Arabic Sign Language Recognition with 3D Convolutional Neural Networks

Menna ElBadawy
Scientific Computing Department
Ain Shams University
Cairo, Egypt
Menna-elbadawy@hotmail.com

A. S. Elons
Scientific Computing Department
Ain Shams University
Cairo, Egypt
ahmed.new80@hotmail.com

Howida A. Shedeed
Scientific Computing Department
Ain Shams University
Cairo, Egypt
Dr_howida@yahoo.com

M.F.Tolba
Scientific Computing Department
Ain Shams University
Cairo, Egypt
fahmytolba@gmail.com

***Abstract*— Sign Language recognition is very important for communication purposes between Hearing Impaired (HI) people and hearing ones. Arabic Sign Language Recognition field became widespread because of its difficult nature and numerous details. Most researchers employed different input sensors, features extractors, and classifiers on static and dynamic data. These different ways were customized and employed in our previous work in the Arabic Sign Language Recognition field. In this paper, features extractor with deep behavior was used to deal with the minor details of Arabic Sign Language. 3D Convolutional Neural Network (CNN) was used to recognize 25 gestures from Arabic sign language dictionary. The recognition system was fed with data from depth maps. The system achieved 98% accuracy for observed data and 85% average accuracy for new data. The results could be improved as more data from more different signers are included.**

Keywords*— *Arabic Sign Language, Deep models, Convolutional Neural Network, Sign Language Recognition.

I. Introduction

Sign language is the mean of communication for the HI people. It intended to be the principle way to communicate with the surrounding community. Also, sign language differs from country to another based on its mother language, as it is a descriptive language.

Sign language recognition is the field of research that gained much focus in past few years. In the past, the HI people needed human experts in sign language translation to communicate with the world. This way was very expensive and inconvenient. Therefore, the need for automatic Sign Language Recognition system increased. But the recognition task is still a challenging problem in spite of the researchers' attempt to find a reasonable solution.

Arabic Sign language recognition is a very difficult task. As the language contains hundreds of words and the words could be very similar in the hand poses. The gestures are not only played with hands, also it includes other different nonverbal communication mean such as facial expressions and body movements. These additional nonverbal communication means may be distinguished between many gestures which are similar in hand poses.

In this paper, we focus on using Convolutional Neural Network to build a 3D vision for the dataset used. The dataset which was used contains 25 words of the Arabic Sign Language and was collected from two different signers. We represent each gesture as a class and its postures build its own 3D CNN model. 3D CNN will help extract spatial-temporal features and motion information encoded in multiple contiguous frames. The system achieved high accuracy in comparison to the relative classical Artificial Neural Network that used to be in such researches.

II. Related Work

Sign language recognition is an attracting field to research. The nature of the sign language is that it contains many parameters that are involved in the word meaning. Many types of research were conducted to study and show the effect of these parameters.

The hand gesture is the key parameter in the Sign language. So, it gained a lot of focus and many types of research were conducted that discuss its different input ways: - The digital camera is the first and axiomatic way to input [1, 2, 3, 4, 5, 6, 7], and the data gloves [8, 9]. Also, the modern sensors are used and customized to use their benefits on the recognition purpose such as Microsoft Kinect and Leap Motion Sensors [10, 11, 12, 13, 14, 15, 16, 17, 18].

Recently, the Deep learning models gained the focus due to their deep layers architecture with the iterative nature. Byeongkeun Kang, Subarna Tripathi, and Truong Q. Nguyen [19] used CNN to recognize the fingerspelling on American Sign Language dataset. They used a dataset of 31 alphabets and numbers. The system was fed with depth maps for the data which was captured by depth sensor and digital camera. The system achieved 83.5% to 85.5% accuracy. CNN was used to recognize Italian Sign Language [20]. The dataset that was used consists of 20 Italian gestures which were recorded by Microsoft Kinect. The system consisted of two CNNs; one for hand features extraction and another for upper-body features extraction. The results then were concatenated to enter a classical Neural Network Classifier. The system could achieve a validation accuracy 91.7%. Also, the CNN was used to

978-1-5386-0821-0/17/$31.00 ©2017 IEEE

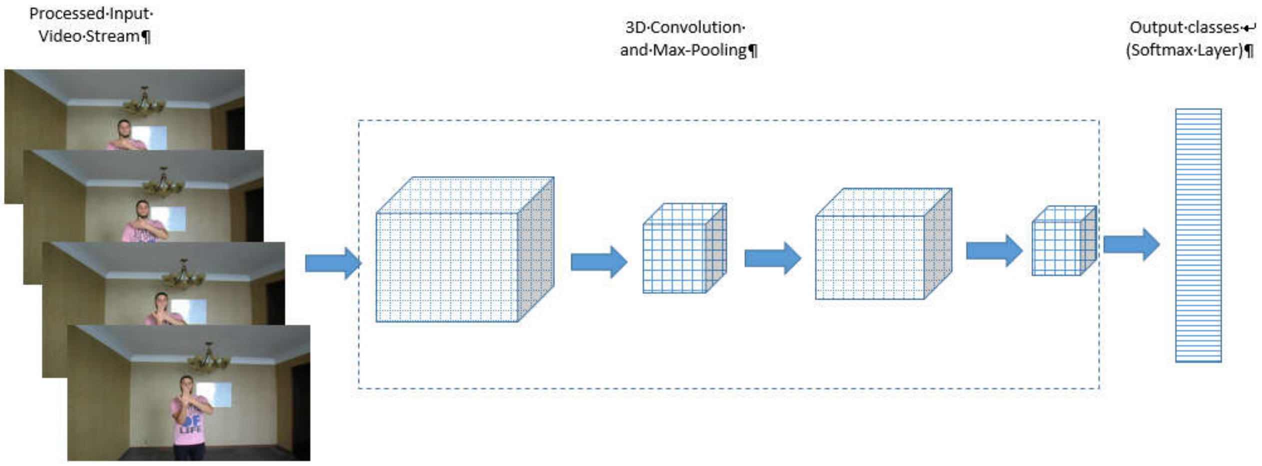


Fig. 1. Arabic Sign Language Recognition System with 3D CNN.

recognize the Indian Sign Language [21]. The dataset contained the 26 alphabets as static images. The system could achieve the maximum accuracy of 100% for all the alphabet. Su Yang and Qing Zhu [22] combined the CNN with Long Short-Term Memory (LSTM) network to recognize Chinese Sign Language. The video is regarded as a sequence of frames, then entered into the LSTM model that is fully connected with CNN. The system was tested with 40 words and had high recognition rate.

Multichannel CNN is a model of the normal CNN but it used cubic kernel rather than the normal 2D kernel. The multichannel CNN concept was applied on static images to recognize the hand postures [23]. Three channels were used for the raw image, and Sobel filtered image on vertical and horizontal dimensions. The system was evaluated using two different datasets of static postures and obtained accuracy rate greater than 90%. 3D CNN was used to recognize the sign language in a different data format [24]. The system was fed with different inputs extracted from video streams such as color information, depth, and body joint positions. The system was evaluated with a dataset of 25 vocabularies, and the system achieved a maximum accuracy of around 94%.

III. Proposed System

This section presents an Arabic Sign Language Recognition system that utilizes depth and intensity channels based on 3D Convolutional Neural Network. **Error! Reference source not found.** shows the system architecture.

A. Data Specifications

Our dataset has 25 words from the Unified Arabic Sign Language dictionary [25]. Each word is played with two signers in two different backgrounds. Every signer plays 4 times per each word. So each word has 8 samples and the dataset then contains 25x8=200 samples total. The dataset then is divided into 125 samples for training and the remaining 75 samples are used for testing. The dataset was recorded by a digital camera with resolution 1280x720. **Error! Reference source not found.** shows sample frames that were extracted from the dataset words in the two different backgrounds used.

B. Preprocessing

The data enter the system as a video contains one gesture. As each video has a different period depending on the number of postures inside it, a unique number of frames is chosen to represent each posture. And these frames are processed to get reasonable dimensions and the most effective frames.

The video is divided into frames with different frame rate. The very low frame rate can cause significant data loss as some postures' frames can be missed while high frame rate causes redundancy that leads to characterize a word with some of its postures and these postures may be common in a number of words. Also, high frame rate requires a heavy computational power. The experiment was conducted using different frame rates; 10, 30 and 50 frames per second.

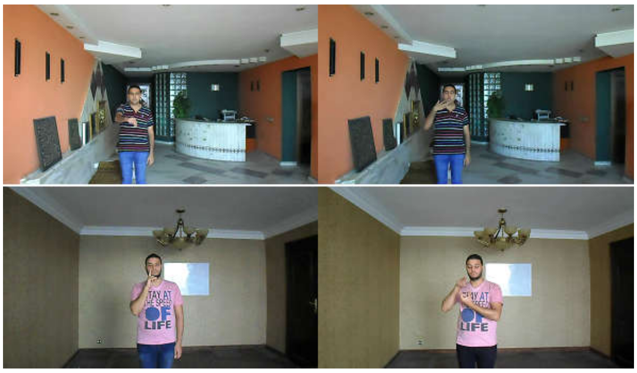


Fig. 2. Sample images from the Dataset used.

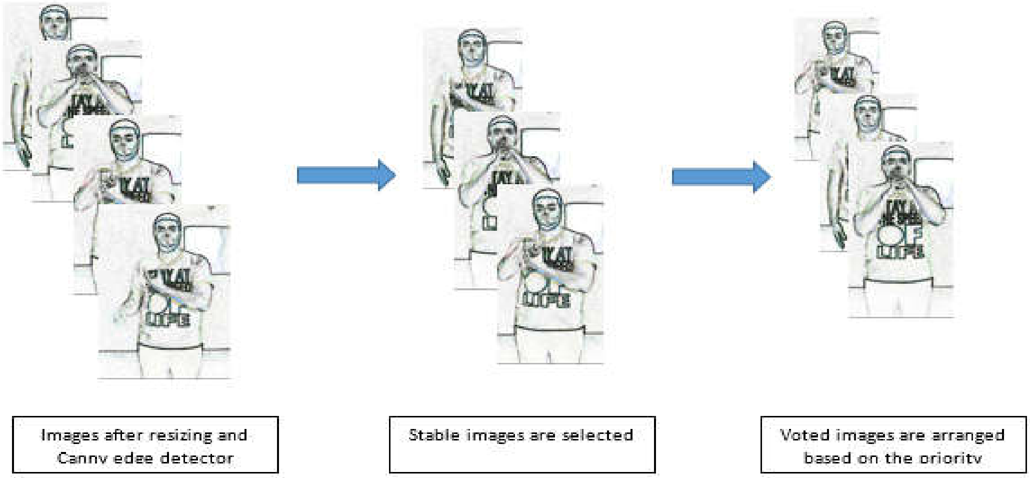
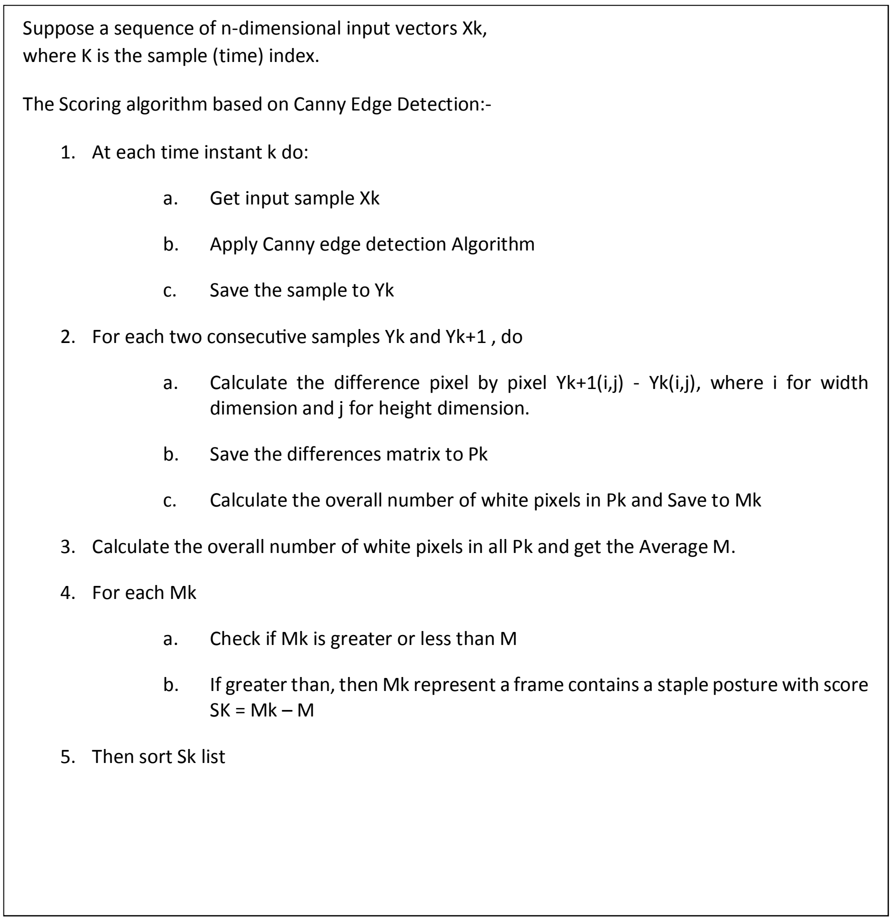

Images after resizing and Canny edge detector
Stable images are selected
Voted images are arranged based on the priority

Fig. 3. The steps of scoring algorithm.

After dividing the video, the frames then are downsampled and scored. The frames resized to 104x116 that will save the computational power. To prepare the word to enter the developed system, we select some frames to each word according to the priority calculated from the Scoring Algorithm **Error! Reference source not found.** . Then the frames with the highest priority are selected to represent the word.

The Scoring Algorithm mentioned above uses the frames after dividing the video. These frames are resized and processed



```
Suppose a sequence of n-dimensional input vectors Xk,
where K is the sample (time) index.

The Scoring algorithm based on Canny Edge Detection:-

    1. At each time instant k do:
            a. Get input sample Xk
            b. Apply Canny edge detection Algorithm
            c. Save the sample to Yk
    2. For each two consecutive samples Yk and Yk+1 , do
            a. Calculate the difference pixel by pixel Yk+1(i,j) - Yk(i,j), where i for width
               dimension and j for height dimension.
            b. Save the differences matrix to Pk
            c. Calculate the overall number of white pixels in Pk and Save to Mk
    3. Calculate the overall number of white pixels in all Pk and get the Average M.
    4. For each Mk
            a. Check if Mk is greater or less than M
            b. If greater than, then Mk represent a frame contains a staple posture with score
               SK = Mk – M
    5. Then sort Sk list
```

Fig. 4. Scoring Algorithm

with Canny edge detector [26] to get the binary images contain sharp edges. **Error! Reference source not found.** shows the Algorithm steps. Depending on these binary images, measurements are calculated that compare the percentage of the white pixels to the black pixels of each frame to the total gesture frames. These measurements are used to conclude if this frame is stable or not. If stable, then make a vote for it with a number indicating its priority. If not stable, then it is neglected. **Error! Reference source not found.** shows the steps of the Scoring Algorithm started after the raw video's frames are entered into the algorithm.

C. Training

In this step, we already have the samples arranged and ready to enter the Recognition System. The depth of the 3D CNN depends on the number of frames that were selected. **Error! Reference source not found.** shows the different values for the data depth according to the frame rate used.

TABLE I. CONFIGURATION TABLE FOR THE INPUT DEPTH

Configure Number	Frame Rate	Selected number of frames (depth)
1	10	3
2	10	7
3	30	10
4	30	20
5	50	30
6	50	15
7	50	40

Then, the input enters the 3D CNN that extracts the spatial-temporal features of that singular this class from the remaining classes even if there are common postures. The last layer of the 3D CNN is a Softmax layer to classify these features to one of 25 classes that represent the dataset used. A number of neurons in the Softmax layer are equaled to the number of classes. The softmax activation function is the best choice in our case that produces a probability value of less than 1.0. This value represents the probability of the node to be the expected output. The output node's value tends to be 1 and all the remaining nodes will tend to be zero.

```
Canny Edge Detection:-
     1. Filter image with derivative of Gaussian
            a. $f_x = \frac{\partial}{\partial x}(f * G) = f * \frac{\partial}{\partial x}G = f * G_x$
            b. $f_y = \frac{\partial}{\partial y}(f * G) = f * \frac{\partial}{\partial y}G = f * G_y$
               $G(x,y)$ is the Gaussian Function
               $G_x(x,y)$ is the derivate of $G(x,y)$ with respect to x
               $G_y(x,y)$ is the derivate of $G(x,y)$ with respect to y
     2. Find magnitude
            a. $magn(i,j) = \sqrt{{f_x}^2 + {f_y}^2}$
     3. Non-maximum suppression
            a. For each pixel (x,y) do:
               if magn(i, j) < magn($i_1$, $j_1$)
               or magn(i, j) < magn($i_2$, $j_2$)
                    then $I_N$ (i, j) = 0
               else $I_N$ (i, j) = magn(i, j)
     4. Linking and thresholding
            a. Produce two thresholded images $I_1$(i, j) and $I_2$(i, j).
            b. Link the edges in $I_2$(i, j) into contours
                   i. Look in $I_1$(i, j) when a gap is found.
                  ii. By examining the 8 neighbors in $I_1$(i, j),
                      gather edge points from $I_1$(i, j)until the gap
                      has been bridged to an edge in $I_2$(i, j).
```

Fig. 5. Canny Edge Detection Algorithm [26]

IV. Experimental Results

The Arabic Sign Language Recognition system is evaluated using a dataset consisting of 25 words. The experiments are conducted on a machine with specifications: Core i5-2520M CPU, 2.5 GHz, 3.78 GB memory RAM, Intel HD Graphics 3000, and Windows8 64 bit operating system.

In order to test the developed system, many runs with different values for the CNN parameters are used to get the best results. The 3D CNN was fed with input with different depth values that were mentioned in **Error! Reference source not found.**. These values are used in **Error! Reference source not found.** to show the different trials with the system architecture. Many trials ran on network with parameters' values: decay 0.001, momentum 0.1, 0.5, 0.9, and learning rate 0.1, 0.5, 0.9. But the trials with the learning rate 0.5 and momentum 0.1 showed the highest accuracy rate, so they are listed on the table.

As we see from the table, configuration number 3 obtained the highest accuracy rate. This configuration has a frame rate reasonable enough to get frames for all word's postures and at the same time, not too many frames that might cause getting frames of unstable postures that may lead to misclassification. Also, the depth used is also reasonable as it not very large which consumes computational power to process and not very small to miss some postures' frames.

Out from the computational power issue, the large frame rate and so the depth cause lower accuracy rate. To get a large number of the frames, the threshold that is used in the Scoring Algorithm has to be minimized which may lead to unstable postures to be selected and voted. As the CNN builds 3D features from the depth these frames entered the system, the resulted features would include unnecessary features from these unstable postures. Subsequently, the misclassification occurs. Also, the larger depth used the higher chance for overfitting problem. This problem leads to misclassification too because the system could not recognize any slight change in the testing words.

On the other hand, the less depth causes lower accuracy rate too. The threshold has to be maximized which may lead to miss necessary postures in the word thereby losing some important features in the 3D features map generated from CNN. This will lead to confusion between more than one word. So, misclassification problem occurs.

Finally, after many trials, the most suitable frame rate and depth are used. The average computational power and accuracy rate are obtained.

TABLE I. THE CLASSIFICATION RESULTS WITH THE CNN CONFIGURATION

Input Depth (from table I)	CNN		Accuracy	
Configuration #1	Layer 1	Kernel	5x5x8	91%
		Subsampling	2x2	
	Layer 2	Kernel	5x5x16	
		Subsampling	3x3	
Configuration #1	Layer 1	Kernel	5x5x8	90.3%
		Subsampling	2x2	
	Layer 2	Kernel	5x5x4	
		Subsampling	2x2	
Configuration #2	Layer 1	Kernel	5x5x8	90.4%
		Subsampling	2x2	
	Layer 2	Kernel	5x5x16	
		Subsampling	3x3	
Configuration #2	Layer 1	Kernel	5x5x8	90%
		Subsampling	2x2	
	Layer 2	Kernel	5x5x4	
		Subsampling	2x2	
Configuration #3	Layer 1	Kernel	5x5x8	93%
		Subsampling	2x2	
	Layer 2	Kernel	5x5x16	
		Subsampling	3x3	
Configuration #3	Layer 1	Kernel	5x5x8	91.8%
		Subsampling	2x2	
	Layer 2	Kernel	5x5x4	
		Subsampling	2x2	
Configuration #4	Layer 1	Kernel	**5x5x8**	91%
		Subsampling	2x2	
	Layer 2	Kernel	5x5x16	
		Subsampling	3x3	
Configuration #4	Layer 1	Kernel	5x5x8	90.5%
		Subsampling	2x2	
	Layer 2	Kernel	5x5x4	
		Subsampling	2x2	
Configuration #5	Layer 1	Kernel	5x5x8	88%
		Subsampling	2x2	
	Layer 2	Kernel	5x5x16	
		Subsampling	3x3	
Configuration #5	Layer 1	Kernel	5x5x8	89.2%
		Subsampling	2x2	
	Layer 2	Kernel	5x5x4	
		Subsampling	2x2	
Configuration #6	Layer 1	Kernel	5x5x8	89.7%
		Subsampling	2x2	
	Layer 2	Kernel	5x5x16	
		Subsampling	3x3	
Configuration #6	Layer 1	Kernel	5x5x8	91%
		Subsampling	2x2	
	Layer 2	Kernel	5x5x4	
		Subsampling	2x2	
Configuration #7	Layer 1	Kernel	5x5x8	85%
		Subsampling	2x2	
	Layer 2	Kernel	5x5x16	
		Subsampling	3x3	
Configuration #7	Layer 1	Kernel	5x5x8	86.2%
		Subsampling	2x2	
	Layer 2	Kernel	5x5x4	
		Subsampling	2x2	

In our previous researches, other aspects that affect the recognition are considered such as facial expressions [27]. Also, different input sensors such as Leap Motion Controller are used [13] and combining more than one input sensor to address different aspects mentioned before [12]. And in this research, a new interesting learning model is used and showed promising results.

I. Conclusion

In this research, an efficient system for Arabic Sign Language Recognition with 3D Convolutional Neural Network was developed. The system uses the word's video stream and gets a normalized depth input. The architecture extracts spatial-temporal features from the input. The Softmax layer acts as a classifier for the features. The system could achieve an accuracy rate greater than 90%. This accuracy could be enhanced with more training samples added to the dataset to accept the variety of the signer and the environment surrounded. The experimental results show the effectiveness of the 3D deep architecture.

References

[1] M. Z. Abdo, S. A. E.-R. Salem, A. M. Hamdy and E. . M. Saad, "Arabic Alphabet and Numbers Sign Language Recognition," International Journal of Advanced Computer Science and Applications(IJACSA), vol. 6, no. 11, 2015.

[2] A. Aliaa, A. Youssif, A. E. Aboutabl and H. H. Ali, "Arabic Sign Language (ArSL) Recognition System Using HMM," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 2, no. 11, 2011.

[3] D. J. Singha and K. , "Recognition of Indian Sign Language in Live Video," International Journal of Computer Applications (0975 – 8887), vol. 70, no. 19, 2013.

[4] N. El-Bendary, H. M.zawbaa, M. S.Daoud, A. E. Hassanien and K. Nakamatsu, "ArSLAT: Arabic Sign Language Alphabets Translator," International Journal of Computer Information Systems and Industrial Management Applications, vol. 3, no. 2150-7988, pp. 498-506., 2011.

[5] M. Nachamai, "Alphabet Recogntion of American Sign Language: A Hand Gesture Recogntion Approach Using SIFT Algorithm," International Journal of Artificial Intelligence & Applications(IJAIA), vol. 4, no. 1, 2013.

[6] J. R. Pansare, S. H. Gawande and M. Ingle, "Real Time Static Hand Gesture Recognition for American Sign Language (ASL) in Complex Background," Journal of Signal and Information Processing, 2012.

[7] A. Medhi, M. Doshi, P. Pawar, A. Kesarkar and P. R. Dalvi, "Real-Time Vision Based Sign Language Recognition System," International Journal of Innovative Research in Computer and Communication Engineering, vol. 5, no. 2, 2017.

[8] N. Tubaiz, T. Shanableh and K. Assaleh, "Glove-Based Continuous Arabic Sign Language Recognition in User-Dependent Mode," IEEE TRANSACTIONS ON HUMAN-MACHINE SYSTEMS, vol. 45, no. 4, AUGUST 2015.

[9] K. Anetha and . P. J. Rejina, "Hand Talk-A Sign Language Recognition Based On Accelerometer and SEMG Data," International Journal of Innovative Research in Computer and Communication Engineering, vol. 2, no. 3, 2014.

[10] A.-N. M. A.Almasre and H. , "A Real-Time Letter Recognition Model for Arabic Sign Language Using Kinect and Leap Motion Controller v2," International Journal of Advanced Engineering, Management and Science (IJAEMS), ISSN:2454-1311, vol. 2, no. 5, May 2016.

[11] C.-H. Chuan, E. Regina and C. Guardino, "American Sign Language Recognition Using Leap Motion Sensor," in Proceedings of the 13th International Conference on Machine Learning and Applications, USA, 2014.

[12] M. ElBadawy, A. S. Elons, H. Sheded and M. Tolba, "A Proposed Hybrid Sensor Archeticure For Arabic Sign Language Recognition," in Intelligent Systems'2014 Advances in Intelligent Systems and Computing, 2014.

[13] A. Elons, M. Ahmed, H. Shedid and M. Tolba, "An Arabic Sign Language Recognition Uing Leap Motion Sensor," in Computer Engineering & Systems (ICCES), 2014 9th International Conference, 2014.

[14] S. Lang, M. Block and R. Rojas, "Sign Language Recognition Using Kinect," in Proceedings of the 11th international conference on Artificial Intelligence and Soft Computing, 2012.

[15] L.-E. Potter, J. A. Araullo and L. A. Carter, "The Leap Motion controller: A view on sign language," in The 25th Australian Computer-Human Interaction Conference, 2013.

[16] L. Zheng and B. Liang, "Sign language recognition using depth images," in 2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV), Phuket, Thailand, 2016.

[17] A. Ben Jmaa, W. Mahdi, Y. Ben Jemaa and A. Ben Hamadou, "Arabic sign language recognition based on HOG descriptor," in Proc. SPIE 10225, Eighth International Conference on Graphic and Image Processing (ICGIP 2016), Tokyo, Japan, 2016.

[18] C. Zhang, Y. Tian and M. Huenerfauth, "MULTI-MODALITY AMERICAN SIGN LANGUAGE RECOGNITION," in 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 2016.

[19] B. Kang, S. Tripathi and T. Q. Nguyen, "Real-time Sign Language Fingerspelling Recognition using Convolutional Neural Networks from Depth map," in arXiv:1509.03001, 2015.

[20] L. Pigou, S. Dieleman, P.-J. Kindermans and B. Schrauwen, "Sign Language Recognition Using Convolutional Neural Networks," in Computer Vision - ECCV 2014 Workshops, 2014.

[21] N. and N.Rajeswari, "Sign Language Recognition Using Convolutional Neural Networks," International Journal on Recent and Innovation Trends in Computing and Communication, vol. 5, no. 6.

[22] S. Yang and Q. Zhu, "Continuous Chinese sign language recognition with CNN-LSTM," in Proc. SPIE 10420, Ninth International Conference on Digital Image Processing (ICDIP 2017), Hong Kong, China, 2017.

[23] P. Barros, S. Magg, C. Weber and S. Wermter, "A Multichannel Convolutional Neural Network for Hand Posture Recognition," in The 24th International Conference on Artificial Neural Networks (ICANN 2014),, Hamburg, 2014.

[24] J. Huang, W. Zhou, H. Li and W. Li, "Sign Language Recognition Using 3D Convolutional Neural Networks," in Multimedia and Expo (ICME), 2015 IEEE International Conference , Turin, Italy, 2015.

[25] S. Samreen and M. Benali, "قواعد لغة الاشارة العربية القطرية الموحدة،" 2009.

[26] E. Trucco and J. e. a. , "Edge Detection, Chapter 4 and 5".

[27] A. Elons, M. Ahmed and H. Shedid, "Facial Expressions Recognition for Arabic Sign Language Translation," in Computer Engineering & Systems (ICCES),2014 9th International Conference, 2014.

[28] P. Molchanov, S. Gupta and K. Kim, "Hand Gesture Recognition with 3D Convolutional Neural Networks," in Computer Vision and Pattern Recognition Workshops (CVPRW), 2015 IEEE Conference, Boston, MA, USA, 2015.

V.