Prema Muthusamy, et. al. International Journal of Engineering Research and Applications
www.ijera.com
ISSN: 2248-9622, Vol. 14, Issue 9, September, 2024, pp: 68-80

RESEARCH ARTICLE OPEN ACCESS

An Optimization of Occlusion robust Spatio Temporal Hybrid Cue Network

Prema Muthusamy[1]*, Gomathi Pudupalayam Murugan[1]
[1]*Research Scholar,Department of Computer Science, P.K.R Arts College for Women, Erode- 638476, Tamilnadu, India*
[2]*Associate Profossor and Head,Department of Computer Science, P.K.R Arts College for Women, Erode-638476, Tamilnadu, India*
** mprema.hanuphd@gmail.com*

ABSTRACT
Sign language (SL) is used by deaf-mute individuals for communication but normal people finds hard to understand the SL which causes a communication gap between them. It is crucial to reduce the gap between physically challenged people and normal people. In recent days, Deep Learning (DL) methods are used for the prediction of SLs. Amongst, Occlusion-resistant Spatio-Temporal Hybrid Cue Network (OSTHCN) method is developed which utilizes Skeleton Occupancy Likelihood Map estimation using B-Spline curve in Dynamic Dense Spatio-Temporal Graph Convolutional Neural Network (DDSTGCNN) to refine the skeleton extraction and VGG11+1D-Convolutinal Neural Network (CNN) for full frame data. It employs BLSTM encoders, Connectionist Temporal Classification (CTC), and Self-Attention based LSTM decoders for sequence learning in ISL recognition and translation. However, the networks utilized in the sequence learning are conceived manually by assigning values to a large number of hyper-parameters which results in high computational complexity and lower accuracy performance. Hence, an automated hyperparameter optimization model is developed in this paper using Dove Swarm Optimization (DSO) to lower the computational complexity and enhance the accuracy results. The DSO is adopted for selecting the optimal hyper-parameters in sequence learning like number of neurons, number of hidden units, learning rate, weight decay, number of epochs, batch size, dropout rate, number of partitions, number of clusters per batch, momentum, optimizer and loss function. The essential principle of DSO are derived from the foraging behaviors of doves adopted in real-time applications. The initial population of dove forages represents the initial hyper-parameters and computation of each dove crumbs subsequent locations represents the search for the best hyper-parameter to find the optimal values for the sequence learning. The BLSTM, CTC and SA-LSTM are completely optimized using DSO and it is termed as Optimized OSTHCN (OOSTHCN). Finally, the test findings revealed that the OOSTHCN model achieves 97.86% of accuracy on ISL-CSLTR dataset compared to the existing models.

Keywords: Sign language, Deep Learning, Bidirectional Long Short-Term Memory, Connectionist Temporal Classification, Dove Swarm Optimization.

Date of Submission: 06-09-2024 Date of acceptance: 21-09-2024

I. INTRODUCTION

The World Health Organization (WHO) reports that over 5% of the global population is deaf, and mute individuals struggle with communication. These people communicate among themselves using sign language (SL) [1]. Deaf and speech-impaired people may benefit greatly from the use of SL, a method of communication based on the use of hand gestures and visual movements [2]. It is a visual language using both manual and non-manual signs without the use of words or phrases. Non-manual signals include things like facial expressions, mouth and head movements, and so on, while manual signs are things like hand and finger motions, hand orientation and gesture, and so forth [3]. Globally, there are several distinct of SLs for communication such as American SL (ASL), Arabic SL (ArSL) and Indian SL (ISL). However normal person rarely knows these signs and this becomes barrier in real world communication [4].

Sign Language Recognition (SLR) aims to automate translating signs into spoken or written language, enabling the hearing-impaired to communicate with the general population [5]. Single-word SLR systems have been formed by researchers, but continuous gestures have proven more difficult. The most difficult part of developing an automated SLR system is creating a modeling

framework that can collect sign gestures and associated phrases [6]. Static and dynamic are two modes of SL, while static signs are unchanging hand and face motions and dynamic signs may be further classified into isolated signs and continuous signs [7]. The isolated Sign Language Recognition (ISLR) has made progress in recognizing single alphabetic signs or words from a given segment of signing video clip [8]. However, the contextual interactions among signals significantly influence the phrase interpretation in ISLR. The Continuous SLR (CSLR) challenge necessitates the prediction of all continuous sign actions from video sequences without prior information of the spatial boundaries among succeeding signs. It is more significant than ISLR as it interprets larger segment of speech. This is more suitable for real-world transcription of SLs [9].

In recent days, the artificial intelligence (AI) models like machine learning (ML) and deep learning (DL) have expanded their potential applications in SLR research enabling significant improvements in the quality of life for people who rely on SL as their primary communication method [10]. ML models like Support Vector Machine (SVM), Naïve Bayes (NB), and Random Forest (RF), have been used to provide accurate SLR [11]. These models attempt to comprehend continuous parts of SL gestures, however they largely analyze static information, such as individual signals separated in time and place. [12]. However, due to their simplicity, these models struggle to capture sophisticated semantic hints making advanced models necessary for SL prediction.

Deep Learning (DL) models are innovative tools for solving SLR problems. These models employs data from multiple sources to recognize SL terms with success varying on the dataset employed for the recognition task [13]. The DL models includes Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long-Short Time Memory (LSTM), and Deep Belief Network (DBN). DL-based applications enable SL translation to text, improving communication between signers and non-signers. However, challenges in SLR applications like as continuous speech interpretation or real-time translation may need the addition of additional layers [14]. DL models are considered the safest option for CSLR applications in the future improving the predictability of SLs for deaf or hearing-impaired individuals [15].

Many DL models have presented for CSLR system. Typically, DL models typically emphasize on the most discriminative traits while neglecting relatively non-trivial and beneficial content. These features severely limit their capacity to acquire latent visual grammars in sign videos based on their interaction with several visual signals (e.g., hand form, face expression, and body position) [16]. The Spatial-Temporal Multi-Cue (STMC) network [17] is a video-based SL that was developed to address vision-based sequence learning difficulties. It represents numerous cues using a spatial multi-cue (SMC) module and a temporal multi-cue (TMC) module. The SMC module is responsible for learning spatial representations, while the TMC module is responsible for modeling temporal adjustments from both intra- and inter-cue perspectives. The STMC model outperforms single-cue approaches on large SL datasets, but requires higher pre-processing time and key-point annotation supervision for end-to-end training.

In order to solve the above-mentioned issues, the STHCN method [18] is developed for recognizing and translating ISL from videos using DDSTGCNN and VGG11+1D-CNN + BLSTM. The DDSTGCNN learns spatial features while DSTCNM captures temporal features through 1D convolution layers. The extracted features are fed into a BLSTM encoder, CTC, and SA-LSTM decoders for sequence learning. The CTC predicts ISL from input videos and sentences. On the other hand, this model finds to match some pixels of the human skeleton structure to network models because of the occlusion of the human body. So, OSTHCN model [19] is developed that addresses occlusions in human skeleton extraction for ISL recognition and translation. The B-Spline curve's skeleton occupancy likelihood map estimation is used to optimize tasks by estimating unconnected skeletal subgraphs due to occlusion by fingers and hands. Heuristic assumptions simulate a 3D probability map, modified based on observed branch clusters across images. The collected features are then fed into the BLSTM encoders, CTC, and SA-LSTM decoders for sequence learning, enhancing ISL recognition and translation. However, the accuracy of the classifier mainly dependent on assigning values for hyper-parameters employed in sequence learning of OSTHCN.

In this paper, an OOSTHCN model is developed to lower the computational complexity and enhance the accuracy performance for ISL recognition and translation. This models adopts for DSO model where their fundamental ideas are derived from the foraging behaviors of doves employed in real-time application. DSO effectively selects the optimal hyper-parameters in sequence learning like number (No.) of neurons, No. of hidden units, learning rate, weight decay, epochs, batch size, dropout rate, No. of partitions, No. of clusters per batch, momentum, optimizer and loss function. The initial population of dove forages represents the initial hyper-parameters and computation of each

dove crumbs subsequent locations represents the search for the best hyper-parameter to find the optimal values for the sequence learning. Comparing to other optimization algorithms, DSO provides effective population diversity and eliminates complexity in the model which automatically increases the accuracy rate in the ISL detection and transcription.

The remainder of this article is organized as follows: the research performed for recognizing the different SLs is presented in Section II. The paradigm is explained in Section III, and its effectiveness is shown in Section IV. Conclusions and suggestions for further research are presented in Section V.

II. LITERATURE SURVEY

Zhou et al. [20] introduced SIGNBERT, a Bert-based DL framework for CSLR which combines BERT bidirectional encoder representations with residual neural network (ResNet) to model SLs and extract spatial features. SignBERT's multimodal version combines hand image input with intelligent feature alignment, narrowing the gap between BERT model recognition scores and CSLR hand photos. However, this strategy has resulted in a lengthy training period.

Gomathi et al. [21] suggested ConvNet-LSTM model for ISL recognition. This method adopts the gesture videos of ISL signs to process and extract spatial features. The videos were converted into image frames and passed to the Inception V3-CNN for feature selection and extraction. The CNN automatically extracts features which are then grouped into feature sequences and fed to the LSTM-RNN network for ISL detection. But still, this model does not suitable to recognize large featured words for recognition tasks.

Natarajan et al. [22] used a Hybrid Deep Neural Architecture (H-DNA) to create a framework for real-time ISL identification, translation, and video creation. Using the MediaPipe package and a hybrid CNN/LSTM model, the model captures posture information and outputs text. It also employs a hybrid Neural Machine Translation (NMT) MediaPipe Dynamic Generative Adversarial Network (GAN) model for sign gesture video generation and text prediction. However, this model results in a slower convergence rate.

Kothadiya et al. [23] proposed a DL model that detects appropriate words based on a person's gestures. Isolated ISL video frames were used as input sources and divided into individual sub-section videos. The InceptionResNetV2 model extracted the gesture characteristics and fed them into a RNN for ISL prediction. However, non-stable and angular input data frames needed to be focused to enhance accuracy.

Katoch et al. [24] created a model that recognizes ISL alphabets and digits in live video streams by combining SVM and CNN. They processed skin color and background removal using the Bag of Visual Words model (BOVW). The SURF characteristics were extracted from pictures, and histograms were created to map signs with appropriate labels. The model was used for the detection task. However, this model takes a long time to compute.

Subramanian et al. [25] constructed a unified Mediapipe-Optimized GRU (MOPGRU) model for ISL recognition. This model was categorized into three stages like data pre-processing and feature extraction; capturing extracted keypoints in a file, training and classifying the gestures using sign motions which have been translated in the form of text on the screen. However, this model was trained on limited dataset and a high temporal complexity.

Mannan et al. [26] developed a hyper-tuned deep convolutional neural network (HDCNN) for SLR. The collected data was augmented and then data generator was applied to expand the size of the training dataset. The features were extracted using pre-trained CNN model. Finally, the HDCNN model was employed for the ISL recognition and classification. However, this model results with high uncertainty issues.

Venugopalan & Reghunadhan, [27] constructed a CNN-BLSTM model for ISL detection utilized for the emergence of deaf COVID-19 patients. The system converts gesture videos into images sequences using ISL words, which are then fed into a CNN model for spatial feature extraction. Feature vectors are concatenated to create a series, which are then classified using an LSTM network for ISL prediction. However, this model results with high time complexity issues.

Sreemathy et al. [28] presented a CSLR recognition using an expert system based on DL model. Two hand gesture recognition systems, SVM with media-pipe and YOLO, are used as feature extractors. The system compares the accuracy of the two models and produces a result based on which model has better performances while training and detection confidence in real-time. However, this model provides lower accuracy results on smaller datasets.

Hu et al. [29] proposed an correlative network (CorrNet) to directly exploit body movements across frames for identifying CSLR activities. The correlation module produced correlation maps between the current and neighboring frames to record cross-frame routes,

creating features as local temporal motions, while the recognition module identified relevant locations in each frame for expressing a sign. However, a high number of instances were needed in the training database.

Buttar et al. [30] constructed a DL approach to detect both stationary and moving signals for SLR. The LSTM with Skeleton model technique was used to sequentially extract characteristics from each frame of the SL videos. Next, the detection tasks' custom dataset was trained using YOLOv6. However, the model's accuracy dropped down sharply when a certain threshold was reached, thus it could only be used with a restricted set of signs.

Alnfiai [31] introduced an automated sign language recognition (SSODL-aSLR) model for deaf and stupid people based on shark smell optimization with DL model. A mask region based convolutional neural network (Mask RCNN) model was used in the SLR procedure. We employed the SSO approach in combination with light boundary SVM (SM-SVM) model to categorize SL. However, the convergence pace was slower with this model.

III. PROPOSED METHODOLOGY

In this section, the complete framework of OOSTHCN is briefly illustrated. The figure 1 depicts the schematic diagram of the suggested model.

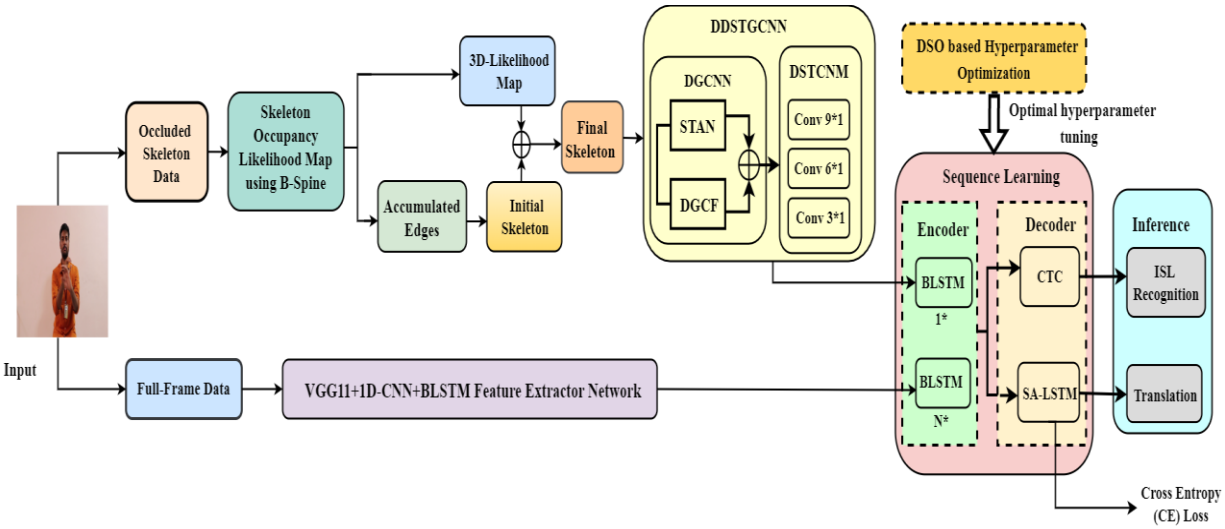

Figure 1. Schematic Representation of OOSTHCN Model

In this model, the BLSTM encoders, CTC, and SA-LSTM decoders are used for sequence learning in OSTHCN [19]. The hyperparameter of sequence learning segments i.e., BLSTM, CTC and SALSTM are the No. of neurons, No. of hidden units, learning rate, weight decay, epochs, batch size, dropout rate, partitions numbers, No. of clusters per batch, momentum, optimizer and loss function. All these hyper parameter values are completely optimized by the DSO model to provide efficient ISL recognition and translation with reduced complexity and enhance the recognition accuracy.

3.1 Dove Swarm Optimization based Hyper-parameter Tuning

Doves generally scavenge in areas where crumbs are visible and hunt for them. Some doves may be content, but not all. Unsatisfied doves loiter in patches, looking for more crumbs. Gradually, it becomes clear that the full doves must have inhabited areas with more crumbs. Dove foraging behavior prompted the development of a revolutionary optimum algorithm. The optimization goal operation in this approach is $F(X) = [f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9, f_{10}, f_{11}, f_{12}]$. Each optimal location is defined by the hyperparameters No. of neurons (f_1), No. of hidden units (f_2), learning rate (f_3), weight decay (f_4), epochs (f_5), batch size (f_6), dropout rate (f_7), No. of partitions (f_8), No. of clusters per batch (f_9), momentum (f_{10}), optimizer (f_{11}) and loss function (f_{12}) at a data collection, every data pattern X is regarded a place containing crumbs, and the number of crumbs at these spots X includes $F(X)$ crumbs. The optimum outcome identifies the region with the greatest number of crumbs.

Step 1: Determine the number of doves that needed to be deployed in the solution space. Consider n as the number of doves. These doves may be scattered arbitrarily in space, but it is recommended to deploy them evenly across a rectangular surface.

Step 2: For dove $D = 1, \dots, n$, modify the number of epoch $e = 0$ and the degree of satiety $\mathcal{S}_D^e$. There are two methods to initialize the position vector $X_D \subset r^k$ for dove D. The most straightforward approach is to dynamically generate the X_D around the solution space. The lattice initialized approach is another option. The appropriate stages are shown below.

- To accelerate up the training task for creating a structurally sorted feature map, two efficient weight initiation procedures are used to create the weight vectors. Based on the initialization method, a distinctive initialization method that is especially appropriate for this algorithm is devised. Assume the parameter space has the smallest hyper-rectangle, which includes the acceptable values for all parameters as $[L_1, U_1], \dots, [L_k, U_k]$ where L_q and U_q represent the lower and upper bounds of the q-dimension in the solution space.
- The key idea underlying the proposed initialization method is to compress the $N-$dimensional hyper-rectangle into a two-dimensional (2D) plane such that a 2D-net may effectively enclose the resultant space. To make things apparent a and b are used to index the rectangular cells from 1 to $U * V$. The following steps are illustrated below.

Step 2.1: Construct the cells in the all four edges. Also, adjust the four neurons on the network's corners with the following weight vectors ($\mathcal{W}$) as given in Eq. (1.1 – 1.4),

$$\mathcal{W}_{1,1} = (L_1, L_2, \dots\dots, L_k)^T \quad (1.1)$$

$$\mathcal{W}_{U,V} = (U_1, U_2, \dots\dots, U_k)^T \quad (1.2)$$

$$\mathcal{W}_{1,V} = (L_1, L_2, , , , , , , L_{\lfloor k/2 \rfloor}, \ U_{\lfloor k/2 \rfloor + 1}, \dots\dots,)^T \quad (1.3)$$

$$\mathcal{W}_{U,1} = (U_1, U_2, , , , , , , U_{\lfloor k/2 \rfloor}, L_{\lfloor k/2 \rfloor + 1}, \dots\dots, L_k)^T \quad (1.4)$$

Step 2.2: Configure the cells on the four edges. The Eq. (2.1 – 2.4) represents the initialled cell values on the four edges,

$$\mathcal{W}_{1,b} = \frac{\mathcal{W}_{1,V} - \mathcal{W}_{1,1}}{V-1} (b-1) + \mathcal{W}_{1,1}$$

$$= \frac{b-1}{V-1} \mathcal{W}_{1,V} + \frac{V-b}{V-1} \mathcal{W}_{1,V} \quad b = 2, \dots, V-1 \quad (2.1)$$

$$\mathcal{W}_{U,b} = \frac{\mathcal{W}_{U,V} - \mathcal{W}_{U,1}}{V-1} (b-1) + \mathcal{W}_{U,1}$$

$$= \frac{b-1}{V-1} \mathcal{W}_{U,V} + \frac{V-b}{V-1} \mathcal{W}_{U,1} \quad b = 2, \dots, V-1 \quad (2.2)$$

$$\mathcal{W}_{a,1} = \frac{\mathcal{W}_{U,1} - \mathcal{W}_{1,1}}{U-1} (a-1) + \mathcal{W}_{1,1}$$

$$= \frac{a-1}{V-1} \mathcal{W}_{U,1} + \frac{U-a}{U-1} \mathcal{W}_{1,1} \quad a = 2, \dots, U-1 \quad (2.3)$$

$$\mathcal{W}_{a,V} = \frac{\mathcal{W}_{U,V} - \mathcal{W}_{1,V}}{U-1} (a-1) + \mathcal{W}_{1,V}$$

$$= \frac{a-1}{U-1} \mathcal{W}_{U,V} + \frac{V-a}{V-1} \mathcal{W}_{1,V} \quad a = 2, \dots, U-1 \quad (2.4)$$

Step 2.3: The weight gradients of the four neurons on the four edges of the network have been generated. The additional neurons are arranged top to bottom and left to right. The following is a pseudo-code description of the residual neuron activation procedure:
Start

For $b = 2\ to\ V-1$
Start
For $a = 2\ to\ U-1$
Initialize

$$\left.\begin{aligned} \mathcal{W}_{a,b} &= \frac{\mathcal{W}_{U,b}-\mathcal{W}_{1,b}}{U-1}\,(a-1)+\mathcal{W}_{1,b} \\ &= \tfrac{a-1}{U-1}\,.\,\mathcal{W}_{U,b}+\tfrac{U-a}{U-1}\,.\,\mathcal{W}_{1,b} \\ &= \tfrac{a-1}{U-1}\left(\tfrac{b-1}{V-1}\,.\,\mathcal{W}_{U,V}+\tfrac{V-b}{V-1}\,.\,\mathcal{W}_{U,1}\right)+ \\ &= \tfrac{U-a}{U-1}\left(\tfrac{b-1}{V-1}\,.\,\mathcal{W}_{1,V}+\tfrac{V-b}{V-1}\,.\,\mathcal{W}_{1,1}\right)+ \\ &= \tfrac{((b-1)(a-1)\mathcal{W}_{U,V}+(b-1)(U-a)\mathcal{W}_{1,V}+(V-b)(a-1)}{(V-1)(U-1)} \end{aligned}\right\} \quad (3)$$

End;
End;
End;

The different sizes are evaluated and analyzed for the final result. The highest numbers of all neurons and their variants are calculated to measure training efficiency. The initial value of the learning rate (ℓ) is set to 0.1 in Eq. (4), and the decreasing rate of ℓ is given.

$$\ell_n = \ell_0 * \left(1-\frac{IN}{100}\right) = 0.1\,(1-\frac{IN}{100}) \quad (4)$$

In Eq. (4), the iterative number is IN, and the initial learning rate is ℓ_0.

Step 3: Compute the total crumbs number at the position of the dove D for all dove fitness functions $F(\mathcal{W}_b^e)$ where $b = 1,\ldots.n$ at epoch e.

Step 4: Discover the dove D_b^e nearest to the maximum crumbs using the highest criteria at epoch e, which is calculated as,

$$D_b^e = argmax\,\{F(\mathcal{W}_b^e)\}\ ,\ b = 1,\ldots n \quad (5)$$

Step 5: Apply the subsequent calculation to each dive's satiety degree.

$$\mathcal{S}_b^e = \text{ë}\mathcal{S}_b^{e-1} + e^{(F(\mathcal{W}_b)-F(\mathcal{W}_{D_F}))},\ \ b = 1,\ldots n \quad (6)$$

Step 6: Determine the highest contented with dove D_S^e with the greatest degree of satiety by utilizing the following maximum criteria, $b = 1,2,\ldots.n$

$$D_S^e = arg\max_{1\le b\le n}\langle\mathcal{S}_b^e\rangle \quad (7)$$

D_S as determined by Eq. (7), symbolizes the dove that demonstrates the best foraging behavior and deserves to be emulated by the rest of the flock.

Step 7: Change the position vector of each dove D using the following maximum criterion

$$\mathcal{W}_b^{e+1} = \mathcal{W}_b^e + \ell\,\text{ä}_b^e\,(\mathcal{W}_{D_s}^e - \mathcal{W}_b^e) \quad (8)$$

Where,

$$\text{ä}_b^e = \left(\frac{\text{ä}_{b_S}^e-\text{ä}_{b_S}^e}{\text{ä}_{b_S}^e}\right)(1-\frac{\left\|\mathcal{W}_b^e-\mathcal{W}_{D_S}^e\right\|}{Max\_Dis}) \quad (9)$$

$$Max\_Dis = \max_{1\le b\le n}\|\mathcal{W}_b - \mathcal{W}_a\| \quad (10)$$

Where, Max_Dis is the maximum distance. The learning rate to adjust the dove position vector is specified by the parameter ℓ . The next step provides extensive justifications of the updating Equations (8) - (10).

Step 8: Return to step 3 and increase the number of epochs by one ($for\ eg,\ e = e+1$) until the terminate condition is met. The following are the termination rules:

$$\left|F_{D_S}^e - T(e)\right| \le \text{ã}\ \ or\ e\ \le max\,epoch\ \big| \quad (11)$$

The complexity order of DSO is Onn_{D^e}, the number of data elements in the dataset is n_D, n is the number of doves and e is the number of epochs. If one of the optimization criteria is to find the least $\mathcal{W}_b^e$, the order of (5) and (6) may be altered accordingly.

$$D_b^e = argmin\,\{F\,(\mathcal{W}_b^e)\} \qquad b = 1,\ldots.,n \quad (12)$$

$$\mathcal{S}_b^e = \begin{cases} \ddot{e}\mathcal{S}_b^{e+1} + e^{\left(F(\mathcal{W}_b) - F\left(\mathcal{W}_{D_F}\right)\right)} & F\left(\mathcal{W}_{D_F}\right) \neq 0 \\ \ddot{e}\mathcal{S}_b^{e-1} + 1, & F\left(\mathcal{W}_{D_F}\right) = 0 \\ b = 1, \dots, n & \end{cases} \quad (13)$$

For better clarity, the updating rules are interpreted as given in Eq. (8)-(10) as follows:

1. Doves in a flock are motivated by the largest individual's accomplishment and seek to duplicate it. They follow the dove with the greatest joy in order to find more food. This social acquisition is duplicated by altering the position vector $\mathcal{W}_{D_S}^e$ to more precisely approach the position vector of the dove with the highest degree of satiety, i.e.,

$$\mathcal{W}_b^{e+1} = \mathcal{W}_b^e + \ell\ddot{a}_b^e \left(\mathcal{W}_{D_S}^e - \mathcal{W}_b^e\right) \quad (4)$$

2. A dove with higher satiety is more cautious and hesitant to change its foraging strategy, while a dove with lower satiety is more likely to modify its strategy and emulate the best individual behavior. This societal effect is demonstrated by comparing modifications to the first term value on the right hand side (RHS) of Eq. (9) is stated as follows,

$$\left(\frac{\mathcal{S}_{j_S}^e - \mathcal{S}_j^e}{\ddot{a}_{j_S}^e}\right) \quad (15)$$

3. The extent of social influence diminishes with distance, demonstrating that a dove's effect is proportionally related to the distances among it and the flock's best dove. This kind of societal influence is mimicked by making the degree of adaptation equivalent to the value of the third variable on the right-hand side of Eq. (9) i.e., $\left((1 - \frac{\left\| w_b^e - w_{D_s}^e \right\|}{Max\_Dis}\right)$.

Hence, the optimal hyper-parameters in sequence learning (BLSTM, CTC and SALSTM) are optimized using DSO by substituting their population locations at the initialization stage. The DSO helps to lower the computational complexity and enhances the classification accuracy for ISL recognition and translation. The pseudocode of DSO for hyperparameter tuning the BLSTM, CTC and SALSTM is described in Algorithm 1. Also, an overall workflow of the DSO hyperparameter tuning is shown in Figure 2.

Algorithm 1: Hyperparameter tuning using DSO

Input: Set of hyperparameters for BLSTM, CTC and SALSTM model (i.e., No. of neurons, No. of hidden units, learning rate, weight decay, number of epochs, batch size, dropout rate, number of partitions, No. of clusters per batch, momentum, optimizer and loss function)

Output: Optimal hyperparameters

Begin

Initialize the No. of doves, initial dove locations, degree of satiety ($\mathcal{S}_D^e$), and the No. of epochs (e);

while($e < e\_max$)

Compute the fitness value of all doves;

Place the dove closest to the greatest quantity of crumbs;

Modify all doves' satiety degree values;

Choose the most satisfied dove with the maximum degree of satiety;

Modify all doves' location vector;

end while

Find the best dove (assigning optimal value for hyperparameters of BLSTM, CTC and SALSTM) in the search space, and the optimal fitness value;

End

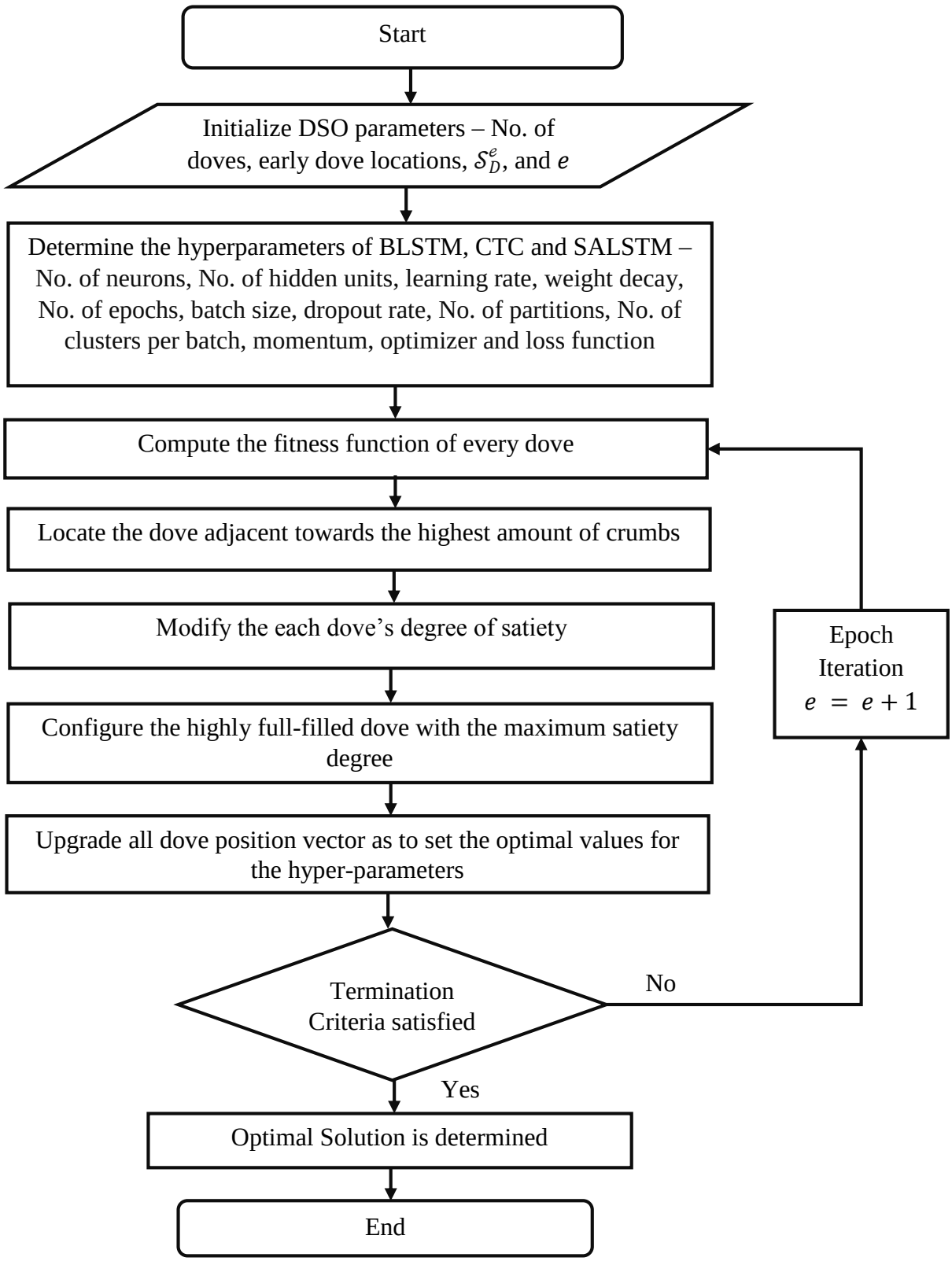

Figure 2 Flowchart of DSO for hyper-parameter selection of sequence learning

3.2 Model Training

The sequence learning model (BLSTM, SALSTM and CTC) for ISL-recognition and translation is trained with a set of optimal hyperparameters by the structure of DSO listed in Table 1.

Table 1. List of Optimal Hyperparameters for sequence learning models

Parameters	Search Space	Optimal Range
Sequence Learning		
Number of neurons	[16, 32, 48, 64, 80, 96, 112, 128]	96
Number of hidden units	[64, 128, 256, 512]	256
Learning rate	[0.01, 0.001, 0.0001]	0.001
Weight decay	[0.0001, 0.001]	0.0003
Number of epochs	[100, 500]	300
Batch size	[32, 64, 128, 512]	64
Dropout rate	[0.1, 0.2, 0.3, 0.5]	0.5
Number of partitions	[50, 100, 150, 200]	100
Number of clusters per batch	[1, 2, 3, 4]	3
Momentum	[0, 1]	0.7
Optimizer	[Stochastic gradient descent, Adam,]	Adam
Loss Function	[Cross-entropy, Mean Squared Error]	Cross-entropy
DSO		
Number of population	[80, 90, 100, 110, 120]	100
Maximum Number of cluster Iterations	[50, 60, 70, 80, 90]	80
Step size	[0.45, 0.55, 0.65, 0.75]	0.65
ë	[0.5, 0.6, 0.7, 0.8, 0.9]	0.9
ℓ	-	0.18~0.375

IV. RESULT AND DISCUSSION

4.1 Dataset Description

For the experimental purposes, The Indian SL Dataset for Continuous SL Translation and Recognition (ISL-CSLTR) dataset is employed [32]. The ISL-CSLTR corpus is a vast collection of 700 videos, 18863 sentence-level frames, and 1036 word-level images for 100 spoken language sentences performed by 7 different Signers. It is publicly available and aims to explore research outcomes in SLTR, helping researchers develop a framework for converting spoken language sentences into SL and vice versa. The corpus addresses challenges in SLRT and significantly improves translation and recognition performance.

4.2 Performance Analysis

In this section, the efficiency of the OOSTHCN model is examined by implementing it in Python using the dataset which is discussed in Section 4.1. For the experimental purposes, 610 videos have been finalized for the collected dataset, 60% (366) data are taken for training and the rest 40% (244) are taken for testing. From the collected dataset 86 sentences (labels) and 25 per second time frame rate have been determined for the final output. Further a comparative analysis is carried out to understand the improvement of the OOSTHCN, model contrasted to the existing models including STMC [17], SVM-CNN [24], CNN-BLSTM [27], LSTM-YOLOv6 [30], SSODL-aSLR [31] STHCN [18] and OSTHCN [19]. The assessment measures used to assess the effectiveness of the proposed and current models are shown briefly below.

4. 1. Accuracy: Accuracy is the ratio of successfully identified signs to the overall number used for classification, indicating the model's overall performance and training.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (14)$$

In above Eq. (14), If the model accurately identified the class of the sign, we say that we have a high proportion of True Positives (TP). When a True Negative (TN) is produced, it means that the model accurately predicted that a given symbol does not belong to a certain class. When a model predicts a class of signs that does not exist in reality, the

prediction is termed as false positive (FP). When the true class of a sign is expected to be false, this is called a False Negative (FN).

4.2 Precision: It is used in SLR to evaluate model performance; it primarily informs about FP results in the dataset. A higher accuracy score means fewer FP values.

$$Precision = \frac{TP}{TP+FP} \quad (15)$$

4.3 Recall: The recall score is used in SL identification to evaluate model performance; this score primarily informs about erroneous FN values in the dataset. A higher recall score means fewer false negative values.

$$Recall = \frac{TP}{TP+FN} \quad (16)$$

4.4 F1-Score: It is the average of the two measures of precisions and recall

$$F1 - score = 2 \times \frac{Precision \cdot Recall}{Precision+Recall} \quad (17)$$

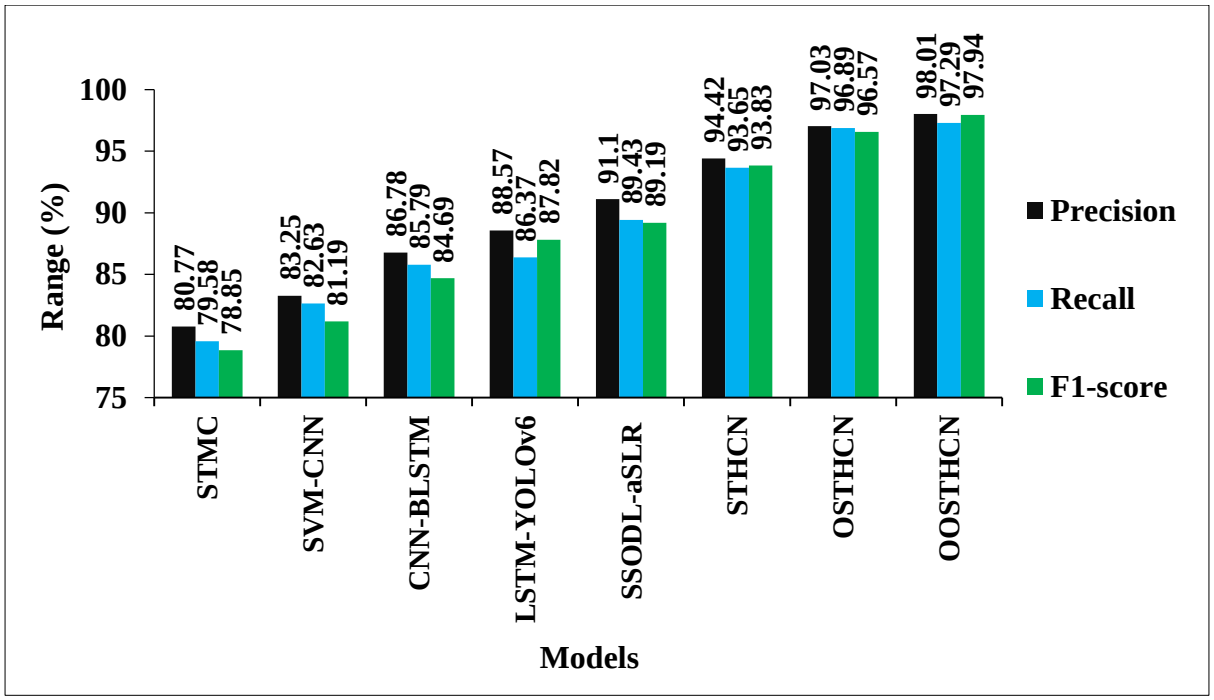

Figure 3. Comparison of Precision, Recall and F1-score for OOSTHCN against Existing SLR Classification Models on ISL-CSLRT Dataset

Figure 3 portrays the performance of OOSTHCN against different existing models on the test ISL-CSLRT dataset in terms of precision, recall and F1-score. It is noticed that the precision of OOSTHCN is increased up to 21.34%, 17.73%, 12.94%, 10.66%, 7.59%, 3.80% and 1.01% respectively contrasted with the STMC, SVM-CNN, CNN-BLSTM, LSTM-YOLOv6, SSODL-aSLR, STHCN and OSTHCN respectively. The recall of OOSTHCN is improved by 22.25 %, 17.74%, 13.40%, 12.64%, 8.79%, 3.89% and 0.41% compared to the STMC, SVM-CNN, CNN-BLSTM, LSTM-YOLOv6, SSODL-aSLR, STHCN and OSTHCN algorithms, accordingly. The F1-score of OOSTHCN is enhanced by 24.21%, 20.63%, 15.65%, 11.52%, 9.81%, 4.38% and 1.42% compared to the STMC, SVM-CNN, CNN-BLSTM, LSTM-YOLOv6, SSODL-aSLR, STHCN and OSTHCN, respectively. This is because of optimizing the model hyperparameters with faster convergence and less computational complexity by the DSO model. This improves the model performance while dealing with the ISL-CSLRT dataset for ISL recognition and translation.

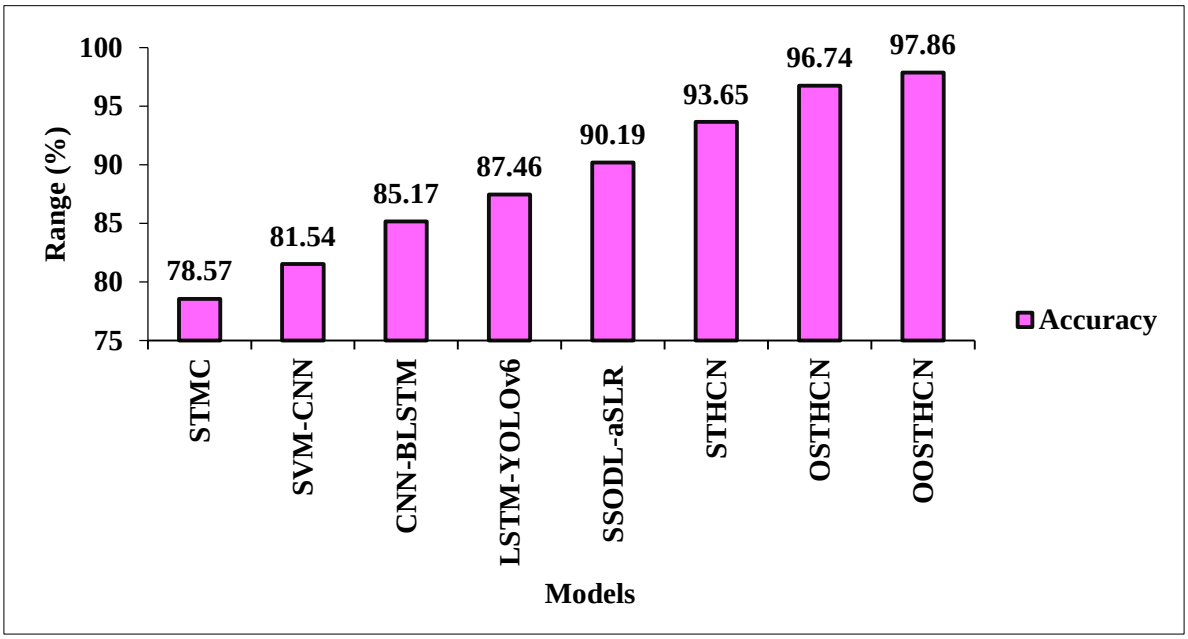

Figure 4. Accuracy Results for OOSTHCN and Existing SLR Models

As shown in Figure 4, the accuracy of the OOSTHCN model is by 24.55%, 20.01%, 14.89%, 11.89%, 8.50%, 4.49% and 1.16 STMC, SVM-CNN, CNN-BLSTM, LSTM-YOLOv6, SSODL-aSLR, STHCN and OSTHCN respectively test ISL-CSLRT dataset. Therefore, it is inferred that the OOSTHCN model is more effective than other SLR models. This is achieved by adopting the DSO for well fine-tuning the hyperparameters of BLSTM, CTC and SA-LSTM with lower computational complexity and enhanced accuracy results to get effective global solutions for ISL recognition and translation.

V. CONCLUSION

This paper presents an OOSTHCN model using DSO to reduce computational complexity and improve accuracy for ISL recognition and translation. The DSO is adopted to select optimal hyper-parameters like learning like No. of neurons, No. of hidden units, learning rate, weight decay, No. of epochs, batch size, dropout rate, No. of partitions, No. of clusters per batch, momentum, optimizer and loss function. This model is based on real-time dove foraging behaviors with the initial population representing hyper-parameters and subsequent locations representing the search for optimal values. The OOSTHCN is completely optimized using DSO and achieves a 97.86 % accuracy on the ISL-CSLTR dataset compared to existing models. In future, mobile applications based DL model will be developed for real-time sign recognition and translation.

REFERENCES

[1]. Naseribooriabadi, T., Sadoughi, F., & Sheikhtaheri, A. (2017). Barriers and facilitators of health literacy among D/deaf individuals: A review article. Iranian journal of public health, 46(11), 1465.

[2]. Rastgoo, R., Kiani, K., & Escalera, S. (2021). Sign language recognition: A deep survey. Expert Systems with Applications, 164, 113794.

[3]. Yang, H. D., & Lee, S. W. (2011, July). Combination of manual and non-manual features for sign language recognition based on conditional random field and active appearance model. In 2011 International Conference on Machine Learning and Cybernetics (Vol. 4, pp. 1726-1731). IEEE.

[4]. Sahoo, A. K., Mishra, G. S., & Ravulakollu, K. K. (2014). Sign language recognition: State of the art. ARPN Journal of Engineering and Applied Sciences, 9(2), 116-134.

[5]. Nimisha, K. P., & Jacob, A. (2020, July). A brief review of the recent trends in sign language recognition. In 2020 International Conference on Communication and Signal Processing (ICCSP) (pp. 186-190). IEEE.

[6]. Obi, Y., Claudio, K. S., Budiman, V. M., Achmad, S., & Kurniawan, A. (2023). Sign language recognition system for communicating to people with disabilities. Procedia Computer Science, 216, 13-20.

[7]. Susitha, A., Geetha, N., Suhirtha, R., & Swetha, A. (2022). Static and Dynamic Hand Gesture Recognition for Indian Sign Language. In Machine Learning and Big Data Analytics (Proceedings of International Conference on Machine Learning and Big Data Analytics (ICMLBDA) 2021) (pp. 48-66). Springer International Publishing.

[8]. Dawod, A. Y., & Chakpitak, N. (2019, August). Novel technique for isolated sign language based on fingerspelling recognition. In 2019 13th International Conference on Software, Knowledge, Information Management and Applications (SKIMA) (pp. 1-8). IEEE.

[9]. Papastratis, I., Dimitropoulos, K., Konstantinidis, D., & Daras, P. (2020). Continuous sign language recognition through cross-modal alignment of video and text embeddings in a joint-latent space. IEEE Access, 8, 91170-91180.

[10]. Sreemathy, R., Turuk, M., Kulkarni, I., & Khurana, S. (2023). Sign language recognition using artificial intelligence. Education and Information Technologies, 28(5), 5259-5278.

[11]. Adeyanju, I. A., Bello, O. O., & Adegboye, M. A. (2021). Machine learning methods for sign language recognition: A critical review and analysis. Intelligent Systems with Applications, 12, 200056.

[12]. Munnaluri, V., Pandey, V., & Singh, P. (2022, June). Machine Learning based Approach for Indian Sign Language Recognition. In *2022 7th International Conference on Communication and Electronics Systems (ICCES)* (pp. 1128-1132). IEEE.

[13]. Al-Qurishi, M., Khalid, T., & Souissi, R. (2021). Deep learning for sign language recognition: Current techniques, benchmarks, and open issues. IEEE Access, 9, 126917-126951.

[14]. Al-Hammadi, M., Muhammad, G., Abdul, W., Alsulaiman, M., Bencherif, M. A., Alrayes, T. S., ... & Mekhtiche, M. A. (2020). Deep learning-based approach for sign language gesture recognition with efficient hand gesture representation. Ieee Access, 8, 192527-192542.

[15]. Mittal, A., Kumar, P., Roy, P. P., Balasubramanian, R., & Chaudhuri, B. B. (2019). A modified LSTM model for continuous sign language recognition using leap motion. IEEE Sensors Journal, 19(16), 7056-7063.

[16]. Cui, R., Liu, H., & Zhang, C. (2019). A deep neural framework for continuous sign language recognition by iterative training. IEEE Transactions on Multimedia, 21(7), 1880-1891.

[17]. Zhou, H., Zhou, W., Zhou, Y., & Li, H. (2021). Spatial-temporal multi-cue network for sign language recognition and translation. IEEE Transactions on Multimedia, 24, 768-779.

[18]. Prema, M., & Gomathi, P.M (2023) Recognition of Indian Continuous Sign Language Using Spatio-Temporal Hybrid Cue Network. International Journal of Inventive Engineering and Sciences.

[19]. Prema, M., & Gomathi, P.M (2023) Occlusion Resistant Spatio-Temporal Hybrid Cue Network For Indian Sign Language Using Recognition.

[20]. Zhou, Z., Tam, V. W., & Lam, E. Y. (2021). SIGNBERT: a Bert-based deep learning framework for continuous sign language recognition. IEEE Access, 9, 161669-161682.

[21]. Gomathi, V. (2021). Indian Sign Language Recognition through Hybrid ConvNet-LSTM Networks. EMITTER International Journal of Engineering Technology, 9(1), 182-203.

[22]. Natarajan, B., Rajalakshmi, E., Elakkiya, R., Kotecha, K., Abraham, A., Gabralla, L. ., & Subramaniyaswamy, V. (2022). Development of an End-to-End Deep Learning Framework for Sign Language Recognition, Translation, and Video Generation. IEEE Access, 10, 104358-104374.

[23]. Kothadiya, D., Bhatt, C., Sapariya, K., Patel, K., Gil-González, A. B., & Corchado, J. M. (2022). Deepsign: Sign language detection and recognition using deep learning. Electronics, 11(11), 1780.

[24]. Katoch, S., Singh, V., & Tiwary, U. S. (2022). Indian Sign Language recognition system using SURF with SVM and CNN. Array, 14, 100141.

[25]. Subramanian, B., Olimov, B., Naik, S. M., Kim, S., Park, K. H., & Kim, J. (2022). An integrated mediapipe-optimized GRU model for Indian sign language recognition. Scientific Reports, 12(1), 11964.

[26]. Mannan, A., Abbasi, A., Javed, A. R., Ahsan, A., Gadekallu, T. R., & Xin, Q. (2022). Hypertuned deep convolutional neural network for sign language recognition. Computational intelligence and neuroscience, 2022.

[27]. Venugopalan, A., & Reghunadhan, R. (2023). Applying Hybrid Deep Neural Network for the Recognition of Sign Language Words Used by the Deaf COVID-19 Patients. Arabian Journal for Science and Engineering, 48(2), 1349-1362.

[28]. Sreemathy, R., Turuk, M. P., Chaudhary, S., Lavate, K., Ushire, A., & Khurana, S. (2023). Continuous word level sign language recognition using an expert system based on machine learning. International Journal of Cognitive Computing in Engineering, 4, 170-178.

[29]. Hu, L., Gao, L., Liu, Z., & Feng, W. (2023). Continuous Sign Language Recognition with Correlation Network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 2529-2539).

[30]. Buttar, A. M., Ahmad, U., Gumaei, A. H., Assiri, A., Akbar, M. A., & Alkhamees, B. F. (2023). Deep Learning in Sign Language Recognition: A Hybrid Approach for the Recognition of Static and Dynamic Signs. Mathematics, 11(17), 3729.

[31]. Alnfiai, M. M. (2023). Deep Learning-Based Sign Language Recognition for Hearing and Speaking Impaired People. Intelligent Automation & Soft Computing, 36(2).

[32]. https://data.mendeley.com/datasets/kcmpdxky7p/1#:~:text=The%20ISLCSLTR%20corpus%20consists%20of%20a%20large%20vocabulary,annotated%20details%20and%20it%20is%20made%20available%20publicly.