Sensing and Imaging (2019) 20:4
https://doi.org/10.1007/s11220-019-0225-3

ORIGINAL PAPER

Multiple Proposals for Continuous Arabic Sign Language Recognition

Mohamed Hassan[1] · **Khaled Assaleh**[2] · **Tamer Shanableh**[3]

Received: 9 May 2017 / Revised: 25 September 2018 / Published online: 17 January 2019

Abstract
The deaf community relies on sign language as the primary means of communication. For the millions of people around the world who suffer from hearing loss, interaction with hearing people is quite difficult. The main objective of sign language recognition (SLR) is the development of automatic SLR systems to facilitate communication with the deaf community. Arabic SLR (ArSLR) specifically did not receive much attention until recent years. This work presents a comprehensive comparison between two different recognition techniques for continuous ArSLR, namely a Modified k-Nearest Neighbor which is suitable for sequential data and Hidden Markov Models (HMMs) techniques based on two different toolkits. Additionally, in this work, two new ArSL datasets composed of 40 Arabic sentences are collected using Polhemus G4 motion tracker and a camera. An existing glove-based dataset is employed in this work as well. The three datasets are made publicly available to the research community. The advantages and disadvantages of each data acquisition approach and classification technique are discussed in this paper. In the experimental results section, it is shown that classification accuracy for sign sentences acquired using a motion tracker are very similar the classification accuracy for sentences acquired using sensor gloves. The modified KNN solution is inferior to HMMs in terms of the computational time required for classification.

Keywords Arabic sign language recognition · Pattern classification · Feature extraction · Motion detectors

1 Introduction

Sign language recognition (SLR) is closely related to speech recognition (SR). Therefore, most of the analysis and classification techniques used in SLR have been borrowed from the speech recognition literature which has been established decades

✉ Mohamed Hassan
b00065775@alumni.aus.edu

Extended author information available on the last page of the article

ago and has reached an adequate level of maturity. SLR, on the other hand, is still a relatively new but active area of research. Since sign language is primarily a set of gestures, it is similarly affected by the advances in gesture recognition. Nonetheless, not all techniques of gesture and speech recognition are adequate for SLR. Accordingly over the years SLR has developed its own literature.

Compared to other gestures, sign language is the most structured one. It has a large set of signs where each sign has a specific meaning. The majority of signs are associated with words while some are for finger spelling. For instance, American sign languages (ASL) has approximately 6000 signs [1].

Data availability is one of the main challenges facing researchers in SLR. The number of publicly available datasets is quite limited both in terms of quantity and quality. Manually annotated datasets are severely scarce. Moreover, since sign language is not universal, some sign languages (e.g., English and Chinese) have more datasets available than others (e.g., Arabic). Some publicly available datasets are in [2–4].

Another issue in sign language is co-articulation or epenthesis which is also encountered in speech recognition. It means that a given sign in a sentence is affected by the signs before and after it. It is a well-known problem in speech recognition, but in sign language it happens over a longer period and affects different aspect of the sign at the same time. This poses a lot of troubles in continuous recognition. Yang and Sarkar [5] used conditional random fields (CRF) to detect co-articulation in sign language. Other approaches for handling co-articulation can be found in [6, 7]. The way in which each person performs signs might be different; this is another problem known as signer dependency.

Signs of any given sign language consist of two components: manual and non-manual components. Manual components are hand position, orientation, shape and trajectory. Non-manual components are body movement and facial expressions. Most of the information is conveyed through manual components [8], thus most of the researches focused only on them [9]. While Non-manual components can form signs by themselves, mostly they emphasize the meaning of manual components; for example, raising an eyebrow indicates a question. Other popular non-manual components include lip shape and head pose are popular non-manual components as well.

The two main approaches for SLR are vision-based and sensor-based approaches. Vision-based SLR uses cameras only to capture gestures (signs). It has the advantage of user friendliness since the user is not required to wear any devices such as data gloves or motion trackers. However, the computational cost is normally high for this approach. Moreover, it can be quite sensitive to variations in the background or changing illumination conditions. The sensor-based approache makes use of wearable devices to accurately capture the signs. Although it might not be as convenient as the vision-based, it comes with huge improvement in recognition accuracy. Gloves and motion trackers are the most popular wearable devices for SLR. A comparison between the various gloves available in the market is provided in [10].

The rest of this paper is organized as follows. Section 2 presents a short survey of the current state-of-the-art in SLR. A description of the datasets used and their collection procedures is in Sect. 3, followed in Sect. 4 by an explanation of the feature

extraction techniques used. Our adopted classification techniques are discussed in Sect. 5. Results of our experiments are in Sect. 6. Concluding remarks and future works are given in Sect. 7.

2 Literature Review

A recent and thorough survey of SLR was done by Cooper et al. in [8]. They covered the key components of SLR, and discussed the pros and cons of the different types of data available. The manual and non-manual components of signs were also explored, as well as the recent researches in the area. The survey discussed some of the current research frontiers such as continuous recognition, signer independency, the work towards combining different modalities of sign, and the development of unconstrained real-life SLR systems. A more recent survey with a focus on Indian sign language (ISL) is found in [11].

Recognition of alphabets is in general easier than recognizing words. Usually alphabets are static gestures, this allows the use of conventional classification and clustering techniques. Color gloves were used in [12] to collect data of ArSL alphabets from multiple users, where adaptive Neuro-Fuzzy Inference System (ANFIS) was the recognition approach. The same data and feature extraction techniques were used by Assaleh et al. [13], but they used polynomial classier and reported better results than the previous ANFIS approach. Depth camera was used as the input device for real time recognition of ASL alphabets in [14].

The problem of coarticulation is not present in recognition of isolated gestures, which makes it simpler than continuous sign recognition. Nonetheless, isolated gestures involve some motion which makes their recognition more difficult than alphabet recognition. Oz et al. [15] collected a dataset of 50 isolated right handed words of ASL. After extracting some global features, artificial neural networks were used for classification. The system was tested on multiple users as well as on new words, and they reported accuracy of 90%. Different spatio-temporal feature-extraction techniques were used in [16] for recognizing isolated ArSL words. Accuracy of 97% was reported upon using K nearest neighbor (KNN) classifier. Their proposed feature extraction and classification yielded results comparable to conventional HMM.

HMMs are the most commonly used classification technique in SLR. For instance, Gaussian Hidden Markov Model (GHMM) was used on the SIGNUM database in [17]. In addition to appearance-based features extracted directly from the videos, multilayer perceptron (MLP) features were used achieving word error rate (WER) of 11.9%. Then, Principal Component Analysis (PCA) was used for dimensionality reduction. The same group investigated combining different sign modalities for the same database in [18]. They studied different combinations of five modalities, and were able to decrease WER to 10.7%. A promising approach of end-to-end embedding of a Convolutional Neural Network (CNN) into an HMM was recently proposed in [19]. In [20] HMMs were used to model the hand trajectory for large isolated Chinese SLR. They claimed to achieve better performance compared to normal coordinate features with HMM.

In [21] Kong and Ranganath presented promising results in terms of signer independency. Their system was tested on new signs as well as new users. Accuracies of 95.7% and 86.6% respectively were reported. They used a segmentation algorithm proposed in their previous work [22]. A major contribution in signer independency was done by Koller and colleagues in [23], where they worked on two publicly available large vocabulary databases representing lab-data (SIGNUM database:25 signers, 455 sign vocabulary, 19k sentences) and unconstrained real-life sign language (RWTH-PHOENIX-Weather database: 9 signers, 1081 sign vocabulary, 7k sentences). The earlier works of Gao et al. [24] and Fang et al. [25] are also examples of research on signer independency and large vocabulary. Fang et al. tried to tackle co-articulation by modeling the transition between signs using transition-movement models (TMMs).

In vision-based SLR, hand tracking is still a challenge especially in unconstrained environment where the background is cluttered and illumination conditions vary. Several researches have tried to tackle this issue with the use of Kinect [26–29]. Kinect simplifies hand tracking by providing depth and color data simultaneously. Many depth cameras are now available in the market with a variety of prices and accuracies. ZED and Kinect are the most commonly used ones. A team from Microsoft research Asia has developed Kinect-based SLR system and has reported very promising results [27]. Zafrulla et al. compared their old copycat system which used a colored glove and embedded accelerometer to a system based in Kinect in [28].

Until recently Arabic sign language recognition (ArSLR) has not received much attention. A survey of the contributions in ArSLR up to 2014 using both sensor-based and vision-based approaches can be found in [30]. The majority of the literature is concerned with isolated sign recognition. For instance, a system based on adaptive neuro fuzzy inference system (ANFIS) networks was proposed by Al-Jarrah and Halawani to recognize 30 Arabic alphabets. They managed to achieve an accuracy of 93.55% [31]. A vision-based posture recognition called AndroSpell was proposed in [32] where the authors made use of a camera phone to recognize 10 postures with 97% accuracy.

It was not until 2010 that the first continuous ArSLR system was proposed by Assaleh et al. [33]. They used their novel spatio-temporal feature-extraction technique [16] and reported 6.0% WER on a dataset of 40 sentences. A modified version of k-Nearest Neighbor (MKNN) was proposed by Tubaiz et al. in [34] and tested on the same dataset but collected using DG5-VHand data gloves instead of the camera. Their system achieved 2.0% WER. The data collected in [34] was used by Tuffaha et al. in [35] where modified polynomial classifier with augmented statistical features was proposed. The paper reported 85.0% sentence recognition rate.

3 Data Collection

Two datasets are collected in this work; both of which are composed of 40 Arabic sign language sentences created from 80 words lexicon. Each sentence was repeated 10 times. The list of sentences is shown in Table 1. We also use an existing dataset which was collected using DG5-VHand data gloves [34]. We refer to this dataset

Table 1 List of sentences

Arabic Sentence with English Meaning	Hands Used
ذهبت الى نادي كرة القدم I went to the soccer club	Both
انا احب سباق السيارات I love car racing	Both
اشتريت كرة ثمينة I bought an expensive ball	Both
يوم السبت عندي مباراة كرة قدم On Saturday I have a soccer match	Both
في النادي ملعب كرة قدم There is a soccer field in the club	Both
غدا سيكون هناك سباق دراجات There will be a bike race tomorrow	Both
وجدت كرة جديدة في الملعب I found a new ball in the field	Both
كم عمر اخيك؟ How old is your brother?	Both
اليوم ولدت امي بنتا My mom had a baby girl today	Right
اخي لا يزال رضيعا My brother is still on breastfeeding	Both
ان جدي في بيتنا My grandfather is at our home	Both
اشترى ابني كرة رخيصة My kid bought an inexpensive ball	Both
قرأت اختي كتابا My sister read a book	Both
ذهبت امي الى السوق في الصباح My mother went to the market this morning	Both
هل اخوك في البيت؟ Is your brother home?	Both
بيت عمي كبير My brother's house is big	Both
سيتزوج اخي بعد شهر In one month my brother will get married	Both
سيطلق اخي بعد شهرين In two months my brother will get divorced	Both
اين يعمل صديقك؟ Where does your friend work?	Both
اخي يلعب كرة سلة My brother plays basketball	Both
عندي أخوين I have two brothers	Right
ما اسم ابيك؟ What is your father's name?	Right
كان جدي مريضا في الامس Yesterday my grandfather was sick	Right
مات ابي في الامس Yesterday my father died	Right
رأيت بنتا جميلة I saw a beautiful girl	Right
صديقي طويل My friend is tall	Both
انا لا آكل قبل النوم I do not eat close to bedtime	Both
اكلت طعاما لذيذا في المطعم I ate delicious food at the restaurant	Both
انا احب شرب الماء I like drinking water	Both
انا احب شرب الحليب في المساء I like drinking milk in the evening	Both
انا احب اكل اللحم اكثر من الدجاج I like eating meat more than chicken	Both
اكلت جبنة مع عصير I ate cheese and drank juice	Both
يوم الاحد القادم سيرتفع سعر الحليب Next Sunday the price of milk will go up	Both
أكلت زيتونا صباح الامس Yesterday morning I ate olives	Right
ساشتري سيارة جديدة بعد شهر I will buy a new car in a month	Both
هو توضأ ليصلي الصبح He washed for morning prayer	Both
ذهبت الى صلاة الجمعة عند الساعة العاشرة I went to Friday prayer at 10:00 o'clock	Both
شاهدت بيتا كبيرا بالتلفاز I saw a big house on TV	Both
في الامس نمت عند الساعة العاشرة Yesterday I went to sleep at 10:00 o'clock	Both
ذهبت الى العمل في الصباح بسيارتي I went to work this morning in my car	Both

henceforth as dataset 1. The DG5-VHand data glove comes with five bend sensors; one for each finger. It also has an embedded accelerometer. Dataset 2 is collected in this work using two Polhemus G4 motion trackers which provides 6 measurements: the Cartesian position coordinates (x, y, z) and the Euler angles coordinates: Azimuth, elevation and roll (a, e, r). Dataset 3 is also collected in this work using a camera only; no wearable sensors are used during the collection of this dataset. Table 2 summarizes the three datasets and equipment used for collecting each. The first two datasets are expected to result in higher recognition accuracy due to the use of accurate sensors; the third dataset has the advantage of being more user-friendly since the user is not required to wear any device.

Table 2 Datasets and equipment used

Datasets	Equipment
Dataset 1	DG5-VHand data gloves
Dataset 2	Two Polhemus G4 motion trackers
Dataset 3	Camera

In the labeling phase, all the sensor readings belonging to each word in a sentence are labeled accordingly. In continuous SLR, the boundaries between adjacent words in a sentence are not clear. For manual labeling in vision-based SLR, a human can decide the boundaries by examining the videos visually. However, for sensor-based SLR, the word boundaries cannot be determined visually. Therefore, a camera was used in the data collection phase and it was synchronized with gloves and tracker recordings in order to detect word boundaries. Figure 1 shows a male user wearing the DG5-VHand data gloves and Polhemus G4 motion trackers. The synchronized camera is also shown in the figure.

4 Feature Extraction

Minimal feature extraction is required for sensor-based datasets. On the other hand, vision-based datasets require extensive feature extraction techniques. Below is a discussion of the feature extraction techniques used for the sensor-based datasets (dataset 1 and dataset 2) and the vision-based dataset (dataset 3).

Fig. 1 Data collection setup

4.1 Feature Extraction for Sensor-Based Datasets

Window-based statistical features extraction techniques are used to compute distinctive features. Those features are then appended to the raw data to form the final feature vectors. The classification systems were tested using both raw data and raw data augmented with the extracted statistical features.

Statistical features used in this work include window-based means and standard deviations. These features are extracted using a sliding window-based approach. The purpose of using a sliding window is to capture contextual information. In Sect. 6 we show that such an approach greatly enhanced classification accuracy. Equations (1) and (2) show the calculation of the window-based means and standard deviations respectively for a window size of w.

$$\tilde{x}_i = \frac{1}{w} \sum_{k=i-\frac{w-1}{2}}^{i+\frac{w-1}{2}} x_k \tag{1}$$

$$s_i = \left(\frac{1}{w-1} \sum_{k=i-\frac{w-1}{2}}^{i+\frac{w-1}{2}} (x_k - \tilde{x}_i) \right)^{\frac{1}{2}} \tag{2}$$

4.2 Feature Extraction for Vision-Based Datasets

Given the raw video, pixel-based difference for successive images is performed to detect the motion. The image differences are then converted into binary images by applying an appropriate threshold. The threshold is given by (3)

$$\mathrm{TH} = \mu + x\sigma \tag{3}$$

where μ is the mean pixel intensity of the image difference; σ is the corresponding standard deviation; x is a weighting parameter.

x is to be empirically determined based on subjective evaluation whose criterion is to retain enough motion information and discard noisy data. Figure 2 shows an illustration of performing image difference and thresholding. Next, a 2D Discrete Cosine Transform (DCT) is applied to the binary image differences. The top left DCT coefficients are zigzag scanned (zonal coding) to form a 1D vector. The number of DCT coefficients in the vector is known as the DCT cutoff. The feature extraction algorithm is depicted in Fig. 3. In Sect. 6, we experiment with different DCT cutoff values. Similar feature extraction techniques were used in our previous works as reported in [33, 34].

An illustration of the proposed data collection, feature extraction and labeling is shown in Fig. 4.

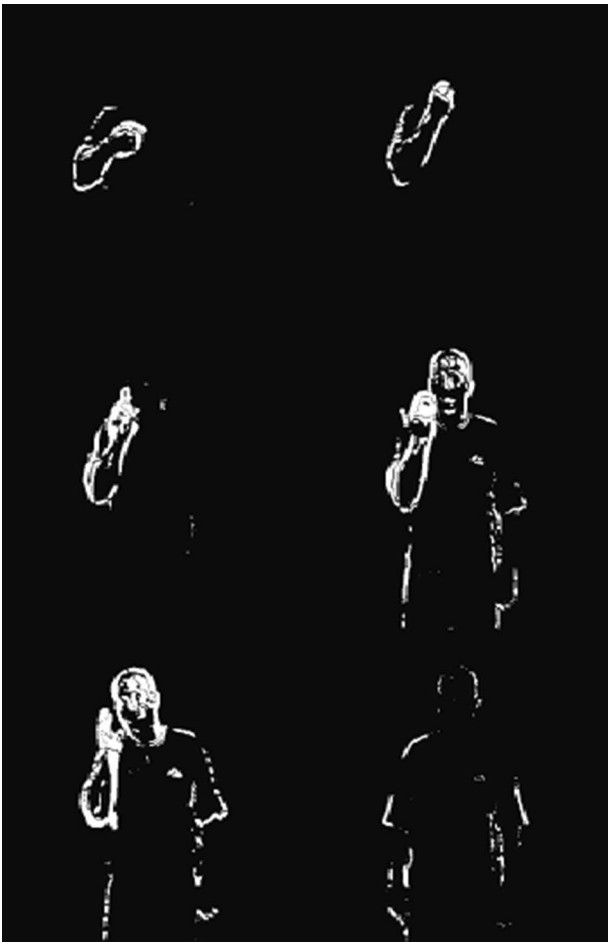

Fig. 2 Thresholded image differences

5 Classification

Three different classification approaches are used in this work; modified KNN suitable for sequential data and two different HMM toolkits. A brief review of each is presented next.

5.1 Modified KNN

In our previous work, we proposed a modification to the K-Nearest Neighbors (KNN) classifier to make it suitable for classifying sequential data [34]. The modified algorithm was called the Modified KNN or MKNN for short. The core modification is to consider the context prior to predicting the label of each feature vector.

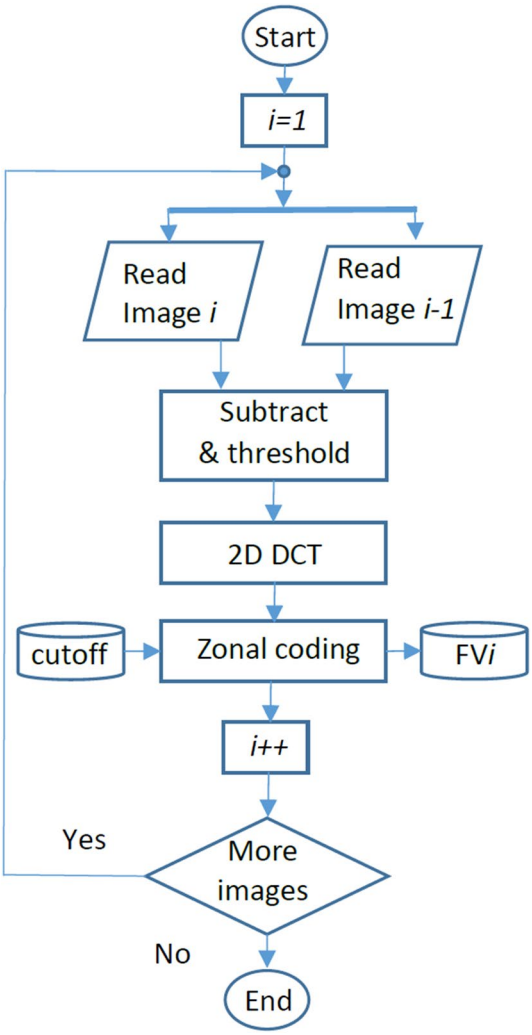

Fig. 3 Vision-based feature extraction

Our approach was to replace the predicted label by the most common label in a surrounding window of labels. After predicting all labels of a given sentence, each label was replaced with the statistical mode of its surrounding labels. For example, if the statistical mode window is of size 5 and k (number of nearest neighbors) is 3 then 5*3 labels are considered in predicting the label of a feature vector. We refer to the window size in this case as ModeW. An illustration of the MKNN for ModeW of 3 and k of 3 is shown in Fig. 5.

Formally, for each class of label L, $g(L)$ is the number of neighbors of the k nearest neighbors that belong to class L. $g(L)$ can be formulated as in (4).

$$g(L) = \sum_{i=1}^{k} \delta\left(L, \text{label}_i(\text{FV}_t)\right) \tag{4}$$

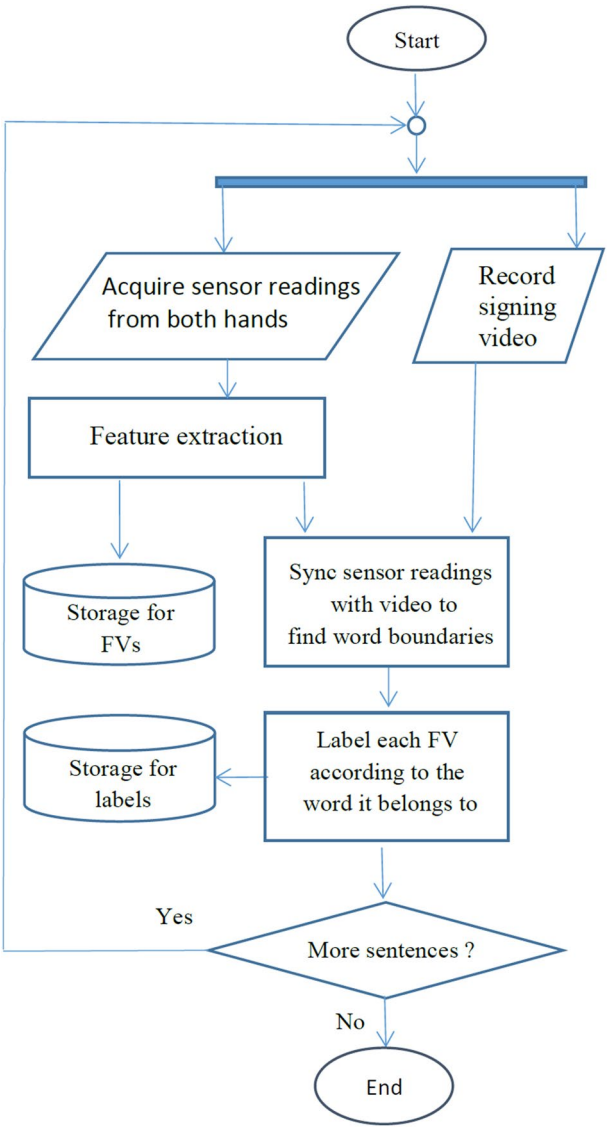

Fig. 4 Flowchart of data collection, feature extraction and labeling

where

$$\delta\left(L, \text{label}_i(\text{FV}_t)\right) = \begin{cases} 1, & \text{if } \text{label}_i(\text{FV}_t) = L \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

where FV_t is a feature vector acquired at time t.

The class label of the ith neighbor of the feature vector acquired at time t FV_t is given by (6).

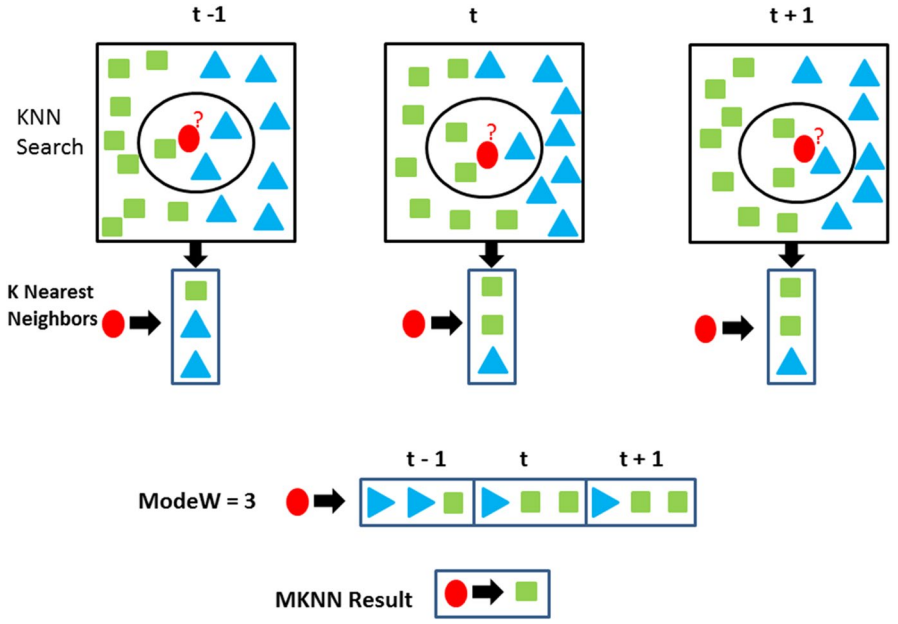

Fig. 5 Modified KNN to accomedate sequential data

$$\text{label}_i(\text{FV}_t) = \arg\min_{\forall i} \left\|\text{FV}_t - \text{FV}_i\right\|, \forall \text{FV}_i \in T \tag{6}$$

where T is a set of labeled training feature vectors.

In our MKNN the class label L^* is found as in (7).

$$L^* = \underset{L}{\text{argmin}} \sum_{j=-\frac{w}{2}}^{\frac{w}{2}} \sum_{i} \delta\left(L, \text{label}_i(\text{FV}_{t+j})\right) \tag{7}$$

The k nearest neighbors of the surrounding FVs, in a windows size w, are taken into account in the prediction of the class FV_t. After predicting a label for each feature vector, similar labels are grouped to form a sign language word.

5.2 Hidden Markov Models (HMMs)

Hidden Markov Models (HMMs) are widely used for sequential data classification in general, and for speech recognition in particular. They are also adopted for SLR and gesture recognition. The majority of SLR toolkits are developed originally for speech recognition then adapted for SLR. CMU Sphinx [36], HTK toolkit [37], Julius [38], Kaldi [39] and RASR [40] are all examples of open source speech recognition toolkits.

Gesture recognition tools are either built from scratch or built based on existing speech recognition tools. For instance the Georgia Tech gesture toolkit GT^{2}K [41]

was created based on a popular speech recognition toolkit known as HTK to provide tools that support gesture recognition research. Additionally, although RASR toolkit was originally developed for speech recognition, it has proved to be flexible and could be easily adapted for different applications such as SLR [23, 42] and optical character recognition [43]. An example of a toolkit created specifically for gesture recognition is the gesture recognition toolkit GRT [44] created by Gillian and Paradiso in 2014 with emphasis on real time recognition.

The GT^{2}K and RASR are selected for our work because they are more suitable for SLR and have been used before in similar applications.

The GT^{2}K toolkit was created based on the HTK to provide tools that support gesture recognition. It can be used for training models in both real-time and off-line modes. To use the toolkit, the user start by building gesture models, specify appropriate grammar and provide labeled examples for training. The tool will then train models for each gesture. The trained models are used for recognition of new data. More details and examples are available in [41].

The RWTH Aachen University Open Source Speech Recognition Toolkit (RASR) on the other hand is an open source version of speech recognition toolkit developed by a group from RWTH Aachen University. It comes with comprehensive documentation, examples and tutorials. RASR proved to be applicable for real-life applications; recently it has been used for numerous large vocabulary speech recognition systems by research group all over the world [45–48]. The toolkit proved to be suitable for SLR as reported in [23, 42].

The toolkit support strict left-to-right HMM topologies. All HMMs have the same number of states, except for silence which is modeled by a single state. Gaussian mixture models (GMMs) are used to model the emission probability. It uses the standard maximum likelihood estimation as well as discriminative training using the minimum phone error (MPE) [49] for Gaussian mixtures estimation. The toolkit itself does not have a module for the estimation of language models; nonetheless the decoder supports N-gram language models in the ARPA format generated by other toolkits.

Step by step examples, several tutorials and training recipes are available in the wiki [50].

6 Experimental Results

In this section we discuss the classification results achieved using the sensor-based and the vision-based datasets. Throughout this section, the results of the three classification tools described in Sect. 5 are discussed. Namely, MKNN, GT^{2}K and RASR. Results are reported in terms of word recognition rate and sentence recognition rate. Word recognition rate is given by (8).

$$\text{Word Recognition Rate} = 1 - \frac{D + S + I}{N}. \tag{8}$$

where D is the number of deletions, S is the number of substitutions, I is the number of insertions, and N is the total number of words. Sentence recognition rate is the

ratio of correctly recognized sentences to the total number of sentences. A sentence is considered to be correctly recognized if and only if all words in this sentence have been correctly recognized without any word being inserted, substituted, or deleted.

6.1 Sensor-Based Datasets

We start by comparing the performance of two HMM toolkits (RASR and GT^{2}K) on manually labeled datasets. Manually labeled means that word boundaries are manually annotated by a human. Both datasets (tacker-based and DG5-VHand-based) are augmented with the statistical features as explained in Sect. 4. Figures 6 and 7 show the classification accuracies for the sensor-based data using the two HMMs toolkits. Classification results are presented for both raw data and raw data augmented with statistical features. It is apparent from the classification results that RASR performance is better than that of the GT^{2}K. For instance RASR sentence recognition rate for the augmented DG5-VHand dataset was 96.7% while it was only 86.0% when GT2K is used. We also note that the motion tracker proved to be more accurate than the DG5-VHand glove, however classification accuracy pertaining to the augmented DG5-VHand FVs surpasses that which uses the augmented tracker FVs. A summary of all recognition rates is presented in Table 3. Note that the average of word and sentence recognition rates shown in the table also confirms the superiority of RASR over GT^{2}K.

The performance of RASR on automatically generated labels has been investigated. Auto labeling refer to the use of a tool to automatically estimate word boundaries. This could be done using RASR alignment module which automatically assigns each feature vector to a HMM state. This is advantageous because it allows the recognition of sentence-level labeled datasets where only sentence boundaries are annotated. Automatic labeling is of high practical gain since manual labeling is a

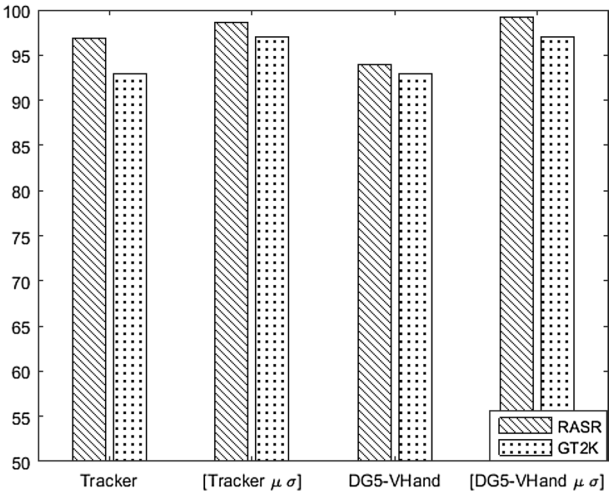

Fig. 6 Word recognition rates of manually labeled sensor-based datasets

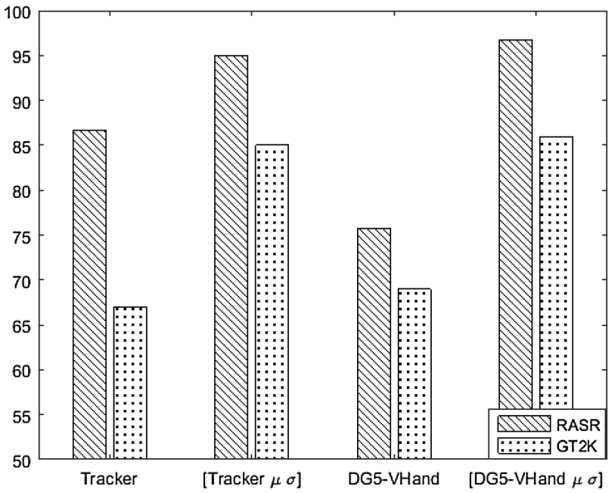

Fig. 7 Sentence recognition rates of manually labeled sensor-based datasets

Table 3 RASR and GT^{2}K comparison on manually labeled datasets

Dataset	RASR		GT^{2}K	
	Word	Sentence	Word	Sentence
Tracker, raw data	96.88	86.67	93.00	67.00
Tracker, augmented data	98.64	95.00	97.00	85.00
DG5-VHand, raw data	94.00	75.80	93.00	69.00
DG5-VHand, augmented data	99.20	96.70	97.00	86.00
Average	97.18	88.54	95.00	76.75

daunting task. Naturally this gain comes at the expense of lower accuracy as shown in Figs. 8 and 9 where manual labeled datasets always result in higher recognition rates. The accuracy of the auto labeling depends on the accuracy of the raw sensor readings. Since tracker data is highly precise, recognition rates of its manual and auto labeled dataset were almost the same, which is around 96% for both.

In the following experiments we compare the results of RASR against MKNN classification solution. The core of the MKNN is considering the context prior to predicting the label of each feature vector. The algorithm replaces the predicted label by the most common label in a surrounding window of labels. The algorithm depends on 2 parameters: the number of nearest neighbors called K and the size of the window of labels called ModeW. K was set to 3 for all experiments. And ModeW was set empirically similar to previous work [34]. Best results have been achieved when ModeW is set to 26. RASR generated better word recognition rates as shown in Fig. 10. On the other hand, in comparison to existing work, Fig. 11 shows that the MKNN surpasses RASR in 3 out of 4 tests in terms of sentence recognition rates which goes up to 97% for both augmented datasets.

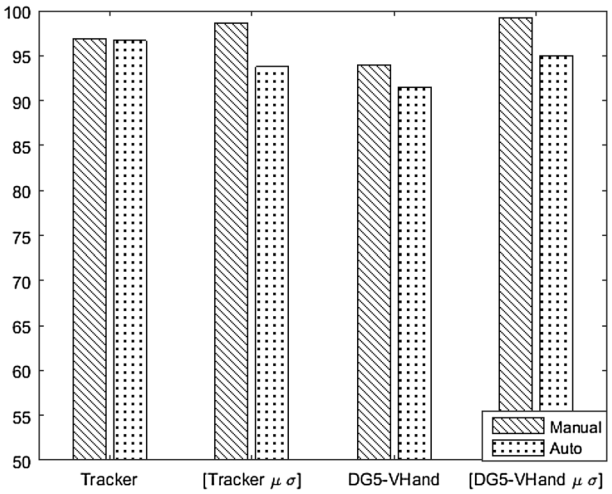

Fig. 8 Word recognition rates of auto and manually labeled sensor-based datasets

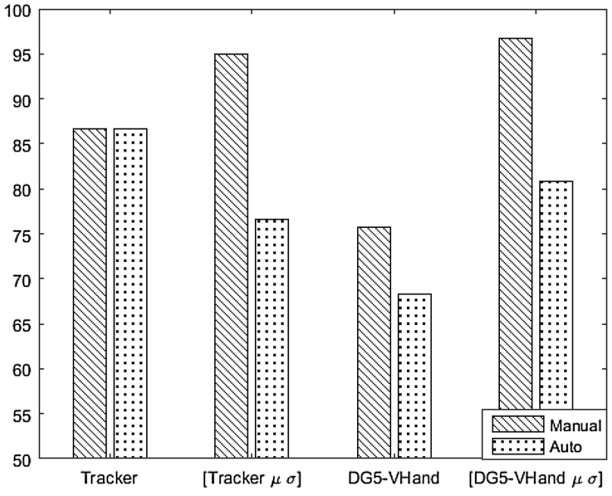

Fig. 9 Sentence recognition rates of auto and manually labeled sensor based datasets

As mentioned previously, in MKNN, a sliding window is used as a post-process to replace each predicted label with the statistical mode of its surroundings. For completeness, in Fig. 12 we show the effect of varying this mode window (ModW) size on classification accuracy. The figure shows that the increment in window size enhances the recognition rate as it captures more contextual information. Classification rates decrease rapidly for large window sizes as such windows include FVs belonging to other sign words and will therefore reduce the accuracy of the classifier.

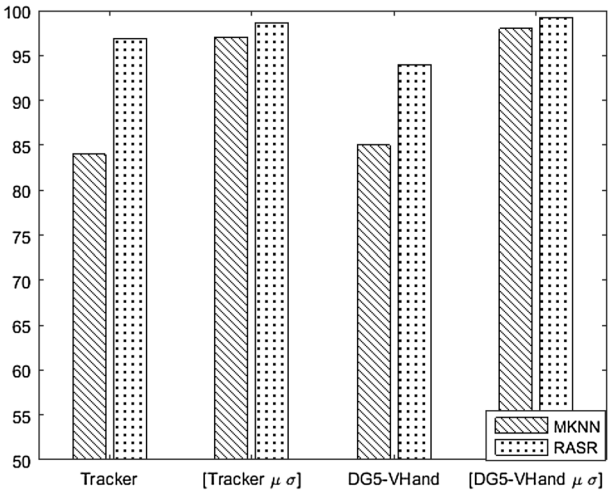

Fig. 10 Word recognition rates of MKNN [34] and RASR

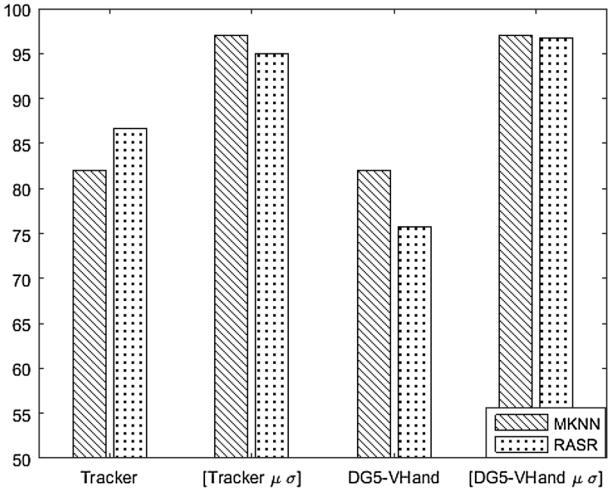

Fig. 11 Sentence recognition rates of MKNN [34] and RASR

The computational time for each classification approach is listed in Table 4. Results were recorded from 64-bit PC, 4.00 GB RAM, Intel Core i5, running Ubuntu 14.04. RASR computational time is 2.03 s, it is closely followed by the GT^2K. However considering both train and test times, our MKNN is advantageous since it does not require any training.

Lastly, to verify the reported results and make sure that they are not user specific, we carried out another set of experiments with a second signer. Using Polhemus G4, another user performed the 40 sentences with 10 repetitions each. The classification

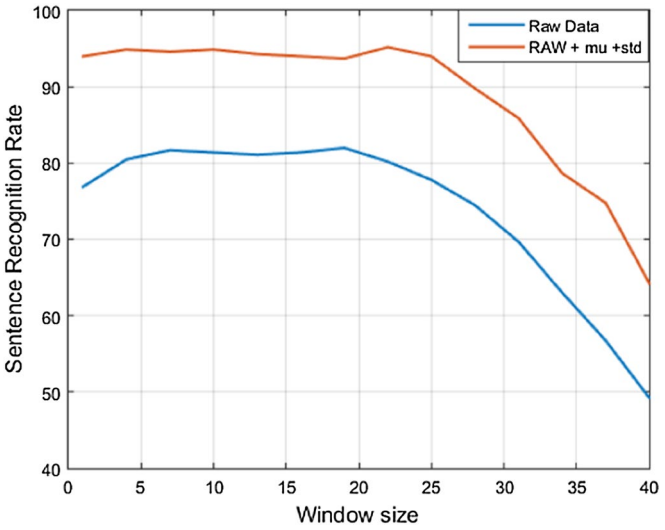

Fig. 12 Effect of the MKNN mode window size on sentence recognition rate for the Polhemus G4 tracker dataset

Table 4 Computational time comparison

Approach	Train time (s)	Classification time (s)
RASR	55.72	2.03
GT^2K	60.4	2.42
MKNN	–	18.87

results using RASR are shown in Fig. 13. It is apparent that the performance of the Polhemus G4 and RASR on both users is close. For the first user, word and sentence recognition rates are 96.9% and 86.7% respectively, compared to 95% and 84% word and sentence recognition rates for the second user.

Another experiment was performed on the datasets of both users combined. Seventy percent of the combined dataset was used for training using RARS and the rest used for testing. The word and sentence recognition results are reported in Table 5. The average word recognition rate is 94.5% and the average sentence recognition rate is 81.2% using the combined dataset. Although the recognition rates slightly decreased, they are still considered very high.

6.2 Vision-Based Datasets

This section is devoted for the discussion of recognition results of the third dataset which was collected using only a camera.

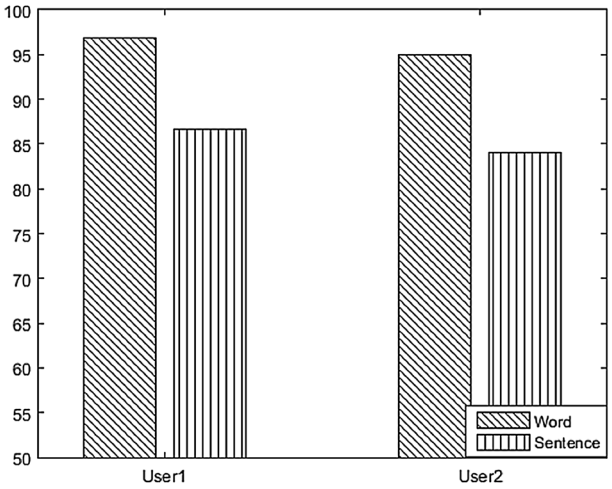

Fig. 13 Word and sentence recognition results for two signers using G4 tracker and RASR

Table 5 RASR Classification results on 2 users with features collected using Polhemus G4

User 1		User 2		combined	
Word	Sentence	Word	Sentence	Word	Sentence
96.90	86.70	95.00	84.00	94.50	81.20

The feature extraction phase, as explained in Sect. 4, depends on two empirical parameters that are determined prior to classification. The first one is the DCT cutoff, which is the number of DCT coefficients to retain in a feature vector. Figure 14 shows sentence recognition rates achieved using RASR for various DCT cutoffs. As expected, the recognition rate increases as the number of coefficients increase. This is due to the fact that DCT coefficients are not correlated. Thus increasing the number of DCT coefficients increases the information content in the feature vector. Nevertheless recognition rates in general decrease as the dimensionality of the feature vector increases beyond a certain threshold; thus there is normally a point after which any increment in the DCT cutoff will cause the recognition rates to decrease . In our case, the best classification rate is achieved with 100 DCT coefficients as shown in Fig. 14

The second parameter to be determined empirically is the weighting parameter x of Eq. (3). Figure 15 shows its effect on word recognition rate using MKNN. The highest rate achieved was with a value of $x = 1$.

We start by forming feature vectors using DCT coefficients of raw images instead of image differences. We apply 2D DCT transformation to raw images and retain the top left DCT coefficients using zigzag scanning. The feature vectors are then fed to the three classification approaches; MKNN, RASR and GT^{2}K. The word and sentence recognition rates are shown in Table 6. It is shown that the highest classification results are attained by RASR.

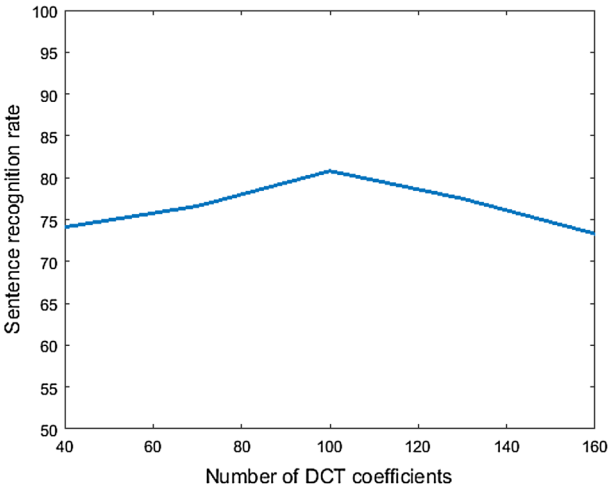

Fig. 14 DCT cutoff versus sentence recognition rate

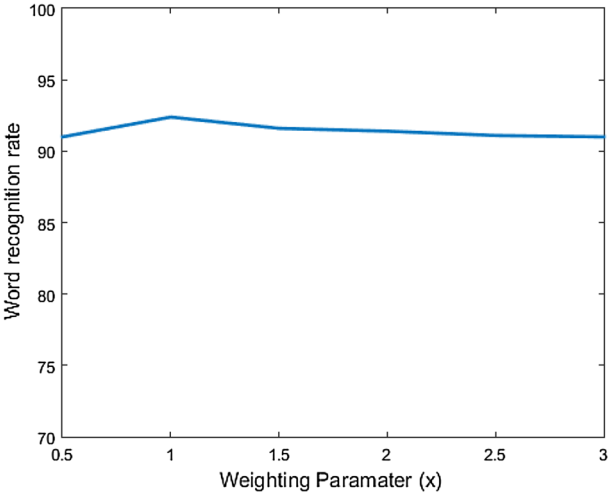

Fig. 15 Values of weighting parameter versus recognition rate

The last set of experimental results examine the effect of computing 2D DCT on thresholded image differences [33]. Comparing Tables 6 and 7 we notice the improvement as a result of using the thresholded image differences. Recognition rates of all approaches used had increased. For instance RASR sentence recognition rate increased from 80.8 to 85.0%. This is due to the motion between successive frames being emphasized by the thresholded image differences approach.

To summarize, we list the best recognition rates of sensor and vision based datasets in Tables 8 and 9. It is shown that MKNN always achieves the best sentence

Table 6 Recognition rates of raw vision data

MKNN		RASR		GT^2K	
Word	Sentence	Word	Sentence	Word	Sentence
82.50	77.20	94.18	80.80	93.10	73.00

Table 7 Recognition rates of thresholded image difference

MKNN		RASR		GT^2K	
Word	Sentence	Word	Sentence	Word	Sentence
91.60	89.17	95.60	85.00	94.00	80.00

Table 8 Best word recognition rates

Dataset	Approach	Rate
[Tracker $\mu\,\sigma$]	RASR	98.64
[DG5-VHand $\mu\,\sigma$]	RASR	99.20
Vision	RASR	95.60

Table 9 Best sentence recognition rates

Dataset	Approach	Rate
[Tracker $\mu\,\sigma$]	MKNN	97.00
[DG5-VHand $\mu\,\sigma$]	MKNN	97.78
Vision	MKNN	89.17

recognition rate. On the other hand, in terms of word recognition rates, RASR yields the best rates. Additionally, the summarized results reveal that data acquisition through motion trackers on their own could be very useful for sign language recognition. This is an interesting finding taking into account that no data gloves are needed. Lastly, the results in Tables 8 and 9 confirm that sensor-based data acquisition results in higher recognition rates in comparison to the camera-based approach.

7 Conclusion

This paper examined various data acquisition approaches and various classification techniques for Arabic sign language recognition. Two datasets are introduced using motion detectors and a camera. A third data set is acquired using data-gloves which is reused from previous work. Three tools are used for classification; MKNN, RASR and GT^2K. The paper also used various feature extraction approaches including window-based statistical features and 2D DCT transformation. The experimental results revealed that our adopted feature extraction techniques enhanced the recognition rates for both sensor and vision-based datasets. The results also revealed that RASR

is superior to GT^2K in terms of word and sentence recognition rates and computational time. The modified KNN achieved the best sentence recognition rates for all datasets exceeding both HMM toolkits. Additionally, sensor-based data turned out to be more precise than vision-based data. Although Polhemus G4 motion tracker only measures hand position and orientation, it achieved higher recognition rates than DG5-VHand data gloves, which measure both hand position and configuration. We conclude that motion trackers could be very useful for SLR.

Acknowledgements The authors gratefully acknowledge the American University of Sharjah for supporting this research through Grant FRG14-2-26.

References

1. Starner, T., Weaver, J., & Pentland, A. (1998). Real-time American sign language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(12), 1371–1375.
2. Dgs-corpus. (2015). http://www.sign-lang.uni-hamburg.de/dgs-korpus/.
3. Dictasign project. (2016). http://www.sign-lang.uni-hamburg.de/dicta-sign.
4. Bsl corpus project. (2016). http://www.bslcorpusproject.org/.
5. Yang, R., & Sarkar, S. (2006). Detecting coarticulation in sign language using conditional random fields. In *18th international conference on pattern recognition (ICPR'06)* (Vol. 2, pp. 108–112).
6. Yang, R., Sarkar, S., & Loeding, B. (2007). Enhanced level building algorithm for the movement epenthesis problem in sign language recognition. In *IEEE conference on computer vision and pattern recognition* (pp. 1–8).
7. Yang, R., Sarkar, S., & Loeding, B. (2010). Handling movement epenthesis and hand segmentation ambiguities in continuous sign language recognition using nested dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(3), 462–477.
8. Cooper, H., Holt, B., & Bowden, R. (2011). Sign language recognition. In *Visual analysis of humans* (pp. 539–562). London: Springer.
9. Ong, S. C., & Ranganath, S. (2005). Automatic sign language analysis: A survey and the future beyond lexical meaning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(6), 873–891.
10. Dipietro, L., Sabatini, A. M., & Dario, P. (2008). A survey of glove-based systems and their applications. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, *38*(4), 461–482.
11. Agrawal, S. C., Jalal, A. S., & Tripathi, R. K. (2016). A survey on manual and non-manual sign language recognition for isolated and continuous sign. *International Journal of Applied Pattern Recognition*, *3*(2), 99–134.
12. Al-Rousan, M., & Hussain, M. (2001). Automatic recognition of Arabic sign language finger spelling. *International Journal of Computers and Their Applications*, *8*, 80–88.
13. Assaleh, K., & Al-Rousan, M. (2005). Recognition of Arabic sign language alphabet using polynomial classifiers. *EURASIP Journal on Applied Signal Processing*, *2005*, 2136–2145.
14. Uebersax, D., Gall, J., den Bergh, M. V., & Gool, L. V. (2011). Real-time sign language letter and word recognition from depth data. In *IEEE international conference on computer vision workshops (ICCV Workshops)* (pp. 383–390).
15. Oz, C., & Leu, M. C. (2011). American sign language word recognition with a sensory glove using artificial neural networks. *Engineering Applications of Artificial Intelligence*, *24*(7), 1204–1213.
16. Shanableh, T., Assaleh, K., & Al-Rousan, M. (2007). Spatio-temporal feature-extraction techniques for isolated gesture recognition in Arabic sign language. *IEEE Transactions on Systems, Man, and Cybernetics Part B (Cybernetics)*, *37*(3), 641–650.
17. Gweth, Y. L., Plahl, C., & Ney, H. (2012). Enhanced continuous sign language recognition using PCA and neural network features. In *IEEE computer society conference on computer vision and pattern recognition workshop* (pp. 55–60).

18. Forster, J., Oberdörfer, C., Koller, O., & Ney, H. (2013). Modality combination techniques for continuous sign language recognition. In *Pattern recognition and image analysis. IbPRIA 2013. Lecture notes in computer science* (Vol. 7887, pp. 89–99). Berlin, Heidelberg: Springer.
19. Koller, O., Zargaran, O., Ney, H., & Bowden, R. (2016). Deep sign: Hybrid CNN-HMM for continuous sign language recognition. In *British machine vision conference*.
20. Pu, J., Zhou, W., Zhang, J., & Li, H. (2016). Sign language recognition based on trajectory modeling with HMMs. In *Multimedia modeling. MMM 2016. Lecture notes in computer science* (Vol. 9516, pp. 686–697). Cham: Springer.
21. Kong, W., & Ranganath, S. (2014). Towards subject independent continuous sign language recognition: A segment and merge approach. *Pattern Recognition*, *47*(3), 1294–1308.
22. Kong, W. W., & Ranganath, S. (2008). Automatic hand trajectory segmentation and phoneme transcription for sign language. In *8th IEEE international conference on automatic face & gesture recognition* (pp. 1–6). Netherlands.
23. Koller, O., Forster, J., & Ney, H. (2015). Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. *Computer Vision and Image Understanding*, *141*, 108–125.
24. Gao, W., Fang, G., Zhao, D., & Chen, Y. (2004). A Chinese sign language recognition system based on SOFM/SRN/HMM. *Pattern Recognition*, *37*(12), 2389–2402.
25. Fang, G., Gao, W., & Zhao, D. (2007). Large-vocabulary continuous sign language recognition based on transition-movement models. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *37*(1), 1–9.
26. Chai, X., Li, G., Lin, Y., Xu, Z., Tang, Y., Chen, X., & Zhou, M. (2013). Sign language recognition and translation with Kinect.
27. Chen, X., et al. (2013). Kinect sign language translator expands communication possibilities.
28. Zafrulla, Z., Brashear, H., Starner, T., Hamilton, H., & Presti, P. (2011). American sign language recognition with the Kinect. In *Proceedings of the 13th international conference on multimodal interfaces* (pp. 279–286). Spain.
29. Lang, S., Block, M., & Rojas, R. (2012). Sign language recognition using Kinect. In *Artificial intelligence and soft computing. ICAISC 2012. Lecture notes in computer science* (Vol. 7267, pp. 394–402). Berlin: Springer.
30. Mohandes, M., Deriche, M., & Liu, J. (2014). Image-based and sensor-based approaches to Arabic sign language recognition. *IEEE Transactions on Human-Machine Systems*, *44*(4), 551–557.
31. Al-Jarrah, O., & Halawani, A. (2001). Recognition of gestures in Arabic sign language using neuro-fuzzy systems. *Artificial Intelligence*, *133*(1–2), 117–138.
32. Elhenawy, I., & Khamiss, A. (2014). The design and implementation of mobile Arabic fingerspelling recognition system. *International Journal of Computer Science and Network Security (IJCSNS)*, *14*(2), 149.
33. Assaleh, K., Shanableh, T., Fanaswala, M., Amin, F., & Bajaj, H. (2010). Continuous Arabic sign language recognition in user dependent mode. *Journal of Intelligent Learning Systems and Applications*, *2*(01), 19.
34. Tubaiz, N., Shanableh, T., & Assaleh, K. (2015). Glove-based continuous Arabic sign language recognition in user-dependent mode. *IEEE Transactions on Human-Machine Systems*, *45*(4), 526–533.
35. Tuffaha, M., Shanableh, T., & Assaleh, K. (2015). Novel feature extraction and classification technique for sensor-based continuous Arabic sign language recognition, pp. 290–299.
36. Walker, W., Lamere, P., Kwok, P., Raj, B., Singh, R., Gouvea, E., et al. (2004). *Sphinx-4: A flexible open source framework for speech recognition*. Mountain View, California: Sun Microsystems, Inc.
37. Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., et al. (2002). *The HTK book* (Vol. 3, p. 175). Cambridge: Cambridge University Engineering Department.
38. Lee, A., Kawahara, T., & Shikano, K. (2001). Julius—An open source real-time large vocabulary recognition engine. In *European conference on speech communication and technology (EUROSPEECH)*.
39. Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., et al. (2011). The kaldi speech recognition toolkit, no. EPFL-CONF-192584.
40. Rybach, D., Gollan, C., Heigold, G., Hoffmeister, B., Lööf, J., Schlüter, R., et al. (2009). The RWTH AACHEN university open source speech recognition system. In *10th annual conference of the international speech communication association* (pp. 2111–2114). Brighton, UK.

41. Westeyn, T., Brashear, H., Atrash, A., & Starner, T. (2003). Georgia tech gesture toolkit: Supporting experiments in gesture recognition. In *5th international conference on multimodal interfaces* (pp. 85–92). New York.
42. Dreuw, P., Rybach, D., Deselaers, T., Zahedi, M., & Ney, H. (2007). Speech recognition techniques for a sign language recognition system. In *8th annual conference of the international speech communication association* (p. 80). Belgium.
43. Dreuw, P., Rybach, D., Heigold, G., & Ney, H. (2012). RWTH OCR: A large vocabulary optical character recognition system for Arabic scripts. In *Guide to OCR for Arabic scripts* (pp. 215–254). London: Springer.
44. Gillian, N., & Paradiso, J. A. (2014). The gesture recognition toolkit. *The Journal of Machine Learning Research*, *15*(1), 3483–3487.
45. Lööf, J., Gollan, C., Hahn, S., Heigold, G., Hoffmeister, B., Plahl, C., et al. (2007). The RWTH 2007 TC-STAR evaluation system for European English and Spanish. In *8th annual conference of the international speech communication association* (pp. 2145–2148). Belgium.
46. Rybach, D., Hahn, S., Gollan, C., Schluter, R., & Ney, H. (2007). Advances in Arabic broadcast news transcription at RWTH. In *IEEE workshop on automatic speech recognition & understanding (ASRU)* (pp. 449–454). Koyoto, Japan.
47. Sundermeyer, M., Nußbaum-Thom, M., Wiesler, S., Plahl, C., Mousa, A. E.-D., Hahn, S., et al. (2011). The RWTH 2010 Quaero ASR evaluation system for English, French, and German. In *IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 2212–2215). Prague, Czech Republic.
48. Plahl, C., Hoffmeister, B., Hwang, M., Lu, D., Heigold, G., Lööf, J., et al. (2008). Recent improvements of the RWTH GALE mandarin LVCSR system. In *9th annual conference of the international speech communication association* (pp. 2426–2429). Brisbane, Australia.
49. Povey, D., & Woodland, P. C. (2002). Minimum phone error and i-smoothing for improved discriminative training. In *IEEE international conference on acoustics, speech, and signal processing* (pp. I-105). Orlando, FL, USA.
50. RASR manual. (2017). http://www.hltpr.rwth-aachen.de/rasr/manual

Affiliations

Mohamed Hassan[1] · Khaled Assaleh[2] · Tamer Shanableh[3]

Khaled Assaleh
k.assaleh@ajman.ac.ae

Tamer Shanableh
tshanableh@aus.edu

[1] Mechatronics Engineering Program, American University of Sharjah, Sharjah, UAE

[2] Department of Electrical Engineering, Ajman University, Ajman, UAE

[3] Department of Computer Science and Engineering, American University of Sharjah, Sharjah, UAE