Indian Sign Language Recognition using the Leap Motion Controller

S.Saritha
AP/Department of ECE
Nandha College of Technology,Erode

B.Aravind, M.Madhumadhi
AP/Department of ECE
Nandha College of Technology,Erode

Abstract—Sign language is important for facilitating communication between hearing impaired and the rest of society. Two approaches have traditionally been used in the literature: image-based and sensor-based systems. Sensor-based systems require the user to wear electronic gloves while performing the signs. The glove includes a number of sensors detecting different hand and finger articulations. Image-based systems use camera(s) to acquire a sequence of images of the hand. Each of the two approaches has its own disadvantages. The sensor-based method is not natural as the user must wear a cumbersome instrument while the image-based system requires specific background and environmental conditions to achieve high accuracy. In this paper, we propose a new approach for Arabic Sign Language Recognition (ArSLR) which involves the use of the recently introduced Leap Motion Controller (LMC). This device detects and tracks the hand and fingers to provide position and motion information. We propose to use the LMC as a backbone of the ArSLR system. In addition to data acquisition, the system includes a preprocessing stage, a feature extraction stage, and a classification stage. We compare the performance of Multilayer Perceptron (MLP) neural networks with the Nave Bayes classifier. Using the proposed system on the Arabic sign alphabets gives 98% classification accuracy with the Nave Bayes classifier and more than 99% using the MLP.

Keywords—Arabic sign langauge recognition; leap motion controller; finger articulation, electronic glove, image-based system.

I. INTRODUCTION

Sign language is the most important means of communication with hearing impaired and integrating them into the rest of society. The problem is that, most vocal people do not understand sign language. Hence, the need to develop automated systems capable of translating sign languages into words and sentences is becoming a necessity. In the following, a brief overview of previously used methods in sign language recognition is presented. There are mainly two sign language recognition approaches: image-based and sensor-based. The main advantage of image-based systems is that signers do not need to use complex devices. However, substantial computations are required in the pre-processing stage. Instead of cameras, sensor-based systems use instrumental gloves equipped with sensors. Sensor-based systems have also their own challenges, including the cumbersome gloves worn by the signer. Traditionally, there have been three categories of image-based ArSLR systems: alphabet, isolated word, and continuous recognition. Several attempts have been made on the different categories of Arabic sign language recognition. This paper focuses on Arabic alphabet sign recognition systems. In [1], Mohandes introduced an automatic recognition of the Arabic sign language letters. For feature extraction, Hus moments are used. For classification, the moment invariants are fed to support vector machines. A correct recognition rate of 87% was achieved. Al-Jarrah and Halawani [2], developed a neuro-fuzzy system. The main steps of the system include: image acquisition, filtering, segmentation, hand outline detection followed by feature extraction. Bare hands were considered in the experiments achieving a recognition accuracy of 93.6%. In [3], Al-Rousan and Hussain built an adaptive neuro-fuzzy inference system for alphabet sign recognition. A colored glove was used to simplify segmentation and geometric features were extracted from the hand region. The achieved recognition accuracy was 95.5%. Assaleh and Al-Rousan [4], used a polynomial classifier to recognize alphabet signs. A glove with 6 different colors was used: 5 for fingertips and one for the wrist region. Different geometric measures such as lengths and angles were used as features. A recognition rate of about 93.4% was achieved on a database of more than 200 samples representing 42 gestures. In [5], Mohandes et al. used a cost effective off-the-shelf device to implement a robust ArSLR system. Statistical features are extracted from the acquired signals and used with an SVM classifier. With a database of 120 signs, a recognition accuracy of over 90% was achieved. In [6], a first attempt at two-handed Arabic sign recognition was made. The database consisted of 20 samples from each of 100 two-handed signs performed by two signers. Second order statistics from sub-frames of the signs were used as features. The length of the feature vector was then reduced using PCA. For classification, the SVM was used, achieving an accuracy of 99.6% with 100 signs. In [7], Maraqa and Abu-Zaiter used recurrent neural networks for alphabet recognition. A database of 900 samples, covering 30 gestures performed by 2 signers, was used in their experiments. Colored gloves similar to the ones in [4], were used in their experiments. The Elman network achieved an accuracy rate of 89.7% while a fully recurrent network improved the accuracy to 95.1%. In [8], El-Bendary et al. developed a sign language recognition system for the Arabic alphabet achieving an accuracy of 91.3%. In their system, the images of bare hands are processed. The input to the system is a set of features extracted from a video of signs and the output is simple text. For each frame, the hand outline is first extracted. Using a centroid point, the distances to the

outline of the hand covering 180 degrees are extracted as a 50 dimensional features vector. These features are rotation, scale, and translation invariant. In the feature segmentation stage, they assumed a small pause between letters. Such pauses are used to separate the letter numbers and the related video frames. The signs of the alphabet are divided into three different categories before feature extraction. At the recognition stage, a multilayer perceptron (MLP) neural network and a minimum distance classifier (MDC) are used. Hemayed and Hassanien [9] discussed an Arabic sign language alphabet recognition system which converts signs into voice. The technique is much closer to real life setup however; recognition is not performed in real time. The system focuses on static and simple moving gestures. The inputs are color images of the gestures. To extract the skin blobs, the YCbCr space is used. The Prewitt edge detector is used to extract the hand shape. To convert the image area into feature vectors, Principal Component Analysis (PCA) is used with a K-Nearest Neighbor Algorithm (KNN) in the classification stage. In [10], Mohandes et al used a Hidden Markov Model (HMM) to identify isolated Arabic signs from images. A Gaussian skin color model was used to find the signers face which is then taken as a reference for hands movement. Two colored gloves were used for the right and left hands for ease of hand region segmentation. A simple region growing technique was used for hands segmentation. They used a dataset consisting of 500 samples of 300 signs, achieving a recognition accuracy of 95%. Naoum et al. [11], developed an image-based sign language alphabet recognition with an accuracy of 50% for naked hand, 75% for hand with a red glove, 65% for hand with a black glove and 80% for hand with a white glove. The system starts by finding histograms of the images. Profiles extracted from such histograms are then used as input to a KNN classifier. Yang [12], proposed a Chinese sign language recognition system which is based on extracting the temporal and spatial characteristic of video sequence recorded while the signer is performing the signs. Using SVM classification method, with 30 groups of the Chinese manual alphabet images, an average recognition rate of 95.55% was achieved. Yang and Peng [13], proposed a system for integration of improved sign language recognition system into intelligent building to enhance a barrier-free environment for the deaf-mute. They proposed a bidirectional language/speech system which can remove communication barrier between the deaf-mute and the vocal people. Normalized rotational inertia NMI and the 7-Hu moments are integrated to characterize the sign language gestures in the system. However translation of the gesture is not in real-time. Most recently, an approach using the Microsoft Kinect camera to capture images [14], [15], [16], has been introduced. Using computer vision techniques, a characteristic depth and motion profile for each gesture was developed [14]. In this paper, we introduce a completely different approach. In particular, we use the recently introduced Leap Motion Controller (LMC) as our interface to acquire data from the hand and finger while performing signs. The LMC has been shown to have 0.7mm precision with regard to gesture-based user interface. The LMC can focus on hands and fingers motion [17]. The remainder of this paper is organized as follows; Section II describes the LMC, in Section III, data acquisition and feature extraction is presented. Section IV presents the classification

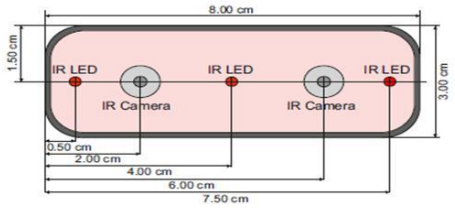


Fig. 1: Schematic view of leap motion controller (LMC)

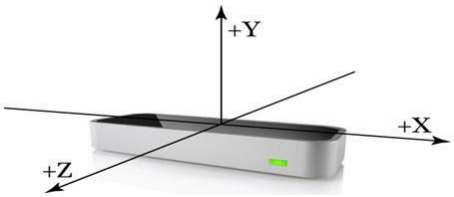


Fig. 2: LMC right-handed coordinate system.

of the Arabic alphabets signs and section V concludes the paper.

II. THE LEAP MOTION CONTROLLER (LMC)

The leap motion controller (LMC) is a device recently developed by Leap Motion Company [18]. It detects and tracks hands, fingers and finger-like objects reporting discrete position and motion. It operates in a close proximity at a rate of 200 frames per second [18]. The LMC field of view is an inverted pyramid of about 8 cubic feet centered on the device [19]. The effective range of the LMC extends from approximately 1 inch to 2 feet above the device [20]. The LMC uses two high precision infrared cameras and three LEDs to capture hand information within its active range. However, it does not provide pictures of detected images. Its driver software processes the acquired data, extracts position and other information using complex mathematics [21]. Figure 1, shows a schematic view of the LMC [17]. The LMC employs a right-handed Cartesian coordinate system as shown in Figure 2. Values reported are in units of real-world millimeters. The origin is the center of the device. The x- and z-axes lie in the horizontal plane, with the x-axis running parallel to the long edge of the device. The y-axis is vertical, with positive values increasing upwards. The z-axis has positive values increasing away from the computer screen [19]. As the LMC tracks hands and fingers in its field of view, it provides updates as a set, or frames of data. Each frame contains a list of the basic tracking data that describes the overall motion in the scene. When it detects hand and fingers, the Leap Motion software assigns it a unique ID tag. The ID remains the same as long as that entity remains visible within the device's field of view. If tracking is lost and regained, the software may assign for it a new ID. Java program was written, using the NetBeans IDE, to collect the motion tracking data.

Fig. 3: Arabic Alphabet signs.

III. DATA ACQUISITION AND FEATURES EXTRACTION

The developed system recognizes the twenty-eight Arabic alphabet signs from Ø - @ as shown in Figure 3. It should be noted that all Arabic alphabet signs are static and are performed using a single hand. For training and testing of the developed system, ten samples were collected for each letter. Each sample includes 10 frames of data making a total of 100 frames per letter. Therefore, a total of 2800 frames of data were collected and imported to MATLAB for analysis and processing.The LMCs data acquisition stage returns twenty-three (23) features for each frame of data. However, in our experiments, we selected the 12 most relevant features to our purpose. These include: finger length, finger width, average tip position with respect to x, y, and z-axis, hand sphere radius, palm position with respect to x, y, and z-axis, hand pitch, roll and yaw. To analyze the relevance of the extracted features, we estimated the mean of each feature across the 10 frames of each sample. As examples, these measures for the signs of

are plotted against sample number in figures

letters @ and H.
4,5.

Figures 4,5 show that there are variations on the values of each feature related to the same letter. This is due to the fact that usually people do not repeat a sign exactly the same. Subsequently, that makes the classification process a challenging task and machine learning algorithms have to be used for better recognition. Similar to the finger length feature, we estimated the mean values across the 10 frames of the remaining 11 features. These include: finger width, average tip position with respect to x, y, and z-axis, hand sphere radius, palm position with respect to x, y, and z-axis, hand pitch, roll and yaw. Similar figures to the ones shown in figures 4,5 were obtained for all features. The mean of each of the extracted features across the 100 frames (10 frames from each of 10

samples) of letters H. , @ are shown in Figures 6,7, respectively. Similar trend was found for each of the 28 letters. The values in the x-axis of figures 6,7, represent the following features: finger length, finger width, average tip position with respect to (wrt) x-axis of the LMC, average tip position wrt y-axis, average tip position wrt z-axis, hand sphere radius, palm position wrt x-axis, palm position wrt y-axis, palm position wrt z-axis, hand pitch, roll, and yaw respectively. Analysis of the mean values of the 12 features for all 28 letters indicates that features 2 (finger width) and 6 (hand sphere radius) are not

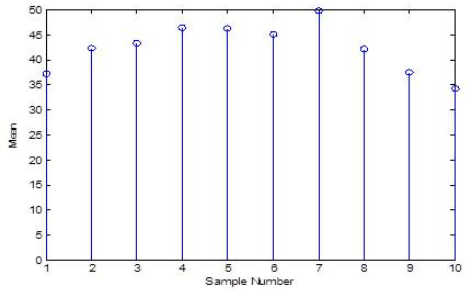


Fig. 4: The mean value (across the 10 frames) of the feature finger length for each of the 10 samples of letter @

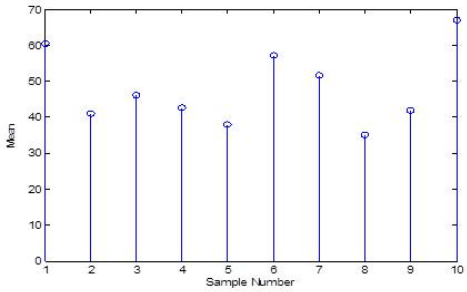


Fig. 5: The mean value (across the 10 frames) of the feature finger length for each of the 10 samples of letter H.

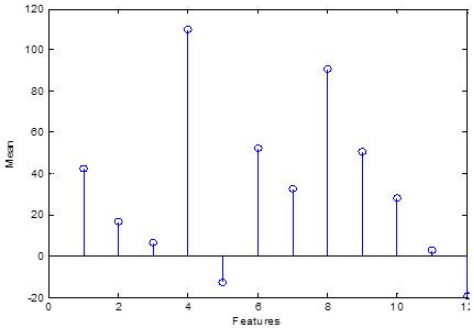


Fig. 6: The mean value of each of the 12 features for letter @

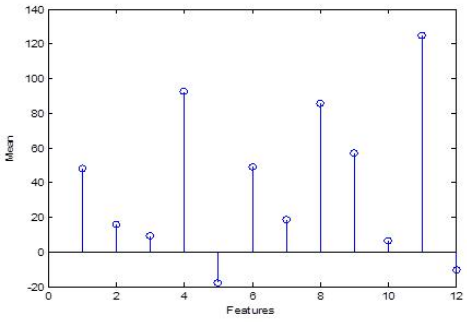


Fig. 7: The mean value of each of the 12 features for letter

discriminative enough across the different letters hence may not be significant in the classification stage. However, in this paper all 12 features are used in the classification stage.

IV. CLASSIFICATION OF ARABIC ALPHABET SIGNS

Using the features discussed above, we compared the per-formance of two classifiers, namely, the Nave Bayes Classifier (NBC) and the Multilayer Perceptron (MLP) which are briefly described below.

A. Nave Bayes Classifier (NBC)

The Bayesian classification approach is based on quantifying the trade-offs between various classification decisions using probability and the costs that accompany such decisions [22]. The classifier was developed based on Bayes probability rule which states:

$$P(Y|X_1, \ldots, X_n) = \frac{P(X_1, \ldots, X_n|Y)P(Y)}{P(X_1, \ldots, X_n)}$$

This is the probability of obtaining Y given conditions X_1 to X_n or the posterior probability of Y given a prior probability of Y and X_1 to X_n likelihood. In our case, using the letters, this is interpreted as posterior probability of any letter given the likelihood parameter and a prior probability of that letter. The decision is taken based on the greatest of the calculated probabilities to achieve minimum error classification. From the Bayes rule, the NBC is mathematically stated as:

$$\prod_{i=1}^{n}$$

The NBC assumes that all features are independent given class label Y, though this assumption is not true in all cases. Hence equation 1 reduces to multiplication (see equation 2). Basically, the NBC ignores the possibilities of correlation among the inputs and reduces a multivariate problem to a univariate problem [23]. In our case, the different sign letters are treated as classes in the classifier. One common rule used by the classifier is to select the class that results in maximum a posterior probability (MAP). The decision rule is to select a class C_k if:

$$P(C_k|x) > P(C_j|x); j \neq k \quad (1)$$

$$P(X_1, \ldots, X_n|Y) = P(X_i|Y) \quad (2)$$

B. Mutilayer Perceptron Neural networks

The current interest in artificial neural networks (ANNs) is largely due to their ability to mimic natural intelligence in its learning from experience [24]. They learn from examples by constructing an input-output mapping without explicit derivation of the model equation. ANNs have been used in a broad range of applications including: pattern classification, function approximation, optimization, prediction and automatic control and many others [25], [26]. An artificial neural network consists of many interconnected identical simple processing units called neurons. Each connection to a neuron has an adjustable weight factor associated with it. Every neuron in the network sums its weighted inputs to produce an internal activity level:

$$a_i = \sum_{j=1}^{n} w_{ij} x_{ij} - w_{io} \quad (6)$$

where w_{ij} is the weight of the connection from input j to neuron i, x_{ij} is input signal number j to neuron i, and w_{io} is the threshold associated with unit i. The threshold is treated as a normal weight with the input clamped at -1. The internal activity is passed through a nonlinear function ' to produce the output of the neuron y_i, where:

$$y_i = '(a_i) \quad (7)$$

The weights of the connections are adjusted during the training process to achieve the desired input/output relation of the network. A multilayer feed forward network has its neurons organized into layers with no feedback or lateral connections. Layers of neurons other than the output layer are called hidden layers. The input signal propagates through the network in a forward direction, on a layer-by-layer basis. The back propagation algorithm [24] is a supervised iterative training method for multilayer feed forward nets with sigmoidal nonlinear threshold units. It uses training data consisting of P input-output pairs of vectors that characterizes the problem. Using a generalized Least-Mean-Square algorithm the back propagation algorithm minimizes the mean square difference between the real network output and the desired output [26]. The error function that the back

propagation algorithm minimizes is the average of the square difference between the output of (3)

Assuming that all a prior probability of classes is equal, equation (3) is simplified further.

$$P(C_{kj}\, x) = \frac{P(xjC_k)P(C_k)}{P(x)} \quad (4)$$

From equation (4), to maximize $P(C_k jx)$, we only need to maximize $P(xjC_k)$ since $P(x)$ is independent of classes. Hence equation (4) becomes:

$$P(xjC_k) > P(xjC_j);\ j \neq k \quad (5)$$

This implies that C_k is selected if the condition in equation 5 is met. For the NBC, the classifier needs to learn two functions, the likelihood and prior. An advantage of the NBC is that it requires a small amount of training data to estimate the parameters necessary for classification [22].

each neuron in the output layer and the desired output. The error function can be

expressed as:

$$E = \frac{1}{P}\sum_{p}\sum_{k}(d_{pk} - o_{pk})^2 \quad (8)$$

where p is the index of the P training pair of vectors, k is the index of elements in the output vector, d_{pk} is the k^{th} element of the p^{th} desired pattern vector, and o_{pk} is the k^{th} element of the output vector when pattern p is presented as input to the network.

Minimizing the cost function represented in equation (8) results in an updating rule to adjust the weights of the connections between neurons. The weight adjustment of the connection between neuron i in layer m and neuron j in layer m+1 can be expressed as:

$$w_{ji} = \ _j o_i \quad (9)$$

TABLE I: SOME OF THE MISCLASSIFIED LETTERS FROM NBC

Actual letter	Misclassified as	Number of times
H	H	
H	•	
h	..	5
h	H	
x	H	

TABLE II: SOME OF THE MISCLASSIFIED LETTERS FROM MLP

Actual letter	Misclassified as	Number of times
H		
H	•	4
H		
•	H	
x	..	2

where i is the index of units in layer m, is the learning rate, o_i is the output of unit i in the m^th layer, and $_j$ is the delta error term back propagated from the j^th unit in layer m+1 defined by:

$$_j = [d_j \quad o_j]o_j [1 \quad o_j] \qquad (10)$$

neuron j is an output,

$$X$$

$$_j = y_j [1 \; y_j] \; _k w_{kj} \; (11) \; _k$$

neuron j is in a hidden layer and k is index of neurons in the layer (m+2), ahead of the layer of neuron j.

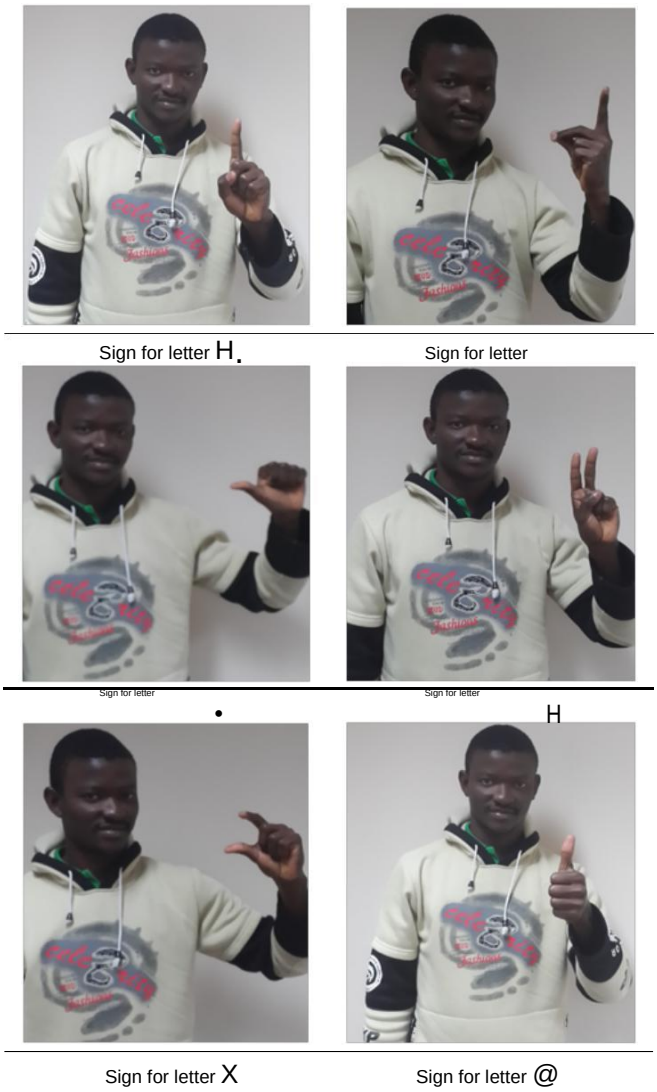
C. Classification Results

Before classification is done, the classifier is trained with part of the data. This was done using five- fold cross validation. Cross validation provides a frame work for creating several train and test splits and guaranteeing that each data points appears in the test set at least once [23]. The procedure is as follows:

Splits the data into n-equal sized groups, For i = 1 to n, a) Select group i to be the test set and all other (n-i) to be training set b) Train the model on the training set and evaluate on the test set.

Among the 2800 samples, it was found that the NBC mis-classified 76 samples with an overall accuracy of 98.3% while the MLP misclassified 26 samples with an overall accuracy of 99.1%. Table I shows some of the misclassified letter signs using the NBC and Table II shows some of the misclassified letter signs using the MLP as a classifier. Table III, shows the sign symbol of some of the misclassified letters. By analyzing the misclassified signs we notice that not all misclassified signs are similar to the signs they are classified to. This indicates that the problem could be due to the fact that the LMC detects the hands and fingers movement from one side. Thus some of the fingers may be occluded by others which reduce the reparability of the some signs. In future work we will use two LMCs, one in the front and the other on the side of the area of

TABLE III: Some of the misclassified letter signs



sign articulations. Several methods for combining the features from both LMCs will be investigated to analyze.

V. Conclusion

In this paper, we have developed a system for Arabic alphabet sign recognition using the newly introduced leap motion controller (LMC). This system releases the users from wearing a cumbersome electronic gloves or performing the signs under restrictive environmental conditions by using the two already utilized methods. Ten samples of each of the 28 Arabic alphabet signs were collected from a single signer. Ten frames were acquired from each sample letter sign, to provide a total of 2800 frames of data. Twelve features were selected from 23 values provided by the LMC for the representations of each frame in the coverage area of the LMC. For classification we compared the performance of the Nave Bayes Classifier (NBC) with a Multilayer Perceptron (MLP) trained by the back-propagation algorithm. The overall accuracy of the signs recognition using the NBC was about

98.3% while the accuracy using the MLP was about 99.1%. Analysis of the misclassified signs (48 for the NBC and 26 for the MLP out of 2,800 frames) revealed that not all misclassified letter signs are similar to the signs they are classified to. This is due to the fact that the LMC field of view is an inverted pyramid of about 8 cubic feet centered on the device. This causes some of the fingers to be occluded by the hand palm or other fingers. To solve this problem we will investigate the use of two LMC devices; one in front of the signer and the other at his side. Several methods of combining the features from the two LMC units will be investigated. Further study on the use of two LMC for Arabic word signs and full sentence recognition will be investigated in the future.

ACKNOWLEDGMENT

The authors acknowledge the support of King Fahd university of Petroleum & Minerals. This work was funded by Deanship of Scientific Research, grant number: FT131016.

REFERENCES

[1] M. Mohandes, "Arabic sign language recognition," in International conference of imaging science, systems, and technology, Las Vegas, Nevada, USA, vol. 1, 2001, pp. 753–9.

[2] O. Al-Jarrah and A. Halawani, "Recognition of gestures in arabic sign language using neuro-fuzzy systems," Artificial Intelligence, vol. 133, no. 1, pp. 117–138, 2001.

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

[4] K. Assaleh and M. Al-Rousan, "Recognition of arabic sign language alphabet using polynomial classifiers," EURASIP Journal on Applied Signal Processing, vol. 2005, pp. 2136–2145, 2005.

[5] M. Mohandes and S. Buraiky, "Automation of the arabic sign language recognition using the powerglove," AIML Journal, vol. 7, no. 1, pp. 41–46, 2007.

[6] M. A. Mohandes, "Recognition of two-handed arabic signs using the cyberglove," Arabian Journal for Science and Engineering, vol. 38, no. 3, pp. 669–677, 2013.

[7] M. Maraqa and R. Abu-Zaiter, "Recognition of arabic sign language (arsl) using recurrent neural networks," in Applications of Digital Information and Web Technologies, 2008. ICADIWT 2008. First International Conference on the. IEEE, 2008, pp. 478–481.

[8] N. El-Bendary, H. M. Zawbaa, M. S. Daoud, K. Nakamatsu et al., "Arslat: Arabic sign language alphabets translator," in Computer Information Systems and Industrial Management Applications (CISIM), 2010 International Conference on. IEEE, 2010, pp. 590–595.

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

[10] M. Mohandes, M. Deriche, U. Johar, and S. Ilyas, "A signer-independent arabic sign language recognition system using face detection, geometric features, and a hidden markov model," Computers & Electrical Engineering, vol. 38, no. 2, pp. 422–433, 2012.

[11] S. J. Reyadh Naoum, Hussein H. Owaied, "Development of a new arabic sign language recognition using k-nearest neighbor algorithm," Journal of Emerging Trends in Computing and Information Sciences, vol. 3, pp. 1173–1178, 2012.

[12] Y. Quan, "Chinese sign language recognition based on video sequence appearance modeling," in Industrial Electronics and Applica-tions (ICIEA), 2010 the 5th IEEE Conference on. IEEE, 2010, pp. 1537–1542.

[13] Y. Quan et al., "Application of improved sign language recognition and synthesis technology in ib," in Industrial Electronics and Applications, 2008. ICIEA 2008. 3rd IEEE Conference on. IEEE, 2008, pp. 1629– 1634.

[14] A. Agarwal and M. K. Thakur, "Sign language recognition using microsoft kinect," in Contemporary Computing (IC3), 2013 Sixth In-ternational Conference on. IEEE, 2013, pp. 181–185.

[15] E.-J. Ong, H. Cooper, N. Pugeault, and R. Bowden, "Sign language recognition using sequential pattern trees," in Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012, pp. 2200–2207.

[16] A. Memis and S. Albayrak, "Turkish sign language recognition using spatio-temporal features on kinect rgb video sequences and depth maps," in Signal Processing and Communications Applications Conference (SIU), 2013 21st. IEEE, 2013, pp. 1–4.

[17] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, "Analysis of the accuracy and robustness of the leap motion controller," Sensors (Basel, Switzerland), vol. 13, no. 5, p. 6380, 2013.

[18] http://dartmouthbusinessjournal.com/2013/08/ the-leap-motion-controller-and-touchless-technology.

[19] https://developer.leapmotion.com/documentation/-Languages/C++/ Guides/Leap Overview.html#id1.

[20] http://phys.org/news/2013-08-motion-ready-prime.html.

[21] https://forums.leapmotion.com.

[22] R. O. Duda, P. E. Hart, and D. G. Stork, Pattern classification. John Wiley & Sons, 2012.

[23] M. J. Islam, Q. J. Wu, M. Ahmadi, and M. A. Sid-Ahmed, "Investigating the performance of naive-bayes classifiers and k-nearest neighbor classifiers," in Convergence Information Technology, 2007. International Conference on. IEEE, 2007, pp. 1541–1546.

[24] J. L. McClelland, D. E. Rumelhart, P. R. Group et al., "Parallel distributed processing," Explorations in the microstructure of cognition, vol. 2, 1986.

[25] P. Wasserman, "Advanced methods in neural computing, 1993."

[26] R. P. Lippmann, "An introduction to computing with neural nets," ASSP Magazine, IEEE, vol. 4, no. 2, pp. 4–22, 1987.