

# *Android Mobile App for Real-Time Bilateral Arabic Sign Language Translation Using Leap Motion Controller*

Abdulla Eqab

Department of Computer Science and Engineering  
American University of Sharjah  
Sharjahm UAE  
Currently with Ithmaar Bank, Bahrain  
AHEqab@gmail.com

Tamer Shanableh

Department of Computer Science and Engineering  
American University of Sharjah  
Sharjah, UAE  
tshanableh@aus.edu

**Abstract**— This paper introduces an android mobile App for real-time bilateral Arabic sign language translation. The system is designed to operate on isolated sign language words. The mobile App has four main components; sign language capture for training, sign language capture for translation, voice to sign language translation and a sign language quiz game. Therefore, the Mobile App can be used for bilateral communication with the deaf community. Acquisition of sign language words is performed using a portable Leap Motion Controller that is connected to the USB of the mobile phone through an OTG adapter. We use simple feature extraction approaches based on statistical features such as mean, variance and covariance. We also use simple classification approach based on the minimum distance classifier. Classification results performed using 15 different sign language words revealed that excellent classification results are achieved. In comparison to a similar system that uses data gloves, our system has the advantage of higher accuracy and the advantage of real-time processing using a mobile App.

**Keywords**—Sign language recognition; pattern recognition; signal processing

## I. INTRODUCTION

Sign language recognition is a means of communication between the hearing and the deaf societies. Many technical solutions are proposed and implemented for translating sign language into text. Likewise, text and/or speech can be translated into sign language to facilitate bilateral communications.

Research work on sign language recognition is reported for many languages such as Arabic [1], Indonesian [2], American [3], Indian [4] and Thai [5]. Most of sign language recognition research is simulation-based and does not work for real-time applications. However, with the advent of mobile Apps, true deployment of sign language translation systems is profiling. For instance, a Vision-based Android App for Indonesian sign language using the OpenCV computer vision library is proposed in [2]. The App recognizes alphabets of the Indonesian sign language in real-time. Likewise, a mobile App for recognizing alphabets of the American sign language is proposed in [6]. The App uses a vision-based approach for the sign language recognition.

In general, sign language recognition systems can be classified according to the data acquisition used. Typically vision-based system is used [7], however sensor-based acquisition is usually more accurate. Such approaches include data gloves [8] and sEMG and inertial sensors [3].

Noteworthy in the field of Arabic sign language recognition is the work reported in [8]. The authors compiled a dataset of Arabic sentences using data gloves and proposed a novel classification technique suitable for sequential data.

Lastly, the leap motion controller connected to a PC, rather than a mobile phone, was used for alphabet sign language recognition. In [9] a recognition rate of 80% was reported for English alphabet, in [10] an accurate recognition of 100% was reported for the same alphabet. For the Indian and Chinese alphabet, recognition rates ranging from 90%-96% are reported in [11] and for the Arabic alphabet, a recognition rate of 99% was reported in [12].

In this work, we propose and develop a real-time Android App that captures sign language gestures using a sensor device connected to the USB port. Therefore, there is no restriction of wearing data gloves whilst retaining the advantage of sensor-based acquisition which is known to be more accurate than vision-based solutions.

Furthermore, the proposed mobile App has four main components; sign language capture for training, sign language capture for translation, voice to sign language translation and a sign language quiz game.

The rest of the paper is organized as follows. Section 2 introduces the mobile App developed for bilateral sign language translation. Section 3 introduces the employed feature extraction techniques, Section 4 explains the used classification approach, Section 5 presents the classification results and Section 6 concludes the paper.

## II. MOBILE APP FEATURES

We implement our Arabic sign language recognition system using a mobile App. The App is developed using Android Studio

2.2. For the proposed mobile system to capture hand movements, we used a Leap Motion Controller (available online [www.leapmotion.com](http://www.leapmotion.com)) that connects to the USB of the mobile through an OTG adapter. The controller tracks hand movements using infrared cameras with a 150° field of view with 8 cubic feet of interactive space. The controller connected to the mobile App is shown in Figure 1.



Fig. 1. Leap Motion Controller connected to the proposed App

The controller captures and stores the following information about hand movements; Hand's x, y and z directions, hand grab strength and finger tip's x, y and z positions for all hand figures. Therefore, the total number of variables is 19. The controller comes with an SDK version 2.3.2+35031 (alpha) for Android Apps Development.

Our mobile App has four main components; "sign language capture for training", "sign language capture for translation", "voice to sign language translation" and a "sign language quiz game". Therefore, the Mobile App can be used for bilateral communications with the deaf community.

In the "sign language capture for training" part, the user creates a new sign language word and repeats it five times as illustrated in the screen shot of Figure 2.



Fig. 2. Screenshot of adding a sign language word

The user can add and delete as many sign language words as needed.

In the "sign language capture for translation" part, the user acts a sign language word in the field of view of the Leap Motion Controller and the mobile App will recognize it and translate it into text and speech as illustrated in Figure 3.



Fig. 3. Screenshot illustrating the translation of a sign language word into text and speech

In the "voice to sign language translation" part, the user is asked to speak a word and the system will translate it into a sign language video as illustrated in Figure 4.

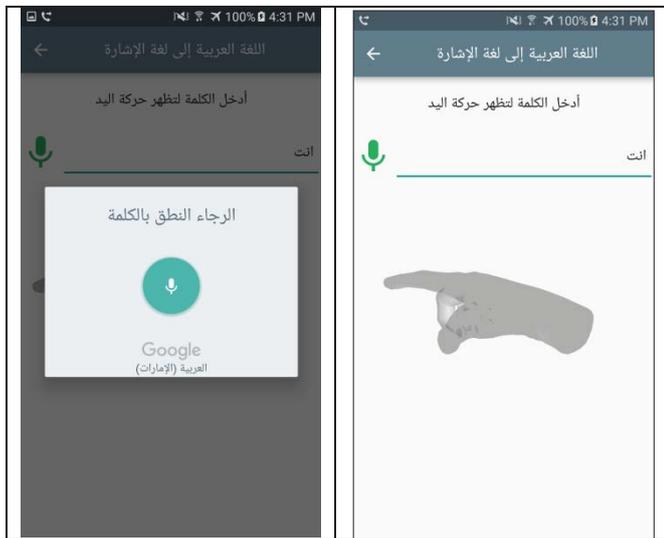


Fig. 4. Screenshot illustrating speech to sign language translation

Lastly, the “sign language quiz game” part is developed to help users learn sign language. In the quiz, a sequence of sign language animated GIFs are displayed and the user selects the correct corresponding word from a list of words as illustrated in figure 5.



Fig. 5. Screenshot illustrating sign language quiz game

### III. FEATURE EXTRACTION

The Leap Motion capture device reads a sequence of 19 variables at a temporal resolution of 60 samples per second. A sequence of  $N$  sensor readings belonging to a certain sign language word  $m$  is represented by the matrix  $S_m$ , as:

$$S_m [s_1 \ s_2 \ \dots \ s_N]^T \quad (1)$$

We use two approaches for feature extraction, namely; a covariance approach and a window-based approach.

#### A. Covariance approach

For feature extraction in this approach, we start by computing the covariance of the  $S_m$  matrix;  $C_{ij} = \text{Cov}(S_m)$ . We then append all the elements on and above the main diagonal into one feature vector  $v_m$  as follows:

$$v_m = [C_{ij}; i \leq j] \quad (2)$$

Where  $v_m$  is the feature vector representation of the word  $m$ . To form the final feature vector, we append the mean vector of  $S_m$  to  $v_m$  such that:

$$v_m = [\mu_{S_m} \ C_{ij}; i \leq j] \quad (1)$$

#### B. Window-based approach

For feature extraction in this approach, we start by dividing the above feature matrix into 3 non-overlapping matrices as follows:

$$S_m = [S_{m1} \ S_{m2} \ S_{m3}]^T \quad (4)$$

To remove the temporal demission from the data, we accumulate the absolute differences of all sensor readings into one vector. We apply the concept of accumulated differences to both submatrices  $S_{m1}$ ,  $S_{m2}$  and  $S_{m3}$  such that one feature vector  $V_m$  is formed:

$$v_m = \left[ \sum_{n=1}^{\frac{N}{3}} |x_{n+1} - x_n| \ \sum_{n=\frac{N}{3}+1}^{\frac{2N}{3}} |x_{n+1} - x_n| \ \sum_{n=\frac{2N}{3}+1}^N |x_{n+1} - x_n| \right] \quad (2)$$

In addition to the accumulated absolute differences, we also experiment with appending the mean and standard deviation vectors of each submatrix. These vectors can be appended to  $S_m$  to form the final feature vector for a given sign language word. For a shorter notation we refer to the accumulated absolute differences of Equation (5) as  $d_{m1}$ ,  $d_{m2}$  and  $d_{m3}$  hence the final feature vector is represented as:

$$v_m = [d_{m1} \ \mu_{S_{m1}} \ \sigma_{S_{m1}} \ d_{m2} \ \mu_{S_{m2}} \ \sigma_{S_{m2}} \ d_{m3} \ \mu_{S_{m3}} \ \sigma_{S_{m3}}]^T \quad (3)$$

The concept of accumulated absolute differences was successfully used in both vision and glove based sign language recognition as reported in [7] and [13].

For the training dataset, once the feature vectors are computed using either Equation (3) or (6), the feature vectors are arranged into one feature matrix. The variables of the matrix are normalized by computing their z-scores. The corresponding means and standard deviation of each variable are stored and used for normalizing a test feature vector prior to classification.

#### IV. CLASSIFICATION

In this App, we use a simple classification approach based on the minimum distance classifier. Denote a feature vector in the training set as  $\mathbf{t}_i$  where  $i=1..M$  and  $M$  is the total number of feature vectors in the training set. We classify a feature vector of a sign word  $\mathbf{v}_j$  by computing its distance with all training feature vectors. The label of the training feature vector corresponding to the minimum distance is then used as a class label for  $\mathbf{v}_j$ .

Formally, the class label is computed as  $label(\arg_i \min(D_E(\mathbf{v}_j, \mathbf{t}_i)))$  Where  $D_E$  is the Euclidian distance and  $label(.)$  is a function that finds the class label of a FV's index from the training set.

#### V. EXPERIMENTAL RESULTS

We test our system using 15 sign language words. In the training phase, which is implemented using the mobile app, each word is repeated 5 times. Feature extraction follows and the resultant feature vectors are stored in the dataset. The normalization parameters are also stored as explained in Section 3.

The sign language words used in our experiments are listed in Table 1. The table lists the English translation of each Arabic Sign language word.

TABLE I. LIST OF SIGN LANGUAGE WORDS USED

Seq.	Word	Seq.	Word
1	All of you	9	Loaner
2	You	10	Mercy
3	Never	11	Water
4	Dirham	12	Maybe
5	Me	13	Mountain
6	Loud	14	Thank You
7	Please Enter	15	Child
8	Cost		

For the testing, we capture a sign language word and compare it against the training dataset as explained in Section 4. The classification accuracy is computed as the percentage of correctly classified test feature vectors.

In Table 2, we present the classification accuracy of the two proposed solutions of Section 3. Each solution is further divided into two sub solutions according to equations 2,3,5 and 6.

TABLE II. CLASSIFICATION ACCURACY OF THE PROPOSED SOLUTIONS

Approach	10 words	15 words
Covariance only; Eq. (2)	88%	85.3%
Covariance and mean; Eq. (3)	94%	93.3%
Window-based, mean and standard deviation Eq. (6)	100%	98.7%

The table presents the classification accuracies for the first 10 and 15 words of Table 1. The lowest classification results are related to the case of computing only the covariance of sensor readings. This is expected as both the variance and means are required for a proper representation of sensor readings. Therefore, the use of Equation (3) in feature extraction enhanced the classification accuracy as shown in the table. On the other hand, the results in the table indicate that the window-based solution that includes the summation of absolute differences, means and standard deviations of sensor readings generate the highest accuracy classification results. This is so because the summation of absolute differences provides a very good representation of sensor readings. When combined with statistical features like the mean and standard deviation, excellent results are generated as reported in the table.

In comparison to existing solutions, the closest work is reported in [13], where 10 Arabic sign language words are used. Data acquisition was performed using data gloves and all processing was done off-line using Matlab<sup>®</sup>. The classification accuracy reported for user dependent classification was 95.3% when using the minimum distance classifier. Hence, the proposed solution in this paper has an advantage in terms of classification accuracy and more importantly, has an advantage in terms of being a real-time usable system in the form a mobile App.

In terms of real-time performance, the app was tested on Samsung Note 5 and the average time required for feature extraction and classification for 40 repetitions was 0.05 seconds.

#### VI. CONCLUSION

A mobile App is proposed and developed using the Android operating system to facilitate the bilateral communication between the hearing and deaf societies. The App captures hand movements using a Leap Motion Controller which is connected to the USB of the mobile through an OTG adapter. The App has a number of functions including system training, sign language to speech, speech to sign language and a sign language quiz game. Feature extraction was based on accumulated differences concatenated with the mean and standard deviation of sensor readings. A simple minimum distance classifier was used to make sure that the App runs in real-time. Experimental results revealed that excellent classification results are achieved for ten and fifteen sign language words. The classification results are 100% and 98.7% respectively.

In future work, we intend to extend the proposed solution into recognizing continuous sign language sentences.

## ACKNOWLEDGMENT

The author Abdulla Eqab participated in this research and development work during his undergraduate study at the Department of Computer Science and Engineering of the American University of Sharjah, UAE.

## REFERENCES

- [1] T. Shanableh, K. Assaleh and M. AL-Rousan, "Spatio-Temporal feature extraction techniques for isolated Arabic sign language recognition," *IEEE Transactions on Systems, Man and Cybernetics Part B*, 37(3), June, 2007.
- [2] I. Kryvonos, I. Krak and W. Wojcik, "Information technologies applications for sign languages investigations," *Computer Science and Information Technologies (CSIT)*, 2015, Yerevan, 2015, pp. 148-150.
- [3] J. Wu, Z. Tian, L. Sun, L. Estevez and R. Jafari, "Real-time American Sign Language Recognition using wrist-worn motion and surface EMG sensors," *2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, Cambridge, MA, 2015, pp. 1-6.
- [4] N. Yadav, S. Thepade and P. H. Patil, "Novel approach of classification based Indian sign language recognition using transform features," *2015 International Conference on Information Processing (ICIP)*, Pune, 2015, pp. 64-69.
- [5] C. Chansri and J. Srinonchat, "Reliability and accuracy of Thai sign language recognition with Kinect sensor," *2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, Chiang Mai, 2016, pp. 1-4.
- [6] C. M. Jin, Z. Omar and M. H. Jaward, "A mobile application of American sign language translation via image processing algorithms," *2016 IEEE Region 10 Symposium (TENSYP)*, Bali, 2016, pp. 104-109.
- [7] T. Shanableh and K. Assaleh, "Telescopic Vector Composition and polar accumulated motion residuals for feature extraction in Arabic Sign Language recognition," *EURASIP Journal on Image and Video Processing*, vol. 2007, Article ID 87929, 10 pages, 2007.
- [8] N. Tobaiz, T. Shanableh and K. Assaleh, "Glove-Based Continuous Arabic Sign Language Recognition in User-Dependent Mode," *IEEE Transactions on Human-Machine System*, 45(4), August, 2015.
- [9] C.-H. Chuan, E. Regina and C. Guardino, "American Sign Language Recognition Using Leap Motion Sensor," *13<sup>th</sup> International Conference on Machine Learning and Applications (ICMLA)*, MI, USA, December, 2014.
- [10] L. Quesad, G. López and L.A. Guerrero, "Sign Language Recognition Using Leap Motion," *Ubiquitous Computing and Ambient Intelligence. Sensing, Processing, and Using Environmental Information, LNCS*, vol 9454, Springer, Cham, 2015.
- [11] R. B. Mapari and G. Kharat, "Analysis of Multiple Sign Language Recognition Using Leap Motion Sensor," *International journal of research in advent technology*, March, 2017.
- [12] M. Mohandes, S. Aliyu, M. Deriche, "Arabic sign language recognition using the leap motion controller," *IEEE International Symposium on Industrial Electronics*, pp. 960-965, Turkey, June, 2014.
- [13] K. Assaleh, T. Shanableh and M. Zourob, "Low Complexity Classification System for Glove-Based Arabic Sign Language Recognition," *Neural Information Processing, Lecture Notes in Computer Science*, volume 7665, 2012, pp 262-268, November, 2012.