

Article

Atom Search Optimization with Deep Learning Enabled Arabic Sign Language Recognition for Speaking and Hearing Disability Persons

Radwa Marzouk ^{1,2}, Fadwa Alrowais ³ , Fahd N. Al-Wesabi ^{4,*}  and Anwer Mustafa Hilal ⁵

¹ Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia

² Department of Mathematics, Faculty of Science, Cairo University, Giza 12613, Egypt

³ Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia

⁴ Department of Computer Science, College of Science and Art at Mahayil, King Khalid University, Muhayel Aseer 63311, Saudi Arabia

⁵ Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam bin Abdulaziz University, AlKharj 16242, Saudi Arabia

* Correspondence: falwesabi@kku.edu.sa

Abstract: Sign language has played a crucial role in the lives of impaired people having hearing and speaking disabilities. They can send messages via hand gesture movement. Arabic Sign Language (ASL) recognition is a very difficult task because of its high complexity and the increasing intraclass similarity. Sign language may be utilized for the communication of sentences, letters, or words using diverse signs of the hands. Such communication helps to bridge the communication gap between people with hearing impairment and other people and also makes it easy for people with hearing impairment to express their opinions. Recently, a large number of studies have been ongoing in developing a system that is capable of classifying signs of dissimilar sign languages into the given class. Therefore, this study designs an atom search optimization with a deep convolutional autoencoder-enabled sign language recognition (ASODCAE-SLR) model for speaking and hearing disabled persons. The presented ASODCAE-SLR technique mainly aims to assist the communication of speaking and hearing disabled persons via the SLR process. To accomplish this, the ASODCAE-SLR technique initially pre-processes the input frames by a weighted average filtering approach. In addition, the ASODCAE-SLR technique employs a capsule network (CapsNet) feature extractor to produce a collection of feature vectors. For the recognition of sign language, the DCAE model is exploited in the study. At the final stage, the ASO algorithm is utilized as a hyperparameter optimizer which in turn increases the efficacy of the DCAE model. The experimental validation of the ASODCAE-SLR model is tested using the Arabic Sign Language dataset. The simulation analysis exhibit the enhanced performance of the ASODCAE-SLR model compared to existing models.

Keywords: quality of life; disabled persons; sign language recognition; deep learning; atom search optimization



Citation: Marzouk, R.; Alrowais, F.; Al-Wesabi, F.N.; Hilal, A.M. Atom Search Optimization with Deep Learning Enabled Arabic Sign Language Recognition for Speaking and Hearing Disability Persons. *Healthcare* **2022**, *10*, 1606. <https://doi.org/10.3390/healthcare10091606>

Academic Editor: Daniele Giansanti

Received: 27 July 2022

Accepted: 22 August 2022

Published: 24 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Communication is the major component of interpersonal relationships that acts as an important connection between individuals and describes human existence. Additionally, it is a prominent basis to promote the growth of the human population. Communication is classified into verbal and nonverbal forms, and its core is to exchange data between the sender and the receiver [1]. As the component of communication, verbal and nonverbal forms are considered spontaneous and disguised spontaneous communications, the initial one is demonstrated as an intentional communication from the motivation emotional state, and the last one is demonstrated as an instinctive intentional strategic

operation [2]. Communication is an indispensable tool in the existence of human beings. It is an effective and fundamental method of sharing opinions, thoughts, and feelings. However, a considerable fraction of the world population lacks this capability [3]. A lot of people suffer from speaking impairment, hearing loss, or both. A complete or partial disability to hearing in one or both ears is called hearing loss. In contrast, being mute is an inability that impairs speaking and makes people have difficulty speaking [4]. During childhood, if deaf–mute happens, their language learning capability could be hindered and leads to language impairment, otherwise called hearing mutism.

Sign language (SL) is the major adaptation for people with hearing and speech disabilities. Additionally, it is called a visual language. In general, it contains five key characteristics: orientation, hand shape, location, movement, and components such as eye-brow movements and mouth shape [5]. Studies had been carried out on voice generation using smart gloves that may provide a voice to SL movement. However, those who do not know SL generally reject or undervalue persons with disability due to the lack of proper communication among themselves. The procedure of translating the gestures and signs portrayed by the user into text is called SL detection [6]. It links the communication gap between the general public and people who could not speak. Image processing algorithms and neural networks are used for mapping the gesture to proper text in the training dataset and thus raw videos or images are transformed into relevant text that cannot be understood and read [7].

Existing models have used statistical approaches and machine learning (ML) models for SL recognition. The ML models are based on handcrafted features which could not determine insignificant regions in every frame, and the existence of temporal misalignment makes it difficult for traditional approaches to determine robust features. The derived features encode temporal dependency among frames, position, and orientation of hands, face, etc. Background noise and varying lighting conditions also result in occlusions and clutter, which have to be considered. They are tedious to extract with traditional ML models. Therefore, in this study, we have proposed a deep learning (DL)-based model as a solution for SL recognition. The primary objective of the DL technique is automated feature engineering. The concept behind this is to learn a set of features automatically from raw information that is beneficial during SL detection [8]. In such a way, it prevents the manual method of hand-crafted features by automatically learning as a set of features. With the emergence of the DL method, end to end model has been constructed for numerous challenges that only need the image as input [9]. Lately, a large number of studies have been ongoing in developing a system that is capable of classifying signs of various SLs in the class. This system has found application in natural language communications, games virtual reality environments, and robot controls [10].

This study designs an atom search optimization with a deep convolutional autoencoder-enabled Arabic Sign Language recognition (ASODCAE-SLR) model for speaking and hearing disabled persons. The presented ASODCAE-SLR technique preprocesses the input frames by a weighted average filtering approach. In addition, the ASODCAE-SLR technique employs a capsule network (CapsNet) feature extractor. For the recognition of sign language, the DCAE model is exploited in the study. At the final stage, the ASO algorithm is utilized as a hyperparameter optimizer which in turn increases the efficacy of the DCAE model. The experimental validation of the ASODCAE-SLR model is tested using the Arabic Sign Language dataset.

2. Literature Review

Sruthi and Lijiya [11] proposed a signer-independent DL-based method to build a sign language (SL) static alphabet detection technique. Now, the study examines many prevailing models in SL detection and carries out a CNN structure for ISL static alphabet detection from the binary silhouette of the signer hand area. Wen et al. [12] developed an AI-assisted SL detection and transmission method encompassing a virtual reality interface, sensing gloves, and DL block. Segmentation and non-segmentation enabled DL

algorithm to accomplish the detection of 20 sentences and 50 words. The segmentation technique classifies whole sentence signals into word units. Next, the DL method identifies each word element and reversely recognizes and reconstructs sentences. Khan et al. [13] aimed to illustrate a user-friendly method for Bangla SL for converting text via CNN and personalized ROI segmentation. By utilizing the ROI selection approach, the technique illustrates improved performance when compared to traditional methodologies.

In [14], the authors developed an SL fingerspelling alphabet detection technique with an image processing technique, supervised deep learning, and machine learning. Especially, twenty-four alphabetical symbols are developed by different integrations of static gestures (not including two motion gestures Z and J). Local binary pattern (LBP) and histogram of oriented gradients (HOG) features of every gesture would be extracted from the training image. Next, the multi-class support vector machine (SVM) is employed for training the extracted dataset. Mannan et al. [15] applied a deep convolution neural network for ASL alphabet detection to resolve ASL detection problems. The study proposed an ASL detection technique with a DCNN. The efficiency of the DCNN method enhances the quantity of the given dataset; for these purposes, we employed the data augmentation method for expanding the size of the trained dataset from the current dataset.

Sharma et al. [16] introduce a DCNN method for recognizing different symbols in ISL, which belongs to thirty-five classes. Such classes comprise cropped images of hand gestures. Different from other feature selection-based models, DCNN has the benefit of automated feature extraction in the training. It is named end-to-end learning. A lightweight transfer learning (TL) structure makes the model training faster which provides 100% accuracy. Furthermore, a web-based method was proposed which could simply decode the symbol. In [17], a proposed novel architecture for signer-independent SL detection with different DL architectures encompassing DRNN, hand semantic segmentation, and hand shape feature representation. Extracting hand shape features can be accomplished by a single-layer convolution self-organizing map (CSOM) rather than depending on the TL of pretrained DCNN. Then, the series of extracted feature vectors are identified by utilizing deep BiLSTM-RNN.

Though several ML and DL models for sign language recognition are available in the literature, it is still needed to enhance classification performance. Owing to the continual deepening of the model, the number of parameters of DL models also increases quickly which results in model overfitting. Since the trial and error method for hyperparameter tuning is a tedious and erroneous process, metaheuristic algorithms can be applied. Therefore, in this work, we employ the ASO algorithm for the parameter selection of the DCAE model.

3. The Proposed Model

In this study, a new ASODCAE-SLR technique has been developed for recognizing sign languages to assist the communication of speaking and hearing disabled persons. The ASODCAE-SLR technique initially pre-processes the input frames by a weighted average filtering approach. Next, the ASODCAE-SLR technique employed a CapsNet feature extractor to produce a collection of feature vectors. To identify and classify sign language, ASO with the DCAE model is exploited in the study.

3.1. Image Pre-Processing

The ASODCAE-SLR technique initially pre-processes the input frames by a weighted average filtering approach. The weighted average filter was planned to pre-process that suppresses noise and improves spatial domain features efficiently [9]. This filter W^η was determined as a matrix, whereas η refers to the odd number. All the element values of the matrix were defined as the distance between the present place and the center of the matrix, as demonstrated in Equation (1). The center of the matrix was determined as

$w_{(\eta+1)/2,(\eta+1)/2} = 2/\eta^2$. The presented filter continues edges but suppresses speckle noise related to another filter namely the mean filter and maintains the continuity of images.

$$w_{ij} = \frac{1}{\eta^2 \sqrt{(\frac{\eta+1}{2} - i)^2 + (\frac{\eta+1}{2} - j)^2}}; i = 1, 2, \dots, \eta; j = 1, 2, \dots, \eta; \tag{1}$$

whereas $I_1, I_2 \in R^{N_r \times N_c}$, the convolutional of all the images utilizing W^η is obtained for acquiring 2 images $I_1^w(\eta) = I_1 * W^\eta$ and $I_2^w(\eta) = I_2 * W^\eta$, whereas $*$ signifies the 2D convolutional function.

3.2. Feature Extraction: CapsNet Model

Next to pre-processing, the ASODCAE-SLR technique employs the CapsNet model to generate feature vectors. A major benefit of CapsNet is that they hold the characteristics of more concrete features which can be interpreted to understand what and how is the network learning. The CapsNet has the ability for encoding spatial data and differentiate among several poses, textures, and orientations [18]. The capsule is a set of neurons, thus all the capsules have an activity vector connected to it that captures several instantiation parameters to recognition of a certain kind of object or its part. The length and orientation of the vector present the probability or possibility of the presence of that object and its generalization pose. These vectors were passed on to the upper-level capsules in lower-layer capsules. The coupling coefficients occur among these layers of capsules. When the forecast by the lower-level capsule equals the outcome of current capsules, the value of coupling coefficient amongst them improves, calculated with utilize of softmax function. Specifically, when the present capsule identifies a tight cluster of preceding prediction, strongly representing the occurrence of that object, its outcomes in a higher probability is also recognized as routing by agreement. Figure 1 depicts the framework of the CapsNet method.

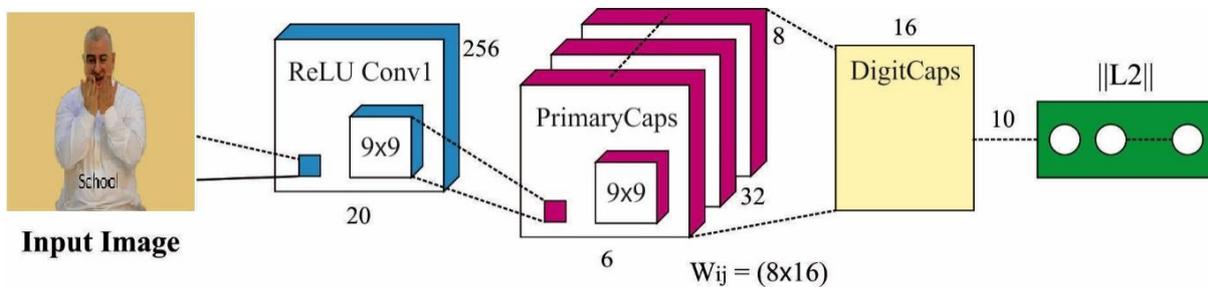


Figure 1. Structure of CapsNet model.

Initially, the prediction vector (Equation (2)) was calculated as:

$$\hat{u}_{j_i} \wedge = W_{ij} u_i, \tag{2}$$

whereas \hat{u}_{j_i} refers to the outcome of the forecast vector of upper-level j th capsule, W_{ij} and u_i implies the weighted matrix and forecast vector of capsules i from the lower layer correspondingly. It can capture spatial connections and interactions among sub-objects and objects. In Equation (3), dependent upon the degree of agreement amongst neighboring layer capsules, the coupling coefficients were calculated using the softmax function,

$$c_{ij} = \exp(b_{ij}) / \sum \exp(b_{ik}), \tag{3}$$

In which b_{ij} signifies the *log* probability amongst two capsules, initialization to zero, and k represents the number of capsules. The input vector s_j to j th layer capsule that a weighted sum of $\hat{u}_{j_i} \wedge$ vectors learned by routing technique is computed as:

$$s_j = \sum_i c_{ij} \hat{u}_{j_i} , \quad (4)$$

Lastly, a squashing function that integrates squashing and unit scaling (Equation (5)) was executed for confining the value of results from the range amongst *zero* and one, therefore calculating the probability as,

$$\|s_j\|^2 v_j = \frac{1 + \|s_j\|^2 \frac{s_j}{\|s_j\|}}{\|s_j\|} , \quad (5)$$

The loss function (as calculated by Equation (6)) was connected to capsules from the final layer, whereas m^+ and m^- are fixed to 0.9 and 0.1 resp.

$$l_k = T_k \max(0, m^+ - \|v_k\|)^2 + \lambda(1 - T_k) \max(0, \|v_k\| - m^-)^2, \quad (6)$$

whereas the value T_k is 1 for correct labels and 0 else, λ refers to the constant whose value is 0.5. The 1st term is calculated to correct labels, and the second term calculates to incorrect labels. If T_k will be 1, the second term develops 0, and for T_k as 0, the first term develops 0. Similarly, the loss value l_k is 0 for correct forecasts with v_k being superior to 0.9 and non-zero otherwise.

3.3. Sign Language Recognition: DCAE Model

To identify and classify sign language, the DCAE model is exploited in the study. AE is a conventional DNN structure that makes use of its input as a label. Later, the network attempts to recreate its input in the learning mechanism [19]; for these purposes, it generates and automatically extracts the representation feature in suitable time iterations. This kind of network is created by stacking deep layers in AE forms consisting of two major parts of decoder and encoder. DCAE is a kind of AE applying a convolution layer to determine the inner data of an image. In CAE, structure weight is shared amongst each location within every feature map, thereby reducing parameter redundancy and preserving the spatial locality. For extracting deep features, consider D , W , and H as the depth, width, and height of the dataset, correspondingly, and n refers to the pixel count. For every member of the X set, the image patches with the size $7 \times 7 \times D$ are extracted, where χ_j denotes the central pixel. Consequently, the X set is characterized as an image patch, every patch, x_i^* , is given into the encoder blocks. For an input x_i^* , the hidden layer mapping of k th feature map is shown below:

$$h^k = \sigma(x_i^* * W^k + b^k) \quad (7)$$

In Equation (7), b refers to the bias; σ denotes an activation function, and the symbol $*$ corresponds to the 2D convolution layer and it is attained by the following expression:

$$y = \sigma\left(\sum_{k \in H} h^k * \tilde{W}^k + \tilde{b}^k\right) \quad (8)$$

In Equation (8), there exists bias \tilde{b} for every input channel, and h denotes the set of latent feature maps. The \tilde{W} corresponding to the flip operation over both dimensions of weight W . y denotes the prediction value. In order to define the parameter vector

depicting the complete DCAE architecture, one could minimize the subsequent cost function signified as follows:

$$E(\theta) = \frac{1}{n} \sum_{i=1}^n \|x_i^* - y_i\|_2^2 \quad (9)$$

For minimizing this function, we need to evaluate the gradient of cost function concerning the convolutional kernel (W, \tilde{W}) and bias (b, \tilde{b}) parameter:

$$\frac{\partial E(\theta)}{\partial W^k} = x^* * \delta h^k + h^k * \delta y \quad (10)$$

$$\frac{\partial E(\theta)}{\partial b^k} = \delta h^k + \delta y \quad (11)$$

Now, δh and δy denote the deltas of the hidden state and the reconstruction, correspondingly. Then, the weight is upgraded by the optimization methodology. At last, the DCAE parameter is evaluated when the loss function convergence is accomplished. The output feature map of the encoder block is regarded as a deep feature. In the study, batch normalization (BN) was employed for tackling the internal covariant shift phenomenon and enhancing the efficiency of the network via the normalization of input layers by re-centering and rescaling. The BN assists to increase accuracy and learn faster.

3.4. Hyperparameter Tuning: ASO Algorithm

In this study, the ASO algorithm is exploited to finely adjust the hyperparameter values related to the DCAE model. The molecular dynamics simulate the mathematical process of the ASO technique. In ASO, the place of all the atoms from the searching space that is affected by their mass signifies the solutions [20]. ASO begins the optimization by creating a group of arbitrary particles from N -dimensional space. Afterward, the solution of all the atoms was estimated as dependent upon the main function. Atoms upgrade their place and velocity from all the iterations, and the place of the optimum atom was upgraded from all the iterations. The velocity of particles is a function of their acceleration, and the acceleration of atoms is estimated based on Newton's second law dependent upon the ratio of forces executed to the mass of particles. The mass of i^{th} atom from the iteration of t , $m_i(t)$ was computed by the subsequent formulas:

$$M_i(t) = e^{\frac{Fit_i(t) - Fit_{Best}(t)}{Fit_{Best}(t) - Fit_{Worst}}} \quad (12)$$

$$m_i(t) = \frac{M_i(t)}{\sum_{j=1}^N M_j(t)} \quad (13)$$

whereas $Fit_{Best}(t)$ and Fit_{Worst} signifies atoms with optimum and worse values from the t^{th} iteration and $Fit_i(t)$ implies the value of i^{th} atom main function from the T^{th} iteration, correspondingly. Regarding the minimize problems, Fit_{Best} and Fit_{Worst} were assumed dependent upon the subsequent connections:

$$Fit_{Best}(t) = \min (Fit_i(t)), i \in \{1, 2, \dots, N\} \quad (14)$$

$$Fit_{Worst}(t) = \max (Fit_i(t)), i \in \{1, 2, \dots, N\} \quad (15)$$

During all the periods, the count of neighbors of all the atoms that interact is defined utilizing Equation (16):

$$K(t) = N - (N - 2) \times \left[\frac{T}{T} \right] \quad (16)$$

In which T defines the entire amount of iterations of the technique, or in another word, the life of systems. As is noted, the parameter K is a function of time, slowly reducing

the iterations. The forces executed on all the particles contain two kinds of interaction forces and internal constraint forces. The interaction force that is determined utilizing the Lennard–Jones potential method and the internal constraint force that is connected to the bond length potential and differs depending upon the distance amongst all the atoms to optimum atoms were computed utilizing Equations (17) and (18), correspondingly.

$$F_i^d(t) = \sum_{j \in K_{Best}} rand_j F_{ij}(t)^d \tag{17}$$

$$F_{ij}(t) = -\alpha \left(1 - \frac{t-1}{T}\right)^3 e^{-\frac{20t}{T}} [2(h_{ij}(t))^{13} - (h_{ij}(t))^7]$$

$$G_i^d(t) = -\lambda(t) (x_{best}^d(t) - x_i^d(t)), \lambda(t) = \beta e\left(-\frac{20T}{T}\right) \tag{18}$$

whereas F and G define the communication and internal constrain forces correspondingly, $rand_j$ depicts an arbitrary number amongst 0 and 1, and K_{Best} refers to the subset of the atom population containing K atoms with optimum main function values. Additionally, $x_{best}^d(t)$ demonstrates the place of an optimum atom from the t^{th} iteration from the d dimensional space, $\lambda(t)$ illustrates the Lagrangian coefficient, α stands for the depth coefficient, and β implies the weighted coefficient. Figure 2 illustrates the flowchart of the ASO technique.

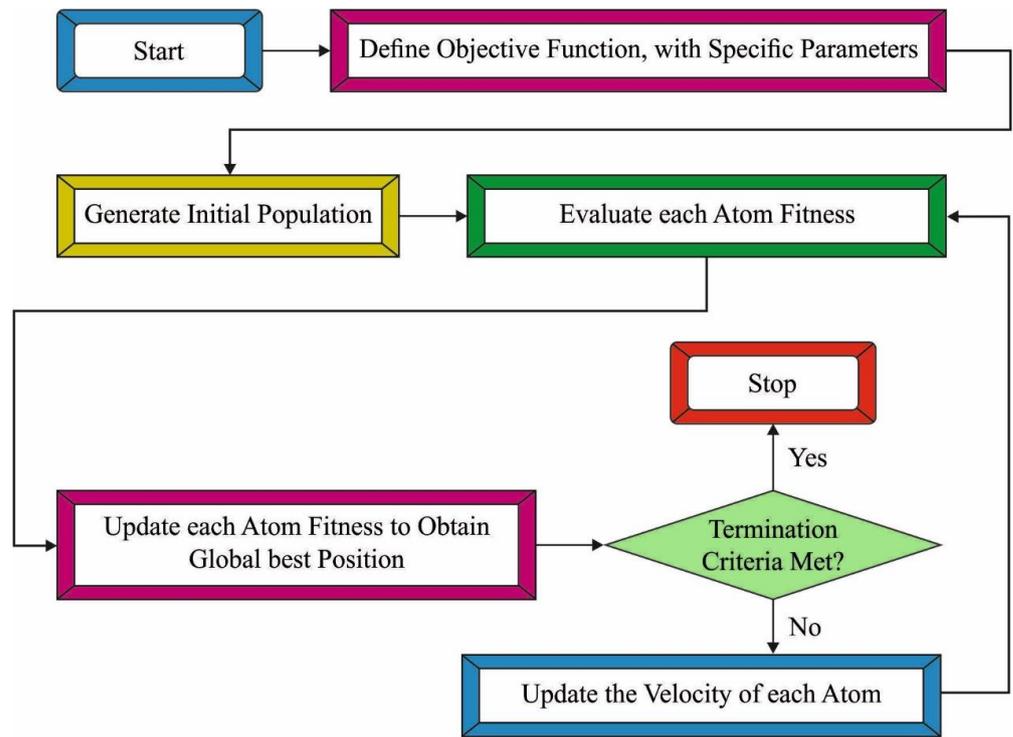


Figure 2. Flowchart of ASO technique.

As follows, the acceleration of i particle from the dimensional d and period τ was computed in Equation (19):

$$a_i^d(t) = \frac{F_i^d(t)}{m_i^d(t)} + \frac{G_i^d(t)}{m_i^d(t)} =$$

$$-\alpha \left(1 \cdot e^{-\frac{20T}{T}}\right) \times \sum_{j \in K_{best}} \frac{r_j [2 \times ((h_{ij}(t))^{13} - h_{ij}(t))^7]}{m_i(t)} \frac{(X_j^d(t) - X_i^d(t))}{\|X_i(t), X_j(t)\|_2}$$

$$+ \frac{\beta e^{-\frac{20T}{T}} (X_{best}^d(t) - X_i^d(t))}{m_i(t)} \tag{19}$$

The last step in all the iterations is for updating the particle velocity and location that is achieved in the subsequent formulas:

$$v_i^d(t+1) = rand_i^d v_i^d(t) + a_i^d(t) \quad (20)$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \quad (21)$$

Every update and compute were carried out constantly still the termination condition is met. Lastly, the location and value of the main function of an optimum atom were assumed as the optimum estimate of problems.

The ASO approach derives a fitness function to achieve an enhanced performance of the classification. It defines a positive integer to signify the performance of the candidate solution. The minimization of the classification error rate is considered as the fitness function in this study as follows.

$$\begin{aligned} fitness(x_i) &= ClassifierErrorRate(x_i) \\ &= \frac{\text{number of misclassified samples}}{\text{Total number of samples}} * 100 \end{aligned} \quad (22)$$

4. Result Analysis

The proposed model is simulated using Python 3.6.5 tool on PC i5-8600k, GeForce 1050Ti 4GB, 16GB RAM, 250GB SSD, and 1TB HDD. The parameter settings are given as follows: learning rate: 0.01, dropout: 0.5, batch size: 5, epoch count: 50, and activation: ReLU. This section inspects the sign language recognition outcomes of the ASODCAE-SLR model using the Arabic Sign Language dataset. In this study, a total of 1100 samples under 11 class labels are used. Table 1 depicts the detailed description of the dataset.

Table 1. Dataset details.

Label	Arabic Word	Meaning	No. of Samples
1	صديق	Friend	100
2	جار	Neighbor	100
3	ضيف	Guest	100
4	هدية	Gift	100
5	عدو	Enemy	100
6	يشم	To Smell	100
7	يساعد	To Help	100
8	شكرا	Thank You	100
9	تفضل	Come in	100
10	عيب	Shame	100
11	بيت	House	100
Total Number of Samples			1100

The confusion matrix generated by the ASODCAE-SLR model on the entire dataset is demonstrated in Figure 3. The figure depicted that the ASODCAE-SLR model has accurately recognized all the 11 class labels on the entire dataset.

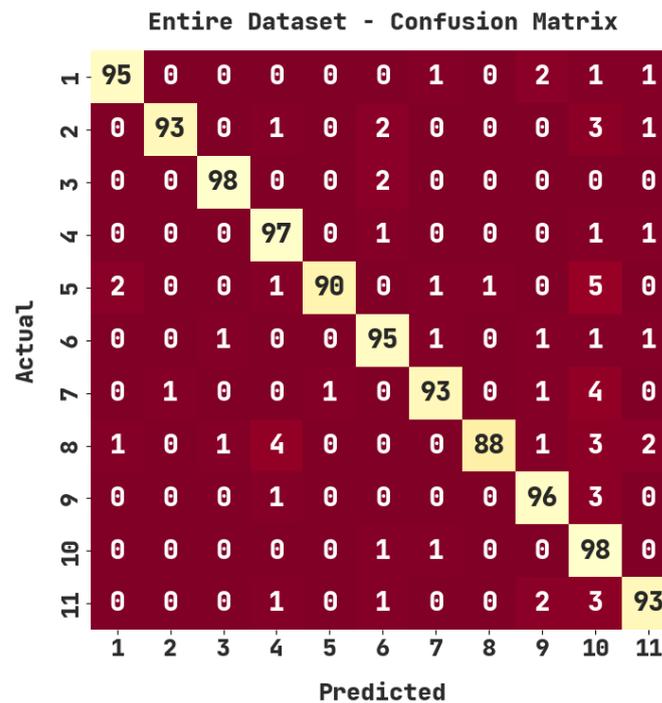


Figure 3. Confusion matrix of ASODCAE-SLR approach under entire dataset.

Table 2 report the sign language recognition outcomes of the ASODCAE-SLR model on the entire dataset. The ASODCAE-SLR model has recognized samples under class 1 with $accu_y$, $prec_n$, $reca_l$, $F1_{score}$, and $Jaccard\ index$ of 99.27%, 96.94%, 95%, 95.96%, and 99.23%. Additionally, the ASODCAE-SLR system has recognized samples under class 2 with $accu_y$, $prec_n$, $reca_l$, $F1_{score}$, and $Jaccard\ index$ of 99.27%, 96.94%, 93%, 95.88%, and 92.08%. In line with this, the ASODCAE-SLR method has recognized samples under class 3 with $accu_y$, $prec_n$, $reca_l$, $F1_{score}$, and $Jaccard\ index$ of 99.64%, 98%, 98%, 98%, and 96.08%. Next, the ASODCAE-SLR system has recognized samples under class 4 with $accu_y$, $prec_n$, $reca_l$, $F1_{score}$, and $Jaccard\ index$ of 99%, 92.38%, 97%, 94.63%, and 89.81%.

Table 2. Result analysis of ASODCAE-SLR approach with distinct class labels under entire dataset.

Entire Dataset					
Labels	Accuracy	Precision	Recall	F1-Score	Jaccard Index
1	99.27	96.94	95.00	95.96	92.23
2	99.27	98.94	93.00	95.88	92.08
3	99.64	98.00	98.00	98.00	96.08
4	99.00	92.38	97.00	94.63	89.81
5	99.00	98.90	90.00	94.24	89.11
6	98.91	93.14	95.00	94.06	88.79
7	99.00	95.88	93.00	94.42	89.42
8	98.82	98.88	88.00	93.12	87.13
9	99.00	93.20	96.00	94.58	89.72
10	97.64	80.33	98.00	88.29	79.03
11	98.82	93.94	93.00	93.47	87.74
Average	98.94	94.59	94.18	94.24	89.19

The confusion matrix generated by the ASODCAE-SLR approach on 70% of training (TR) data is displayed in Figure 4. The figure depicted that the ASODCAE-SLR model has accurately recognized all the 11 class labels on 70% of TR data.

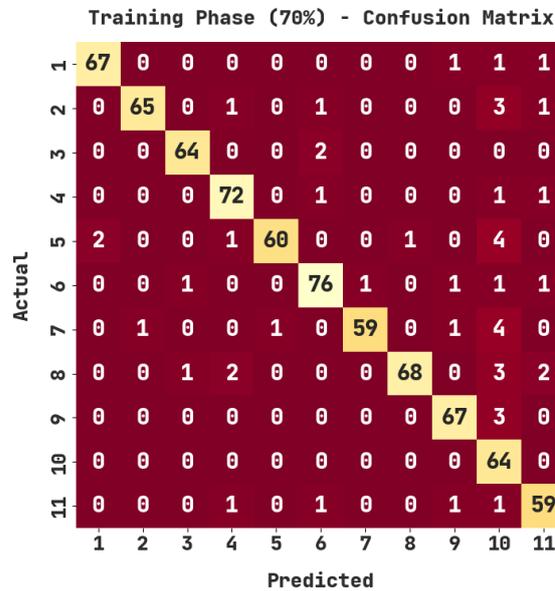


Figure 4. Confusion matrix of ASODCAE-SLR approach under 70% of TR data.

Table 3 illustrates the sign language recognition outcomes of the ASODCAE-SLR methodology on 70% of TR data. The ASODCAE-SLR technique has recognized samples under class 1 with $accu_y$, $prec_n$, $reca_l$, $F1_{score}$, and $Jaccard\ index$ of 99.35%, 97.10%, 95.71%, 96.40%, and 93.06%. Additionally, the ASODCAE-SLR algorithm has recognized samples under class 2 with $accu_y$, $prec_n$, $reca_l$, $F1_{score}$, and $Jaccard\ index$ of 99.09%, 98.48%, 91.55%, 94.89%, and 90.28%. Similarly, the ASODCAE-SLR approach has recognized samples under class 3 with $accu_y$, $prec_n$, $reca_l$, $F1_{score}$, and $Jaccard\ index$ of 99.48%, 96.97%, 96.97%, 96.97%, and 94.12%. At last, the ASODCAE-SLR system has recognized samples under class 4 with $accu_y$, $prec_n$, $reca_l$, $F1_{score}$, and $Jaccard\ index$ of 98.96%, 93.51%, 96%, 94.74%, and 90%.

Table 3. Result analysis of ASODCAE-SLR approach with distinct class labels under 70% of TR data.

Training Phase (70%)					
Labels	Accuracy	Precision	Recall	F1-Score	Jaccard Index
1	99.35	97.10	95.71	96.40	93.06
2	99.09	98.48	91.55	94.89	90.28
3	99.48	96.97	96.97	96.97	94.12
4	98.96	93.51	96.00	94.74	90.00
5	98.83	98.36	88.24	93.02	86.96
6	98.70	93.83	93.83	93.83	88.37
7	98.96	98.33	89.39	93.65	88.06
8	98.83	98.55	89.47	93.79	88.31
9	99.09	94.37	95.71	95.04	90.54
10	97.27	75.29	100.00	85.91	75.29
11	98.70	90.77	93.65	92.19	85.51
Average	98.84	94.14	93.68	93.67	88.23

The confusion matrix generated by the ASODCAE-SLR approach on 30% of testing (TS) data is represented in Figure 5. The figure depicted that the ASODCAE-SLR technique has accurately recognized all the 11 class labels on 30% of the TS dataset.

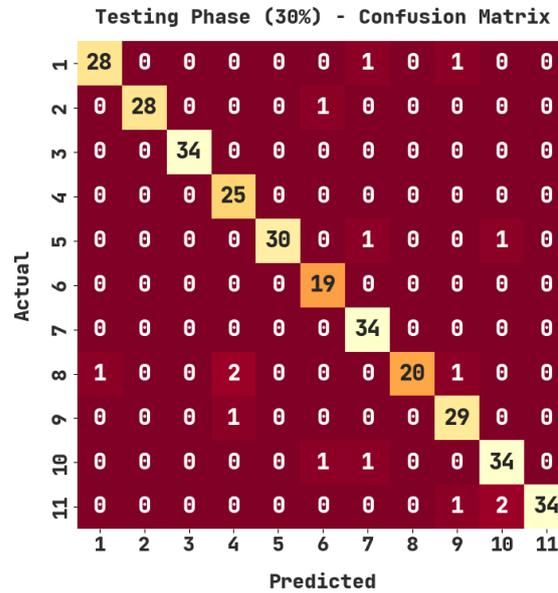


Figure 5. Confusion matrix of ASODCAE-SLR approach under 30% of TS data.

Table 4 demonstrates the sign language recognition outcomes of the ASODCAE-SLR technique on 30% of TS data. The ASODCAE-SLR approach has recognized samples under class 1 with $accu_y$, $prec_n$, $reca_l$, $F1_{score}$, and $Jaccard\ index$ of 99.09%, 96.55%, 93.33%, 94.92%, and 90.32%. Furthermore, the ASODCAE-SLR methodology has recognized samples under class 2 with $accu_y$, $prec_n$, $reca_l$, $F1_{score}$, and $Jaccard\ index$ of 99.70%, 100%, 96.55%, 98.25%, and 96.55%. In addition, the ASODCAE-SLR system has recognized samples under class 3 with $accu_y$, $prec_n$, $reca_l$, $F1_{score}$, and $Jaccard\ index$ of 100%, 100%, 100%, 100%, and 100%. Afterward, the ASODCAE-SLR system recognized samples under class 4 with $accu_y$, $prec_n$, $reca_l$, $F1_{score}$, and $Jaccard\ index$ of 99.09%, 89.29%, 100%, 94.34%, and 89.29%.

Table 4. Result analysis of ASODCAE-SLR approach with distinct class labels under 30% of TS data.

Testing Phase (30%)					
Labels	Accuracy	Precision	Recall	F1-Score	Jaccard Index
1	99.09	96.55	93.33	94.92	90.32
2	99.70	100.00	96.55	98.25	96.55
3	100.00	100.00	100.00	100.00	100.00
4	99.09	89.29	100.00	94.34	89.29
5	99.39	100.00	93.75	96.77	93.75
6	99.39	90.48	100.00	95.00	90.48
7	99.09	91.89	100.00	95.77	91.89
8	98.79	100.00	83.33	90.91	83.33
9	98.79	90.62	96.67	93.55	87.88
10	98.48	91.89	94.44	93.15	87.18
11	99.09	100.00	91.89	95.77	91.89
Average	99.17	95.52	95.45	95.31	91.14

The training accuracy (TRA) and validation accuracy (VLA) acquired by the ASODCAE-SLR approach on the test dataset is shown in Figure 6. The experimental result stated that the ASODCAE-SLR technique has achieved improved values of TRA and VLA. Particularly the VLA seemed greater than TRA.

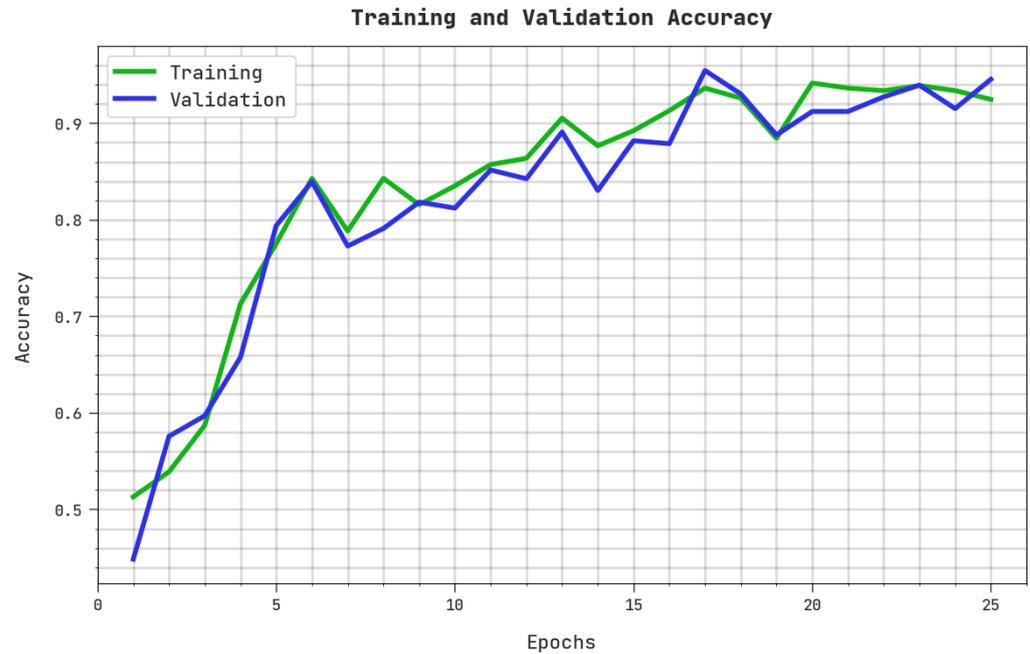


Figure 6. TRA and VLA analysis ASODCAE-SLR approach.

The training loss (TRL) and validation loss (VLL) accomplished by the ASODCAE-SLR system on the test dataset are depicted in Figure 7. The experimental result revealed that the ASODCAE-SLR approach has obtained minimal values of TRL and VLL. Certainly, the VLL is lesser than TRL.



Figure 7. TRL and VLL analysis ASODCAE-SLR approach.

A clear precision–recall examination of the ASODCAE-SLR algorithm on the test dataset is illustrated in Figure 8. The figure represented that the ASODCAE-SLR approach has resulted in higher values of precision—recall values under all classes.

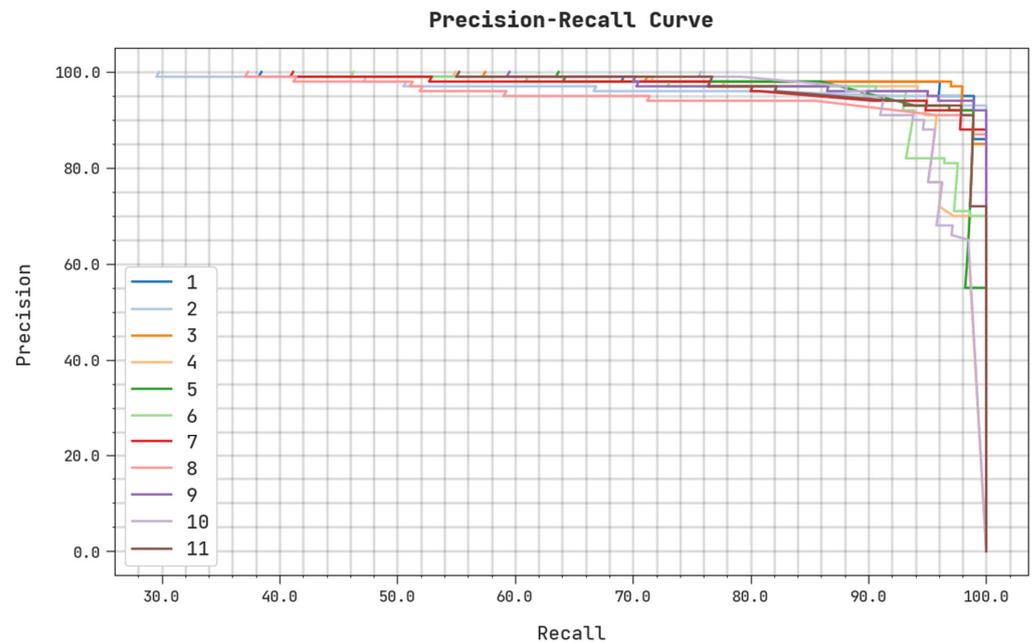


Figure 8. Precision-recall analysis ASODCAE-SLR approach.

A detailed ROC analysis of the ASODCAE-SLR system on the test dataset is illustrated in Figure 9. The outcomes represented by the ASODCAE-SLR algorithm have demonstrated their capability in categorizing distinct classes on the test dataset.

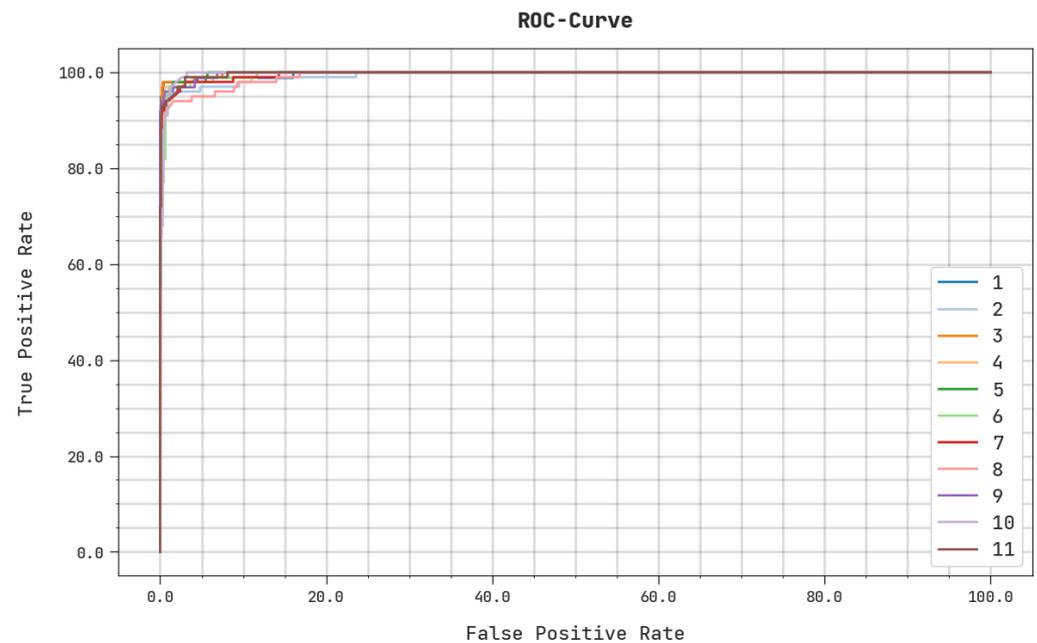


Figure 9. ROC analysis ASODCAE-SLR approach.

At last, comprehensive comparative results of the ASODCAE-SLR model with recent models are given in Figure 10 [21]. The figure indicated that the GRU-LSTM model has attained reduced classification results compared to existing techniques.

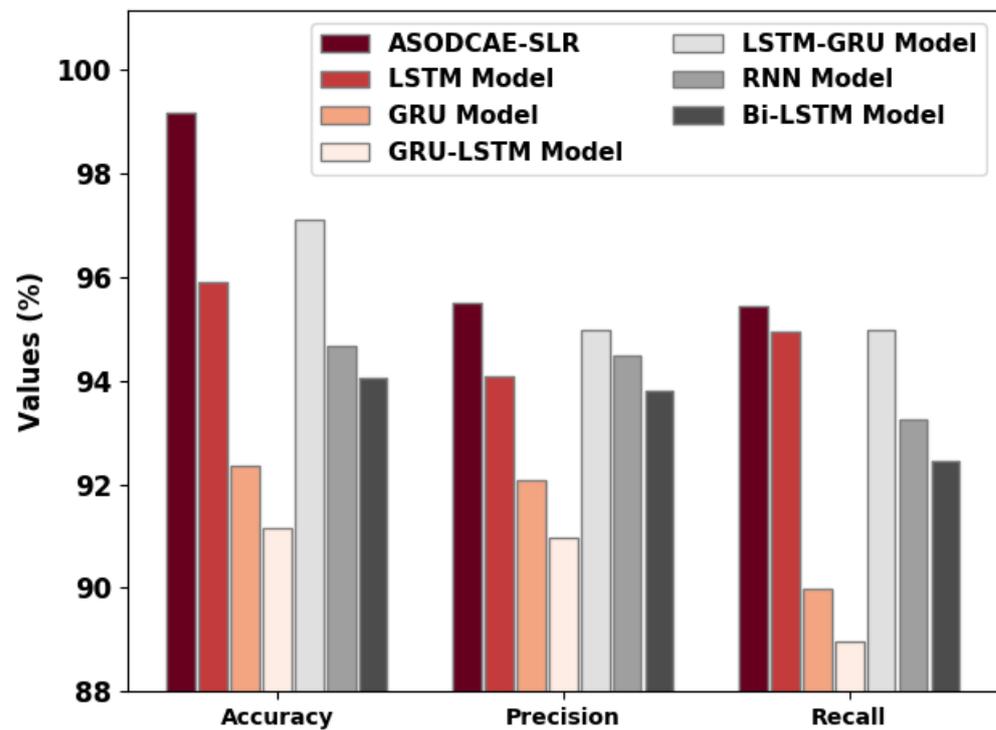


Figure 10. Comparative analysis of ASODCAE-SLR approach with existing algorithms.

Next, the GRU, RNN, and BiLSTM models have reported slightly enhanced classification performance whereas the LSTM model has shown reasonable classification performance. Moreover, the LSTM-GRU model has accomplished near-optimal performance. However, the obtained values implied that the ASODCAE-SLR model has accomplished improved performance over other models.

5. Conclusions

In this study, a new ASODCAE-SLR technique has been developed for recognizing sign languages to assist the communication of speaking and hearing disabled persons. The ASODCAE-SLR technique initially pre-processes the input frames by a weighted average filtering approach. Next, the ASODCAE-SLR technique employed a CapsNet feature extractor to produce a collection of feature vectors. To identify and classify sign language, the DCAE model is exploited in the study. At the final stage, the ASO algorithm is utilized as a hyperparameter optimizer which in turn increases the efficacy of the DCAE model. The experimental validation of the ASODCAE-SLR model is tested using the Arabic Sign Language dataset. The simulation analysis exhibit the enhanced performance of the ASODCAE-SLR model compared to existing models. Therefore, the proposed model can be employed to assist communication between deaf and dumb people with ordinary people. The proposed model can be extended to sign board recognition in real-time applications. In the future, the performance of the proposed model can be tested on a real-time large-scale dataset. In addition, a fusion of DL models can be derived to boost the SL recognition performance.

Author Contributions: Conceptualization, R.M. and F.A.; methodology, F.N.A.-W.; software, A.M.H.; validation, F.A., F.N.A.-W. and R.M.; formal analysis, A.M.H.; investigation, R.M.; resources, A.M.H.; data curation, A.M.H.; writing—original draft preparation, R.M., F.A. and F.N.A.-W.; writing—review and editing, A.M.H. and F.N.A.-W.; visualization, A.M.H.; supervision, F.N.A.-W.; project administration, F.A.; funding acquisition, R.M. All authors have read and agreed to the published version of the manuscript.

Funding: The authors extend their appreciation to the King Salman center for Disability Research for funding this work through Research Group no KSRG-2022-017.

Data Availability Statement: Data sharing is not applicable to this article as no datasets were generated during the current study.

Conflicts of Interest: The authors declare that they have no conflict of interest. The manuscript was written with the contribution of all authors. All authors have approved the final version of the manuscript.

References

1. Kamruzzaman, M.M. Arabic Sign Language Recognition and Generating Arabic Speech Using Convolutional Neural Network. *Wirel. Commun. Mob. Comput.* **2020**, *2020*, 1–9. [[CrossRef](#)]
2. Boukdir, A.; Benaddy, M.; Ellahyani, A.; El Meslouhi, O.; Kardouchi, M. Isolated Video-Based Arabic Sign Language Recognition Using Convolutional and Recursive Neural Networks. *Arab. J. Sci. Eng.* **2021**, *47*, 2187–2199. [[CrossRef](#)]
3. Batnasan, G.; Gochoo, M.; Otgonbold, M.E.; Alnajjar, F.; Shih, T.K. ArSL21L: Arabic Sign Language Letter Dataset Benchmarking and an Educational Avatar for Metaverse Applications. In Proceedings of the 2022 IEEE Global Engineering Education Conference (EDUCON), Gammarth, Tunisia, 28–31 March 2022; IEEE: Piscataway, NJ, USA; pp. 1814–1821.
4. More, V.; Sangamnerkar, S.; Thakare, V.; Mane, D.; Dolas, R. Sign language recognition using image processing. *JournalNX* **2021**, 85–87.
5. AlKhuraym, B.Y.; Ismail, M.M.B.; Bchir, O. Arabic Sign Language Recognition Using Lightweight CNN-Based Architecture. *Int. J. Adv. Comput. Sci. Appl.* **2022**, *13*, 438. [[CrossRef](#)]
6. Tolentino, L.K.S.; Juan, R.O.S.; Thio-Ac, A.C.; Pamahoy, M.A.B.; Forteza, J.R.R.; Garcia, X.J.O. Static Sign Language Recognition Using Deep Learning. *Int. J. Mach. Learn. Comput.* **2019**, *9*, 821–827. [[CrossRef](#)]
7. Islalm, M.S.; Rahman, M.M.; Rahman, M.H.; Arifuzzaman, M.; Sassi, R.; Aktaruzzaman, M. Recognition bangla sign language using convolutional neural network. In Proceedings of the 2019 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), Sakhir, Bahrain, 22–23 September 2019; IEEE: Piscataway, NJ, USA; pp. 1–6.
8. Ismail, M.H.; Dawwd, S.A.; Ali, F.H. Dynamic hand gesture recognition of Arabic sign language by using deep convolutional neural networks. *Indones. J. Electr. Eng. Comput. Sci.* **2022**, *25*, 952–962. [[CrossRef](#)]
9. Song, Y.; Liu, J. An improved adaptive weighted median filter algorithm. *J. Phys. Conf. Ser.* **2019**, *1187*, 042107. [[CrossRef](#)]
10. Alsaadi, Z.; Alshamani, E.; Alrehaili, M.; Alrashdi, A.A.D.; Albelwi, S.; Elfaki, A.O. A Real Time Arabic Sign Language Alphabets (ArSLA) Recognition Model Using Deep Learning Architecture. *Computers* **2022**, *11*, 78. [[CrossRef](#)]
11. Sruthi, C.J.; Lijiya, A. Signet: A deep learning based indian sign language recognition system. In Proceedings of the 2019 International Conference on Communication and Signal Processing (ICCSP), Melmaruvathur, India, 4–6 April 2019; IEEE: Piscataway, NJ, USA; pp. 0596–0600.
12. Wen, F.; Zhang, Z.; He, T.; Lee, C. AI enabled sign language recognition and VR space bidirectional communication using triboelectric smart glove. *Nat. Commun.* **2021**, *12*, 1–13. [[CrossRef](#)] [[PubMed](#)]
13. Khan, S.A.; Joy, A.D.; Asaduzzaman, S.M.; Hossain, M. An efficient sign language translator device using convolutional neural network and customized ROI segmentation. In Proceedings of the 2019 2nd International Conference on Communication Engineering and Technology (ICCET), Nagoya, Japan, 12–15 April 2019; IEEE: Piscataway, NJ, USA; pp. 152–156.
14. Nguyen, H.B.; Do, H.N. Deep learning for american sign language fingerspelling recognition system. In Proceedings of the 2019 26th International Conference on Telecommunications (ICT), Hanoi, Vietnam, 8–10 April 2019; pp. 314–318.
15. Mannan, A.; Abbasi, A.; Javed, A.R.; Ahsan, A.; Gadekallu, T.R.; Xin, Q. Hypertuned Deep Convolutional Neural Network for Sign Language Recognition. *Comput. Intell. Neurosci.* **2022**, *2022*, 1–10. [[CrossRef](#)] [[PubMed](#)]
16. Sharma, C.M.; Tomar, K.; Mishra, R.K.; Chariar, V.M. Indian sign language recognition using fine-tuned deep transfer learning model. In Proceedings of the International Conference on Innovations in Computer and Information Science (ICICIS), Ganzhou, China, 15–16 September 2021; pp. 62–67.
17. Aly, S.; Aly, W. DeepArSLR: A Novel Signer-Independent Deep Learning Framework for Isolated Arabic Sign Language Gestures Recognition. *IEEE Access* **2020**, *8*, 83199–83212. [[CrossRef](#)]
18. Mazzia, V.; Salvetti, F.; Chiaberge, M. Efficient-capsnet: Capsule network with self-attention routing. *Sci. Rep.* **2021**, *11*, 1–13. [[CrossRef](#)] [[PubMed](#)]
19. Zheng, Q.; Zhao, P.; Zhang, D.; Wang, H. MR-DCAE: Manifold regularization-based deep convolutional autoencoder for unauthorized broadcasting identification. *Int. J. Intell. Syst.* **2021**, *36*, 7204–7238. [[CrossRef](#)]
20. Barshandeh, S.; Haghzadeh, M. A new hybrid chaotic atom search optimization based on tree-seed algorithm and Levy flight for solving optimization problems. *Eng. Comput.* **2020**, *37*, 3079–3122. [[CrossRef](#)]
21. Kothadiya, D.; Bhatt, C.; Sapariya, K.; Patel, K.; Gil-González, A.B.; Corchado, J.M. Deepsign: Sign Language Detection and Recognition Using Deep Learning. *Electronics* **2022**, *11*, 1780. [[CrossRef](#)]