Interpreting Arabic Sign Alphabet by Using the Deep Learning

Ahmed Saeed Ibrahim Al-Obaidi[1], **Raid Rafi Omar Al-Nima**[1], **and Tingting Han**[2]
[1] Department of Medical Instrumentation Technology Engineering, Technical Engineering College of Mosul, Northern Technical University, Iraq
[2] Department of Computer Science and Information Systems, Birkbeck, University of London, United Kingdom

ABSTRACT

Sign Language (SL) is a communication method between people. It is an essential language; especially for people who are speech impaired and hearing impaired, it can be considered as their mother tongues. Hand gestures form the nonverbal communication of this language. We focus on interpreting *Arabic* Sign Alphabet (ASA) in this study and, as a case study, the recognition of alphabet in *Iraqi* Sign Language (IrSL) is carried out with the help of specialists from the "Al-Amal Institute for the Deaf and Dumb". A new ASA dataset of various hand gestures was created and adopted. In addition, a deep learning model named the Deep Arabic Sign Alphabet (DASA) is proposed, which is a developed version of the Convolutional Neural Network (CNN). It can efficiently interpret the ASA, achieving a high interpretation accuracy of 95.25%.
Keywords: *Arabic Sign Alphabet, Deep Learning, Convolutional Neural Network*

1. INTRODUCTION

Sign Language (SL) is utilized to help the people with speech and hearing impairments as a tool of communication. It consists of a set of gestures and body languages to denote different meanings [1] . There are usually two types of SL gestures: static and dynamic for the hands, body and face [2]. Studying the SL is so valuable as the number of people with speech and hearing impairments has been increased according to the World Health Organization (WHO) and reached 432 million until the year of 2021, among which around 34 million cases are children and the remaining are adults [3]. The WHO has warned the increasing risk of hearing loss due to genetic reasons, complications of childbirths, infectious diseases and chronic ear infections [3][4]. By 2050, it is estimated that about 2.5 billion (or 1 out of 4) people will have some degrees of hearing loss and at least 700 million people will need hearing rehabilitation. Hearing loss becomes more common when people get older — about 25% of people over the age of 60 have such issue; and nearly 80% of people who suffer from hearing loss live in low- and middle-income nations [3].

The SL has been improved over different countries but in Iraq it is almost non-existed. A unified dictionary for the Iraqi Sign Language (IrSL) is under development according to the new curriculum, and this would be beneficial for the Special Needs Welfare Department (SNWD) of the Ministry of Labor and Social Affairs (MLSA) in Iraq. Almost every country has its own SL such as the American Sign Language (ASL) [5], British Sign Language (BSL) [6] and Arabic Sign Language (ArSL) [7]. Unfortunately, there is no unified global SL for all countries over the world.

Many academic subjects including image recognitions, analyses and classifications have recently benefited from the Deep Learning (DL) techniques as in [8][9][10][11][12][13][14][15]

[16][17][18]. There are two types of methods that can be applied to the SL recognition: the first type is by using sensors and the second type is based on images [19]. For the first type, it requires people to wear hand gloves with sensors to recognize hands' gestures. The problem with such methods is the inconvenient use of gloves attaching to wires and sensors [20][21]. The second type does not require wearing sensors, but rather focuses on image processing. These methods are developed using various machine learning methods such as Artificial Neural Networks (ANNs) [22], Support Vector Machines (SVMs) [23] and Elastic the Graph Matching (EGM) [24]. The ArSL recognition studies are mainly of the second type [7][25].

This paper is aiming to provide a study to automatically interpret the Arabic Sign Alphabet (ASA) to the alphabet in natural language. As a case study, our work identified the alphabet SL from Iraq and translated the letters to natural language. This work was done under the help of *Al-Amal Institute for the deaf and dumb*. The main contributions of this paper are as follows:

- We collected a set of images and created a new dataset (*Arabic Sign Alphabet*, ASA) for hand gestures. To the best of our knowledge, this is the first dataset of this sort in Iraq.
- We invented and developed a deep learning model called the *Deep Arabic Sign Alphabet* (DASA) which can interpret the alphabet in Arabic sign language to that in natural language.

The remaining sections in this paper are organized as follows: Section 2 reviews the related literature review, Section 3 presents the methodology of the proposed DASA approach, Section 4 illustrates ASA dataset and discusses experimental results, and Section 5 provides the conclusion of the study.

2. LITERATURE REVIEW

2.1 Previous work on non-Arabic sign languages

In 2010, Zafrulla *et al.* suggested an educational interactive adventure game to learn about the ASL called the CopyCat project, which is a collection of educational adventure game for children with hearing impairment. Special gloves with accelerometers were worn to help with the segmentations of hands. Eleven children with hearing impairment participated in the database acquisition during the game by wearing two different colored gloves such as red and purple. Each child must use the ASL to play, then the video images were taken as data. The game supported 19 vocabularies from the ASL. Hidden Markov Model (HMM) was exploited for categorizations the data [26].

In 2012, Lang *et al.* offered an open-source framework for gesture recognitions of the German sign language (GSL) in order to be presented to the public. The framework was called Dragon and it was tested by separate signs of the GSL. A three-dimensional (3D) camera was used to capture body parts such as head and hands to distinguish their gestures. HMM model was used for the recognition [27].

In 2013, Gunasekaran and Manikandan focused on the technology that could recognize and understand the SL. The work was done in India. The proposed technology produced voices (sounds) from hand gestures by using sensors which were located on the palm of a hand, a sound storage unit, a processing unit and a wireless communication unit. The language was translated by

utilizing flow sensors of type APR9600 with a microcontroller of type PIC16F877A. By this technique, different languages could be provided without having to change the microcontroller codes. The data was collected directly from the sensors placed on the glove, which produced different values according to changeable resistances. The microcontroller converted the signals from the elastic sensors into digital signals and provided sound output [28].

In 2015, Simonyan and Zisserman focused on the effect of a deep CNN in the case of precise definition and classification for a larger scale of images — the ImageNet Large Scale Visual Recognition Challenge (ILSVRC-2012) dataset. The work shows the benefit of using small filters in evaluating networks of increasing depth, where an improvement in evaluating ultra-deep CNN of 19 weight layers is obtained [29].

In 2018, Ahmed *et al.* surveyed the studies conducted in the field of SL translations from the years between 2007 and 2017. This study collected SL data from people wearing different types of sensors on the gloves such as flex and accelerometer. The advantage of using the sensors (gloves) was that it could directly acquire information of different cases such as multiple degrees of deflection, direction and hand movement. Furthermore, this method was not affected by external factors such as locations, background conditions and lightings. The disadvantage of using gloves was that it was difficult to understand some movements of the hand and fingers as there were interferences between them [30].

In 2019, Luqman and Mahmoud built a grammar-based machine translation system between Arabic scripts and ArSL. The proposed method analyzed the input of Arabic sentences by applying morphological, grammatical and semantic terms to output ArSL presented as sequences of Graphics Interchange Format (GIF) images. The sentences are separated into words to be worked on separately. The problem was that the Arabic language and ArSL were not compatible, because their structure and grammar are different [31]. Thus, translating into the ArSL required the help from two language experts [32].

In 2020, Hossain *et al.* created an application called the Kotha Bondhu, which could translate the Bengali Sign Language (BaSL) into Bengali audio. The suggested application contains videos, audios and gestures. This work was carried out in Bangladesh, not only for speech and hearing impaired people, but also for people who were not so familiar with the BaSL or SL speakers. Separate interviews were conducted between the interpreters and a group called the Tablighi Jama'at, who used their native language (Bengali language). The application interpreted the BaSL into Bengali audio [33].

In 2021, Martin and Espejo proposed a new alphabet decoding technique for the Spanish Sign Language (SSL). It included images showing upper limbs. SSL dataset was created by using a camera attached to the head of a robot. Since the focus should be on the hands and arms, an open-source library was utilized to identify the anatomical key points of the individual images. The strategy here required thorough examinations of both spatial and temporal aspects. Therefore, two types of architectures were investigated: the Convolutional Neural Network (CNN) for the spatial dimension and Recurrent Neural Network (RNN) to alter the temporal sequences. . It was found

through experiments that the CNN achieved higher performance than the RNN, which signifies the importance of the spatial dimension over the temporal dimension of signal interpretation [34].

2.2 Previous work on Arabic sign languages

There is a limited number of studies that consider the Arabic SL compared to the American SL and British SL.

In 2005, Assaleh and Al-Rousan introduced the use of polynomial classifiers to recognize the ArSL alphabet. The authors developed a system that involved three main stages: collecting images, applying image processing and extracting the features. An ArSL database for 30 people with hearing impairments was collected in Jordan. The participant wears colored gloves of six colors to perform special ArSL signs [19].

In 2020, Elsayed and Fathy built a system that translated the ArSL into Arabic scripts using the power of DL techniques and semantic web facilities. Ontology had helped solve some of the challenges in sign language translation. Recent data for ten Arabic words were collected using a mobile phone camera, in addition to the previously available ArSL2018 data. Semantic Deep Learning (SDL) was used, which was a novel variant of a neural network model. The proposed model is used to train and test previously existing Arabic alphabet data and newly collected data. To further test the application of the proposed method, it was used to translate Arabic languages (not only alphbet) into Arabic texts [4].

It can be observed from the literature review that so far there has no work concentrated on establishing an SL dataset where Iraq is considered as a case study. In addition, only a few studies applied DL techniques on the ArSL translation. This study will further address this problem by proposing a new deep learning network.

3. THE PROPOSED DL MODEL

First of all, a new Arabic Sign Alphabet (ASA) dataset is established and employed taking into account the IrSL as the case study. Fundamentally, ASA hand gestures represent Arabic letter shapes, where a human's hand forms a letter shape in the SL. There are 29 Arabic alphabet letters (from Alif to Yā, in addition to Tā marbūtah), each letter has its own unique sign. Images of the ASA, each with its represented Arabic alphabet letter, can be seen in Figure 1.

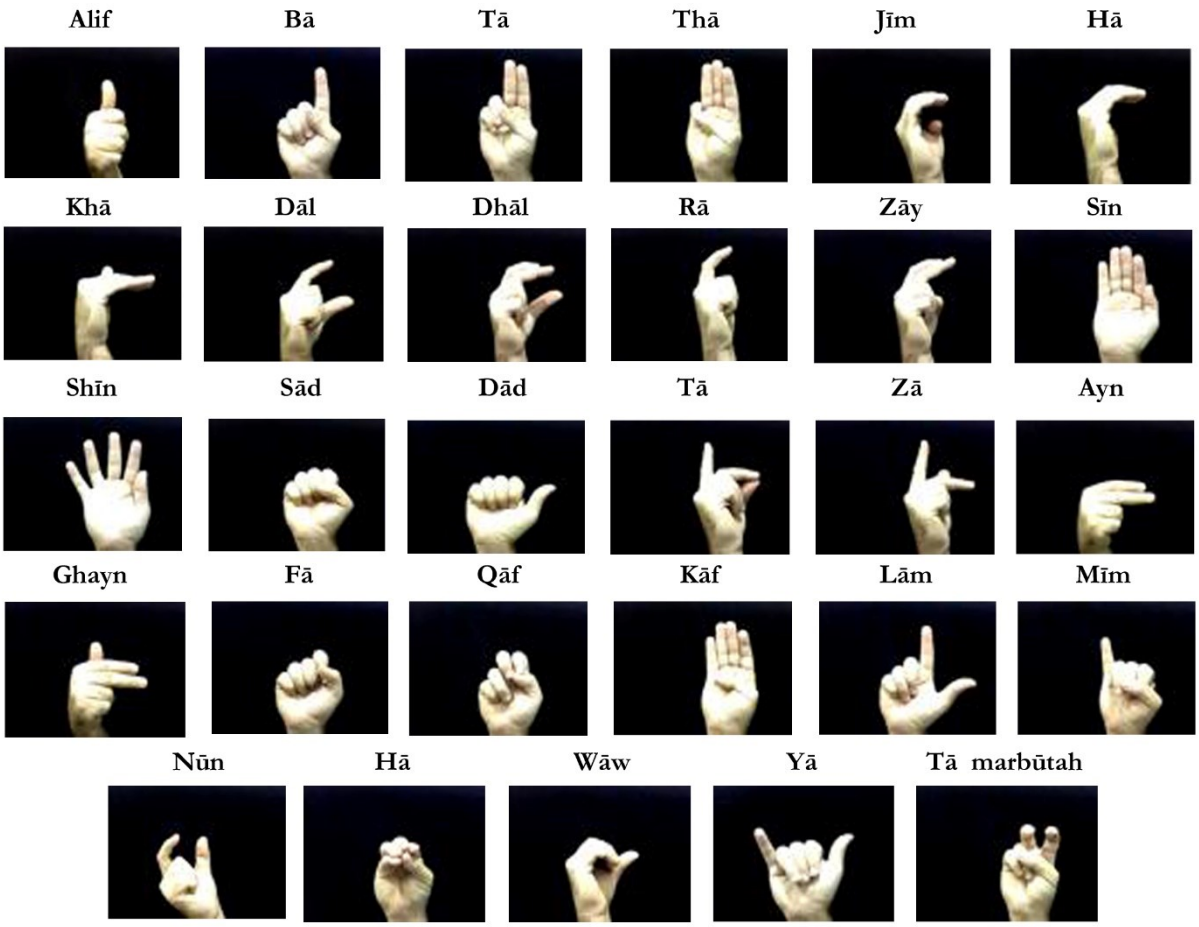

Figure 1: Images from the ASA dataset, each with its represented Arabic alphabet letter [35]

Now that we have the dataset, we propose a novel deep learning model *Deep Arabic Sign Alphabet* (DASA). It is based on the Convolutional Neural Network (CNN), however, it has been adapted to interpret the ASA of IrSL. It is focused on recognizing 29 Arabic letters as each letter has its own unique sign.

Given the input images, DASA will process in two phases. The first phase is feature extraction (FE), which consists of three layers: Convolutional, ReLU and Pooling layers. The second phase is classification, which also has three layers: Fully Connected (FC), Softmax and Classification Layers. These layers appear sufficient for the subject of this paper as there are no detailed inputs to be further analyzed for reaching their desired outputs. This suggested DL model has been adapted to input images of the ASA from Iraq and output categories represent Arabic alphabet letters. The DASA architecture is illustrated in Figure 2.

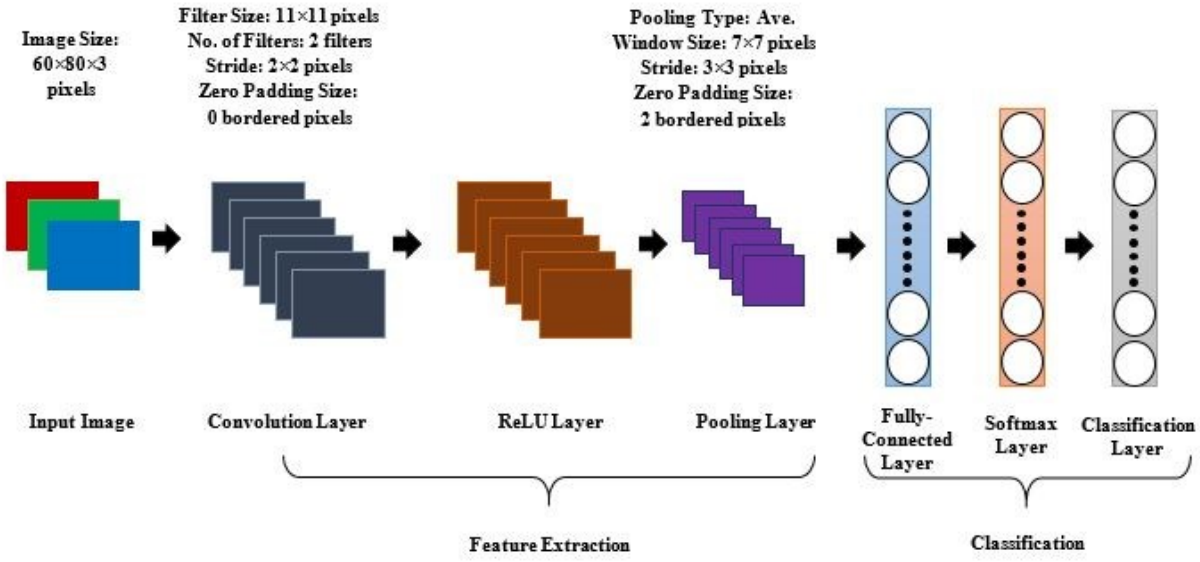

Figure 2: The architecture of the proposed DASA model

The essential layers of the CNN are detailed in [8][11][12][16][36]. The DASA layers can be described as follows:

A. *Input layer:*

This layer deals with the ASA input images. Each image is of format Joint Photographic Group (JPG) and of type Red, Green and Blue (RGB) with the dimensions (height × width × 3) pixels. In other words, the inputs are colored images, each one having three channels of the RGB [37].

B. *Convolution layer:*

This layer represents the first physical process in the DASA network. Two-dimensional (2D) convolution is performed here for each input channel. This layer has a group of image channels known as feature maps (or filters). By training the convolutional weights (or kernels), different feature maps are obtained [8]. Generally, the kernel size (or filter size) is $k_h \times k_w \times C$ pixels, where k_h refers to the kernel's height, k_w refers to the kernel's width and C refers to the channel's number. $W_{i,j,c^{l-1}}^{c^l}$ refers to the components of the kernel weights, B_{c^l} refers to the bias of the convolution layer, $l-1$ refers to the prior layer and l refers to the current layer. D_{z,t,c^l} is the value of a spatial pixel at (z, t) in channel c^l of the layer l. The following equation can be used to calculate the values of this layer:

$$D_{z,t,c^l} = B_{c^l} + \sum_{i=-k_h^l}^{k_h^l} \sum_{j=-k_w^l}^{k_w^l} \sum_{c^{l-1}=1}^{C^{l-1}} W_{i+k_h^l, j+k_w^l, c^{l-1}}^{c^l} D_{z+i,t+j,c^{l-1}} \quad (1)$$

where D_{z,t,c^l} refers to the output of convolution layer node [38].

C. *ReLU layer*:

A ReLU transfer function is used in this layer. It preserves the positive values and removes the negative values of the previous feature maps. Thus, it provides a non-linear computation. Moreover, the ReLU transfer function can be represented by the following equation:

$$G_{z,t,c^l} = f(D_{z,t,c^l}) = \max(0, D_{z,t,c^l}) \quad (2)$$

where G_{z,t,c^l} refers to the ReLU layer's output and max refers to the maximum operation [39].

D. *Pooling layer:*

This layer has important role of reducing the size of received channels. It usually considers maximum or average windowed values of previous channels. In the average pooling, all features are taken into accounts. Therefore, it considers more information than the maximum. Moreover, the average pooling computation is carried out according to the following equation:

$$P_{ave}(G) = \frac{1}{W \times W} \sum_{i=1}^{w} G_{z,t,c^l} \quad (3)$$

where P_{ave} refers to the output of the average pooling layer and w refers to the height/width of a pooling region [40].

E. *FC layer:*

Each node of a previous layer is connected to all nodes in this layer. It can adjust the number of nodes in the output layer according to the number of nodes in the previous layer. Its outcome can be obtained by utilizing the following equation:

$$FD_r = \sum_{a=1}^{mc_1^{l-1}} \sum_{b=1}^{mc_2^{l-1}} \sum_{c=1}^{mc_3^{l-1}} W_{a,b,c,r}^{l} \ (\mathbf{P}o)_{a,b} \qquad \forall 1 \le r \le mc^l \quad (4)$$

where FD_r refers to the output of the FC layer, mc_1^{l-1} refers to the height and mc_2^{l-1} refers to the width of a pooling layer channel, respectively, mc_3^{l-1} refers to the number of created channels in the pooling layer, $W_{a,b,c,r}^{l}$ refers to the linking weights between the pooling

layer and the fully connected layer, $\mathbf{P}o$ is the vector of pooling layer's outputs, and mc^l refers to the number of required neurons in the FC layer (which can be the same as the number of required classes or outputs) [41].

F. *Softmax layer:*

Usually, the softmax layer is put before the last layer. It is utilized to provide the corresponding relationships between a given input to all output classes. Mathematically, it can be calculated by the following equation:

$$YO_r = \frac{\exp(FD_r)}{\sum_{s=1}^{mc^{l-1}} \exp(FD_s)}, \qquad r = 1, 2, \dots, mc^l \tag{5}$$

where YO_r refers to an outcome of the softmax layer. The softmax normalizes its outputs, where each one of them will have a value between 0 and 1.

G. *Classification layer:*

The classification layer is used at the end to achieve the recognition or classification decision. The winner-takes-all rule is employed in this layer. This rule can be expressed by the following equation:

$$CD_r = \begin{cases} 1 & \text{if } YO_r = \text{Max} \\ 0 & \text{otherwise} \end{cases}, \quad r = 1,2, \dots mc^l \tag{6}$$

where CD_r refers to layer's output and Max is the maximum softmax output value [42].

4. RESULTS AND DISCUSSIONS

4.1 Established Dataset

Initially, the experiments were conducted using an established dataset called the ASA. There were 34 images are used for each Arabic alphabet letter and the total number of images from the ASA dataset used in our experiments was 928.

Our own dataset has been collected for this study. A large number of hand gesture images were captured between the 6th of October 2021 and 9th of November 2021. For each letter, 42 samples are acquired. As there are 29 letters, a total of 1218 images were acquired, for right-hand gestures with a black background. All the images were taken under the same lighting environment. A webcam of type Kisonli was used and the images were of standard dimensions of 240 × 320 × 3 pixels and JPG format. Four movements were taken into account during the acquisition. These were the vertical rotations of angles between 10 to 30 degrees, horizontal rotations of angles between 35 to 45 degrees, scaling for up to 10 centimeters and translations for up to 4 centimeters.

4.2 Standardizations

All experiments were implemented on a laptop computer that had the following specifications: Dell, Intel Core i7 processor with 2.20 GHz speed, 8 GB computer memory, NVIDIA external graphics card, GF117 graphics processor and 2 GB display memory. The total number of images that is actually considered from the ASA dataset is 928 images, where 34 images are used for each Arabic alphabet letter. Moreover, all experiments used the following training parameters: optimizer type of Stochastic Gradient Descent with Momentum (SGDM), momentum value of 0.9, weight decay value of 0.0001, with fixed learning rate 0.0001, mini-batch size of 128 and maximum epochs of 300. The ASA dataset was randomly partitioned into three groups: 50% for the training phase (following [43][44]), 25% for the testing phase and 25% for the validation phase. The collected input images have empirically been resized to the dimensions of 60 × 80 × 3 pixels, which will be detailed later. The number of output classes was fixed to the 29 classes, each one of them corresponding to an ASA letter.

4.3 DASA Parameters

Extensive experiments are performed to investigate the suitable parameters of the proposed DASA network. Table 1 shows the performances of those experiments for the established ASA dataset.

Table 1: The performances of extensive experiments that are implemented to obtain the appropriate parameters of the proposed DASA network for the established ASA dataset

No. of Stages	Convolution				Pooling				Accuracy (%)
	Filter Size (pixels)	No. of Filters (filters)	Stride Size (pixels)	Padding Size (bordered pixels)	Type	Window Size (pixels)	Stride Size (pixels)	Padding Size (bordered pixels)	
Stage 1	3×3	2	1×1	0	Max.	3×3	3×3	0	85.78
	5×5	2	1×1	0	Max.	3×3	3×3	0	87.93
	7×7	2	1×1	0	Max.	3×3	3×3	0	86.64
	9×9	2	1×1	0	Max.	3×3	3×3	0	87.06
	11×11	2	1×1	0	Max.	3×3	3×3	0	90.09
	13×13	2	1×1	0	Max.	3×3	3×3	0	87.93
Stage 2	11×11	2	1×1	0	Max.	3×3	3×3	0	90.09
	11×11	4	1×1	0	Max.	3×3	3×3	0	88.36
	11×11	6	1×1	0	Max.	3×3	3×3	0	89.66
	11×11	8	1×1	0	Max.	3×3	3×3	0	85.34
	11×11	10	1×1	0	Max.	3×3	3×3	0	87.50
	11×11	12	1×1	0	Max.	3×3	3×3	0	86.21
	11×11	14	1×1	0	Max.	3×3	3×3	0	84.91
Stage 3	11×11	2	5×5	0	Max.	3×3	3×3	0	77.59
	11×11	2	4×4	0	Max.	3×3	3×3	0	84.91
	11×11	2	3×3	0	Max.	3×3	3×3	0	86.64

	11×11	2	2×2	0	Max.	3×3	3×3	0	90.95
	11×11	2	1×1	0	Max.	3×3	3×3	0	90.09
Stage 4	11×11	2	2×2	0	Max.	3×3	3×3	0	90.95
	11×11	2	2×2	1	Max.	3×3	3×3	0	87.93
	11×11	2	2×2	2	Max.	3×3	3×3	0	87.07
	11×11	2	2×2	3	Max.	3×3	3×3	0	89.22
	11×11	2	2×2	4	Max.	3×3	3×3	0	90.09
	11×11	2	2×2	same	Max.	3×3	3×3	0	88.79
Stage 5	11×11	2	2×2	0	Max.	3×3	3×3	0	90.95
	11×11	2	2×2	0	Ave.	3×3	3×3	0	93.97
Stage 6	11×11	2	2×2	0	Ave.	3×3	3×3	0	93.97
	11×11	2	2×2	0	Ave.	5×5	3×3	0	93.10
	11×11	2	2×2	0	Ave.	7×7	3×3	0	94.40
	11×11	2	2×2	0	Ave.	9×9	3×3	0	90.52
	11×11	2	2×2	0	Ave.	11×11	3×3	0	74.14
	11×11	2	2×2	0	Ave.	13×13	3×3	0	59.05
Stage 7	11×11	2	2×2	0	Ave.	7×7	7×7	0	88.36
	11×11	2	2×2	0	Ave.	7×7	6×6	0	90.52
	11×11	2	2×2	0	Ave.	7×7	5×5	0	90.95
	11×11	2	2×2	0	Ave.	7×7	4×4	0	92.67
	11×11	2	2×2	0	Ave.	7×7	3×3	0	94.40
	11×11	2	2×2	0	Ave.	7×7	2×2	0	93.10
	11×11	2	2×2	0	Ave.	7×7	1×1	0	93.10
Stage 8	11×11	2	2×2	0	Ave.	7×7	3×3	0	94.40
	11×11	2	2×2	0	Ave.	7×7	3×3	1	93.10
	11×11	2	2×2	0	Ave.	7×7	3×3	2	95.26
	11×11	2	2×2	0	Ave.	7×7	3×3	3	94.40
	11×11	2	2×2	0	Ave.	7×7	3×3	4	93.10
	11×11	2	2×2	0	Ave.	7×7	3×3	Same	94.40

The parameters that require evaluating/tuning are distributed into two essential layers: the convolution and pooling layers. Convolution layer involves the parameters: filter size, number of filters, stride size and zero padding size. Pooling layer has the parameters: type, window size, stride size and zero padding size. The experiments are divided into eight stages, and in each stage only one parameter is sufficiently changed and the values of other parameters are preserved. Best accuracies (best in the stage and above 90%) are always recorded and observed in order to be used for the next stage, until stage 8 where the best percentage can be benchmarked.

For the convolution layer, the filter size is gradually changed between 3×3 pixels to 13×13 pixels. The highest accuracy of 90.09% is recorded for 11×11 pixels. The number of filters is tuned from 2 to 14 filters and the best number of filters is found to be 2 filters where it can preserve the same highest result as mentioned before. The stride size is tuned from 1×1 pixels to 5×5 pixels, the highest performance of 90.95% is reported for 2×2 pixels. Zero padding size is adjusted from 0 to 4 bordered pixels and same original channel size. It is observed that the exact previous best percentage value of 90.95% is benchmarked for 0 bordered pixels.

For pooling layer, two types of pooling are firstly investigated. These are the Maximum (Max.) and Average (Ave.). Best result of 93.97% is recorded for the Ave. Window size is tuned from 3×3 pixels to 13×13 pixels, the best performance of 94.39% appears at 7×7 pixels. Stride size of pooling layer is altered between 1×1 pixels to 7×7 pixels, the best percentage value of 94.40% is benchmarked at 3×3 pixels. Zero padding sizes are changed from 0 to 4 bordered pixels, in addition to the 'same' pixels. It has been reported the highest result of 95.26% at 2 pixels.

Among all the experiments, the highest accuracy is recorded for the following parameters: convolution filter size equal to 11×11 pixels, number of convolution filters equal to 2, convolution stride size equal to 2×2 pixels, convolution padding size equal to 0 bordered pixels, pooling type Ave., pooling window size equal to 3×3 pixels, pooling stride size equal to 3×3 pixels and pooling size equal to 2 bordered pixels (each pixel includes a zero value). These values are benchmarked for the proposed DASA model.

Additional experiments are carried out to investigate the performances of changing the initial learning rate value. Table 2 displays the accuracies and Equal Error Rates (EERs) of such experiments.

Table 2: Additional experiments for investigating the performances of changing the initial learning rate value

Initial Learning Rate Value	Accuracy (%)	EER (%)
0.0001	95.25	4.75
0.0002	95.69	4.31
0.0003	96.12	3.88
0.0004	96.12	3.88
0.0005	96.12	3.88

According to this table, as the initial learning rate value varies from 0.0001 to 0.0005, the highest accuracy of 96.12% and lowest error of 3.87% appear at the value of 0.0003. The same highest accuracy remains constant even after increasing the initial learning rate value, therefore, it has been adopted here.

4.4 Resizing Input Images

Further experiments are performed to choose the best input image size for the DASA. Table 3 shows relationships between the different input image sizes and their accuracies.

Table 3: Relationships between the different input image sizes and their accuracies

Input Image Size (pixels)	Accuracy (%)
240×320×3	3.88
180×240×3	88.79
120×160×3	89.66
60×80×3	96.12

30×40×3	83.19

We have investigated five image sizes (original size, reduced original size to three quarters, a half, a quarter and an eighth). For the original input image size of 240×320×3 pixels, a very low percentage value of 3.88% is recorded. Then, the accuracy is significantly increased after changing the original size to three quarters as a value of 88.79% is obtained. Consequently, reducing the original size to half achieves slightly higher accuracy of 89.66%. The highest accuracy 96.12% is obtained when the original size is changed to a quarter to 60×80×3. A further reduction to one eighth of the original size brings down the accuracy significantly to 83.19%.

This indicates that it is necessary to adjust the input image size. This is because interpreting the ASA information does not need complicated analysis as the SL concentrates on global movements instead of small details. The low original size of a half quarter losses useful input information, so, its percentage value is decreased.

The proposed DASA model is adapted for the input size of 60×80×3 pixels as this size achieves the highest compared result.

4.5 Training

As mentioned, 50% of the total images in the ASA dataset are randomly sampled for the training phase and 25% of the total images in the same dataset are randomly sampled for the validation phase. Training and validation performances of the proposed DASA network are demonstrated in Figure 3.

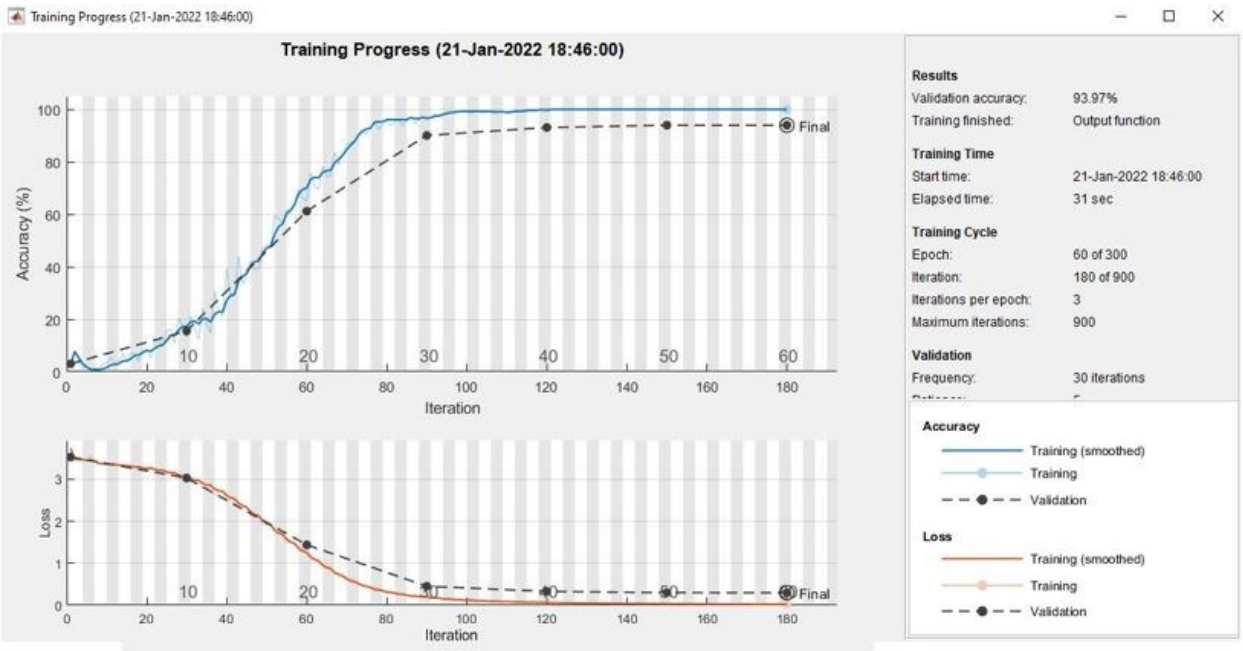

Figure 3: Training process (accuracy and loss) for the Dataset of hand gesture.

This figure mainly displays the validation accuracy, number of iterations used per epochs and training loss. It has two curves, first curve shows the relationships between the percentage accuracy and iteration, whilst the second curve provides the relationships between the loss/error and iteration. The loss/error could significantly be reduced, and the accuracy could dramatically be increased to the highest value. So it is appropriate to conclude that training the proposed DASA is successfully implemented.

4.6 Testing and Comparisons

As mentioned, the remaining 25% of the total images in the same dataset are used in the testing phase. The proposed DASA model is evaluated in the testing phase. Also, it is compared to various state-of-the-art DL network architectures, where they are simulated and tested for the images from the ASA dataset. Table 3 shows the comparison between the testing accuracies of various DL network architectures.

Table 3: Comparison between the testing accuracies of various DL network architectures

References	DL Network	Accuracy (%)
Ibrahim *et al.* [11]	DFCN	87.5
AL-Hatab *et al.* [16][17]	XCM	89.22
	YCM	75.86
	ZCM	89.22
Albak *et al.* [12]	PCNN	88.36
Our approach	DASA	96.12

The state-of-the-art network architectures that have been considered for the comparison are the Deep Fingerprint Classification Network (DFCN) [11], X Axis Classification Model (XCM) [16][17], Y Axis Classification Model (YCM) [16][17], Z Axis Classification Model (ZCM)[16][17] and Palm Convolutional Neural Network (PCNN) [12]. From Table 3, it is noted that all the compared networks have not obtained satisfactory accuracies. This is because their architectures could not be well adapted to the ASA, where each one of them has different parameter values that are not so fit to the goal of this work. On the other hand, our proposed DASA approach achieves the highest result of 96.12%. This is due to its architecture which is carefully designed and adapted for the ASA.

5. CONCLUSIONS

This work presented two main contributions. Firstly, we collected the ASA dataset as a part of the IrSL from scratch. Secondly, a DL model termed the DASA was carefully designed and adopted to interpret the ASA. As a case study, the alphabet SL from Iraq was focused as this work may help the deaf and dumb people in this country.

A big number of hand gesture images have been acquired for the ASA dataset, where total of 1218 images have been collected. Four movements were considered during the acquisition. These are the vertical rotations, horizontal rotations, scaling and translations. SL of 29 ASA letters

(from Alif to Yā, in addition to Tā marbūtah) were taken into accounts, 42 samples are captured for each letter.

The proposed DASA model was evaluated for the established ASA dataset. Extensive experiments were provided for reaching best DASA parameters. Additional experiments were performed for investigating the appropriate initial learning rate value. Further experiments were implemented for finding the suitable adapted input size. As a result, the DASA network attained a high accuracy of 96.12% and a low EER of 3.88%. The DASA architecture performance could surpass other state-of-the-art DL network architectures, where related comparison was established.

REFERENCES

[1] S. Yang and Q. Zhu, "Video-based Chinese sign language recognition using convolutional neural network," *2017 9th IEEE International Conference on Communication Software and Networks, ICCSN 2017*. pp. 929–934, 2017, doi: 10.1109/ICCSN.2017.8230247.

[2] S. Saqib and S. Asad, "Repository of Static and Dynamic Signs," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 11, pp. 101–105, 2017, doi: 10.14569/ijacsa.2017.081113.

[3] W. H.Organization, "Deafness and hearing loss." https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss (accessed Feb. 22, 2022).

[4] E. K. Elsayed and D. R. Fathy, "Sign language semantic translation system using ontology and deep learning," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 1. pp. 141–147, 2020, doi: 10.14569/ijacsa.2020.0110118.

[5] M. M. Rahman, M. S. Islam, M. H. Rahman, R. Sassi, M. W. Rivolta, and M. Aktaruzzaman, "A new benchmark on american sign language recognition using convolutional neural network," *2019 International Conference on Sustainable Technologies for Industry 4.0, STI 2019*. doi: 10.1109/STI47673.2019.9067974.

[6] S. Liwicki and M. Everingham, "Automatic recognition of fingerspelled words in british sign language," *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, pp. 50–57, 2009, doi: 10.1109/CVPR.2009.5204291.

[7] R. El Rwelli, O. R. Shahin, and A. I. Taloba, "Gesture based Arabic Sign Language Recognition for Impaired People based on Convolution Neural Network," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 12, pp. 574–582, 2021, doi: 10.14569/IJACSA.2021.0121273.

[8] S. O. Ali, R. R. O. Al-Nima, and E. A. Mohammed, "Individual Recognition with Deep Earprint Learning," *International Conference on Communication and Information Technology, ICICT 2021*, pp. 304–309, 2021, doi: 10.1109/ICICT52195.2021.9568410.

[9] S. M. M. Najeeb and and M. L. A.-D. Raid Rafi Omar Al-Nima, "Reinforced Deep Learning for Verifying Finger Veins," *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 17, no. 07, 2021, pp. 19–27, 2021, [Online]. Available: https://doi.org/10.3991/ijoe.v17i07.24655.

[10] R. R. O. Al-Nima, T. Han, S. A. M. Al-Sumaidaee, T. Chen, and W. L. W. Woo, "Robustness and performance of Deep Reinforcement Learning," *Elsevier, Applied Soft Computing*, vol. 105, 2021, 2021, [Online]. Available: https://doi.org/10.1016/j.asoc.2021.107295.

[11] Abdulsattar M. Ibrahim, A. K. Eesee, and R. R. O. Al-Nima, "Deep fingerprint

classification network.pdf," *TELKOMNIKA Telecommunication, Computing, Electronics and Control*, vol. 19, no. 3, June 2021, p. 893~901, 2021, doi: DOI: 10.12928/TELKOMNIKA.v19i3.18771.

[12] L. H. Albak, R. R. O. Al-Nima, and A. H. Salih, "Palm print verification based deep learning," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 19, no. 3. pp. 851–857, 2021, doi: 10.12928/TELKOMNIKA.v19i3.16573.

[13] S. Q. H. and S. E. Raid Rafi Omar Al-Nima, "Exploiting the Deep Learning with Fingerphotos to Recognize People," *International Journal of Advanced Science and Technology*, vol. 29, no. 7. pp. 13035–13046, 2020.

[14] M. M. A. Abuqadumah, M. A. M. Ali, and R. R. O. Al-Nima, "Personal Authentication Application Using Deep Learning Neural Network," *Proceedings - 2020 16th IEEE International Colloquium on Signal Processing and its Applications, CSPA 2020*, no. Cspa, pp. 186–190, 2020, doi: 10.1109/CSPA48992.2020.9068706.

[15] R. R. Omar, T. Han, S. A. M. Al-Sumaidaee, and T. Chen, "Deep finger texture learning for verifying people," *IET Biometrics*, vol. 8, no. 1, pp. 40–48, 2019, doi: 10.1049/iet-bmt.2018.5066.

[16] M. M. AL-Hatab, R. R. O. Al-Nima, I. Marcantoni, C. Porcaro, and L. Burattini, "Comparison Study Between Three Axis Views of Vision, Motor and Pre-Frontal Brain Activities," *Journal of Critical Reviews*, vol. 7, no. 5. pp. 2598–2607, 2020.

[17] M. M. AL-Hatab, R. R. O. Al-Nima, I. Marcantoni, C. Porcaro, and L. Burattini, "CLASSIFYING VARIOUS BRAIN ACTIVITIES BY EXPLOITING DEEP LEARNING TECHNIQUES AND GENETIC ALGORITHM FUSION METHOD," *TEST Engineering & Management*, vol. 83, pp. 3035–3052.

[18] A. S. Anaz and R. R. O. Al-nima, "Multi-Encryptions System Based on Autoencoder Deep Learning Network," *Solid State Technology*, vol. 63, no. 6, pp. 3632–3645, 2020.

[19] K. Assaleh and M. Al-Rousan, "Recognition of arabic sign language alphabet using polynomial classifiers," *Eurasip Journal on Applied Signal Processing*, pp. 2136–2145, 2005, doi: 10.1155/ASP.2005.2136.

[20] M. Mohandes, S. Aliyu, and M. Deriche, "Arabic sign language recognition using the leap motion controller," *IEEE International Symposium on Industrial Electronics*, pp. 960–965, 2014, doi: 10.1109/ISIE.2014.6864742.

[21] R. Alzohairi, R. Alghonaim, W. Alshehri, S. Aloqeely, M. Alzaidan, and O. Bchir, "Image based Arabic Sign Language recognition system," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 3, pp. 185–194, 2018, doi: 10.14569/IJACSA.2018.090327.

[22] S. K. Yewale and P. K. Bharne, "Hand Gesture Recognition Using Different Algorithms Based on Artificial Neural Network," *International Journal of Electronics Signals and Systems*, no. 1998, pp. 128–133, 2011, doi: 10.47893/ijess.2011.1025.

[23] M. HafizurRahman and J. Afrin, "Hand Gesture Recognition using Multiclass Support Vector Machine," *International Journal of Computer Applications*, vol. 74, no. 1, pp. 39–43, 2013, doi: 10.5120/12852-9367.

[24] J. Triesch and C. Von Der Malsburg, "A system for person-independent hand posture recognition against complex backgrounds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 12, pp. 1449–1453, 2001, doi: 10.1109/34.977568.

[25] H. Luqman and E. S. M. El-Alfy, "Towards hybrid multimodal manual and non-manual arabic sign language recognition: Marsl database and pilot study," *Electronics*

(Switzerland), vol. 10, no. 14, pp. 1–16, 2021, doi: 10.3390/electronics10141739.
[26] Z. Zafrulla, H. Brashear, P. Yin, P. Presti, T. Starner, and H. Hamilton, "American sign language phrase verification in an educational game for deaf children," *Proceedings - International Conference on Pattern Recognition*, pp. 3846–3849, 2010, doi: 10.1109/ICPR.2010.937.
[27] S. Lang, M. Block, and R. Rojas, "Sign language recognition using kinect," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7267 LNAI, no. PART 1, pp. 394–402, 2012, doi: 10.1007/978-3-642-29347-4_46.
[28] K. Gunasekaran and R. Manikandan, "Sign language to speech translation system using PIC microcontroller," *International Journal of Engineering and Technology*, vol. 5, no. 2, pp. 1024–1028, 2013.
[29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–14, 2015.
[30] M. A. Ahmed, B. B. Zaidan, A. A. Zaidan, M. M. Salih, and M. M. Bin Lakulu, "A review on systems-based sensory gloves for sign language recognition state of the art between 2007 and 2017," *Sensors (Switzerland)*, vol. 18, no. 7, 2018, doi: 10.3390/s18072208.
[31] A. M. Abdel-Fattah, "Arabic Sign Language: A Perspective," *Journal ofDeafStudies and DeafEducation*, vol. 10, pp. 213–221, 2005, doi: 10.1093/deafed/eni007.
[32] H. Luqman and S. A. Mahmoud, "Automatic translation of Arabic text-to-Arabic sign language," *Universal Access in the Information Society*, vol. 18, no. 4, pp. 939–951, 2019, doi: 10.1007/s10209-018-0622-8.
[33] M. Hossain, A. Mahmood, M. R. Rahman, S. Rahman, R. J. Rony, and N. Ahmed, "Kotha Bondhu: Translating Sign Languages into Bengali Voice," *PervasiveHealth: Pervasive Computing Technologies for Healthcare*, pp. 21–24, doi: 10.1145/3391203.3391208.
[34] E. Martinez-Martin and F. Morillas-Espejo, "Deep Learning Techniques for Spanish Sign Language Interpretation," *Computational Intelligence and Neuroscience*, vol. 5532580, pp. 1–10, 2021, doi: 10.1155/2021/5532580.
[35] "Mongolian alphabet | Britannica." https://www.britannica.com/topic/Mongolian-alphabet (accessed Mar. 01, 2022).
[36] S. O. Ali, R. R. O. Al-Nima, and E. A. Mohammed, *Earprint Authentication for Communicating Purpose*. LAP Lambert Academic Publishing, 2021.
[37] P. Khaire, P. Kumar, and J. Imran, "Combining CNN streams of RGB-D and skeletal data for human activity recognition," *Pattern Recognition Letters*, vol. 115, pp. 107–116, 2018, doi: 10.1016/j.patrec.2018.04.035.
[38] E. Simo-Serra, S. Iizuka, K. Sasaki, and H. Ishikawa, "Learning to Simplify: Fully Convolutional Networks for Rough Sketch Cleanup," *ACM Trans. Graph.*, vol. 35, no. 4, Article 121, p. 11, doi: http://dx.doi.org/10.1145/2897824.2925972.
[39] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural," *Handbook of Approximation Algorithms and Metaheuristics*. pp. 1–1432, 2007, doi: 10.1201/9781420010749.
[40] Q. Zhao, S. Lyu, B. Zhang, and W. Feng, "Multiactivation Pooling Method in Convolutional Neural Networks for Image Recognition," *Wireless Communications and Mobile Computing*, vol. 8196906, pp. 1–16, 2018, doi: 10.1155/2018/8196906.
[41] D. Stutz, "Neural Codes for Image Retrieval," *Proc. of the Comput. VisionECCV*, pp. 584–

599.
[42] J. Wu, “Introduction to Convolutional Neural Networks,” *National Key Lab for Novel Software Technology Nanjing University. China 5*, pp. 1–31, 2017.
[43] M. T. Al-Kaltakchi, R. R. Omar, H. N. Abdullah, T. Han, and J. A. Chambers, “Finger Texture Verification Systems Based on multiple spectrum Lighting Sensors with Four Fusion Levels,” *Iraqi Journal of Information & Communications Technology*, vol. 1, no. 3, pp. 1–16, 2019, doi: 10.31987/ijict.1.3.28.
[44] R. R. O. Al-nima, S. S. Dlay, and J. A. Chambers, “EFFICIENT FINGER SEGMENTATION ROBUST TO HAND ALIGNMENT IN IMAGING WITH APPLICATION TO HUMAN VERIFICATION,” *5th IEEE International Workshop on Biometrics and Forensics (IWBF)*, pp. 1–6, 2017.