Journal of Disability Research

2023 | Volume 2 | Issue 4 | Pages: 31–39
DOI: 10.57197/JDR-2023-0047

Sign Language Recognition Using Artificial Rabbits Optimizer with Siamese Neural Network for Persons with Disabilities

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 Abdul Rahman University, 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 Abdul Rahman University, Riyadh 11671, Saudi Arabia
[4]Department of Computer Science, College of Science and Arts at Mahayil, King Khalid University, Muhayil Aseer, Saudi Arabia
[5]Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam bin Abdulaziz University, AlKharj, Saudi Arabia

Correspondence to:
Radwa Marzouk*, e-mail: rmMarzouk@pnu.edu.sa

Received: May 21 2023; Revised: June 7 2023; Accepted: October 21 2023; Published Online: November 23 2023

ABSTRACT

Sign language recognition is an effective solution for individuals with disabilities to communicate with others. It helps to convey information using sign language. Recent advances in computer vision (CV) and image processing algorithms can be employed for effective sign detection and classification. As hyperparameters involved in Deep Learning (DL) algorithms considerably affect the classification results, metaheuristic optimization algorithms can be designed. In this aspect, this manuscript offers the design of Sign Language Recognition using Artificial Rabbits Optimizer with Siamese Neural Network (SLR-AROSNN) technique for persons with disabilities. The proposed SLR-AROSNN technique mainly focused on the recognition of multiple kinds of sign languages posed by disabled persons. The goal of the SLR-AROSNN technique lies in the effectual exploitation of CV, DL, and parameter tuning strategies. It employs the MobileNet model to derive feature vectors. For the identification and classification of sign languages, Siamese neural network is used. At the final stage, the SLR-AROSNN technique makes use of the ARO algorithm to get improved sign recognition results. To illustrate the improvement of the SLR-AROSNN technique, a series of experimental validations are involved. The attained outcomes reported the supremacy of the SLR-AROSNN technique in the sign recognition process.

KEYWORDS

deep learning, disabled persons, computer vision, artificial rabbits optimizer, sign language recognition

INTRODUCTION

Sign languages are categorized as natural languages and display every feature of other natural languages. The spatial nature and visual of sign languages and their changeability provide a challenge for study in numerous domains, like computer vision (CV), linguistics, machine learning, computer graphics, natural language processing (NLP), and medicine (Bora et al., 2023). Interpretation and linguistics of sign languages were considered with the meaning taken with the use of the sign language. In the early 1980s and late 1970s with the detection of sign languages as a natural language, linguist research got deep into this domain (Katoch et al., 2022). Neural aspects were regarded for completely grabbing the connection between phonetic and sign languages. NLP is concerned with analysis, a task similar to comprehension and interpretation issues (Novopoltsev et al., 2023). Visualization and sign language synthesis is an area that manages the creation of signed speech and visualization problems of sign languages. Sign language recognition is the scientific area accountable for capturing and translating sign speech utilizing artificial intelligence and CV methods (Mannan et al., 2022). In this concern, the effect of digital divide on the urban knowledge of people with disabilities becomes a critical problem that needs to be solved. Considering the elderly in the global population, the percentage of susceptible, underrepresented people in the monarchy of smart city practices can be high (Das et al., 2023).

Deep learning (DL) remains a class of learning methods developed to define difficult structures by merging several nonlinear adjustments (Rwelli et al., 2022). The NNs related to building deep neural networks are the vital building blocks

of DL (Duy Khuat et al., 2021). Such approaches have enabled progression in picture and sound processing, encompassing automated language processing, face identification, CV, voice recognition, spam identification, and various other fields like genomics and drug diagnosis. There are numerous potential uses (Aarthi et al., 2023). First, DL enabled computational processes with numerous processing layers to obtain a representation of various abstracted dimensions (Herath and Ishanka, 2022). DL finds unpredictability in large datasets with the use of the backpropagation approach to express how a mechanism must alter its inner variables that can be used to achieve a presentation in all levels from the symbolization in the preceding layer (Grover et al., 2021).

This manuscript offers the design of Sign Language Recognition using Artificial Rabbits Optimizer with Siamese Neural Network (SLR-AROSNN) technique for persons with disabilities. The proposed SLR-AROSNN technique mainly focused on the recognition of multiple kinds of sign languages posed by disabled persons. The goal of the SLR-AROSNN technique lies in the effectual exploitation of CV, DL, and parameter tuning strategies. It employs the MobileNet model to derive feature vectors. For the identification and classification of sign languages, Siamese neural network (SNN) is used. At the final stage, the SLR-AROSNN technique makes use of the ARO algorithm to get improved sign recognition results. To illustrate the improvement of the SLR-AROSNN technique, a series of experimental validations are involved.

RELATED STUDIES

Obi et al. (2023) present American Sign Language (ASL) data and the CNN system. During the classifying process, the hand image can be sent over a filter and after filter was adopted, the hand was sent through a classifier that forecasts the class of the hand gesture. AlKhuraym et al. (2022) intend to address the ASL recognition issue and make sure tradeoffs are made between enhancing the classifier performance and reducing the design of deep networks to diminish the computing cost. To be specific, we modified EfficientNet methods and produced lightweight DL methods for categorizing Arabic Sign Language gestures. Alsaadi et al. (2022) present a real-time ArSLA detection method with the use of the DL structure. As a method, the proceeding steps were followed. Initially, trusted scientific ArSLA data are located. Then, better DL structures are selected by inspecting related studies. Next, an experiment is directed to test the earlier DL selected structure. Then, the DL structure is constructed on extracted outcomes. Eventually, a real-time detection system is designed.

Athira et al. (2022) modeled a signer-independent vision-relevant gesture detection mechanism that can detect single-handed dynamic and static gestures, double-handed static gestures, and finger spelling words of the Indian Sign Language. In the preprocessing stage, utilizing skin color segmentation, the signs are extracted from real videos. A suitable feature vector was mined from the gesture series after the co-articulation elimination stage. The acquired features are utilized for classification with SVM. Galván-Ruiz et al. (2023) described one method devised for transcribing Spanish Sign Language. In this study, a leap motion volumetric sensor was leveraged because of its capability to find hand movement in three dimensions.

Li et al. (2020) explored the temporal semantic structure of sign videos for learning discriminatory features. To this end, the author developed new sign video segment representations that consider many temporal granularities, therefore easing the necessity for precise video segmentation. Using the presented segment representation, the author designed a new hierarchical sign video feature learning approach through a temporal semantic pyramid network named TSPNet. Jiang et al. (2020) devised a new fingerspelling detection approach that tested four different configurations of TL. Moreover, Adam algorithm was compared with root mean square propagation methods, and SGD with momentum, and comparison of utilizing data augmentation against not utilizing DA was implemented to get higher performance.

THE PROPOSED MODEL

This manuscript is aimed to develop an automated sign language detection model, named SLR-AROSNN technique, for persons with disabilities. The proposed SLR-AROSNN technique mainly focused on the recognition of multiple kinds of sign languages posed by disabled persons. The goal of the SLR-AROSNN technique lies in the effectual exploitation of CV, DL, and parameter tuning strategies. Figure 1 illustrates the overall process of the SLR-AROSNN method.

Phase I: MobileNet model

Primarily, the presented model employs the MobileNet model to derive feature vectors. The MobileNet architecture is designed for possibly running in mobile devices or embedded devices and is designed for efficiency (Taufiqurrahman et al., 2020). The DW convolution is the initial layer of this model for reducing the amount of features. The DW separable convolution consists of two layers: pointwise and DW convolutions. With DW and separable convolutions, MobileNet significantly decreased the parameter of the convolutional layer while retaining better classification performance. Also, a residual bottleneck block was introduced for reusing the feature map. Thus, MobileNetV2 could accomplish good classification performance with a lesser training time. In the current study, we applied both pretrained networks. MobileNetV2 can be the same as original MobileNet, except that it exploits an inverted residual block with bottle necking feature. Compared to original MobileNet, it has a considerably lesser parameter number. MobileNet supports any input size bigger than 32×32, with bigger image sizes providing best performance. The major difference between both versions lies in the convolution layer.

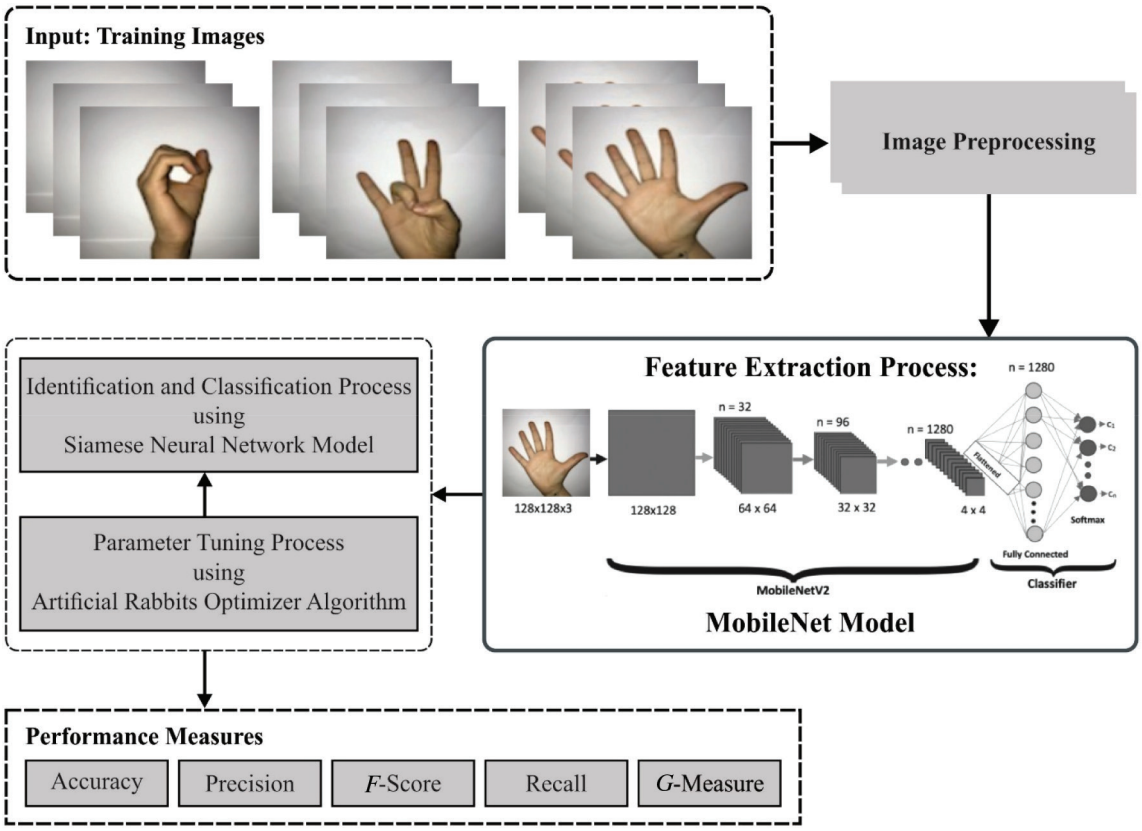

Figure 1: Overall process of the SLR-AROSNN method. Abbreviation: SLR-AROSNN, Sign Language Recognition using Artificial Rabbits Optimizer with Siamese Neural Network.

Phase II: the SNN model

For the identification and classification of sign languages, the SNN classifier is used. The SNN comprises a set of interconnected NN with similar parameters that are trained simultaneously, where learning and unified architecture can be directed by the contrastive loss function (Vasconcellos et al., 2023).

$$CLoss = (1-y)\frac{1}{2}(d) + (y)\frac{1}{2}\max\{0,\ \alpha - d\}, \tag{1}$$

where

$$d(x_1,\ x_2) = g(x_1) - g(x_2). \tag{2}$$

The contrastive loss function's aim is to enhance classification accuracy by evaluating the difference between pairs of samples, represented by the binary label y, where $y = 1$ indicates dissimilarity and y = 0 represents similarity between samples. The function takes in two input samples, x_1 and x_2, and evaluates the Euclidean distance between their corresponding embedding's ($g(x_1)$ and $g(x_2)$ mapping functions). Next, the similarity between embeddings is normalized and calculated to form the $d(x_1, x_2)$ distance measure. In the training, the parameter of $g(x_n)$ is adjusted to minimalize the error rate (loss) by decreasing the distance measure between identical samples when maximizing it between different examples. This can be accomplished by setting a threshold value (α) so that the distance between identical samples is lesser than α, whereas the distance between different samples is greater than α. This architecture is same as binary cross-entropy loss function that is widely applied in binary classification problems whose primary benefit is that it enables the multiclass problem to be transformed into binary classification problems. This allows to implementation of flexible comparisons namely comparing one class to several classes. In case of arthropod species, this flexibility can be highly effective in problems where there is countless classes and comparisons between classes are significant.

Phase III: ARO model

At the final stage, the SLR-AROSNN technique makes use of the ARO algorithm to get improved sign recognition results. ARO algorithm is based on the survival technique used by rabbits, namely random sheltering and detour searching (Pop et al., 2022). The exploration (detour searching) compels a bunny to eat the grass nearby, toward other bunnies' nests. A rabbit selects a shelter from the network of tunnels to hide in using the exploitation (random hiding) method. Additionally, energy levels of the rabbits are diminishing which makes them abandon the detour foraging strategy for transition (randomized hiding).

The detour foraging represents that every search agent is likely for updating the location toward another randomly selected individual and contribute perturbation. The

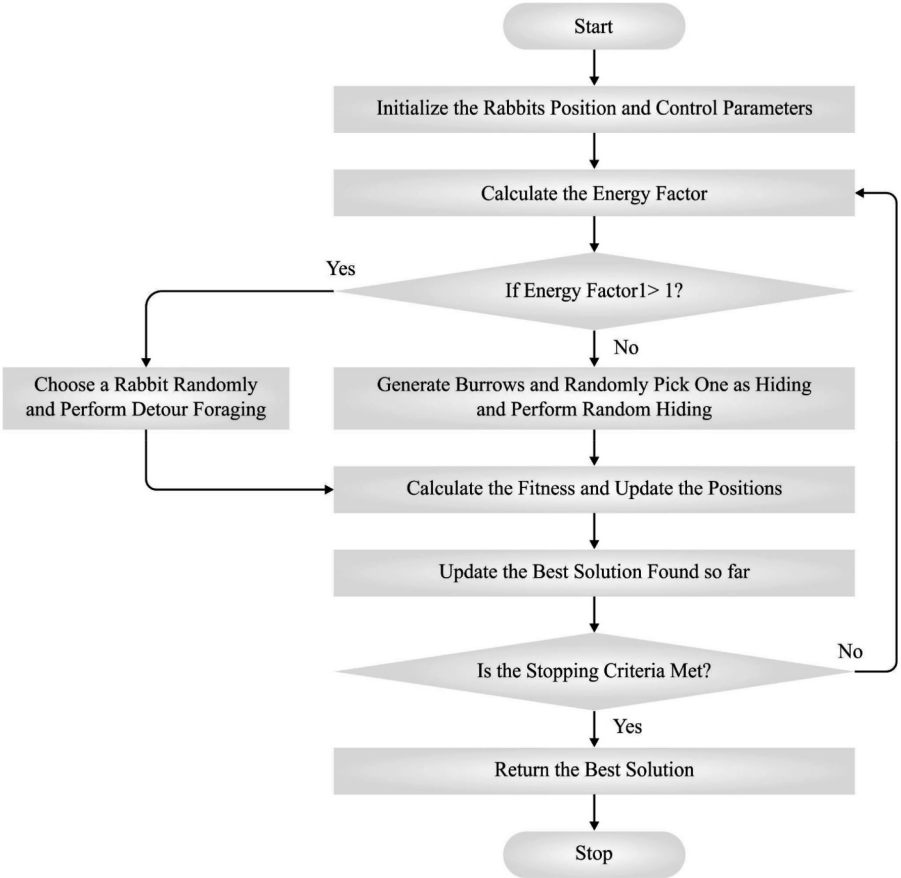

Figure 2: Flowchart of the ARO algorithm. Abbreviation: ARO, Artificial Rabbits Optimizer.

mathematical expression of detour foraging can be given as follows:

$$\vec{V}_i(t+1) = \vec{x}_j(t) + R\cdot(\vec{x}_i(t) - \vec{x}_j(t)) + round(0.025 + 0.5.r_1)\cdot n_1, \quad i,j = 1,\ldots,n \qquad R = L\cdot c \tag{3}$$

$$L = \left(e - e\left(\frac{f-1}{T}\right)^2\right)\cdot sin(2\pi r_2) \tag{4}$$

$$c(k) = \begin{cases} 1 \; if \; k == g(l) \\ 0 \quad else \end{cases} k = 1,\ldots,d \; and \; l = 1,\ldots,r_3\cdot d \tag{5}$$

$$g = randdperm\ (d) \tag{6}$$

Where $\vec{V}_i(t+1)$ represents the updated ith candidate location, n indicates the population size, L indicates the running length that defines the velocity of motion while conducting detour foraging, t represents the existing iteration, T characterizes the count of iterations, d indicates the problem dimension, *randperm* produces a random permutation from 1 to problem dimension, *round* indicates the rounding to the nearest integer, r_1, r_2, and r_3 show the random value in the interval of [0,1], and n_1 indicates the standard distribution number between. The perturbation in Equation (3) assists ARO to conduct a global search and escape local minima or maxima. This unique methodology of foraging includes going toward other bunny's nests rather than their own, which considerably assists in exploration and ensures a global search.

At all the iterations of ARO, a rabbit continuously generates d holes along all the dimensions, randomly choosing a tunnel for hiding to diminish the probability of being intruded upon. The jth burrows of the ith rabbits are shown below.

$$\vec{b}_{i,j}(t) = \vec{X}_i(t) + H\cdot g\cdot\vec{X}_i(t), \quad j = 1,\ldots,d \tag{7}$$

$$H = \frac{T-t+1}{T}\cdot r_4 \tag{8}$$

Table 1: Details of the database.

Class	No. of samples
0	100
1	100
2	100
3	100
4	100
5	100
6	100
7	100
8	100
9	100
Total number of samples	1000

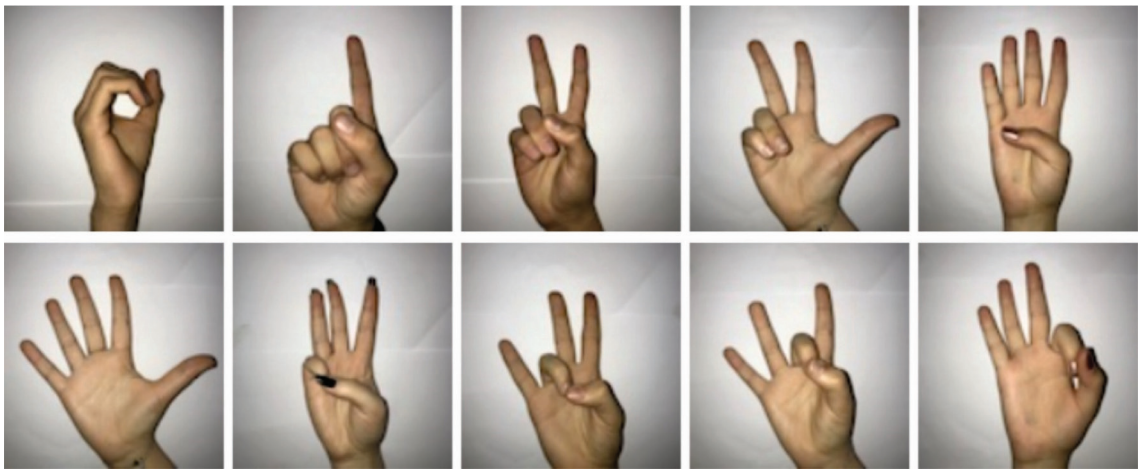

Figure 3: Sample images.

Table 2: Recognition outcome of the SLR-AROSNN approach under varying epochs.

Class	$Accu_y$	$Prec_n$	$Reca_l$	F_{score}	$G_{measure}$
500 epochs					
0	98.50	91.26	94.00	92.61	92.62
1	99.40	96.08	98.00	97.03	97.03
2	99.20	96.00	96.00	96.00	96.00
3	99.70	98.99	98.00	98.49	98.49
4	99.00	96.88	93.00	94.90	94.92
5	98.90	95.88	93.00	94.42	94.43
6	99.80	99.00	99.00	99.00	99.00
7	99.20	95.10	97.00	96.04	96.04
8	98.60	93.00	93.00	93.00	93.00
9	99.10	95.05	96.00	95.52	95.52
Average	99.14	95.72	95.70	95.70	95.71
1000 epochs					
0	92.70	62.86	66.00	64.39	64.41
1	93.80	70.65	65.00	67.71	67.77
2	93.40	69.77	60.00	64.52	64.70
3	92.10	58.68	71.00	64.25	64.55
4	91.50	60.56	43.00	50.29	51.03
5	90.60	52.78	57.00	54.81	54.85
6	94.50	68.60	83.00	75.11	75.45
7	92.20	62.22	56.00	58.95	59.03
8	92.80	64.58	62.00	63.27	63.28
9	91.80	58.18	64.00	60.95	61.02
Average	92.54	62.89	62.70	62.42	62.61
1500 epochs					
0	95.50	76.70	79.00	77.83	77.84
1	96.10	82.11	78.00	80.00	80.03
2	95.90	79.80	79.00	79.40	79.40
3	93.40	65.74	71.00	68.27	68.32
4	93.10	67.42	60.00	63.49	63.60
5	93.90	70.10	68.00	69.04	69.04
6	95.70	77.14	81.00	79.02	79.05
7	93.60	68.37	67.00	67.68	67.68
8	95.10	72.97	81.00	76.78	76.88
9	94.50	73.68	70.00	71.79	71.82
Average	94.68	73.40	73.40	73.33	73.37
2000 epochs					
0	96.60	81.73	85.00	83.33	83.35
1	98.10	90.10	91.00	90.55	90.55
2	97.50	84.40	92.00	88.04	88.12
3	97.60	89.58	86.00	87.76	87.77
4	96.50	85.71	78.00	81.68	81.77
5	96.30	81.82	81.00	81.41	81.41
6	98.60	93.00	93.00	93.00	93.00
7	96.40	81.37	83.00	82.18	82.18
8	97.50	87.13	88.00	87.56	87.56
9	97.50	88.66	86.00	87.31	87.32
Average	97.26	86.35	86.30	86.28	86.30
2500 epochs					
0	97.60	85.85	91.00	88.35	88.39
1	97.20	88.30	83.00	85.57	85.61
2	97.70	86.67	91.00	88.78	88.81
3	97.30	86.14	87.00	86.57	86.57
4	96.30	84.62	77.00	80.63	80.72
5	96.20	81.63	80.00	80.81	80.81
6	98.40	92.00	92.00	92.00	92.00
7	96.80	82.08	87.00	84.47	84.50
8	97.60	88.00	88.00	88.00	88.00
9	96.90	84.85	84.00	84.42	84.42
Average	97.20	86.01	86.00	85.96	85.98
3000 epochs					
0	96.60	80.56	87.00	83.65	83.72
1	97.60	89.58	86.00	87.76	87.77
2	97.40	87.00	87.00	87.00	87.00
3	96.00	78.85	82.00	80.39	80.41
4	96.00	83.33	75.00	78.95	79.06
5	95.90	82.42	75.00	78.53	78.62
6	97.10	82.57	90.00	86.12	86.20
7	95.70	79.38	77.00	78.17	78.18
8	97.50	85.71	90.00	87.80	87.83
9	96.80	84.00	84.00	84.00	84.00
Average	96.66	83.34	83.30	83.24	83.28

Abbreviation: SLR-AROSNN, Sign Language Recognition using Artificial Rabbits Optimizer with Siamese Neural Network.

$$g(k) = \begin{cases} 1 \text{ if } k == j \\ 0 \quad else, \end{cases} \quad k = 1, \ldots, d, \tag{9}$$

where d is the burrows generated in the region of a rabbit, H indicates the hiding parameter and the value is declined

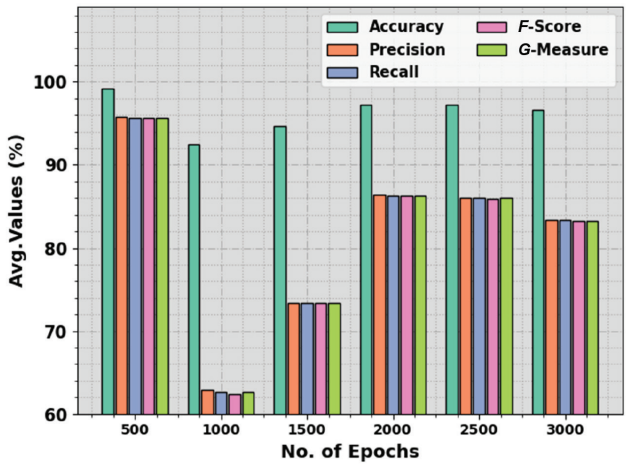

Figure 4: Average outcome of the SLR-AROSNN approach under varying epochs. Abbreviation: SLR-AROSNN, Sign Language Recognition using Artificial Rabbits Optimizer with Siamese Neural Network.

linearly from 1 to $1/T$ with the random perturbation through iteration. Mainly, hole was generated in a larger area of the bunny. This neighborhood decreases as the amount of iterations increases. Equation (10) is recommended to model the random hiding method.

$$\vec{V}_i(t+1) = \vec{X}_i(t) + R \cdot (r_4 \cdot \vec{b}_{i,j}(t) - \vec{X}_i(t)) \tag{10}$$

$$g(k) = \begin{cases} 1 \text{ if } & k == r_5 \cdot d \\ 0 & else, \end{cases} \quad k = 1, \ldots, d \tag{11}$$

$$\vec{b}_{i,j}(t) = \vec{X}_i(t) + H \cdot g_r \cdot \vec{X}_i(t), \tag{12}$$

where r_4 and r_5 denote the random numbers within [0, 1]. $\vec{b}_{i,j}(t)$ denotes a randomly chosen burrow for hiding, Afterward either random hiding or detour foraging, the location of the ith rabbit is upgraded using the following equation:

$$\vec{x}(t+1) = \begin{cases} \vec{X}_i(t) & f(\vec{X}_i(t)) \le f(\vec{V}_i(t+1)) \\ \vec{V}_i(t+1) & f(\vec{X}_i(t)) > f(\vec{V}_i(t+1)) \end{cases} \tag{13}$$

If the candidate fitness of the ith rabbit is higher than the existing location fitness, the bunny departs the existing position and remains at the candidate location generated by using Equation (3) or Equation (10). A factor that aims to imitate the transition from exploration to exploitation is given below:

$$A(t) = 4\left(1 - \frac{t}{T}\right) ln \frac{1}{r}. \tag{14}$$

In Equation (14), r indicates the random integer within [0, 1]. Once the factor $A(t) > 1$, the process finds the exploration (solution globally); once the factor $A(t) \le 1$, the process finds the exploitation (solution locally). Figure 2 depicts the flowchart of the ARO algorithm.

The ARO method not only derives a fitness function to accomplish better classification performance but also defines a positive integer to represent the best performance of the candidate solution. The decline of the classification error rate is regarded as the fitness function, as given in Equation (15).

$$fitness(x_i) = \frac{No.of\ misclassified\ samples}{Total\ No.of\ samples} * 100 \tag{15}$$

EXPERIMENTAL VALIDATION

Comprising 1000 samples with 10 classes as demonstrated in Table 1. Figure 3 represents the sample images.

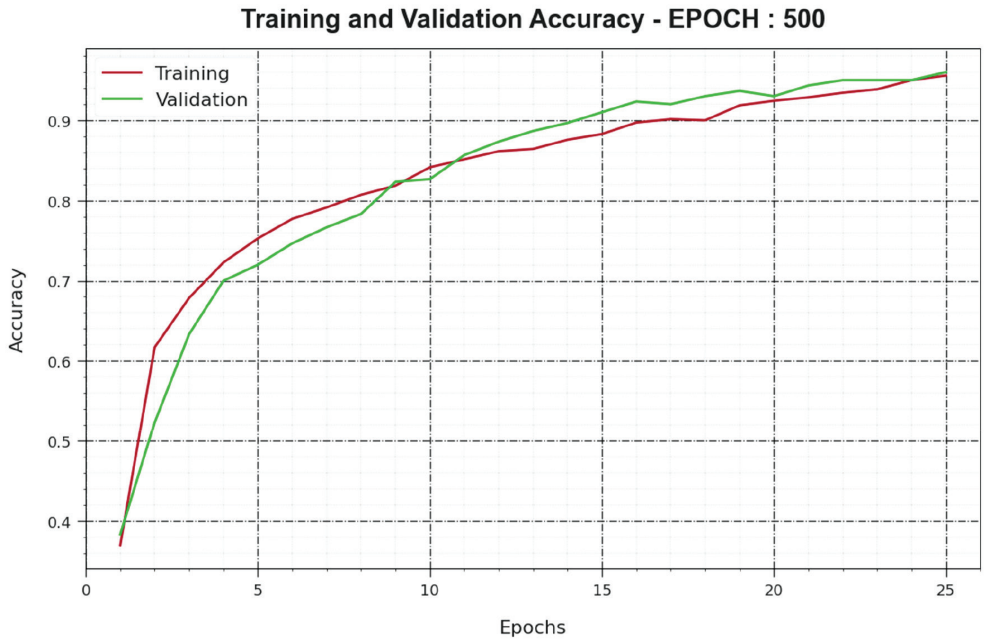

Figure 5: Accuracy curve of the SLR-AROSNN approach under 500 epochs. Abbreviation: SLR-AROSNN, Sign Language Recognition using Artificial Rabbits Optimizer with Siamese Neural Network.

In Table 2 and Figure 4, the recognition results of the SLR-AROSNN method are investigated under several epochs. The results highlighted that the SLR-AROSNN technique gains improved results under each epoch. On 500 epochs, the SLR-AROSNN technique obtains average $accu_y$, $prec_n$, $reca_1$, F_{score}, and $G_{measure}$ of 99.14, 95.72, 95.70, 95.70, and 95.71%, respectively. Simultaneously, on 1500 epochs, the SLR-AROSNN technique obtains average $accu_y$, $prec_n$, $reca_1$, F_{score}, and $G_{measure}$ of 94.68, 73.40, 73.40, 73.33, and 73.37%, respectively. Concurrently, on 2500 epochs, the SLR-AROSNN method obtains average $accu_y$, $prec_n$, $reca_1$, F_{score}, and $G_{measure}$ of 97.20, 86.01, 86, 85.96, and 85.98%, respectively. Lastly, on 3000 epochs, the SLR-AROSNN technique obtains average $accu_y$, $prec_n$, $reca_1$, F_{score}, and $G_{measure}$ of 96.66, 83.34, 83.30, 83.24, and 83.28%, respectively.

Figure 5 inspects the accuracy of the SLR-AROSNN method in the training and validation process on 500 epochs. The figure indicates that the SLR-AROSNN technique reaches increasing accuracy values over increasing epochs.

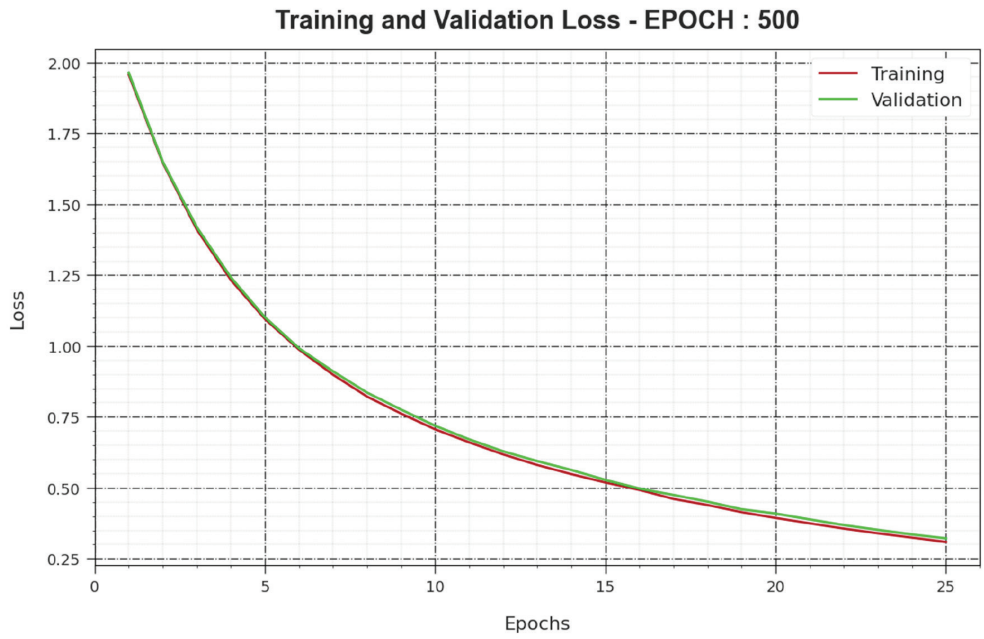

Figure 6: Loss curve of the SLR-AROSNN approach under 500 epochs. Abbreviation: SLR-AROSNN, Sign Language Recognition using Artificial Rabbits Optimizer with Siamese Neural Network.

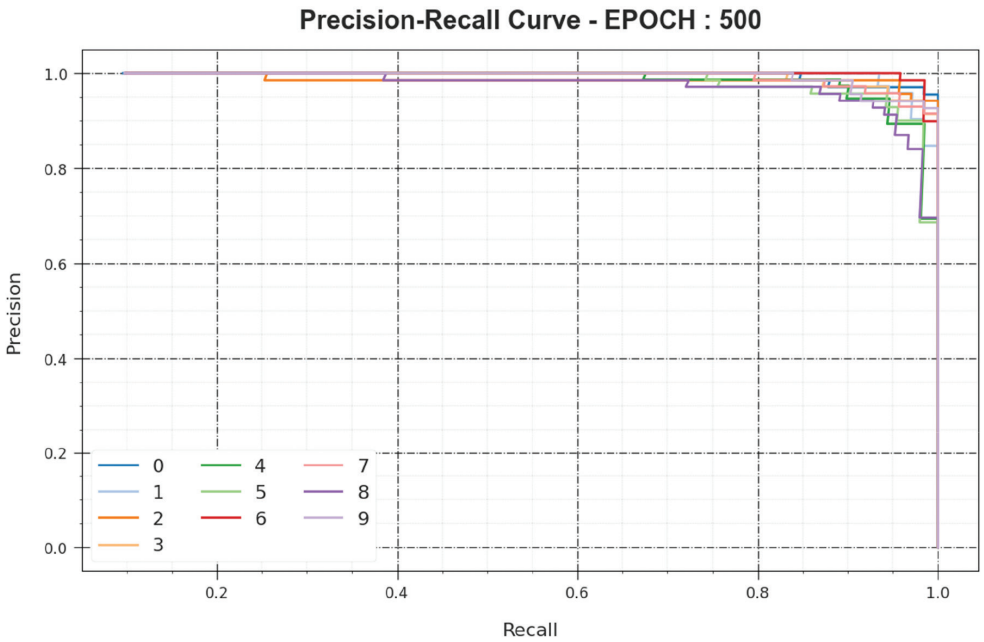

Figure 7: PR curve of the SLR-AROSNN approach under 500 epochs. Abbreviation: SLR-AROSNN, Sign Language Recognition using Artificial Rabbits Optimizer with Siamese Neural Network. PR, precision-recall.

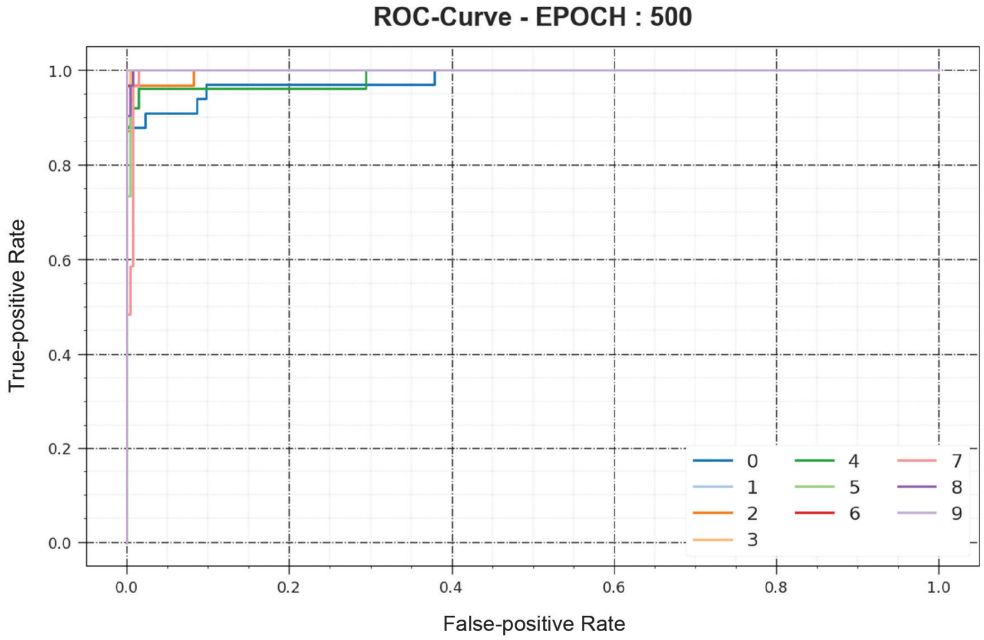

Figure 8: ROC curve of the SLR-AROSNN approach under 500 epochs. Abbreviation: SLR-AROSNN, Sign Language Recognition using Artificial Rabbits Optimizer with Siamese Neural Network.

Furthermore, the increasing validation accuracy over training accuracy shows that the SLR-AROSNN method learns effectively on 500 epochs.

The loss analysis of the SLR-AROSNN method in training and validation is depicted on 500 epochs in Figure 6. The results indicate that the SLR-AROSNN technique obtains closer values of training and validation loss. The SLR-AROSNN method learns efficiently on 500 epochs.

A brief precision-recall (PR) curve of the SLR-AROSNN technique is demonstrated on 500 epochs in Figure 7. The outcomes stated that the SLR-AROSNN method results in increasing values of PR. In addition, it is noticeable that the SLR-AROSNN technique can reach higher PR values in all classes.

In Figure 8, an ROC study of the SLR-AROSNN technique is revealed on 500 epochs. The figure described that the SLR-AROSNN system resulted in improved ROC values. Besides, the SLR-AROSNN technique can extend enhanced ROC values on all classes.

In Table 3 and Figure 9, the overall results of the SLR-AROSNN method are compared with recent approaches (Dhulipala et al., 2022). Based on $accu_y$, the SLR-AROSNN technique attains increasing $accu_y$ of 99.14% while the RF, DT, LSTM, CNN, and SHADU-DL techniques obtain decreasing $accu_y$ of 98.77, 98.62, 95.42, 97.32, and 98.63%, correspondingly. Also, based on $prec_n$, the SLR-AROSNN technique attains increasing $prec_n$ of 99.14% while the RF, DT, LSTM, CNN, and SHADU-DL techniques obtain decreasing $prec_n$ of 98.77, 98.62, 95.42, 97.32, and 98.63%, correspondingly. Similarly, based on $reca_1$, the SLR-AROSNN technique attains increasing $reca_1$ of 99.14% while the RF, DT, LSTM, CNN, and SHADU-DL techniques obtain decreasing $reca_1$ of 98.77, 98.62, 95.42, 97.32, and 98.63%, correspondingly. Finally, based on F_{score}, the SLR-AROSNN technique attains increasing F_{score} of 99.14% while the RF, DT, LSTM, CNN, and SHADU-DL techniques obtain decreasing F_{score}

Table 3: Comparative outcome of the SLR-AROSNN method with other algorithms.

Methods	$Accu_y$	$Prec_n$	$Reca_1$	F_{score}
Random forest	98.77	93.34	94.5	94.03
Decision tree	98.62	94.51	94.78	93.80
LSTM model	95.41	93.22	94.02	93.22
CNN model	97.32	94.37	93.16	93.33
SHADU-DL	98.63	94.86	94.52	93.88
SLR-AROSNN	99.14	95.72	95.70	95.70

Abbreviation: SLR-AROSNN, Sign Language Recognition using Artificial Rabbits Optimizer with Siamese Neural Network.

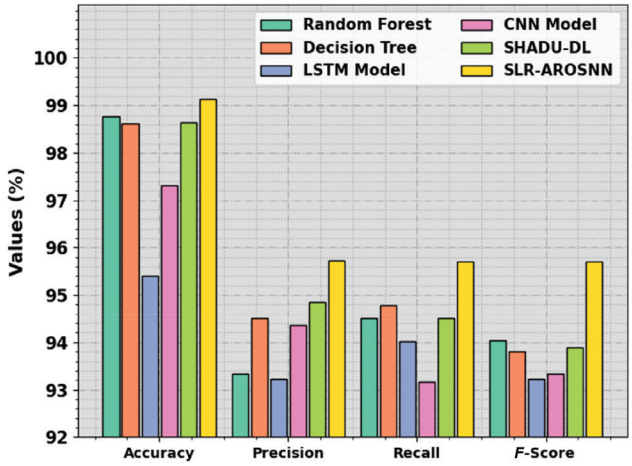

Figure 9: Comparative outcome of the SLR-AROSNN approach with other algorithms. Abbreviation: SLR-AROSNN, Sign Language Recognition using Artificial Rabbits Optimizer with Siamese Neural Network.

of 98.77, 98.62, 95.42, 97.32, and 98.63%, correspondingly. Thus, the SLR-AROSNN technique can be employed for accurate recognition purposes.

CONCLUSION

This manuscript aimed to develop an automated sign language detection model, named the SLR-AROSNN technique for persons with disabilities. The proposed SLR-AROSNN technique mainly focused on the recognition of multiple kinds of sign languages posed by disabled persons. The goal of the SLR-AROSNN technique lies in the effectual exploitation of CV, DL, and parameter tuning strategies. It employs the MobileNet model to derive feature vectors. For the identification and classification of sign languages, the SNN classifier is used. At the final stage, the SLR-AROSNN technique makes use of the ARO algorithm to get improved sign recognition results. To illustrate the improvement of the SLR-AROSNN technique, a series of experimental validations are involved. The attained outcomes exhibited the supremacy of the SLR-AROSNN technique in the sign recognition process.

ACKNOWLEDGEMENTS

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

REFERENCES

Aarthi B., Charitharyan P., Thanmai R. and Kesana M. (2023). Implementation of dynamic gesture interpretation of sign language for impact on hearing and speech impairment. In *Handbook of Research on Advanced Practical Approaches to Deepfake Detection and Applications* pp. 257-274, IGI Global.

AlKhuraym B.Y., Ismail M.M.B. and Bchir O. (2022). Arabic sign language recognition using lightweight CNN-based architecture. *Int. J. Adv. Comput. Sci. Appl.*, 13(4).

Alsaadi Z., Alshamani E., Alrehaili M., Alrashdi A.A.D., Albelwi S. and Elfaki A.O. (2022). A real time Arabic sign language alphabets (ArSLA) recognition model using deep learning architecture. *Computers*, 11(5), 78.

Athira P.K., Sruthi C.J. and Lijiya A. (2022). A signer independent sign language recognition with co-articulation elimination from live videos: an Indian scenario. *J. King Saud Univ. Comput. Inf. Sci.*, 34(3), 771-781.

Bora J., Dehingia S., Boruah A., Chetia A.A. and Gogoi D. (2023). Real-time Assamese sign language recognition using mediapipe and deep learning. *Procedia Comput. Sci.*, 218, 1384-1393.

Das S., Imtiaz M.S., Neom N.H., Siddique N. and Wang H. (2023). A hybrid approach for Bangla sign language recognition using deep transfer learning model with random forest classifier. *Expert Syst. Appl.*, 213, 118914.

Dhulipala S., Adedoyin F.F. and Bruno A. (2022). Sign and human action detection using deep learning. *J. Imaging*, 8(7), 192.

Duy Khuat B., Thai Phung D., Thi Thu Pham H., Ngoc Bui A. and Tung Ngo S. (2021). Vietnamese sign language detection using Mediapipe. In: *2021 10th International Conference on Software and Computer Applications*. ACM, Kuala Lumpur, Malaysia, February 23-26, pp. 162-165.

Galván-Ruiz J., Travieso-González C.M., Pinan-Roescher A. and Alonso-Hernández J.B. (2023). Robust identification system for Spanish sign language based on three-dimensional frame information. *Sensors*, 23(1), 481.

Grover Y., Aggarwal R., Sharma D. and Gupta P.K. (2021). Sign language translation systems for hearing/speech impaired people: a review. In: *2021 International Conference on Innovative Practices in Technology and Management (ICIPTM)*. IEEE, Noida, India, February 17-19, pp. 10-14.

Herath R.J. and Ishanka P. (2022). An approach to Sri Lankan sign language recognition using deep learning with MediaPipe. In: *Digital Technologies and Applications: Proceedings of ICDTA'22, Fez, Morocco*, Volume 1, Springer International Publishing, Cham; pp. 449-459.

Jiang X., Hu B., Chandra Satapathy S., Wang S.H. and Zhang Y.D. (2020). Fingerspelling identification for Chinese sign language via AlexNet-based transfer learning and Adam optimizer. *Sci. Program.*, 2020, 1-13.

Katoch S., Singh V. and Tiwary U.S. (2022). Indian sign language recognition system using SURF with SVM and CNN. *Array*, 14, 100141.

Li D., Xu C., Yu X., Zhang K., Swift B., Suominen H., et al. (2020). Tspnet: hierarchical feature learning via temporal semantic pyramid for sign language translation. *Adv. Neural Inf. Process Syst.*, 33, 12034-12045.

Mannan A., Abbasi A., Javed A.R., Ahsan A., Gadekallu T.R. and Xin Q. (2022). Hypertuned deep convolutional neural network for sign language recognition. *Comput. Intell. Neurosci*, 2022, 1450822.

Novopoltsev M., Verkhovtsev L., Murtazin R., Milevich D. and Zemtsova I. (2023). Fine-tuning of sign language recognition models: a technical report. *arXiv preprint arXiv:2302.07693*.

Obi Y., Claudio K.S., Budiman V.M., Achmad S. and Kurniawan A. (2023). Sign language recognition system for communicating to people with disabilities. *Procedia Comput. Sci.*, 216, 13-20.

Pop C.B., Cioara T., Anghel I., Antal M., Chifu V.R., Antal C., et al. (2022). Review of bio-inspired optimization applications in renewable-powered smart grids: emerging population-based metaheuristics. *Energy Rep.*, 8, 11769-11798.

Rwelli R.E., Shahin O.R. and Taloba A.I. (2022). Gesture based Arabic sign language recognition for impaired people based on convolution neural network. *arXiv preprint arXiv:2203.05602*.

Taufiqurrahman S., Handayani A., Hermanto B.R. and Mengko T.L.E.R. (2020). Diabetic retinopathy classification using a hybrid and efficient MobileNetV2-SVM model. In *2020 IEEE REGION 10 CONFERENCE (TENCON)*, IEEE; pp. 235-240.

Vasconcellos M.E., Ferreira B.G., Leandro J.S., Neto B.F., Cordeiro F.R., Cestari I.A., et al. (2023). Siamese convolutional neural network for heartbeat classification using limited 12-lead ECG datasets. *IEEE Access*, 11, 5365-5376.