
Research article

Deep convolutional neural network-based Leveraging Lion Swarm Optimizer for gesture recognition and classification

Mashaal Maashi^{1,2,*}, Mohammed Abdullah Al-Hagery³, Mohammed Rizwanullah⁴ and Azza Elneil Osman⁴

¹ Department of Software Engineering, College of Computer and Information Sciences, King Saud University, PO box 103786, Riyadh 11543, Saudi Arabia

² King Salman Center for Disability Research, Riyadh, Saudi Arabia

³ Department of Computer Science, College of Computer, Qassim University, Saudi Arabia

⁴ Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam bin Abdulaziz University, AlKharj, Saudi Arabia

* **Correspondence:** Email: mmaashi@ksu.edu.sa.

Abstract: Vision-based human gesture detection is the task of forecasting a gesture, namely clapping or sign language gestures, or waving hello, utilizing various video frames. One of the attractive features of gesture detection is that it makes it possible for humans to interact with devices and computers without the necessity for an external input tool like a remote control or a mouse. Gesture detection from videos has various applications, like robot learning, control of consumer electronics computer games, and mechanical systems. This study leverages the Lion Swarm optimizer with a deep convolutional neural network (LSO-DCNN) for gesture recognition and classification. The purpose of the LSO-DCNN technique lies in the proper identification and categorization of various categories of gestures that exist in the input images. The presented LSO-DCNN model follows a three-step procedure. At the initial step, the 1D-convolutional neural network (1D-CNN) method derives a collection of feature vectors. In the second step, the LSO algorithm optimally chooses the hyperparameter values of the 1D-CNN model. At the final step, the extreme gradient boosting (XGBoost) classifier allocates proper classes, i.e., it recognizes the gestures efficaciously. To demonstrate the enhanced gesture classification results of the LSO-DCNN approach, a wide range of experimental results are investigated. The brief comparative study reported the improvements in the LSO-DCNN technique in the gesture recognition process.

Keywords: human-computer interaction; swarm intelligence; CNN Model; gesture recognition; transfer learning

Mathematics Subject Classification: 11Y40

1. Introduction

Noncontact gesture recognition has made a significant contribution to human-computer interaction (HCI) applications with the enormous growth of artificial intelligence (AI) and computer technology [1]. Hand gesture detection systems, with their natural human-computer interaction features, enable effective and intuitive communication through a computer interface. Furthermore, gesture detection depends on vision and can be broadly implemented in AI, natural language communication, virtual reality, and multimedia [2]. Daily, the demand for and the level of services essential to people is increasing. Hand gestures are a main component of face-to-face communication [3]. Hence, human body language serves a significant part in face-to-face transmission and making hand gestures. In interaction, many things are expressed with hand gestures, and this study presents few visions into transmission itself [4]. Yet, recent automation in this region does not concentrate on using hand gestures in everyday actions. The emerging technology eases the difficulty of processes of different user interfaces and computer programs presented to the user. To make this mechanism less complex and easy to understand, nowadays image processing is utilized [5].

When transmission has to be recognized between a deaf and a normal person, there is a robust necessity for hand gestures. To make the system smarter, there comes a necessity to enter hand gesture imageries into the mechanism and carry out an examination further to determine their meaning [6]. Still, conventional hand gesture detection related to image processing methods was not broadly implemented in HCI due to its complex algorithm, poor real-time capability, and low recognition accuracy [7]. Currently, gesture detection related to machine learning (ML) has advanced quickly in HCI owing to the presentation of AI and image processing graphics processor unit (GPU) [8]. The ML methods like neural networks, local orientation histograms, elastic graph matching, and support vector machines (SVM) were broadly utilized. Due to its learning capability, the NN does not require manual feature setting through simulating human learning processes and can execute training gesture instances to form a network classification detection map [9]. Currently, DL is a frequently utilized approach for HGR. Recurrent neural networks (RNN), CNNs, and stacked denoising auto encoders (SDAE), and are usually utilized in HGR applications [10].

This study leverages the Lion Swarm optimizer with deep convolutional neural network (LSO-DCNN) for gesture recognition and classification. The aim of the LSO-DCNN technique lies in the proper identification and categorization of various categories of gestures that exist in the input images. Primarily, the 1D-convolutional neural network (1D-CNN) method derives a collection of feature vectors. In the second step, the LSO algorithm optimally chooses the hyperparameter values of the 1D-CNN model. At the final step, the extreme gradient boosting (XGBoost) classifier allocates proper classes, i.e., recognizes the gestures efficaciously. To portray the enhanced gesture classification results of the LSO-DCNN algorithm, a wide range of experimental results are investigated. A brief comparative study reports the improvements in the LSO-DCNN technique in the gesture recognition process.

2. Literature survey

Sun et al. [11] suggested a model dependent upon multi-level feature fusion of a two-stream convolutional neural network (MFF-TSCNN) which comprises three major phases. Initially, the Kinect sensor acquires red, green, blue, and depth (RGB-D) imageries for establishing a gesture dataset. Simultaneously, data augmentation is accomplished on the datasets of testing and training. Later, a MFF-TSCNN model is built and trained. Barioul and Kanoun [12] proposed a new classifying model established on an extreme learning machine (ELM) reinforced by an enhanced grasshopper optimization algorithm (GOA) as a fundamental for a weight-pruning procedure. Myographic models like force myography (FMG) present stimulating signals that can construct the foundation for recognizing hand signs. FMG was examined for limiting the sensor numbers to appropriate locations and giving necessary signal processing techniques for observable employment in wearable embedded schemes. Gadekallu et al. [13] presented a crow search-based CNN (CS-CNN) method for recognizing gestures relating to the HCI field. The hand gesture database utilized in the research is an open database that is obtained from Kaggle. Also, a one-hot encoding method was employed for converting the definite values of the data to its binary system. After this, a crow search algorithm (CSA) for choosing optimum tuning for data training by utilizing the CNNs was employed.

Yu et al. [14] employed a particle swarm optimization (PSO) technique for the width and center value optimization of the radial basis function neural network (RBFNN). Also, the authors utilized a Electromyography (EMG) signal acquisition device and the electrode sleeve for gathering the four-channel continuous EMG signals produced by 8 serial gestures. In [15], the authors presented an ensemble of CNN-based techniques. First, the gesture segment is identified by employing the background separation model established on the binary threshold. Then, the contour section can be abstracted and the segmentation of the hand area takes place. Later, the imageries are re-sized and given to three distinct CNN methods for similar training.

Gao et al. [16] developed an effective hand gesture detection model established on deep learning. First, an RGB-D early-fusion technique established on the HSV space was suggested, efficiently mitigating background intrusion and improving hand gesture data. Second, a hand gesture classification network (HandClasNet) was suggested for comprehending hand gesture localization and recognition by identifying the center and corner hand points, and a HandClasNet was suggested for comprehending gesture detection by employing a similar EfficientNet system. In [17], the authors utilized the CNN approach for the recognition and identification of human hand gestures. This procedure workflow comprises hand region of interest segmenting by employing finger segmentation, mask image, segmented finger image normalization, and detection by utilizing the CNN classifier. The segmentation is performed on the hand area of an image from the whole image by implementing mask images.

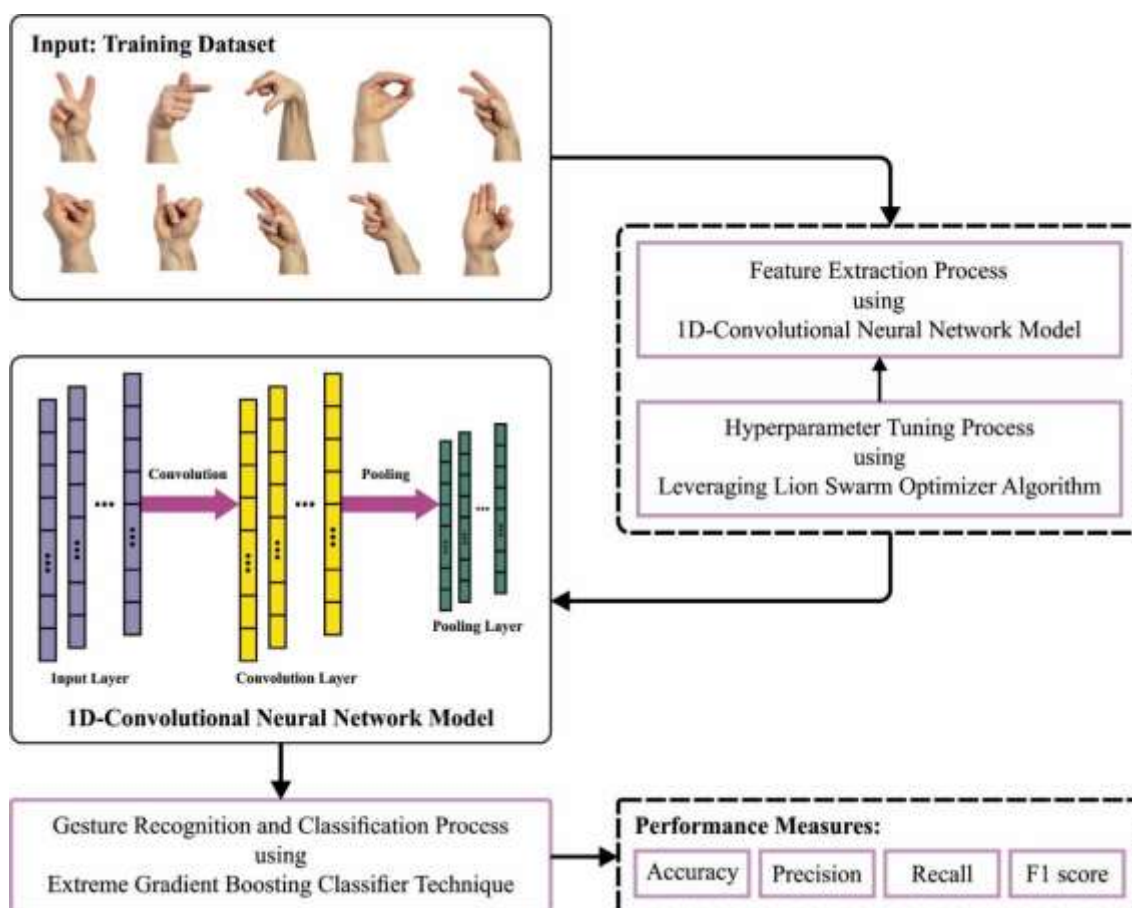


Figure 1. Overall process of the LSO-DCNN approach.

3. Materials and methods

This study has developed a new LSO-DCNN method for automated gesture recognition and classification. The major intention of the LSO-DCNN method lies in the proper identification and categorization of various categories of gestures that exist in the input images. The presented LSO-DCNN model follows a three-step procedure:

Step 1: The 1D-CNN method derives a collection of feature vectors.

Step 2: The LSO method optimally chooses the hyperparameter values of the 1D-CNN model.

Step 3: The XGBoost classifier assigns appropriate classes, i.e., effectively recognizes the gestures.

3.1. Stage I: 1D-CNN based feature extraction

First, the 1D-CNN model derives a collection of feature vectors. The CNN can be referred to as a neural network that exploits convolutional operations in at least one layer of the network instead of normal matrix multiplication operations [18]. Convolution is a special linear operation; all the layers of the convolutional network generally consist of three layers: pooling, convolutional, and activation layers. In the image detection domain, the 2DCNN can be commonly utilized for extracting features from images. The classical CNN models are AlexNet, LeNet, ResNet, VGG, GoogleNet, and so on. The 1D-CNN is used for extracting appropriate features of the data. The input of the 1D-CNN is 1D

data, hence its convolutional kernel adopts a 1D architecture. The output of every convolutional, activation, and pooling layer corresponds to a 1D feature vector. In this section, the fundamental structure of the 1DCNN will be introduced.

3.1.1 Convolution layer

The convolution layer implements the convolution function on the 1D input signals and the 1D convolution filter, and later extracts local features using the activation layer. The data is inputted to the convolution layer of the 1D-CNN to implement the convolutional function.

$$x_k^l = \sum_{i=1}^n \text{conv}(w_{ik}^{l-1}, s_i^{l-1}) + b_k^l \quad (1)$$

Here, x_k^l, b_k^l correspondingly characterize the output and offset of the k -th neurons in layer l ; s_i^{l-1} characterizes the output of i -th neurons in layer $l - 1$; w_{ik}^{l-1} characterizes the convolutional kernels of i -th neurons in the $l - 1$ layer, and the k -th neurons in layer $l, i = 1, 2, \dots, n, n$ denotes the amount of neurons.

3.1.2 Activation layer

The activation layer implements a non-linear conversion on the input signal through a non-linear function to improve the CNN's expressive power. Currently, the typical activation function is ReLU, Sigmoid, and *Tanh*. Since the ReLU function may overcome gradient dispersion and converge quickly, it is extensively applied. Thus, the ReLU function was applied as the activation function, and its equation can be represented as

$$y_k^l = f(x_k^l) = \{0, x_k^l\} \quad (2)$$

where y_k^l denotes the activation value of layer l .

3.1.3 Pooling layer

The pooling layer can generally be employed after the convolution layer. Downsampling avoids over-fitting, decreases the spatial size of parameters and network features, and decreases the calculation count. The typical pooling operations are maximum and average pooling.

$$z_k^{l(j)} = \{y_k^{l(t)}\} \quad (3)$$

Where $z_k^{l(j)}$ signifies the j th value in the k -th neuron of layer l ; $y_k^{l(t)}$ characterizes the t -th activation value in the k -th neuron of layer l ; r denotes the pooling area's width.

3.2. Stage II: LSO-based hyperparameter tuning

In this work, the LSO approach optimally chooses hyperparameter values of the 1D-CNN model. This approach is selected for its capacity for effectively navigating the parameter space, adapting the

nature of the model to local characteristics, and converging toward optimum settings, making the model more appropriate to fine-tune intricate methods. In the LSO algorithm, based on the historical optimum solution, the lion king conducts a range search to find the best solutions [19]. The equation for updating the location is given below:

$$x_i^{k+1} = g^k(1 + \gamma\|p_i^k - g^k\|) \quad (4)$$

A lioness arbitrarily chooses an additional lioness to cooperate with, and the equation for location updating can be represented as

$$x_i^{k+1} = \frac{p_i^k + p_c^k}{2}(1 + \alpha_f \gamma) \quad (5)$$

Follow the lioness, leave the group, or follow the lion king to find an updated position are the three updating approaches for young lions:

$$x_i^{k+1} = \begin{cases} \frac{g^k + p_i^k}{2}(1 + \alpha_c \gamma), & 0 \leq q \leq \frac{1}{3} \\ \frac{p_m^k + p_i^k}{2}(1 + \alpha_c \gamma), & \frac{1}{3} < q \leq \frac{2}{3} \\ \frac{g^k + p_i^k}{2}(1 + \alpha_c \gamma), & \frac{2}{3} < q \leq 1 \end{cases} \quad (6)$$

In Eq (6), x_i^k denotes the i -th individuals at the k^{th} generation population; p_i^k represents the prior optimum location of the i -th individuals from the 1st to k^{th} generation; γ shows the uniform distribution random number $N(0,1)$ p_c^k is randomly chosen from the k^{th} generation lioness group; g^k shows the optimum location of the k^{th} generation population; q denotes the uniform distribution random number $U[0,1]$ $\underline{g} = \underline{low} + \underline{up} - g^k$, p_m^k is arbitrarily chosen from the k^{th} generation lion group; α_f and α_c denotes the disturbance factor, \underline{low} and \underline{up} indicates the minimal and maximal values of all the dimensions within the range of lion activity space

$$\alpha_f = 0.1(\underline{up} - \underline{low}) \times \exp\left(-\frac{30t}{T}\right)^{10} \quad (7)$$

$$\alpha_c = 0.1(\underline{up} - \underline{low}) \times \left(\frac{T-t}{T}\right) \quad (8)$$

where T shows the maximal amount of iterations and t denotes the existing amount of iterations.

The fitness selection becomes a vital component in the LSO method. Solution encoding can be used to evaluate the candidate solution's aptitude. Here, to design a fitness function, the accuracy value is the main condition used.

$$Fitness = \max(P) \quad (9)$$

$$P = \frac{TP}{TP+FP} \quad (10)$$

From the expression, FP means the false positive value and TP denotes the true positive.

3.3. Stage III: XGBoost classification

Finally, the XGBoost classifier allocates proper classes, i.e., recognizes the gestures efficaciously. XGBoost is an ensemble ML technique, a gradient boost method utilized for improving the efficiency of a predictive model, which integrates a series of weak methods as a strong learning approach [20].

The ensemble methods offer optimum outcomes related to a single model. Figure 2 defines the architecture of XGBoost. The steps involved are given as follows.

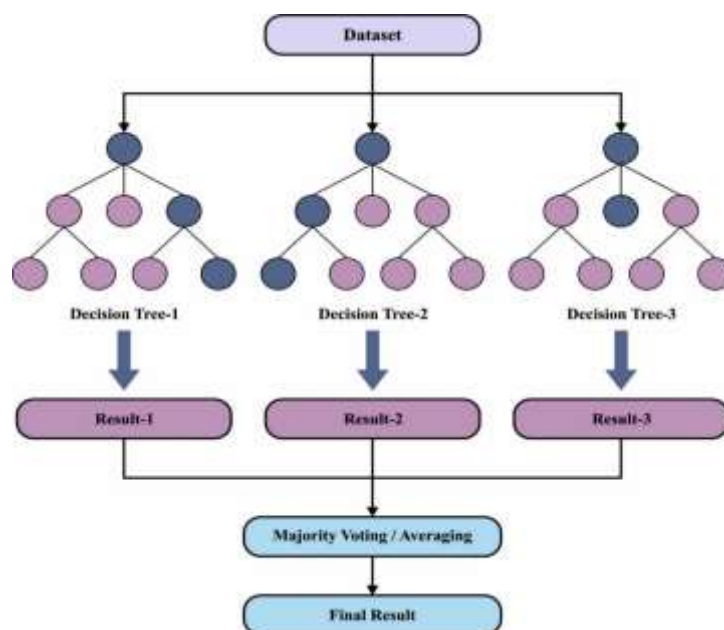


Figure 2. Structure of XGBoost.

Step 1: Initialize

To solve a binary classifier problem, where y_j is the actual label denoted as 1 or 0. Consequently, the commonly exploited log loss function is assumed during this case and is demonstrated as

$$l(y_i, \hat{y}_i^t) = -(y_i \log(P_i) + (1 - y_i) \log(1 - P_i)) \quad (11)$$

where

$$p_i = \frac{1}{1 + e^{-\mathcal{Y}_i^t}} \quad (12)$$

Based on the P_i, y_i , and p values, the g_i and h_j values are evaluated.

$$g_i = P_i - y_i, h_i = p(1 - p_i). \quad (13)$$

From the $(t - 1)th$ tree of instance x_i , the evaluated forecasted value is projected as $\hat{y}_i^{(t-1)}$, in which the actual value of x_i is y_i . But, the predictive value is 0 for the 0^{th} tree, which implies $\hat{y}_i^{(0)} = 0$. **Step 2:** The Gain value of features required for traverse and is computed for determining the splitting mode for the present root node. The Gain value is support to evaluate the feature node with maximal Gain score.

Step 3: During this step, the establishment of the Current Binary Leaf Node setup is performed. Based on the feature with maximal Gain, the sample set can be categorized as 2 parts for obtaining 2 leaf nodes. Moreover, the second step can repeat to 2 leaf nodes assuming a negative gain score and end criteria, correspondingly. This step establishes the entire tree.

Step 4: Whole Leaf Node forecast values are computed in this step. Leaf node ω_j forecast values are computed as

$$\omega_j = -\frac{G_j}{H_j + \lambda} \quad (14)$$

and the second tree forecast outcomes are expressed as

$$y_i^{(2)} = y_i^{(1)} + f_2(x_i) \quad (15)$$

Afterward, this will result in establishing the second tree.

Step 5: The next step is to repeat steps 1 and 2 to set up further trees until a sufficient count of trees can be introduced. The predictive values of model $y_i^{(t)}$ are expressed as $\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_2(x_i)$, whereas $y_i^{(t)}$ refers to the predictive value of t trees on instance x_i . This procedure creates the t^{th} tree.

$$p_i = \frac{1}{1 + e^{-\hat{y}}} \quad (16)$$

Step 6: This equation that is utilized for determining the classifier outcome of an instance is to attain the probability by changing the last forecast value \hat{y} of the instance. If $p_i \geq 0.5$, the probability of the instance is 1; else, it is 0.

4. Results and discussion

In this section, the results of the LSO-DCNN technique are validated using two benchmark datasets: the sign language digital (SLD) dataset and the sign language gesture image (SLGI) dataset.

In Table 1 and Figure 3, the overall comparative recognition results of the LSO-DCNN technique are examined on the SLD dataset [21]. Based on $accu_y$, the LSO-DCNN technique reaches an increased $accu_y$ of 91.32%, while the RF, LR, KNN, XGBoost, and MobileNet-RF models obtain decreased $accu_y$ of 90.19%, 89.29%, 85.79%, 90.18%, and 90.55%, respectively. Next, based on $prec_n$, the LSO-DCNN approach reaches an increased $prec_n$ of 91.18%, while the RF, LR, KNN, XGBoost, and MobileNet-RF techniques obtain decreased $prec_n$ of 45.77%, 50.59%, 35.53%, 49.26%, and 80.97%, correspondingly. At the same time, based on $reca_l$, the LSO-DCNN algorithm attained an increased $reca_l$ of 91.31%, while the RF, LR, KNN, XGBoost, and MobileNet-RF approaches obtained decreased $reca_l$ of 48.67%, 44.55%, 35.83%, 50.12%, and 81.13%, respectively. Finally, based on $F1_{score}$, the LSO-DCNN method reaches an increased $F1_{score}$ of 91.78%, while the RF, LR, KNN, XGBoost, and MobileNet-RF models obtain decreased $F1_{score}$ of 46.75%, 44.56%, 34.07%, 49.31%, and 80.10%, correspondingly.

Table 1. Comparative analysis of the LSO-DCNN approach with other systems on the SLD dataset.

Sign Language Digital Dataset				
Methods	Accuracy	Precision	Recall	F1 score
Random Forest	90.19	45.77	48.67	46.75
Logistic Regression	89.29	50.59	44.55	44.56
K-Nearest Neighbor	85.79	35.53	35.83	34.07
XGBoost	90.18	49.26	50.12	49.31
MobileNet-RF	90.55	80.97	81.13	80.10
LSO-DCNN	91.32	91.18	91.31	91.78

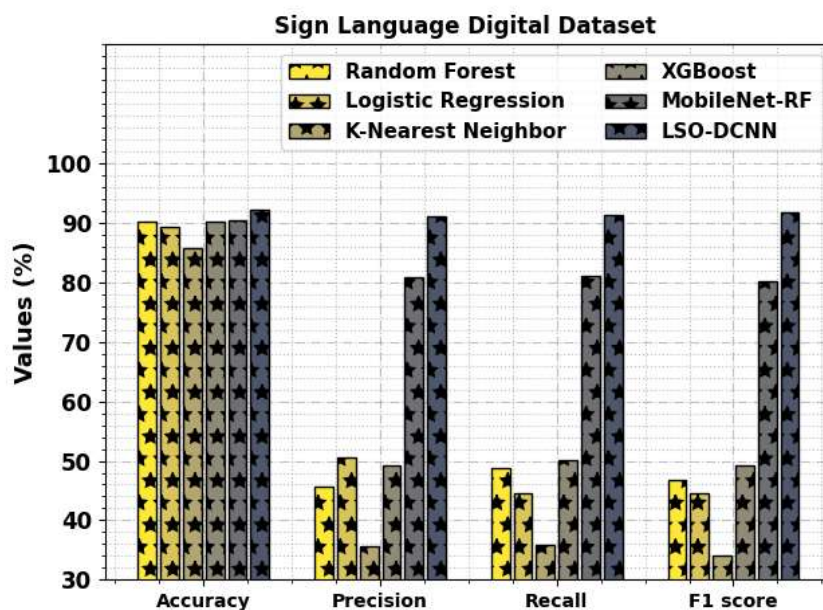


Figure 3. Comparative outcome of LSO-DCNN approach on the SLD dataset.

Figure 4 inspects the accuracy of the LSO-DCNN method in the training and validation of the SLD dataset. The figure notifies that the LSO-DCNN method has greater accuracy values over higher epochs. Furthermore, the higher validation accuracy over training accuracy portrays that the LSO-DCNN approach learns productively on the SLD dataset.

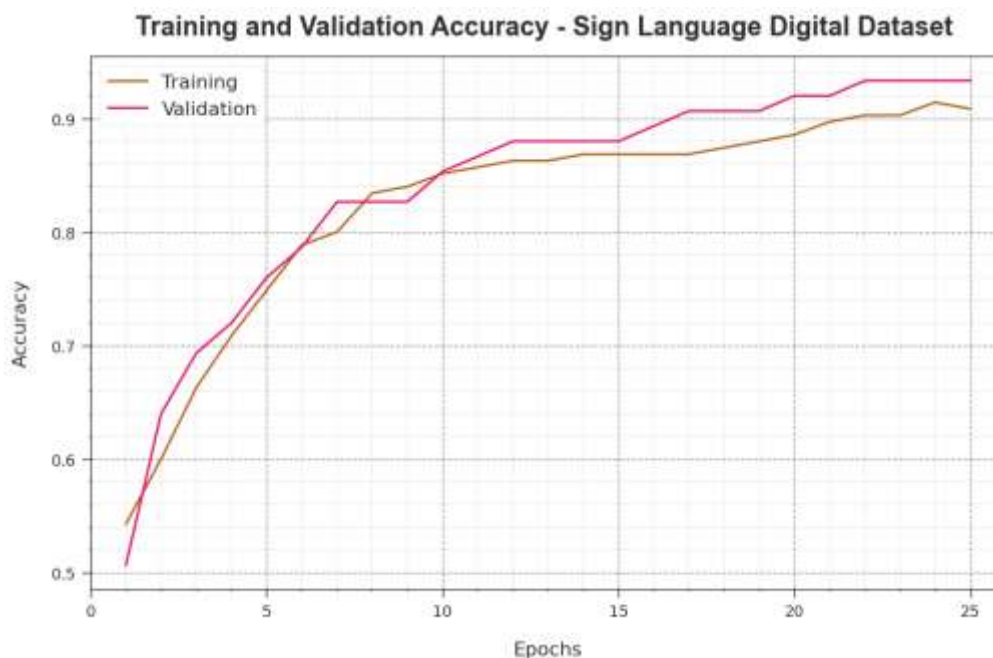


Figure 4. Accuracy curve of LSO-DCNN approach on the SLD dataset.

The loss analysis of the LSO-DCNN technique in the training and validation is given on the SLD dataset in Figure 5. The results indicate that the LSO-DCNN approach attained adjacent values of training and validation loss. The LSO-DCNN approach learns productively on the SLD database.



Figure 5. Loss curve of LSO-DCNN approach on the SLD dataset.

In Table 2 and Figure 6, the overall comparative recognition outcomes of the LSO-DCNN technique are examined on the SLGI dataset. Based on $accu_y$, the LSO-DCNN technique reaches an increased $accu_y$ of 99.09%, while the RF, LR, KNN, XGBoost, and MobileNet-RF approaches gain decreased $accu_y$ of 97.93%, 97.93%, 93.40%, 98.25%, and 98.31%, correspondingly. Next, based on $prec_n$, the LSO-DCNN methodology reaches an increased $prec_n$ of 98.86%, while the RF, LR, KNN, XGBoost, and MobileNet-RF approaches obtain decreased $prec_n$ of 29.08%, 20.49%, 27.34%, 31.15%, and 98.12%, correspondingly. Simultaneously, based on $reca_l$, the LSO-DCNN method reaches an increased $reca_l$ of 99.15%, while the RF, LR, KNN, XGBoost, and MobileNet-RF models obtain decreased $reca_l$ of 30.33%, 23.37%, 27.98%, 31.78%, and 98.11%, correspondingly. Eventually, based on $F1_{score}$, the LSO-DCNN technique reaches an increased $F1_{score}$ of 99.03%, while the RF, LR, KNN, XGBoost, and MobileNet-RF approaches obtain decreased $F1_{score}$ of 29.10%, 19.77%, 27.30%, 30.03%, and 97.89%, correspondingly.

Table 2. Comparative analysis of the LSO-DCNN approach with other methods on the SLGI dataset.

Sign Language Gestures Image Dataset				
Methods	Accuracy	Precision	Recall	F1 score
Random Forest	97.93	29.08	30.33	29.10
Logistic Regression	97.93	20.49	23.37	19.77
K-Nearest Neighbor	93.40	27.34	27.98	27.30
XGBoost	98.25	31.15	31.78	30.03
MobileNet-RF	98.31	98.12	98.11	97.89
LSO-DCNN	99.09	98.86	99.15	99.03

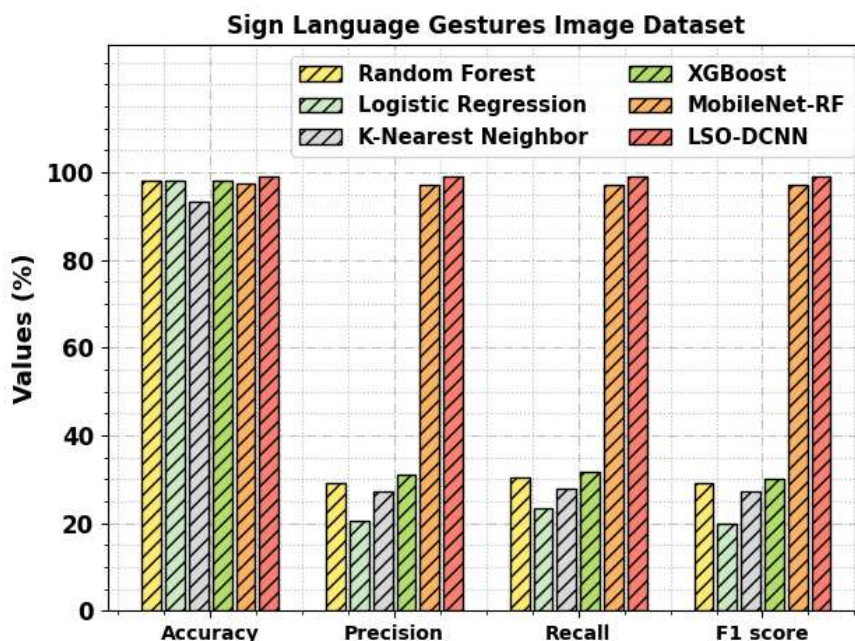


Figure 6. Comparative outcome of LSO-DCNN approach on the SLGI dataset.

Figure 7 portrays the accuracy of the LSO-DCNN method in the training and validation of the SLGI database. The result shows that the LSO-DCNN technique has higher accuracy values over greater epochs. Moreover, the higher validation accuracy over training accuracy shows that the LSO-DCNN technique learns productively on the SLGI database.

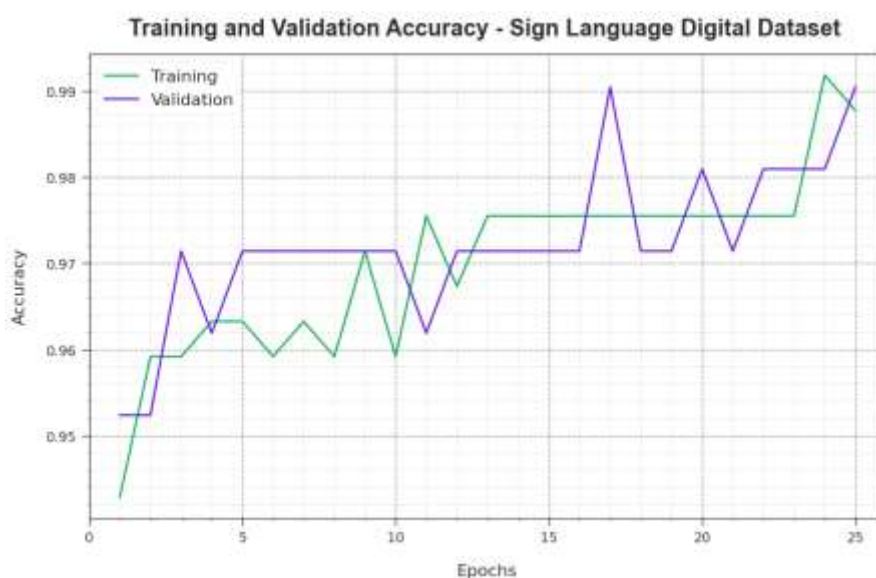


Figure 7. Accuracy curve of LSO-DCNN approach on the SLGI dataset.

The loss analysis of the LSO-DCNN approach in the training and validation is shown on the SLGI dataset in Figure 8. The results indicate that the LSO-DCNN method reaches adjacent values of training and validation loss. The LSO-DCNN method learns productively on the SLGI database.

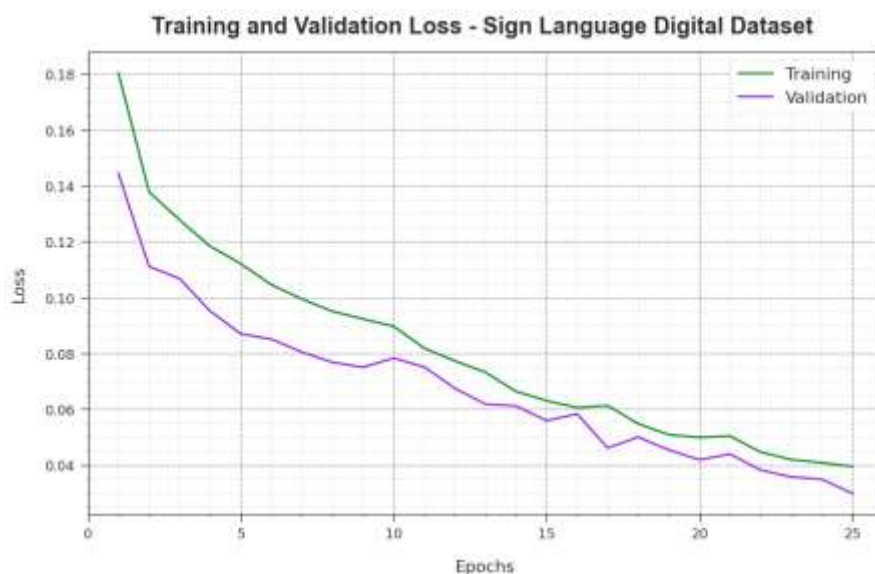


Figure 8. Loss curve of LSO-DCNN approach on the SLGI dataset.

5. Conclusions

This study developed a new LSO-DCNN technique for automated gesture recognition and classification. The major intention of the LSO-DCNN approach lies in the proper identification and categorization of various categories of gestures that exist in the input images. The presented LSO-DCNN model follows a three-step procedure, namely 1D-CNN based feature extraction, LSO-based hyperparameter tuning, and XGBoost classification. In this work, the LSO method optimally chooses the hyperparameter values of the 1D-CNN model and it helps to recognize the gestures efficaciously. To prove the enhanced gesture classification results of the LSO-DCNN approach, a wide range of experimental results are investigated. The brief comparative study reported the improvements in the LSO-DCNN technique in the gesture recognition process. In the future, multimodality concepts can enhance the performance of the LSO-DCNN technique.

Acknowledgments

The authors extend their appreciation to the King Salman center For Disability Research for funding this work through Research Group no KSRG-2023-175.

References

1. C. Dewi, A. P. S Chen, H. J. Christanto, Deep learning for highly accurate hand recognition based on Yolov7 model, *Big Data Cogn. Comput.*, **7** (2023), 53. <https://doi.org/10.3390/bdcc7010053>
2. J. John, S. P. Deshpande, Hand gesture identification using deep learning and artificial neural networks: A review, *Computational Intelligence for Engineering and Management Applications: Select Proceedings of CIEMA 2022*, 2023, 389–400. https://doi.org/10.1007/978-981-19-8493-8_30
3. R. Padmavathi, Expressive and Deployable Hand Gesture Recognition for Sign Way of Communication for Visually Impaired People, 2021.

4. A. Agarwal, A. Das, Facial Gesture Recognition Based Real Time Gaming for Physically Impairment. In *Artificial Intelligence: First International Symposium, ISAI 2022*, 2023, Haldia, India, February 17–22, 2022, Revised Selected Papers (256–264). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-22485-0_23
5. V. Gorobets, C. Merkle, A. Kunz, Pointing, pairing and grouping gesture recognition in virtual reality, In *Computers Helping People with Special Needs: 18th International Conference, ICCHP-AAATE 2022*, Lecco, Italy, July 11–15, 2022, Proceedings, Part I (313–320), Cham: Springer International Publishing. https://doi.org/10.1007/978-3-031-08648-9_36
6. J. Gangrade, J. Bharti, Vision-based hand gesture recognition for Indian sign language using convolution neural network, *IETE J. Res.*, **69** (2023), 723–732. https://doi.org/10.1007/978-3-031-08648-9_36
7. J. Li, C. Li, J. Han, Y. Shi, G. Bian, S. Zhou, Robust hand gesture recognition using HOG-9ULBP features and SVM model, *Electronics*, **11** (2022), 988. https://doi.org/10.1007/978-3-031-08648-9_36
8. D. Ryumin, D. Ivanko, E. Ryumina, Audio-visual speech and gesture recognition by sensors of mobile devices, *Sensors*, **23** (2023), 2284. https://doi.org/10.1007/978-3-031-08648-9_36
9. T. Sahana, S. Basu, M. Nasipuri, A. F. Mollah, MRCS: multi-radii circular signature based feature descriptor for hand gesture recognition, *Multimed. Tools Appl.*, **81** (2022), 8539–8560. <https://doi.org/10.1007/s11042-021-11743-w>
10. S. Pandey, Automated Gesture Recognition and Speech Conversion Tool for Speech Impaired. In *Proceedings of Third International Conference on Advances in Computer Engineering and Communication Systems: ICACECS 2022*, (467–476), Singapore: Springer Nature Singapore, 2023. https://doi.org/10.1007/978-981-19-9228-5_39
11. Y. Sun, Y. Weng, B. Luo, G. Li, B. Tao, D. Jiang, et al., Gesture recognition algorithm based on multi-scale feature fusion in RGB-D images, *IET Image Process.*, **17** (2023), 1280–1290. <https://doi.org/10.1049/ipr2.12712>
12. R. Barioul, O. Kanoun, k-Tournament grasshopper extreme learner for FMG-Based gesture recognition, *Sensors*, **23** (2023), 1096. <https://doi.org/10.1049/ipr2.12712>
13. T. R. Gadekallu, M. Alazab, R. Kaluri, P. K. R Maddikunta, S. Bhattacharya, K. Lakshmana, Hand gesture classification using a novel CNN-crow search algorithm, *Complex Intell. Syst.*, **7** (2021), 1855–1868. <https://doi.org/10.1049/ipr2.12712>
14. M. Yu, G. Li, D. Jiang, G. Jiang, F. Zeng, H. Zhao, et al., Application of PSO-RBF neural network in gesture recognition of continuous surface EMG signals, *J. Intell. Fuzzy Syst.*, **38** (2020), 2469–2480. <https://doi.org/10.3233/JIFS-179535>
15. A. Sen, T. K. Mishra, R. Dash, A novel hand gesture detection and recognition system based on ensemble-based convolutional neural network, *Multimed. Tools Appl.*, **81** (2022), 40043–40066. <https://doi.org/10.3233/JIFS-179535>
16. Q. Gao, Z. Ju, Y. Chen, Q. Wang, C. Chi, An efficient RGB-D hand gesture detection framework for dexterous robot hand-arm teleoperation system, *IEEE T. Hum-Mach Syst.*, 2022. <https://doi.org/10.1109/THMS.2022.3206663>
17. P. S. Neethu, R. Suguna, D. Sathish, An efficient method for human hand gesture detection and recognition using deep learning convolutional neural networks, *Soft Comput.*, **24** (2020), 15239–15248. <https://doi.org/10.1109/THMS.2022.3206663>

18. X. Zhang, P. Han, L. Xu, F. Zhang, Y. Wang, L. Gao, Research on bearing fault diagnosis of wind turbine gearbox based on 1DCNN-PSO-SVM, *IEEE Access*, **8** (2020), 192248–192258. <https://doi.org/10.1109/THMS.2022.3206663>
19. J. Fu, J. Liu, D. Xie, Z. Sun, Application of fuzzy PID based on Stray Lion Swarm Optimization Algorithm in overhead crane system control, *Mathematics*, **11** (2023), 2170. <https://doi.org/10.1109/THMS.2022.3206663>
20. R. Jena, A. Shanableh, R. Al-Ruzouq, B. Pradhan, M. B. A. Gibril, M. A. Khalil, et al., Explainable Artificial Intelligence (XAI) model for earthquake spatial probability assessment in Arabian peninsula, *Remote Sens.*, **15** (2023), 2248. <https://doi.org/10.1109/THMS.2022.3206663>
21. F. Wang, R. Hu, Y. Jin, Research on gesture image recognition method based on transfer learning, *Procedia Comput. Sci.*, **187** (2021), 140–145. <https://doi.org/10.1109/THMS.2022.3206663>



AIMS Press

© 2024 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0>)