشئة" over 10 samples.

Conclusion and Future Work: In this paper, we introduced an automatic system for Arabic sign recognition using Microsoft Kinect V2. The proposed model was applied and tested on 40 Arabic signs which are related to the medical field. Each sign is captured and represented as a depth stream, this stream is analyzed and normalized to overcome the variation of signer's position and size, then the features of both skeleton (32 features) and hand (30 features) are extracted and integrated in one feature vector with (62 features), the data with these features are used to train the two

Table 19: SVM (Support Vector Machine) Misclassification

<table>
<thead>
<tr>
<th>Misclassified Signs</th>
<th>Misclassification Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>72 %</td>
</tr>
<tr>
<td>5</td>
<td>98%</td>
</tr>
<tr>
<td>12</td>
<td>100%</td>
</tr>
<tr>
<td>17</td>
<td>67%</td>
</tr>
<tr>
<td>19</td>
<td>88%</td>
</tr>
<tr>
<td>31</td>
<td>51%</td>
</tr>
<tr>
<td>40</td>
<td>62%</td>
</tr>
</tbody>
</table>

Table 20: KNN (K-Nearest Neighbors) Misclassification

<table>
<thead>
<tr>
<th>Misclassified Signs</th>
<th>Misclassification Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>64%</td>
</tr>
<tr>
<td>19</td>
<td>58%</td>
</tr>
</tbody>
</table>

Table 21: Computation latency

<table>
<thead>
<tr>
<th>Processes</th>
<th>Time in Sec.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sign Capturing</td>
<td>6 sec.</td>
</tr>
<tr>
<td>Preprocessing (Data Normalization)</td>
<td>2 sec.</td>
</tr>
<tr>
<td>KNN Classifier</td>
<td>8 sec.</td>
</tr>
<tr>
<td>SVM Classifier</td>
<td>5 sec.</td>
</tr>
<tr>
<td>Late Fusion + DsmT Fusion</td>
<td>7 sec.</td>
</tr>
<tr>
<td>Total</td>
<td>28 sec.</td>
</tr>
</tbody>
</table>
classifiers KNN and SVM. Finally, DsmT theory is used to fuse the results of these two classifiers. Three different signers performed the signs in order to build required dataset, the collected samples were 1200 samples (800 for training set and 400 for testing set). The accuracy of the classifiers was 89 % for KNN and 79 % for SVM. Classifiers’ accuracy is compared with the fusion results which reaches to 91%. Finally, we compared the fusion model with two algorithms which were used widely in dynamic gestures recognition, these algorithms are DTW and HMM, they achieved accuracy of 82.6 and 79.5 respectively so that our model was more accurate than them. The suggested future work consists of increasing the overall accuracy of the system, improving the model in order to recognize the full sentences and also reducing the computation latency in real time.

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 and comments.

Conflict of Interest: The authors declare that they have no conflict of interest.

Funding: The study has no external funds

REFERENCES


