



Arabic Sign Language Gesture Classification Using Deer Hunting Optimization with Machine Learning Model

Badriyya B. Al-onazi¹, Mohamed K. Nour², Hussain Alshahran³, Mohamed Ahmed Elfaki³,
Mrim M. Alnfai⁴, Radwa Marzouk⁵, Mahmoud Othman⁶, Mahir M. Sharif⁷ and
Abdelwahed Motwakel^{8,*}

¹Department of Language Preparation, Arabic Language Teaching Institute, Princess Nourah Bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia

²Department of Computer Sciences, College of Computing and Information System, Umm Al-Qura University, Saudi Arabia

³Department of Computer Science, College of Computing and Information Technology, Shaqra University, Shaqra, Saudi Arabia

⁴Department of Information Technology, College of Computers and Information Technology, Taif University, Taif P.O. Box 11099, Taif, 21944, Saudi Arabia

⁵Department of Information Systems, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia

⁶Department of Computer Science, Faculty of Computers and Information Technology, Future University in Egypt, New Cairo, 11835, Egypt

⁷Department of Computer Science, Faculty of Computer Science and Information Technology, Omdurman Islamic University, Omdurman, 14415, Sudan

⁸Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam bin Abdulaziz University, AlKharj, Saudi Arabia

*Corresponding Author: Abdelwahed Motwakel. Email: a.ismaeil@psau.edu.sa

Received: 15 August 2022; Accepted: 13 October 2022

Abstract: Sign language includes the motion of the arms and hands to communicate with people with hearing disabilities. Several models have been available in the literature for sign language detection and classification for enhanced outcomes. But the latest advancements in computer vision enable us to perform signs/gesture recognition using deep neural networks. This paper introduces an Arabic Sign Language Gesture Classification using Deer Hunting Optimization with Machine Learning (ASLGC-DHOML) model. The presented ASLGC-DHOML technique mainly concentrates on recognizing and classifying sign language gestures. The presented ASLGC-DHOML model primarily pre-processes the input gesture images and generates feature vectors using the densely connected network (DenseNet169) model. For gesture recognition and classification, a multilayer perceptron (MLP) classifier is exploited to recognize and classify the existence of sign language gestures. Lastly, the DHO algorithm is utilized for parameter optimization of the MLP model. The experimental results of the ASLGC-DHOML model are tested and the outcomes are inspected under distinct aspects. The comparison analysis highlighted that the ASLGC-DHOML method has resulted in enhanced



gesture classification results than other techniques with maximum accuracy of 92.88%.

Keywords: Machine learning; sign language recognition; multilayer perceptron; deer hunting optimization; densenet

1 Introduction

Sign language is important for communicating with deaf and mute people, ordinary people, and themselves. Sign language is a subset of communication utilized as a medium of interaction by the deaf. Dissimilar to other natural languages, it uses body movements for communication, named as gestures or signs. Arabic is the 4th most spoken language in the world. Arabic Sign Language (ArSL) was a certified main language for talking and listening impaired in Arab nations [1]. Although Arabic is one of the global key languages, ArSL was still in its initial levels [2]. The typical issue ArSL patients experience is “diglossia.” Regional dialects were spoken than written languages around every nation. So, several spoken dialects generated various ArSLs. They were as copious as Arab states; hitherto, they shared numerous alphabet and terminologies [3]. “ArSL was reliable on the alphabet.” Arabic is considered one of the Semitic languages spoken by nearly 3.8 million people globally as its primary official language [4].

Sign language (SL) comprises 4 major manual elements: hand orientation, hand figure configuration, hand location, and hand movement relating to the body [5]. Two procedures exist 2 procedures that have an automatic sign-recognition mechanism to identify the features and classify input data. Several techniques were brought for classifying and detecting sign languages for the betterment of act of the automatic SL mechanism. SL was considered an interaction subset utilized as a channel of interaction by deaf [6]. Dissimilar to other natural languages, it employs important body gestures for communicating messages, called signs or gestures. For communicating a message, finger and hand gestures, facial expressions, head nodding, and shoulder gestures were used. Thus, the suggested work will be helpful for deaf people for interaction among deaf and normal individuals or deaf and deaf. If a deaf individual attempts to express anything, they employ gestures for communication. Every symbol indicates a special letter, emotion, or word [7]. A stage was formed by signal combination, and a string of words invokes letters in spoken languages. Therefore, SL was a natural language with sentence and structure grammar [8].

Conversely, DL was a subset of machine learning (ML) in AI that has networks that can perform learning unsupervised from data that were unlabeled or unstructured, which was also called a deep neural network (DNN) or deep neural learning [9]. In DL, a convolutional neural network (CNN) is a class of DNN, most typically implied in the domain of computer vision (CV). The vision-related techniques largely aim at the captured gestures image and receive the primary feature for identifying them. This technique was implied in several tasks, which include semantic segmentation, super-resolution, multimedia systems, and emotion recognition and image classification [10].

Hassan et al. [11] introduce a complete evaluation among 2 different recognition methods for continual ArSLR, such as a Modified k-Nearest Neighbor that suits Hidden Markov Models (HMMs) and sequential data methods based on 2 distinct toolkits. Moreover, in this work, 2 novel ArSL datasets comprising forty Arabic sentences were accumulated using a camera and Polhemus G4 motion tracker. Ibrahim et al. [12] provide an automated visual SLRS which converts isolated Arabic word signs into text. The suggested mechanism has 4 phases: hand segmentation, classification, tracking, and feature extraction. After that, a suggested skin-blob tracking method was utilized to identify and track the

hands. Deriche et al. [13] suggest a dual leap motion controller (LMC)-related Arabic sign language recognition mechanism. To be very specific, the idea of utilizing both side and front LMCs was introduced to cater for the difficulties of missing data and finger occlusions. For feature extraction, an optimal geometric feature set was chosen from both controllers. In contrast, in classification, a Bayesian technique with a Gaussian mixture model (GMM) and a simple linear discriminant analysis (LDA) method was utilized. Combining the information from 2 LMCs introduces evidence-related fusion techniques such as the Dempster-Shafer (DS) evidence theory.

Elpeltagy et al. [14] suggested technique is made up of 3 major phases: sign classification, hand segmentation, hand shape sequence, and body motion description. The hand shape segmenting depended on the position and depth of the hand joints. Histograms of related gradients and principal component analysis (PCA) were implied on segmented hand shapes for obtaining hand shape series descriptors. The co-variance of 3-dimension joints of the upper half of the skeleton, along with the face properties and hand states were implemented for motion sequence description.

This paper introduces an Arabic Sign Language Gesture Classification using Deer Hunting Optimization with Machine Learning (ASLGC-DHOML) model. The presented ASLGC-DHOML technique mainly concentrates on recognising and classifying sign language gestures. The presented ASLGC-DHOML model primarily pre-processes the input gesture images and generates feature vectors using the densely connected network (DenseNet169) model. For gesture recognition and classification, a multilayer perceptron (MLP) classifier is exploited to recognize and classify the existence of sign language gestures. Lastly, the DHO algorithm is utilized for parameter optimization of the MLP model. The experimental results of the ASLGC-DHOML model are tested, and the outcomes are inspected under distinct aspects.

2 Materials and Methods

In this study, a new ASLGC-DHOML technique was developed for recognising and classifying sign language gestures. The presented ASLGC-DHOML model primarily pre-processes the input gesture images and generates feature vectors using the DenseNet169 model. For gesture recognition and classification, the MLP classifier is exploited to recognize and classify the existence of sign language gestures. Lastly, the DHO algorithm is utilized for parameter optimization of the MLP model.

2.1 Level I: Feature Extraction

The presented ASLGC-DHOML model primarily pre-processes the input gesture images and generates feature vectors using the DenseNet169 model. DenseNet is a DL structure where every layer is directly linked, achieving effectual data flow. All the layers get extra inputs in every preceding layer and transmissions their feature map (FM) for each following layer [15]. The resultant FM attained in the existing layer is integrated with the preceding layer utilizing concatenation. All the layers are connected to every subsequent layer of the network, and it can be mentioned that DenseNets. This method needs some parameters related to typical CNNs. It also decreases the overfitting issue with a lesser malware-trained set. Assume that input image x_0 is approved with the presented convolution network. The network comprises N layer, and every layer implements a non-linearity transformation $F_n(.)$. Assume that layer n contains FMs of all the earlier convolutional layers. An input FM of layers 0 to $n - 1$ is concatenated and demonstrated as x_0, \dots, x_{n-1} . So, this method takes $N(N + 1)/2$ links on N layer network. The resultant of n^{th} layer was provided as:

$$x_n = F_n ([x_0, \dots, x_{n-1}]), \quad (1)$$

whereas x_n refers to the present n^{th} layer, $[x_0, \dots, x_{n-1}]$ signifies the concatenation of FMs attained in 0 to $n - 1$ layers and $F_n (\cdot)$ denotes the composite functions of batch normalization (BN) and rectified linear unit (ReLU).

The consecutive functions from the transition layer comprise BN, ReLU, and 3×3 convolutions. The concatenation function could not possible when the size of FMs was altered. Thus, the layers which contain distinct FM sizes were downsampled. The transition layers containing 1×1 Conv and 2×2 average pooling functions were provided amongst 2 neighboring Dense Conv blocks. Afterward, in the last Dense Conv block, the classifier layer containing global average pooling as well as softmax classification were linked. The correct forecast was complete utilizing every FM from the NN. A resultant layer with K neurons provides the correct match of the K malware family. The convolutional function learns the image features and continues the link between the pixels. Next the convolutional was executed on the image, ReLU was executed to the resultant FMs. This function establishes non-linearity from CNNs. The ReLU function was provided as:

$$f(x_0) = \max(0, x_0). \quad (2)$$

The pooling was executed to reduce the dimensionality of resultant FM. This pooling was implemented also utilizing average or max pooling. The max pooling contains taking the biggest element in the enhanced FM. The average pooling divides the input as to the pooling area and estimates the average value of each region. The global average pooling (GAP) calculates the average of every FM, and the outcome vector was obtained from the softmax layer. During this case, the DenseNet-169 method was employed dependent upon the fundamental DenseNet structure, and DenseNet takes $L(L + 1)/2$ direct connection.

2.2 Level II: Sign Language Classification

For gesture recognition and classification, the MLP classifier is exploited to recognize and classify the existence of sign language gestures. MLP comprises three (output, input, and hidden) layers. The trial-and-error mechanism defines the number of neurons in every layer [16]. The primary weight of this neural network is randomly defined. The error backpropagation model is applied for training the NN model, whereby the weight of the network changes in a supervised model depends on the variance among the desired and neural network outputs; hence, for all the inputs, the output is produced using the NN model. The input and output patterns are normalized first through a normalized factor for equalizing the training model's impact in altering the network's weight. For p - the input patterns, the squared error in every neuron is evaluated by the subsequent formula:

$$E_p = \frac{1}{2} (d^p - y^p)^2 = \frac{1}{2} \sum_{j=1}^{n_j} (d_j^p - y_j^p)^2 \quad (3)$$

In Eq. (3), d_j^p and y_j^p are, correspondingly, the value for desired and evaluated outputs in the j - th neurons for p pattern. Also, overall squared errors for each pattern are evaluated by the following equation:

$$E = \sum_{p=1}^N E_p = \frac{1}{2} \sum_{p=1}^{n_p} \sum_{j=1}^{n_j} (d_j^p - y_j^p)^2. \quad (4)$$

$$w_{ij(t+1)} = w_{ij}(t) + \eta d \Delta w_{ij}(t) + \alpha \Delta w_{ij}(t + 1),$$

$$\Delta w_{ij} = - \left(\frac{\delta E_p}{\delta w_{ij}(t)} \right). \quad (5)$$

Here, $w_{ij}(t+1)$ indicates the current weight, $w_{ij}(t)$ denotes the preceding weight, η refers to learning coefficient, and α characterizes the momentary coefficient. In the study, weights are repeatedly upgraded for each learning pattern. The training procedure ends while the overall error values for each pattern achieve a value lower when compared to the defined critical point or once the entire learning period obtains the last point. It should be noted that the training methodology is a BP error model with momentary term that reduces the probability of coordination at local minimal in comparison to the BP error mechanism.

2.3 Level III: Parameter Optimization

Finally, the DHO algorithm is utilized for parameter optimization of the MLP model. DHO approach is a metaheuristic algorithm stimulated by the hunting nature of humans toward deer. Even though the action of the hunter might vary, the strategy of assaulting the deer or buck chiefly relies on the hunting strategy [17]. Thanks to the particular abilities of deer, it could easily escape. The hunting strategy is depending upon the movement of 2 hunters in the best possible location named leader and successor. During deer hunting strategy, the hunter encloses it and moves to the prey. Afterward, each hunter upgrades the location until they find the deer. Likewise, accommodating nature amongst the hunters is indispensable for proficiently making the hunting strategy. Ultimately, they find the prey according to the location of the leader and successor. At first, the population of hunters is represented as follows,

$$Y = \{Y_1, Y_2, \dots, Y_n\}; 1 < j \leq n \quad (6)$$

In Eq. (6), n indicates the hunter count that is regarded as a solution in Y population. If the population is initialized, wind angle and location of deer are the 2 vital features while estimating the optimum location of the hunter. Mainly, the searching region is regarded as a circle and the wind angle follows the circumference of a circle.

$$\theta_i = 2\pi r \quad (7)$$

In Eq. (7), r indicates a random integer lies within zero and one, i represents the existing iteration. At the same time, the angle location of a deer is formulated by,

$$f_i = \theta + \pi \quad (8)$$

In Eq. (8), θ determines a wind angle. If the location of optimum area is not defined, the solution candidate is located nearer to the optimum one and described according to the Fitness Function (FF) denoted as optimal solution. Here, two solutions were taken into account. Fig. 1 illustrates the stages of DHO approach.

Propagation through a leader's position:

When the optimum location is enforced, each individual of a population tries to obtain a consecutive location and iteratively upgrade the location. Then, encircle behavior is labelled by the following equation,

$$Y_{i+1} = Y^{lead} - X \cdot p \cdot |L \times Y^{lead} - Y_i| \quad (9)$$

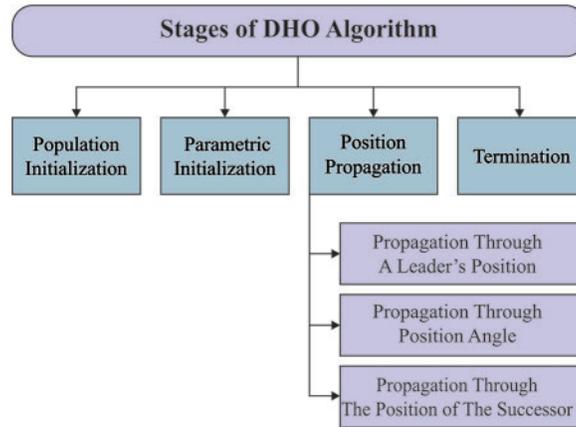


Figure 1: Stages of DHO technique

In Eq. (9), Y_i indicates the location in existing iteration, Y_{i+1} represent the position at following iterations, X and L denotes the coefficient vector, and p decides an arbitrary value positioned by the wind speed whereby the value ranging from $[0, 2]$ and it is assessed by,

$$X = \frac{1}{4} \log \left(i + \frac{1}{i_{\max}} \right) b \quad (10)$$

$$L = 2 \cdot c \quad (11)$$

whereby j_{\max} indicates high iteration, b represents a parameter ranging from $[-1, 1]$ and c indicates an arbitrary integer range within zero and one.

Now, (Y, Z) suggests the initialized location of a hunter that is upgraded by using prey location. Then, agent position is altered until it reaches an effectual location (Y^*, Z^*) and changes the location of X and L . Location updating is functioned by the Eq. (10) where $p < 1$, indicates that an individual is allowable only to move in random way notwithstanding of angle location. Therefore, Eqs. (9) and (10) illustrate the location updating of a hunter randomly within a particular region.

Propagation through position angle:

To increase the searching region, the process gets upgraded through location angle. The angle evaluation is extremely substantial for calculating the location of a hunter whereby the prey is unconscious of the danger and makes the hunting strategy very effectual. The visualization angle can be defined by,

$$a_j = \frac{\pi}{8} \times r \quad (12)$$

Based on the distinctions amongst the visual and wind angles, novel attributes are defined to upgrade the angle location.

$$d_i = \theta_i - a_i \quad (13)$$

In Eq. (13), θ indicates the wind angle. Then, a location angle is upgraded to following iteration as follows,

$$f_{i+1} = f_i + d_i \quad (14)$$

By taking the angle location into account it is updated as follows,

$$Y_{i+1} = Y^{lead} - p \cdot |\cos(v) \times Y^{lead} - Y_i| \quad (15)$$

If $A = f_{i+1}$, Y_i^* indicates the best possible location and p indicates the arbitrary integer. The location of an individual is nearby to the inverse angle location, as a result, the hunter move from the deer sight.

Propagation through the position of the successor:

Here, the same technique of encircling behavior is exploited through expanding the L vector. Assume the searching region as random location, then the value of vector L is less than 1. Therefore, the location updating is depending on the successor position. It allows a global searching in the following,

$$Y_{i+1} = Y^{successor} - X \cdot p \cdot |L \times Y^{successor} - Y_i| \quad (16)$$

In Eq. (16), $Y^{successor}$ represents the successor location of a searching region from existing population.

3 Results and Discussion

The experimental validation of the ASLGC-DHOML method is tested by making use of a sign language dataset, comprising 500 samples under five distinct classes as shown in Table 1. A few sample sign language gesture images were illustrated in Fig. 2.

Table 1: Dataset details

Label	Class	No. of samples
Class-0	Thank you	100
Class-1	Home	100
Class-2	I	100
Class-3	Ate	100
Class-4	Wake up	100
Total no. of instances		500

Fig. 3 highlights the set of confusion matrices created by the ASLGC-DHOML model on the applied data. The figure demonstrated that the ASLGC-DHOML model has resulted ineffectual outcomes under distinct classes and runs. On run-1, the ASLGC-DHOML model has identified 68 samples into class 0, 78 samples into class 1, 81 samples under class 2, 88 samples class 3, and 83 samples into class 4. In addition, on run-3, the ASLGC-DHOML method has detected 85 samples into class 0, 63 samples into class 1, 90 samples under class 2, 88 samples class 3, and 75 samples into class 4. At last, on run-5, the ASLGC-DHOML technique has identified 71 samples into class 0, 77 samples into class 1, 80 samples under class 2, 85 samples class 3, and 84 samples into class 4.



Figure 2: Samples sign language gesture images

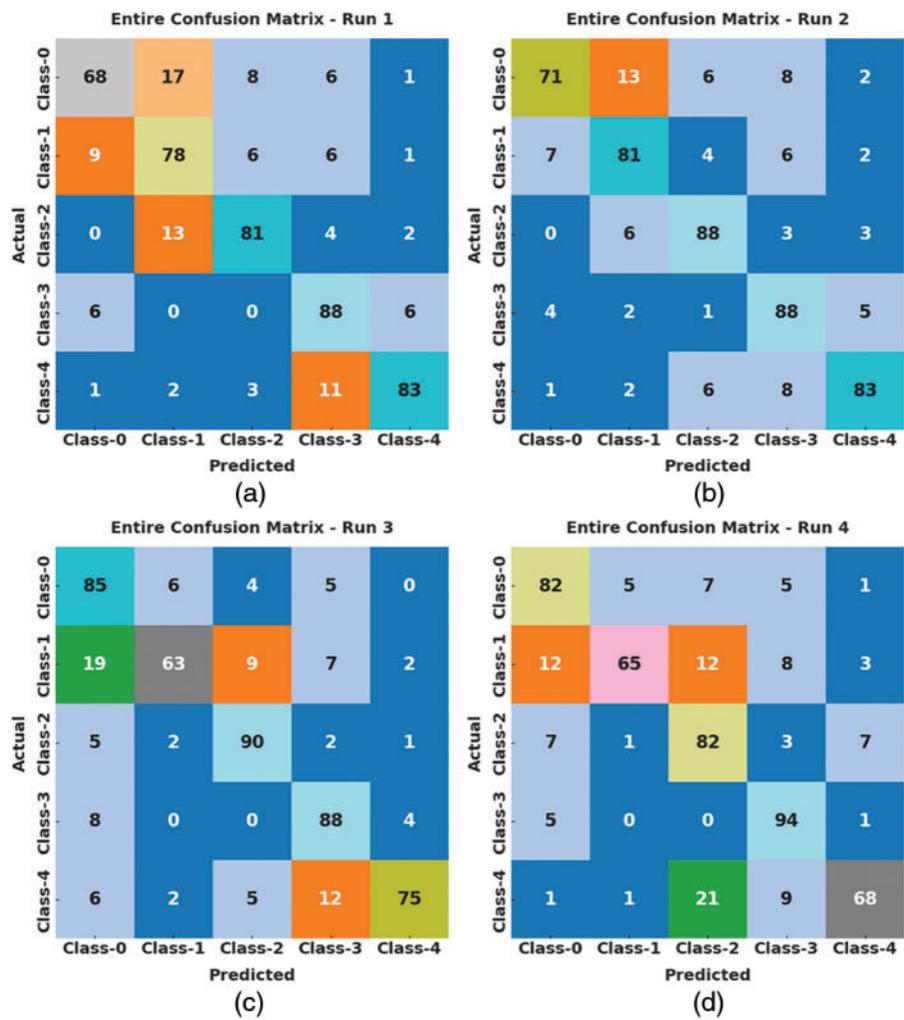


Figure 3: (Continued)

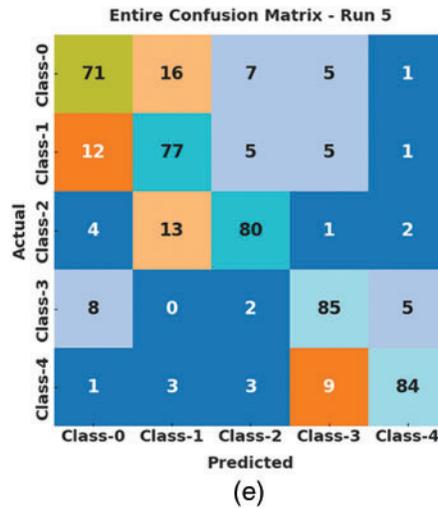


Figure 3: Confusion matrices of ASLGC-DHOML approach (a) Run-1, (b) Run-2, (c) Run-3, (d) Run-4, and (e) Run-5

A brief collection of simulation results provided by the ASLGC-DHOML model on the test data is given in Table 2 and Fig. 4. The results demonstrated that the ASLGC-DHOML model has effectually recognized all the classes under distinct runs. For instance, on run-1, the ASLGC-DHOML model has attained average $accu_y$, $sens_y$, $spec_y$, F_{score} , and $G_{measure}$ of 91.84%, 79.60%, 94.90%, 79.58%, and 79.70% respectively. Besides, on run-3, the ASLGC-DHOML technique has achieved average $accu_y$, $sens_y$, $spec_y$, F_{score} , and $G_{measure}$ of 92.08%, 80.20%, 95.05%, 80.05%, and 80.44% correspondingly. Also, on run-5, the ASLGC-DHOML approach has reached average $accu_y$, $sens_y$, $spec_y$, F_{score} , and $G_{measure}$ of 91.76%, 79.40%, 94.85%, 79.46%, and 79.50% correspondingly.

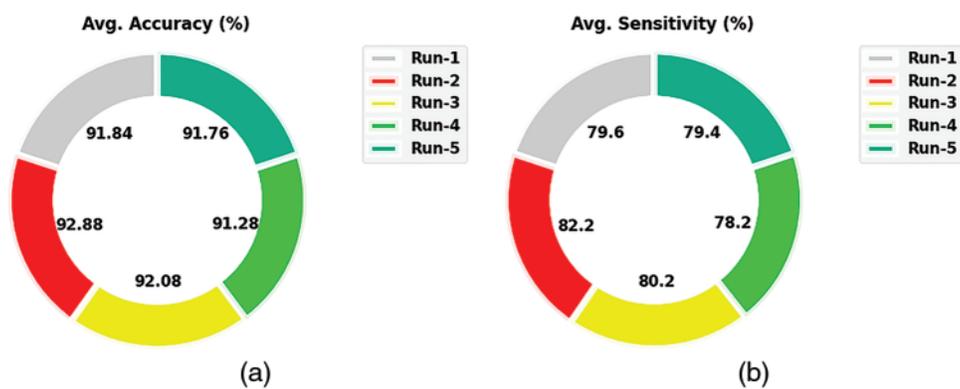
Table 2: Result analysis of ASLGC-DHOML approach with various measures and runs

Labels	Accuracy	Sensitivity	Specificity	F-score	G-measure
Run-1					
Class-0	90.40	68.00	96.00	73.91	74.19
Class-1	89.20	78.00	92.00	74.29	74.37
Class-2	92.80	81.00	95.75	81.82	81.82
Class-3	92.20	88.00	93.25	81.86	82.06
Class-4	94.60	83.00	97.50	86.01	86.07
Average	91.84	79.60	94.90	79.58	79.70
Run-2					
Class-0	91.80	71.00	97.00	77.60	77.93
Class-1	91.60	81.00	94.25	79.41	79.43
Class-2	94.20	88.00	95.75	85.85	85.88
Class-3	92.60	88.00	93.75	82.63	82.78
Class-4	94.20	83.00	97.00	85.13	85.16
Average	92.88	82.20	95.55	82.12	82.24

(Continued)

Table 2: Continued

Labels	Accuracy	Sensitivity	Specificity	F-score	G-measure
Run-3					
Class-0	89.40	85.00	90.50	76.23	76.64
Class-1	90.60	63.00	97.50	72.83	73.74
Class-2	94.40	90.00	95.50	86.54	86.60
Class-3	92.40	88.00	93.50	82.24	82.42
Class-4	93.60	75.00	98.25	82.42	82.82
Average	92.08	80.20	95.05	80.05	80.44
Run-4					
Class-0	91.40	82.00	93.75	79.23	79.27
Class-1	91.60	65.00	98.25	75.58	76.60
Class-2	88.40	82.00	90.00	73.87	74.24
Class-3	93.80	94.00	93.75	85.84	86.17
Class-4	91.20	68.00	97.00	75.56	76.03
Average	91.28	78.20	94.55	78.02	78.46
Run-5					
Class-0	89.20	71.00	93.75	72.45	72.46
Class-1	89.00	77.00	92.00	73.68	73.75
Class-2	92.60	80.00	95.75	81.22	81.23
Class-3	93.00	85.00	95.00	82.93	82.95
Class-4	95.00	84.00	97.75	87.05	87.10
Average	91.76	79.40	94.85	79.46	79.50

**Figure 4:** (Continued)

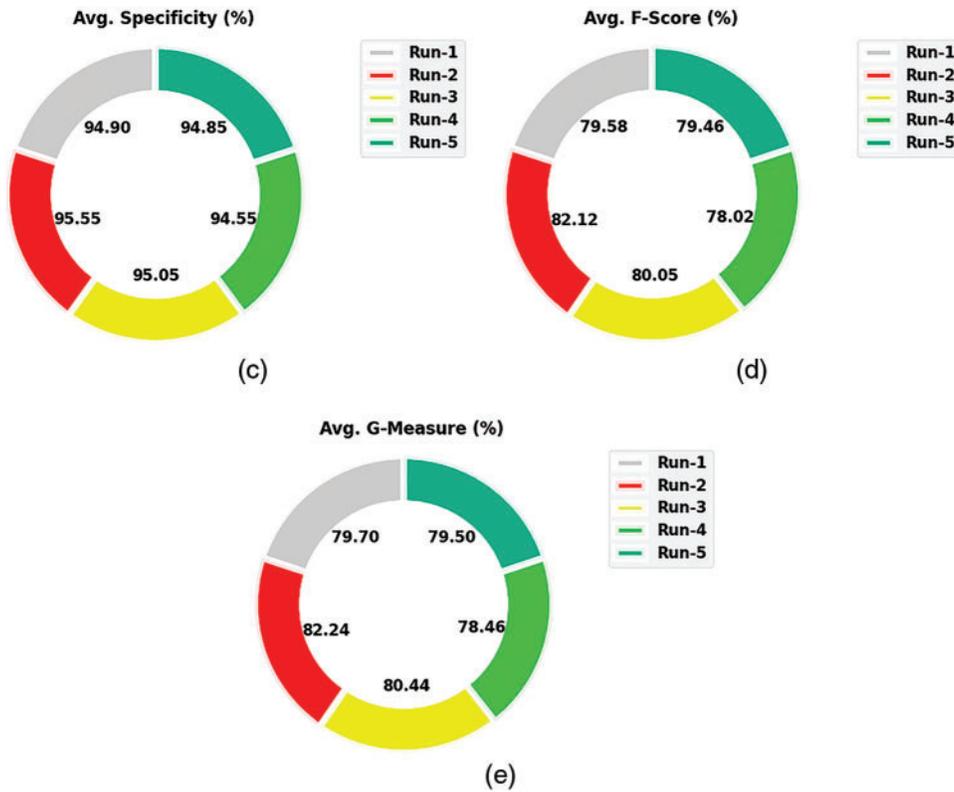


Figure 4: Average analysis of ASLGC-DHOML approach (a) Run-1, (b) Run-2, (c) Run-3, (d) Run-4, and (e) Run-5

The training accuracy (TA) and validation accuracy (VA) obtained by the ASLGC-DHOML method on the test dataset is portrayed in Fig. 5. The experimental outcome denoted the ASLGC-DHOML approach has reached maximal values of TA and VA. Specifically, the VA is greater than TA.

The training loss (TL) and validation loss (VL) gained by the ASLGC-DHOML approach on the test dataset were shown in Fig. 6. The experimental outcome represented the ASLGC-DHOML algorithm has presented least values of TL and VL. In specific, the VL is lesser than TL.

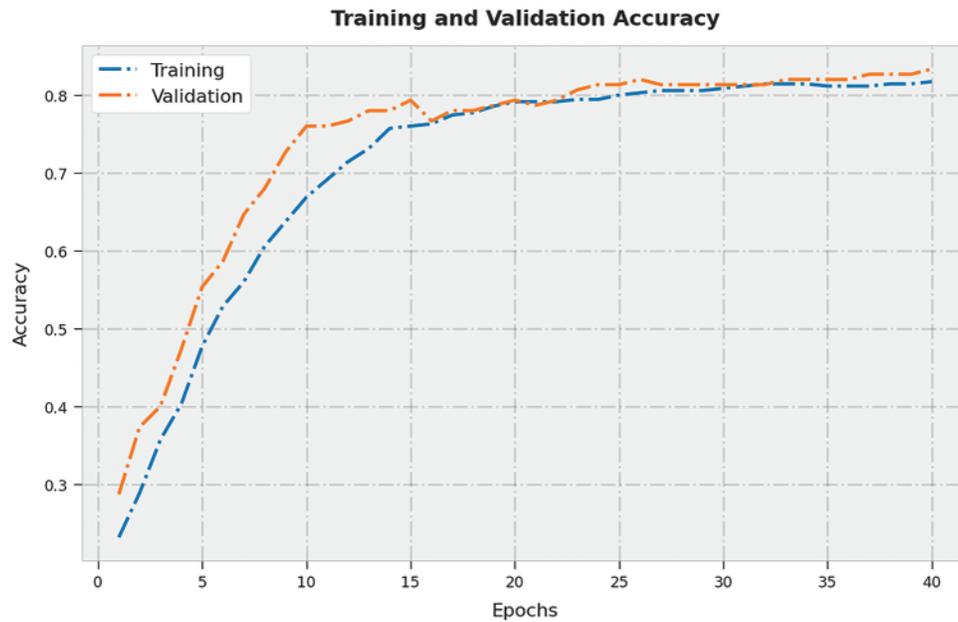


Figure 5: TA and VA analysis of ASLGC-DHOML approach

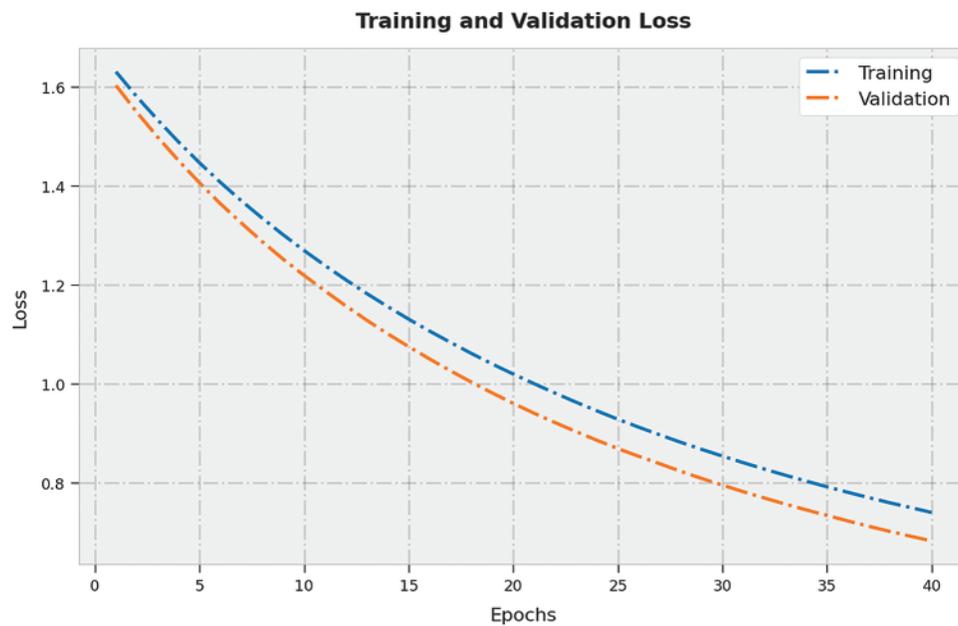


Figure 6: TL and VL analysis of ASLGC-DHOML approach

A clear precision-recall analysis of the ASLGC-DHOML algorithm on the test dataset is portrayed in Fig. 7. The figure denoted the ASLGC-DHOML technique has resulted in enhanced values of precision-recall values under all classes.

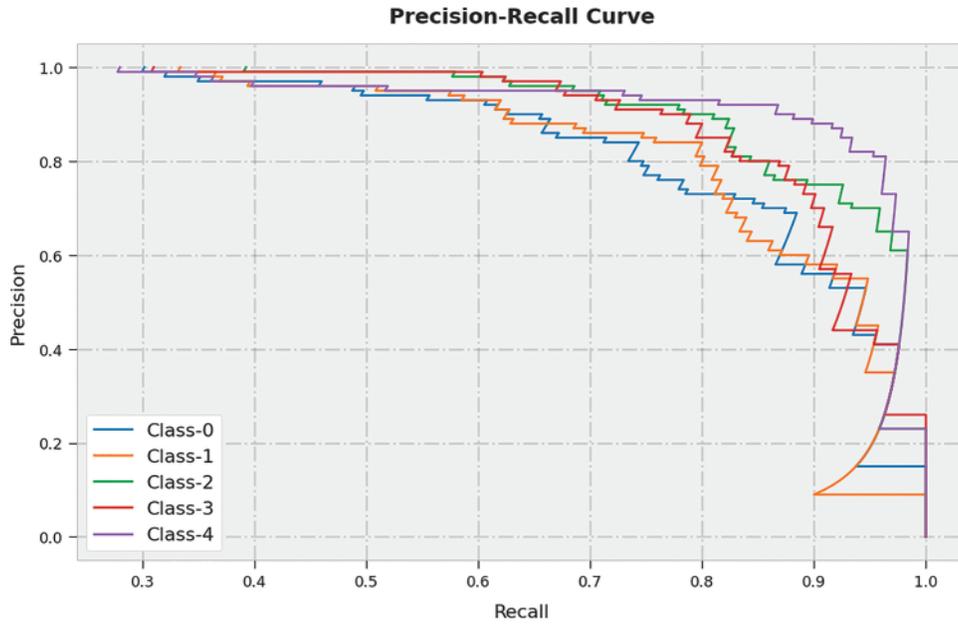


Figure 7: Precision-recall curve analysis of ASLGC-DHOML approach

A brief receiver operating characteristic (ROC) analysis of the ASLGC-DHOML approach on the test dataset is shown in Fig. 8. The results signify the ASLGC-DHOML approach has displayed its ability in categorizing distinct classes on the test dataset.

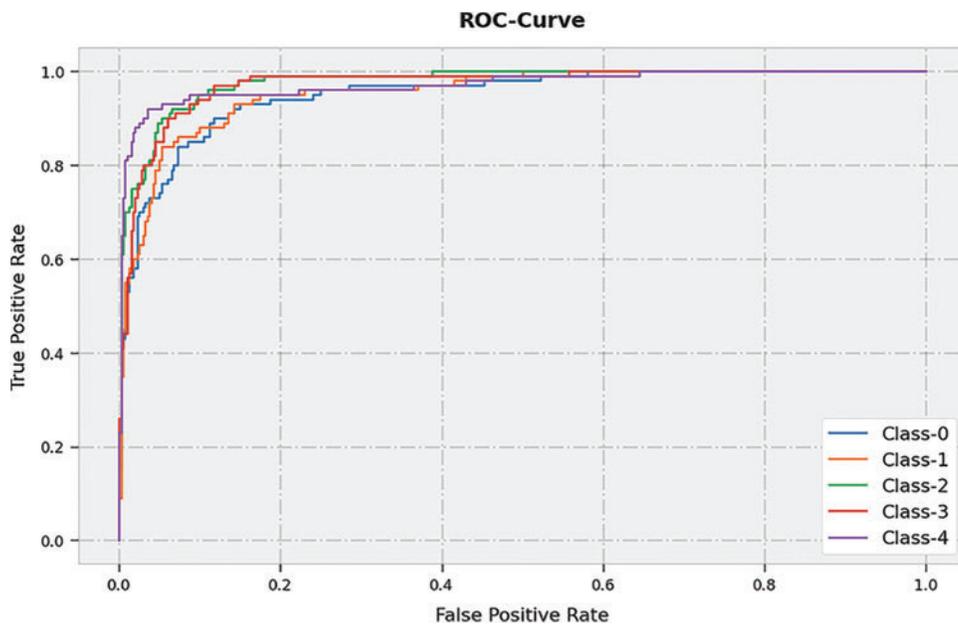


Figure 8: ROC curve analysis of ASLGC-DHOML approach

To emphasize the improvised performance of the ASLGC-DHOML method, a comparative analysis is provided in Table 3 [18]. Fig. 9 exhibits a comparative $accu_y$ inspection of the ASLGC-DHOML model with recent models. The figure demonstrated that the 3D-CNN and DeepLAV3 models have shown lower $accu_y$ values of 85.53% and 85.58% respectively. Followed by, discrete cosine transform with k-nearest neighbour (DCT-KNN) model has offered reasonable $accu_y$ of 87.90%. In the meantime, the BoF-BoP, Gaussian Naïve Bayes (GNB), and CSOM-BiLSTMNet models have reported considerable $accu_y$ of 88.74%, 88.21%, and 88.02% respectively. But the ASLGC-DHOML model has exhibited superior $accu_y$ of 92.88%.

Table 3: Comparative analysis of ASLGC-DHOML approach with existing methodologies [18]

Methods	Accuracy	Sensitivity	Specificity	F-score
ASLGC-DHOML	92.88	82.2	95.55	82.12
DCT-KNN	87.90	78.42	86.11	78.67
BoF-BoP	88.74	75.06	86.81	76.87
3D-CNN	85.53	75.72	88.52	76.65
GNB model	88.21	75.45	85.07	75.07
DeepLAV3	85.58	75.90	86.30	75.62
CSOM-BiLSTMNet	88.02	77.37	86.87	76.10

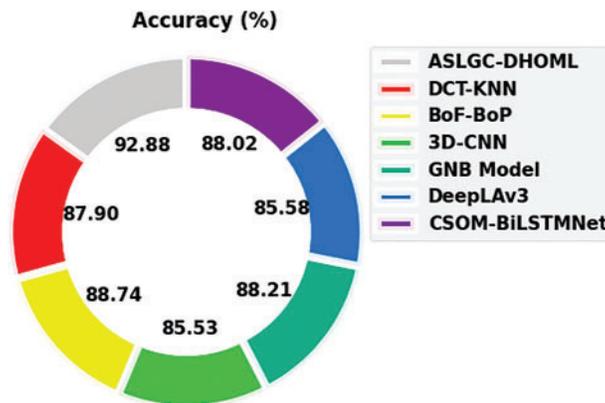


Figure 9: $Accu_{racy}$ analysis of ASLGC-DHOML approach with existing methodologies

Fig. 10 depicts a comparative $sens_y$ analysis of the ASLGC-DHOML method with recent models. The figure demonstrated that the 3D-CNN and DeepLAV3 algorithms have shown lower $sens_y$ values of 75.72% and 75.90% correspondingly. Followed by, DCT-KNN model has rendered reasonable $sens_y$ of 78.42%. Meanwhile, the BoF-BoP, GNB, and CSOM-BiLSTMNet models have reported considerable $sens_y$ of 75.06%, 75.45%, and 77.37% correspondingly. But the ASLGC-DHOML method has displayed superior $sens_y$ of 77.37%.

Fig. 11 exhibits a comparative $spec_y$ inspection of the ASLGC-DHOML model with recent models. The figure demonstrated that the 3D-CNN and DeepLAV3 models have shown lower $spec_y$ values of 88.52% and 86.30% correspondingly. Followed by, DCT-KNN model has provided reasonable $spec_y$ of 86.11%. In the meantime, the BoF-BoP, GNB, and CSOM-BiLSTMNet models have reported

considerable $spec_y$ of 86.81%, 85.07%, and 86.87% respectively. But the ASLGC-DHOML model has shown superior $spec_y$ of 95.55%.

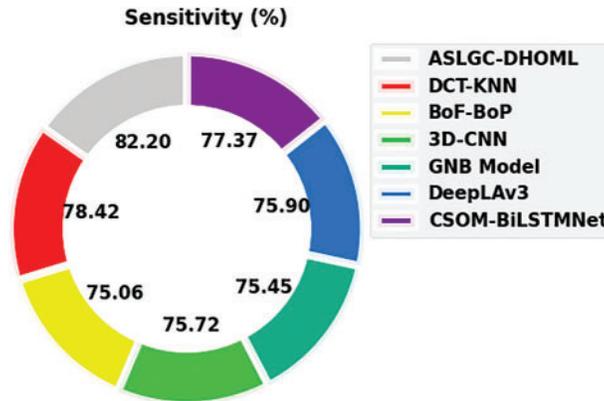


Figure 10: $Sens_y$ analysis of ASLGC-DHOML approach with existing methodologies

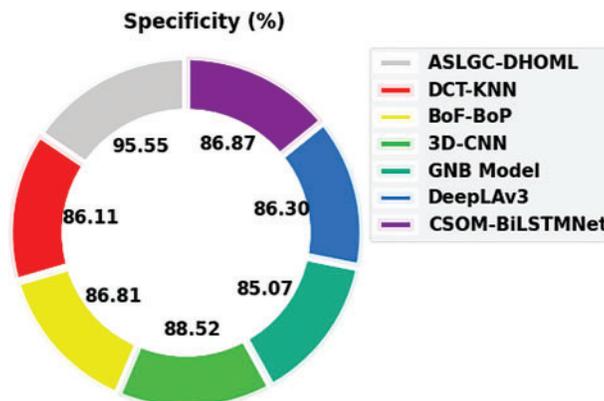


Figure 11: $Spec_y$ analysis of ASLGC-DHOML approach with existing methodologies

Fig. 12 exhibits a comparative F_{score} inspection of the ASLGC-DHOML model with recent models. The figure established that the 3D-CNN and DeepLAv3 models have shown lower F_{score} values of 76.65% and 75.62% correspondingly. Then, DCT-KNN model has provided reasonable F_{score} of 78.67%. In the meantime, the BoF-BoP, GNB, and CSOM-BiLSTMNet methodologies have reported considerable F_{score} of 76.87%, 75.07%, and 76.10% correspondingly. But the ASLGC-DHOML model has exhibited superior F_{score} of 82.12%.

Thus, the ASLGC-DHOML model has accomplishes maximum Arabic sign language gesture recognition performance.

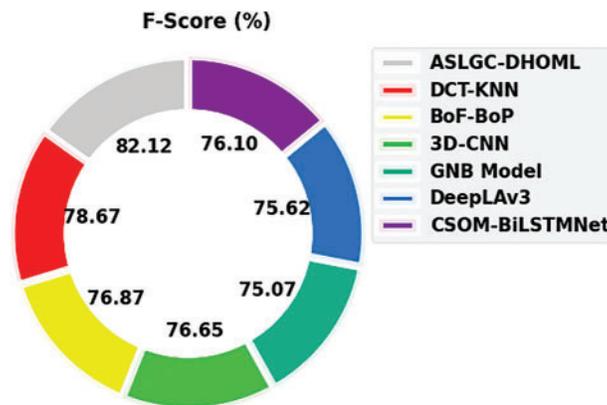


Figure 12: F_{score} analysis of ASLGC-DHOML approach with existing methodologies

4 Conclusion

In this study, a new ASLGC-DHOML technique was developed for the recognition and classification of sign language gestures. The presented ASLGC-DHOML model primarily pre-processes the input gesture images and generates feature vectors using the DenseNet169 model. For gesture recognition and classification, MLP classifier is exploited to recognize and classify the existence of sign language gestures. Lastly, the DHO algorithm is utilized for parameter optimization of the MLP model. The experimental results of the ASLGC-DHOML model are tested and the outcomes are inspected under distinct aspects. The comparison analysis highlighted that the ASLGC-DHOML method has resulted in enhanced gesture classification results than other techniques with higher accuracy of 92.88%. As a part of future scope, the performance of the ASLGC-DHOML model is improved by the utilization of advanced DL classification models.

Funding Statement: Princess Nourah bint Abdulrahman University Researchers Supporting Project Number (PNURSP2023R263), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. The authors would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by Grant Code: 22UQU4310373DSR54.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] M. A. Almasre and H. Al-Nuaim, "A comparison of Arabic sign language dynamic gesture recognition models," *Heliyon*, vol. 6, no. 3, pp. e03554, 2020.
- [2] 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.
- [3] B. Hisham and A. Hamouda, "Arabic sign language recognition using Ada-boosting based on a leap motion controller," *International Journal of Information Technology*, vol. 13, no. 3, pp. 1221–1234, 2021.
- [4] H. Luqman and S. A. Mahmoud, "A machine translation system from Arabic sign language to Arabic," *Universal Access in the Information Society*, vol. 19, no. 4, pp. 891–904, 2020.

- [5] G. Latif, N. Mohammad, R. AlKhalaf, R. AlKhalaf, J. Alghazo *et al.* “An automatic Arabic sign language recognition system based on deep cnn: An assistive system for the deaf and hard of hearing,” *International Journal of Computing and Digital Systems*, vol. 9, no. 4, pp. 715–724, 2020.
- [6] A. Ahmed, R. A. Alez, G. Tharwat, M. Taha, B. Belgacem *et al.* “Arabic sign language intelligent translator,” *The Imaging Science Journal*, vol. 68, no. 1, pp. 11–23, 2020.
- [7] A. S. Al-Shamayleh, R. Ahmad, N. Jomhari and M. A. Abushariah, “Automatic Arabic sign language recognition: A review, taxonomy, open challenges, research roadmap and future directions,” *Malaysian Journal of Computer Science*, vol. 33, no. 4, pp. 306–343, 2020.
- [8] S. M. Elatawy, D. M. Hawa, A. A. Ewees and A. M. Saad, “Recognition system for alphabet Arabic sign language using neutrosophic and fuzzy c-means,” *Education and Information Technologies*, vol. 25, no. 6, pp. 5601–5616, 2020.
- [9] A. A. Samie, F. Elmisery, A. M. Brisha and A. Khalil, “Arabic sign language recognition using kinect sensor,” *Research Journal of Applied Sciences, Engineering and Technology*, vol. 15, no. 2, pp. 57–67, 2018.
- [10] M. H. Ismail, S. A. Dawwd and F. H. Ali, “Static hand gesture recognition of Arabic sign language by using deep CNNs,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 24, no. 1, pp. 178, 2021.
- [11] M. Hassan, K. Assaleh and T. Shanableh, “Multiple proposals for continuous Arabic sign language recognition,” *Sensing and Imaging*, vol. 20, no. 1, pp. 4, 2019.
- [12] N. B. Ibrahim, M. M. Selim and H. H. Zayed, “An automatic Arabic sign language recognition system (ArSLRS),” *Journal of King Saud University-Computer and Information Sciences*, vol. 30, no. 4, pp. 470–477, 2018.
- [13] M. Deriche, S. O. Aliyu and M. Mohandes, “An intelligent Arabic sign language recognition system using a pair of lmcns with gmm based classification,” *IEEE Sensors Journal*, vol. 19, no. 18, pp. 8067–8078, 2019.
- [14] M. Elpeltagy, M. Abdelwahab, M. E. Hussein, A. Shoukry, A. Shoala *et al.* “Multi-modality-based Arabic sign language recognition,” *IET Computer Vision*, vol. 12, no. 7, pp. 1031–1039, 2018.
- [15] J. Hemalatha, S. Roseline, S. Geetha, S. Kadry and R. Damaševičius, “An efficient densenet-based deep learning model for malware detection,” *Entropy*, vol. 23, no. 3, pp. 344, 2021.
- [16] H. Wang, H. Moayedi and L. Kok Foong, “Genetic algorithm hybridized with multilayer perceptron to have an economical slope stability design,” *Engineering with Computers*, vol. 37, no. 4, pp. 3067–3078, 2021.
- [17] Z. Yin and N. Razmjoooy, “PEMFC identification using deep learning developed by improved deer hunting optimization algorithm,” *International Journal of Power and Energy Systems*, vol. 40, no. 2, pp. 189–203, 2020.
- [18] S. Aly and W. Aly, “DeepArSLR: A novel signer-independent deep learning framework for isolated Arabic sign language gestures recognition,” *IEEE Access*, vol. 8, pp. 83199–83212, 2020.