1st International Conference for Pure Science

University of Diyala, College of Science, in a partnership with Western Michigan University

ICPS-2021

Publication Acceptance Letter

Manuscript No. 145

Authors: Hind Ibrahim Mohammed and Jumana Waleed

We are pleased to inform you that your manuscript No. (145) entitled "**Hand Gesture Recognition Using a Convolutional Neural Network for Arabic Sign Language**" has been accepted for presentation in the **"1st International Conference for Pure Science (ICPS-2021)** which will be held at College of Science / University of Diyala, Iraq in partnership with Western Michigan University, USA on 26-27 May 2021. Your paper will be published online by **AIP Conference Proceeding (ISSN: 0094-243X).**

Your interest in ICPS-2021is very much appreciated. We look forward to meet you at this event.

With Best Regards,

Prof. Dr. Tahseen H. Mubarak

Chairman of ICPS-2021

4th National Scientific Conference

Hand Gesture Recognition Using a Convolutional Neural Network for Arabic Sign Language

Hind Ibrahim Mohammed [1, a)] and Jumana Waleed [2, b)]

[1, 2] *Department of Computer Science, College of Science, University of Diyala, Iraq.*

[a] hindimohammed83@gmail.com,

[b]jumanawaleed@sciences.uodiyala.edu.iq

ABSTRACT. Communication is one of the most basic human needs, but for between deaf and dumb people and normal people, communication in their daily lives is a challenge, due to the lack of reliable and easy-to-use assistive devices and skilled sign language interpreters. The need to integrate hard-of-hearing Arab individuals into their societies has recently received greater attention than many public and private institutions. Accordingly, In the fields of artificial intelligence(AI) and machine learning(ML), automating sign language recognition has become a critical technology, this paper presents a proposed system for static hand gestures recognition for the Arabic sign language (ArSL) using a Convolutional Neural Network(CNN) with accuracy 97.42 %.

Keywords Arabic sign language (ArSL), Convolutional Neural Network (CNN).

INTRODUCTION

Gesture recognition in today's technologies is an emerging topic. The major objective of this is to use mathematical algorithms for human-computer interaction (HCI), Typically Gestures may arise from any movement or condition of the body, however generally come from the face or hand [1][2].

The deaf and dumb people have been disconnected from the community, and it is impossible for average people to learn sign language. Not only for deaf and dumb individuals, sign language learning has been adopted, but also as a medium for common people to communicate with them [3] [4]. In addition to concerns regarding differences (languages) of Arabic sign language (ArSL) throughout the Arab countries due to the lack of communication systems in Arabic sign language, which directs our focus to (ArSL) [5].

Generally, gestures can be classified into static gestures and dynamic gestures. Static gestures are usually described in terms of hand shapes and dynamic gestures are generally described according to hand movements [6], as in Figure 1. Vision-based techniques and sensor-based techniques are the two types of hand gesture recognition techniques [7] [8].

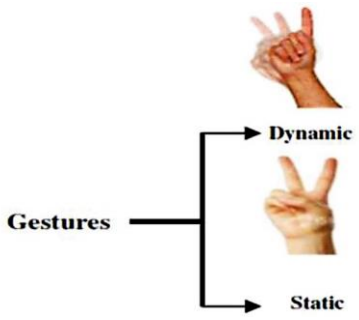

Figure (1): Static and dynamic gestures.

The techniques examined in this paper are divided into four stages: data acquisition, data preprocessing, feature extraction, and gesture recognition, with various algorithms elaborated and their merits compared at each stage. Overall, the study is hoped to provide readers with a thorough introduction to the field of automated gesture and sign language recognition, as well as aid future research efforts in this area [9]. The remaining part of the paper is sketched out as follows: Section (1) contains a review of the literature, Section (2) contains Data Acquisition, Section (3) contains data preprocessing Section (4) contains Architecture**,** Section (5) contains Experimental Result and Discussion, finally, Section (6) contains a conclusion.

REVIEW OF THE LITERATURE

There are lots of domaines that utilized the machine learning and deep learning algrithms [10][11][12][13][14]. In the domain of hand gestures, many achievements are presented by many researchers.

M. Elbadawy et al. (2017) [15], proposed a system for features extractor with deep behavior to deal with the minor details of Arabic Sign Language. 25 gestures from the ArSL dictionary were recognized using 3D CNN. Data from depth maps was fed into the recognition system. For observed data, the method was 98 % accurate, and for new data, it was 85 % accurate on average.

S. Hayani et.al (2019) [16], proposed system using CNN. This machine would automatically recognize the numbers and letters of Arabic sign language if fed a real data set. A comparative analysis was conducted to validate the scheme, demonstrating the feasibility and robustness of the proposed method relative to traditional methods based on k-closest neighbors (KNN) and support vector machines (SVM), as well as the accuracy of CNN 90.02 %.

A. Hasasneh (2020) [17], proposed ArSL method, An unsupervised deep learning algorithm is based on a deep belief network (DBN) combined with the direct use of tiny images to recognize and categorize Arabic alphabetical letters. The use of deep learning has aided in the extraction of the most significant features that are sparsely represented, as well as simplifying the overall recognition process. After resizing and normalization, about 6,000 samples of the 28 Arabic alphabetic signs were used for feature extraction. A soft max regression was used to analyze the classification approach, and an overall precision of 83.32% was achieved. The model had a sensitivity and specificity of 70.5 and 96.2 % respectively.

M. M. Kamruzzaman (2020) [18], proposed a system to detect hand sign with CNN automatically to dataset (ArSL), The RMSProp optimizer was used to train the system for 100 epochs with a cost function based on Categorical Cross Entropy, and it converged well before the 100th epoch, allowing the weights to be saved with the system for use in the next stage. The device is then linked to its signature point, where a hand sign is translated into Arabic speech with 90% accuracy. Since only 100 images were used, it could be enhanced by the amount of images used.

Data Acquisition

The dataset form (Mendeley Data) includes 54,049 images (64x64) of ArSL alphabets performed by more than 40 people for 32 standard Arabic signs and alphabets. Depending on the class, the number of images per class varies. A sample picture of all Arabic Language Signs is also included. The CSV file includes the Label of each corresponding Arabic Sign Language Image based on the image file name [19][20].

This paper's dataset includes 1000 image for training set and 200 image for testing set for each hand sign, as well as 32 letter of Arabic sign language. The proposed system is put to the test by combining hyper parameters in various ways to get the best results in the shortest amount of time.

Data Preprocessing

Data preprocessing is the first step in creating a functioning deep learning model. It is used to transform raw data into a format that is both accessible and effective. A data preprocessing flow diagram is shown in (Figure 2):

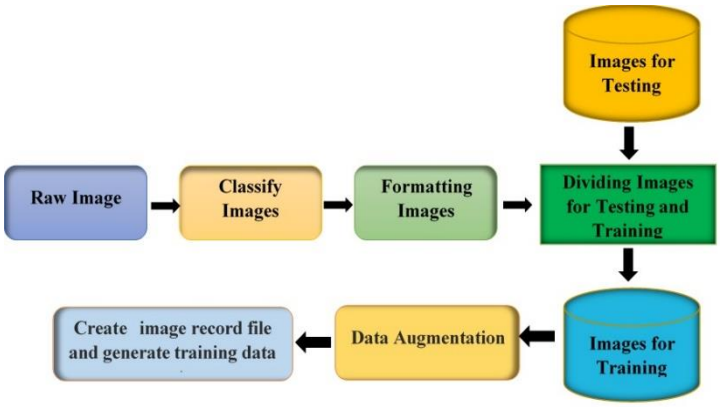

Figure (2): A data preprocessing flow diagram

Raw images of hand signs are taken with a camera and used to incorporate the proposal system. The images were taken in the following setting: from various perspectives, altering the lighting situation, in concentrate and of high quality by altering the scale and distance of the objects. The aim of creating raw images is to create a training and testing dataset. The Arabic Alphabet is depicted in (Figure 3), which is part of the dataset for the proposed scheme.

Figure (3): Arabic Alphabet Sign (data set image).

The proposed system divides images into 32 groups for each of the Arabic alphabet's 32 letters. For device implementation, one subfolder is used to store images from one group. In the proposed system, all subfolders representing Categories are held together in one main folder called "Data Set".

we have applied random transformations like shear, rotation, zoom, width shift, and height shift using the image data generator class from the Keras library. After that, the image is converted to a grayscale image with a gray level intensity range of 0 to 255. To lessen the impact of lighting and scaling shifts. The following equation is used to normalize the grayscale picture. The following equation (1) is used to normalize the grayscale image:

$$y = ((x - min) * 255 / (max - min)) \quad (1)$$

Where x is the gray scale intensity of the original image, y is the gray scale intensity of the output image after normalization, min is the original image's minimum gray scale intensity value, and max is the original image's maximum gray scale intensity value.

Due to the big size of the dataset, we made a validation split of 80:20. Meaning 80% of images (which is equal to 25600 images belonging to 32 classes) will be used for training the model. And the remaining 20% (6400 images belonging to 32 classes) will be used for model validation.

Data Augmentation

Data is absolutely critical to artificial intelligence because deep learning models hunger for it. Adding more training data to your deep learning model is the most effective way to rapidly improve its efficiency.

Data collection and marking, on the other hand, can be time-consuming and costly. As a result, several Deep Learning researchers are looking into Data Augmentation techniques to artificially add training data to these huge, data-hungry models.

For image data, there are a variety of traditional/easy-to-implement data augmentation methods. Horizontal image flipping is one of the most popular techniques. Horizontally rotating pictures and cropping them while holding the eye on the object of interest as (Figure 4) was derived from the propose system software, as discussed below:

```
train_datagen = ImageDataGenerator(
        rescale=1 / 255.0,
        featurewise_std_normalization=True,
        rotation_range=20,
        zoom_range=0.05,
        width_shift_range=0.05,
        height_shift_range=0.05,
        shear_range=0.05,
        horizontal_flip=True,
        fill_mode="nearest",
        validation_split=0.20)
```

Figure (4): Data Augmentation

Follow these steps to build an image record file and a training dataset: We'll need to make a list of all the photos in a separate folder to get information from a mark or filename.

ARCHITECTURE

CNN is one of the most effective tools for representing image data at a higher level. CNN attempts to return the inference about pixels by learning how to extract features from image pixel data that has been provided as input. Convolution is a mathematical linear operation between matrices that gives it its name. In recent years, Automatic feature learning had been successfully implemented using CNN. It performed admirably in image detection, object recognition, and even recognition of human behavior. The availability of large databases containing millions of samples is credited with this outstanding success.

A deep learning technique based on CNN is used to identify each gesture by automatically learning and extracting features. The CNN structure of the proposal system as shown in (Figure 5) explained all layers as following:

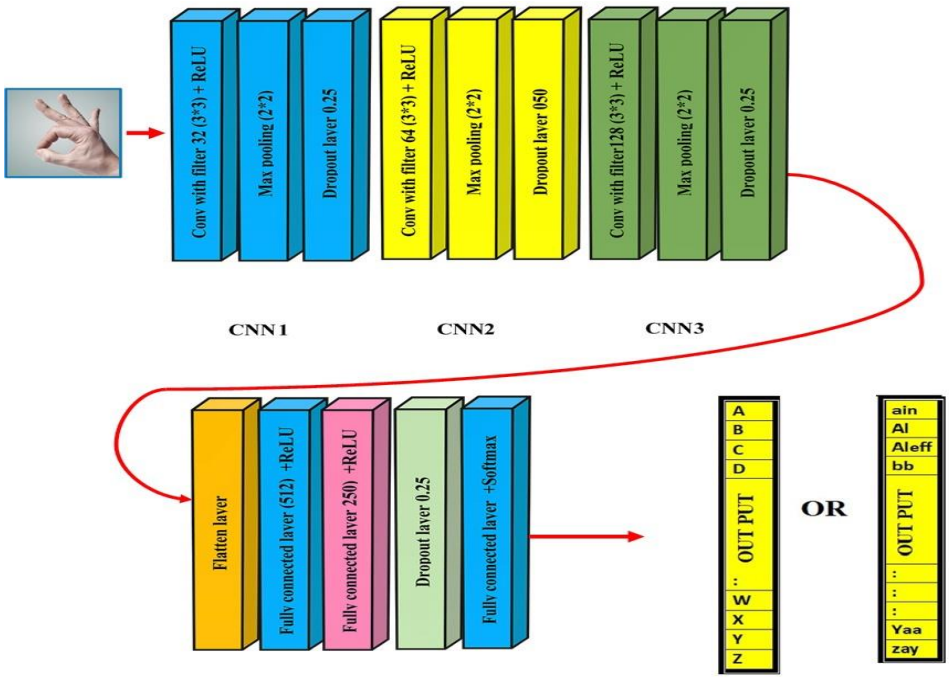

Figure (5): The CNN structure of the proposal system

1. The input images that used in the proposal system for hand gesture recognition whose size 64*64*3.

2. The convolution layer is sometimes called the feature extractor layer because features of the image are extracted within this layer. First of all, a part of the image is connected to the Convolution layer to perform convolution operation by slide the filter over the next receptive field of the same input image by a Stride and do the same operation again. We will repeat the same process again and again until we go through the whole image. The output will be the input for the next layer. Stride denotes how many steps we are moving in each step in convolution. By default, it is one. In the proposed system, three convolution layers are used. In the first convolution layer, 32 filters were used with dimension 3*3, the second filters were used with dimension 3*3, the third convolution layer, 128 filters were used with dimension 3*3, as shown in (Figure 5). Choosing the number of filters in each of the convolution layers, based on several experiments that prove the best result that obtained of these numbers that were used in each level.

3. Non-linear Layer (Activation Function): we are used in the proposed system after each level of the convolution layers, and with fully connected layers after flatten layer

4. Pooling Layer in the proposed system the Max-pooling layer is applied with size (2*2). Afterwards, a layer called Flatten converts the two-dimensional matrix data to a vector, thereby allowing the final output to be processed by standard fully connected layers to obtain the next layers.

6. Fully Connected Layer: Often the last layers of a CNN are fully connected layers, in a traditional neural network. The major difference is that the inputs would be in the form that CNN earlier stages would build. In the two neighboring layers the neurons were connected directly to the neurons within the fully connected network.

The first fully connected layer with the ReLU activation function contains 512neurons. The second fully connected layer containing 250 neurons with the ReLU activation function, this is followed by a dropout layer to exclude 25% of neurons to reduce overfitting.

7. Softmax Layer: In the proposed system, 32 classes for the ArSL dataset will be produced from this process because according to the data used for training and determining the number of classes that have been extracted from this stage, these classes will be in the form of value for each class linked to the fully connected layer image and Softmax will be made for them in the last step**.**

CNN Training

The system reads the dataset and the dataset augmentation will be generated. The system starts to train the network as much as the number of epochs inputted by the user earlier. The training will produce a probability value for the Thirty-two classification classes, where the class with the greatest probability value is the classification class predicted by the program. The training results are then stored in the form of a model file for use later. After completing the training, the system will save the model and plot from the results of the training.

CNN Testing

The testing dataset is used to provide an unbiased final design fit assessment based on the training set of data. At this point, the system employs the groups that were trained in the previous phase in CNN, and the features were extracted by learning the network.

EXPERIMENTAL RESULT AND DISCUSSION

The Arabic Sign Language Recognition system is evaluated using a dataset consisting of 25600 images for training, 6400 images for testing, and 6400 images for validation belonging to 32 classes.

The experiments are implemented under a specific system requirement such as Windows-10 operating system, Hardware processor: Core i7- CPU 8550U, 200 GHz, and (8GB) RAM. Python (3.7.10 64-bit) programming language with Tensor Flow backend, CNN programs implemented on Kaggle server

In order to test the developed system, many runs with different values for the CNN parameters are used to get the best results.

In CNN training there are parameters that are run constantly throughout the procedure, namely learning rate, batch size, and optimizer. The learning rate used is 0.0001, where this parameter states the constants for learning speed from the network layer used. While the batch size parameter serves to determine the total amount of data used in one batch of training, the size of the batch that was applied with size 256.

Determination of batch size is considered from the memory capability of the device used to conduct the training process, and optimizer is (Nadam).

The classification of hand gesture images has been completed. The training stage comes before the testing stage, which means that the network is trained when some image of a hand gesture is inserted. Since the network was previously learned and practiced, the types of hand gestures can be determined. So, the key distinction between training and testing is that test data is unlabeled, while training data is classified. This feature in Python is used in the proposed framework, scores = model. Evaluate (X _test, y _test) assigns each row of the dataset to one of the training classes. Both the sample and training arrays must have the same column size. Training the Group is a group element, and the individual values decide groups, and each factor specifies the group the related training row belongs to. We have arrived at the final stage, which is the classification and recognition of hand gesture images, Table (1) show the main result of CNN structure that has been used in the proposed system.

Table (1): The main structure of the proposed CNN.

Layer (type)	Output Shape	Param #
conv2d_18 (Conv2D)	(None, 62, 62, 32)	896
max_pooling2d_18 (MaxPooling	(None, 31, 31, 32)	0
dropout_24 (Dropout)	(None, 31, 31, 32)	0
conv2d_19 (Conv2D)	(None, 29, 29, 64)	18496
max_pooling2d_19 (MaxPooling	(None, 14, 14, 64)	0
dropout_25 (Dropout)	(None, 14, 14, 64)	0
conv2d_20 (Conv2D)	(None, 12, 12, 128)	73856
max_pooling2d_20 (MaxPooling	(None, 6, 6, 128)	0
dropout_26 (Dropout)	(None, 6, 6, 128)	0
flatten_6 (Flatten)	(None, 4608)	0
dense_18 (Dense)	(None, 512)	2359808
dense_19 (Dense)	(None, 250)	128250
dropout_27 (Dropout)	(None, 250)	0
dense_20 (Dense)	(None, 32)	8032

Total params: 2,589,338
Trainable params: 2,589,338
Non-trainable params: 0

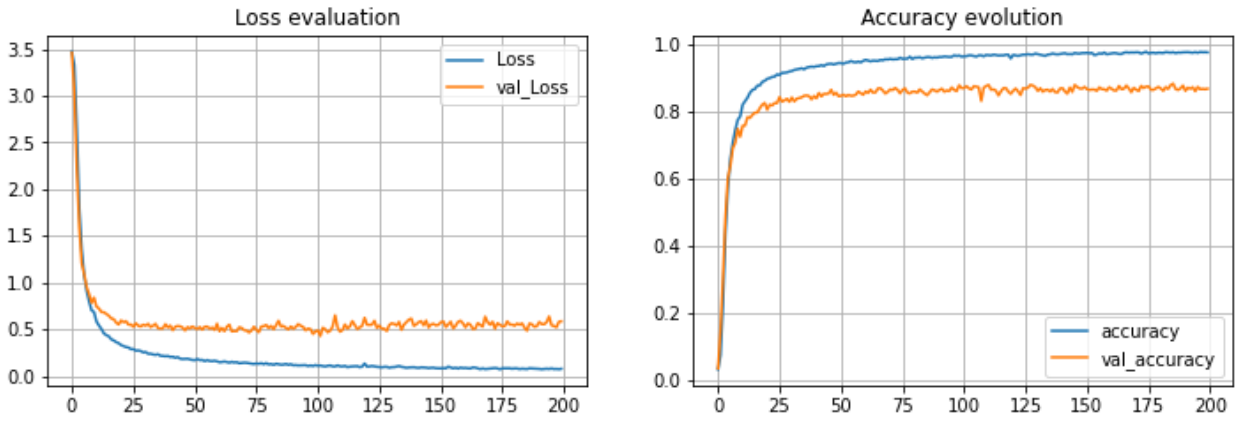

Figure 6: The Proposal System's Loss and Accuracy Evaluation

The confusion matrix (CM) depicts the system's success in terms of established correct and incorrect classifications. As a result, the test predictions in Table CM (2).

Table (2): The confusion matrix of the Proposal system.

		ain	al	aleff	bb	dal	dha	dhad	fa	gaaf	ghain	ha	haa	jeem	kaaf	khaa	la	laam	meem	nun	ra	saad	seen	sheen	ta	taa	thaa	thal	toot	waw	ya	yaa	zay
0	عين ain	200	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	ال al	0	200	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	الف aleff	0	0	180	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	باء bb	0	0	0	200	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	دال dal	0	0	0	1	174	2	0	0	0	0	0	0	6	0	3	0	1	0	0	3	0	0	0	1	0	0	9	0	0	0	0	0
5	ظاء dha	0	0	0	1	0	189	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	8	1	0	0	0	0	0	0	0
6	ضاد dhad	0	0	0	0	0	0	197	0	0	0	0	1	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0
7	فاء fa	0	0	0	0	0	0	1	170	13	0	4	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	1	0	0	0
8	قاف gaaf	0	0	0	0	0	0	0	3	187	0	8	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
9	غين ghain	2	0	0	0	0	0	0	0	0	197	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
10	حاء ha	0	0	0	0	0	0	0	1	4	0	193	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
11	هاء haa	1	0	0	0	0	0	0	0	0	0	0	192	1	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	جيم jeem	0	0	0	1	0	0	0	0	0	0	1	0	196	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
13	كاف kaaf	0	0	0	0	0	0	1	0	0	0	0	0	0	194	0	0	0	0	0	0	0	1	1	0	0	3	0	0	0	0	0	0
14	خاء khaa	0	0	0	0	0	1	0	1	0	0	0	4	0	0	192	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
15	لا la	0	0	1	0	3	1	0	6	0	0	0	0	0	0	0	187	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
16	لام laam	0	1	0	0	0	5	0	0	0	0	0	0	0	0	0	3	188	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0
17	ميم meem	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	195	0	0	0	0	0	4	0	0	0	0	0	0	0	0
18	نون nun	0	2	0	0	8	0	0	0	0	0	0	0	0	0	0	3	0	0	185	1	0	0	0	0	0	0	0	0	0	0	1	0
19	راء ra	0	0	0	0	4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	193	0	0	0	1	0	0	0	0	0	0	0	1
20	صاد saad	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	199	0	0	0	0	0	0	0	0	0	0	0
21	سين seen	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	196	1	0	0	2	0	0	0	0	0	0
22	شين sheen	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	199	0	0	0	0	0	0	0	0	0
23	تاء ta	0	0	0	0	0	5	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	193	0	0	0	0	0	0	0	0
24	تاء taa	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	193	2	0	2	0	0	0	0
25	ثاء thaa	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	195	0	0	0	0	0	0
26	ذال thal	0	0	0	0	7	0	0	3	0	0	0	6	3	0	1	0	0	0	0	0	0	0	0	0	0	0	177	0	0	0	0	3
27	تاء مربوطة toot	0	0	0	0	10	0	0	1	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	3	1	0	179	0	0	0	4
28	واو waw	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	198	0	0	0
29	همزة ya	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	197	0	0
30	ياء yaa	0	3	0	0	1	0	0	0	0	9	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	185	0
31	زاء zay	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	199

The comparison was made for the proposed system of hand gesture recognition (see Table (3)) with the related works.

Table (3): Comparison of Classification Accuracy with Earlier Studies.

Reference	Algorithm	Accuracy
M. Elbadawy et al. (2017)	CNN	85%
S. Hayani et.al (2019)	CNN	90.02%
A. Hasasneh (2020)	DBN	83.32%
M. M. Kamruzzaman (2020)	CNN	90%
Our proposed (2021)	CNN	97.42 %

CONCLUSION

In this Paper, an efficient system for Arabic Sign Language Recognition with a Convolutional Neural Network was developed. The system uses Static Hand Gesture Recognition for Arabic Sign Language Alphabets. The methodology of CNN that extracting features locally by (convolution and pooling layers) to classify Static Hand Gesture images achieved best performance compared with related work.

The major goal of the proposed system is to reduce the gap between deaf and dumb people and normal people who use Arabic sign language, Although the proposed system is in its early stages, it is still successful, with accuracy 97.42 %.

After recognizing Arabic letters based on the hand mark, the result will be entered into the text in the speech engine that generates the sound relative as a result of the Arabic language in future work, and Expanding this work to include building a real-time application that can recognize sign language and includes words and phrases to recognize it instead of just letters, and finally it can be expanded further to recognize dynamic gestures in real time videos.

REFERENCE

[1] Hind Ibrahim Mohammed, Jumana Waleed, Saad Albawi, "An Inclusive Survey of Machine Learning based Hand Gestures Recognition Systems in Recent Applications", IOP Conference Series: Materials Science and Engineering, Volume 1076, 2nd International Scientific Conference of Engineering Sciences (ISCES 2020) 16th-17th December 2020, Diyala, Iraq.

[2] S. C. Mesbahi, J. Riffi, and H. Tairi, "Hand gesture recognition based on convexity approach and background subtraction," pp. 0–4, 2018.

[3] S. Nagarajan and T. S. Subashini, "Static Hand Gesture Recognition for Sign Language Alphabets using Edge Oriented Histogram and Multi Class SVM," *International Journal of Computer Applications*, vol. 82, no. 4. pp. 28–35, 2013, doi: 10.5120/14106-2145.

[4] A. Abraham, P. Krömer, and V. Snášel, "SIFT-based Arabic Sign Language Recognition System," *Afro-European Conf. Ind. Adv. Proc. First Int. Afro-European Conf. Ind. Adv. AECIA 2014 Adv. Intell. Syst. Comput.*, vol. 334, no. November, 2014, doi: 10.1007/978-3-319-13572-4.

[5] "https://www.iwillteachyoualanguage.com/wp-content/uploads/2017/08/children-american-sign-language.jpg, Accessed on 10/1/2020." .

[6] "S.S. Kakkoth, and S. Gharge," Visual Descriptors Based Real Time Hand Gesture Recognition"2018 International Conference On Advances in Communication and Computing Technology (ICACCT). doi:10.1109/icacct.2018.8529663." .

[7] A. Ghotkar, "STUDY OF VISION BASED HAND GESTURE RECOGNITION USING INDIAN SIGN LANGUAGE," no. INTERNATIONAL JOURNAL ON SMART SENSING AND INTELLIGENT SYSTEMS VOL. 7, NO. 1, MARCH 2014. .

[8] M. Oudah, A. Al-naji, and J. Chahl, "Hand Gesture Recognition Based on Computer Vision : A Review of Techniques," 2020.

[9] M. Jin, C. Zaid, O. Mohamed, and H. Jaward, "A review of hand gesture and sign language recognition techniques," *Int. J. Mach. Learn. Cybern.*, vol. 0, no. 0, p. 0, 2017, doi: 10.1007/s13042-017-0705-5.

[10] A. J. Abdullah, T. M. Hasan and J. Waleed, "An Expanded Vision of Breast Cancer Diagnosis Approaches Based on Machine Learning Techniques," 2019 International Engineering Conference (IEC), 2019, pp. 177-181, doi: 10.1109/IEC47844.2019.8950530.

[11] Taha Mohammed Hasan, Sahab Dheyaa Mohammed, Jumana Waleed, "Development of breast cancer diagnosis system based on fuzzy logic and probabilistic neural network", Eastern-European Journal of Enterprise Technologies, Vol. 4, No. 9 (106), 2020, pp. 6-13, DOI: https://doi.org/10.15587/1729-4061.2020.202820.

[12] A. A. Hayawi and J. Waleed, "Driver's Drowsiness Monitoring and Alarming Auto-System Based on EOG Signals," 2019 2nd International Conference on Engineering Technology and its Applications (IICETA), 2019, pp. 214-218, doi: 10.1109/IICETA47481.2019.9013000.

[13] Jumana Waleed, Thekra Abbas, Taha Mohammed Hasan, "Implementation of driver's drowsiness assistance model based on eye movements detection", Eastern-European Journal of Enterprise Technologies, Vol. 5, No. 9 (107), 2020, pp. 6-13. DOI: https://doi.org/10.15587/1729-4061.2020.211755.

[14] M. H. Abdul-Hadi and J. Waleed, "Human Speech and Facial Emotion Recognition Technique Using SVM," 2020 International Conference on Computer Science and Software Engineering (CSASE), 2020, pp. 191-196, doi: 10.1109/CSASE48920.2020.9142065

[15] M. Elbadawy, A. S. Elons, H. A. Shedeed, and M. F. Tolba, "Arabic sign language recognition with 3D convolutional neural networks," *2017 IEEE 8th Int. Conf. Intell. Comput. Inf. Syst. ICICIS 2017*, vol. 2018-Janua, no. Icicis, pp. 66–71, 2018, doi: 10.1109/INTELCIS.2017.8260028.

[16] S. Hayani, M. Benaddy, O. El Meslouhi, and M. Kardouchi, "Arab Sign language Recognition with Convolutional Neural Networks," *Proceedings of 2019 International Conference of Computer Science and Renewable Energies, ICCSRE 2019*. 2019, doi: 10.1109/ICCSRE.2019.8807586.

[17] A. Hasasneh, "ARABIC SIGN LANGUAGE CHARACTERS RECOGNITION BASED ON A DEEP LEARNING APPROACH AND A SIMPLE LINEAR CLASSIFIER," *Jordanian J. Comput. Inf. Technol. (JJCIT), Vol. 06, No. 03, Sept. 2020.*

[18] M. M. Kamruzzaman, "Arabic Sign Language Recognition and Generating Arabic Speech Using Convolutional Neural Network," *Wirel. Commun. Mob. Comput.*, vol. 2020, 2020, doi: 10.1155/2020/3685614.

[19] G. Latif, N. Mohammad, J. Alghazo, R. AlKhalaf, and R. AlKhalaf, "ArASL: Arabic Alphabets Sign Language Dataset," *Data Br.*, vol. 23, p. 103777, 2019, doi: 10.1016/j.dib.2019.103777.

[20] "https://data.mendeley.com/datasets/y7pckrw6z2/1." .