International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Translation of Indian Sign Language to Text-A Comprehensive Review

Seema Sabharwal*[1], Priti Singla[2]

Submitted: 07/12/2023 **Revised**: 18/01/2024 **Accepted**: 28/01/2024

Abstract: Deaf and mute persons across the world uses gestures, non-manual features to interact with fellow persons. This way of communication is called Gesture language or Sign language. Gesture languages are local in nature because of their dependency on geographical area, syntax, pragmatics, and other attributes. The focus of this paper is to present a comprehensive review of conventional as well as contemporary Indian sign language translation system. The process of literature review has been carried out in accordance with Preferred Reporting Items for Systematic reviews and Meta Analysis (PRISMA) guidelines by searching in Scopus, google scholar, Science direct and Lensorg databases. Different articles were included between the years 2010 to 2023 for the purpose of literature review. The study was based on four themes-dataset, technique, result and previous literature reviews. This is the first detailed review conducted in the field of Indian sign language translation system which solely analyses literature related to ISL as per author's knowledge. The findings of this research article may contribute to gain insights and form a blueprint for future areas in the arena of Indian Sign Language translation/recognition system.

Keywords: *Comprehensive review, ISL Translation, ISL Recognition, Indian Sign Language, PRISMA*

1. Introduction

Nonverbal communication encompassing body language, gestures, facial emotions is a vital aspect of human interaction. However, people with special needs are solely reliant on this form of communication. Deaf and dumb persons across the world uses gestures, non-manual features to interact with fellow persons. This way of communication is called Gesture language or Sign language[1]. These languages have evolved over the years because of their natural existence. Due to lack of resources for these especially abled persons, they used sign language to communicate with their families. With technological advances, to aid these persons, schools, medical facilities came into existence. Gesture languages are local in nature because of their dependency on geographical area, syntax, pragmatics, and other attributes[2]. Indian Sign Language (ISL) came into existence in 2018, after a long battle by deaf and dumb community[3]. There are many popular sign languages used in India apart from ISL such as Bangla Sign Language (BSL), Tamil Sign Language (TSL), Panjabi Sign Language (PSL) and Malayalam Sign Language (MSL) etc. Currently, Sign language translation is most popular domain with the potential to provide automatic and effective communication tool for hearing disabled persons[4]. It translates the given input sign language gesture into corresponding text. Promptness and exactness are the important parameters for the decision makers to determine the efficacy of the proposed system[5].

Machine learning is an offshoot of artificial intelligence which aims to simulate human intelligence in machines using various algorithms. However, recently advanced form of machine learning is deep learning which relies on artificial neutral network to emulate human neurons for image processing tasks. Machine Learning (ML) and Deep learning (DL) paradigms can process huge amount of data in a reasonable time limit and build an efficient translation system. Consequently, ML and DL practices are getting immensely popular in the discipline of sign language processing [6].

Although a lot of literature reviews has already been conducted in the field of sign language but dearth of an exhaustive literature review in the field of ISL was one of the major motivation factors for this research article using standard Preferred Reporting Items for Systematic reviews and Meta Analysis (PRISMA) guidelines [7]. The contributions of this research paper are as follows.

- In this paper, we have studied the work done in the field of Indian Sign Language Translation for the vicennial period along with their shortcomings.
- Various datasets available in the domain of Indian Sign Language Translation System (ISLTS) has been explored with the focus on available open access dataset.
- Different levels of translation such as Alphabet level, word level and sentence level has been discussed.
- This paper examines current trends in ISLTS and

[1] *Department of Computer Science and Engineering, Baba Mastnath University, Rohtak, INDIA*
ORCID ID : 00000-0002-3365-9886
[2] *Department of Computer Science and Engineering, Baba Mastnath University, Rohtak, INDIA*
ORCID ID : 0000-0001-8921-6630
** Corresponding Author Email: sabharwalseema@gmail.com*

provide suggestions to researchers for future works.

The rest of the research paper is organized as follows- Section II describes methodology followed by the authors to conduct this literature review. Section III describes the results followed by conclusion in future scope in Section IV.

2. Materials and Methods

We carried out a PRISMA comprehensive review to analyze technological advancements in the field of translation of Indian sign language into text. The research questions were formulated in the first step to initiate the process of conducting literature review.

- RQ1- What is the focus of study in previous literature review in the field of ISLTS?
- RQ2-What are various types of datasets available for researchers in the field of ISLTS?
- RQ3- What are the number of research paper published per year on ISLT/RS?
- RQ4- What are existing techniques for translating ISL gestures and their performance?

2.1 Search Query

Different research papers related to ISLTS/ Indian Sign Language Recognition System (ISLRS) were searched on four popular research databases such as Scopus, Google Scholar, Science Direct and Lensorg. The principal objective of this literature survey is to examine translation/recognition attempts made in the ISL sector. Open Access articles in English language has been selected for this review process from year 2010 to 2023 based on the search queries mentioned in Table 1.

Table 1. Search Query

Name of Dataset	*Query*
Scopus	TITLE-ABS-KEY ("INDIAN SIGN LANGUAGE TRANSLATION" OR "ISL TRANSLATION" OR "ISL RECOGNITION" OR "INDIAN SIGN LANGUAGE RECOGNITION") AND (LIMIT-TO (EXACTKEYWORD , "Indian Sign Languages") OR LIMIT-TO (EXACTKEYWORD , "Indian Sign Language"))
Science Direct LensOrg Google Scholar	"INDIAN SIGN LANGUAGE TRANSLATION" OR "ISL TRANSLATION" OR "ISL RECOGNITION" OR "INDIAN SIGN LANGUAGE RECOGNITION"

The workflow of the literature review process has been shown using **Error! Reference source not found.**. A total of 598 research articles were identified for the purpose of literature review using above mentioned queries from four major research databases. In the next stage, 47 articles were excluded from the study because of redundancy. Upon preliminary literature investigation, every research article's title and abstract were examined manually and then 414 pertinent papers were selected for further assessment criteria. Subsequently, 202 research articles were selected based on the criteria of availability of research paper or whether it is open access. Four eligibility criteria were adopted for this literature survey in the next phase.

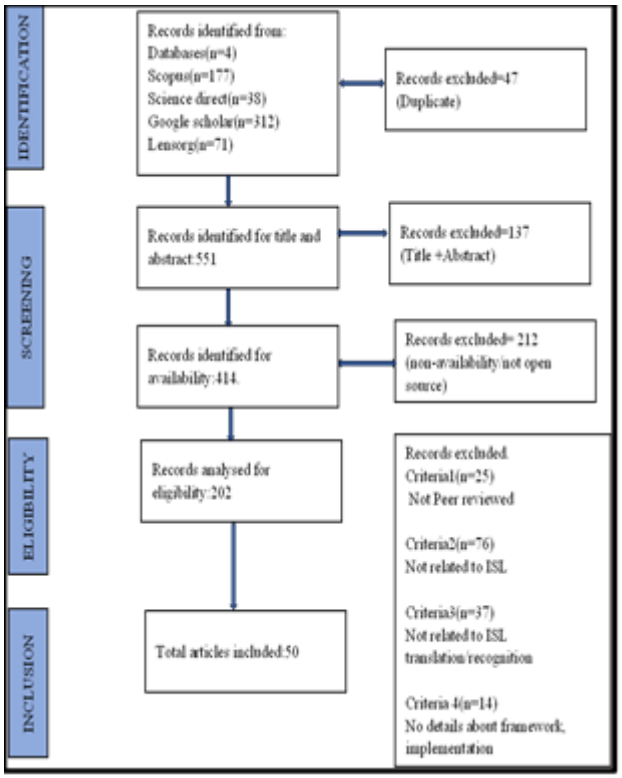


Fig. 1. Flowchart of comprehensive review process using PRISMA guidelines

- The article should be peer reviewed.
- The article should be related to Indian Sign Language.
- The theme of the article should be related to recognition/translation of ISL to text.
- The details about framework/implementation should have been mentioned in the research article.

After stringent eligibility criteria more than 50 articles has been selected for this literature review.

3. Results and Discussion

In this section, we will try to answer all the research questions on the basis of literature.

3.1 RQ1- What is the focus of study in previous literature review in the field of ISLTS?

To analyze previous literature surveys in this field and their focus areas on which review has been performed a list of review articles has been crafted in the field of ISLRS/ISLTS process. Table 2. shows various research articles published along with their year and focus of review from the period of 2010 to 2023.It has been observed that no literature review article has been published in the field of ISLTS/ISLRS using PRISMA guidelines as per the author's knowledge. It has been observed that most of the literature review articles [8]–[12] included less than 10 research articles for their analysis due to lack of standard research work in the domain of ISL. [13] contemplated 29 research articles related to dynamic recognition and compared different methodologies. However, [9], [13] reviewed various methodologies and [12], [14] discussed various feature extraction techniques in ISLTS/ISLRS. [15] examined different articles related to dataset acquisition techniques and concluded that area of non-manual features is yet to be explored in case of ISL. [16] studied few ISLR articles along with other sign languages to conclude there are limited work in alphanumeric recognition, dynamic sign recognition. It has been analyzed that 90% of review papers in ISL translation/recognition process considered less than 20-25.

Table 2. Prior literature reviews in ISL

Ref	*Type*	*Year*	*Criteria for Review*
[8]	J	2015	Gesture set and technique
[9]	J	2023	Methodology
[10]	J	2013	Input, segmentation, Feature vector, classification, recognition rate, platform
[11]	C	2015	Challenges
[12]	C	2021	Feature extraction
[13]	C	2021	Dynamic ISLRS with focus on methodology
[14]	C	2022	Feature Extraction
[15]	C	2022	Dataset acquisition techniques
[16]	C	2019	Input, dataset, segmentation, method, number of gestures, output, limitation, recognition percentage

3.2 RQ2- What are various types of datasets available for researchers in the field of ISLTS?

A standard well annotated dataset is very important for any sign language processing system. In case of ISLTS, lack of standard, open access datasets are few of the major challenges in translation/recognition process. Table 3. describes list of open access datasets available in the domain of ISLTS along with their characteristics. We have also included some dataset with limited access.

Table 3. List of open access datasets available in ISL

Ref	*Year*	*Dataset*	*Type*		
			Alphanumeric	*Word*	*Sentence*
[17]	2010	ISL		22	
[18]	2020	Include	-	15W	
[19]	2021	ISL-CSLTR	-	1036	100
[20]	2021	ISLAN	24 A	-	-
[21]	2021	Emergency	-	8W	-
[22]	2021	INSIGNVID	-	55W	15S
[23]	2022	IISL2020	-	11W	-

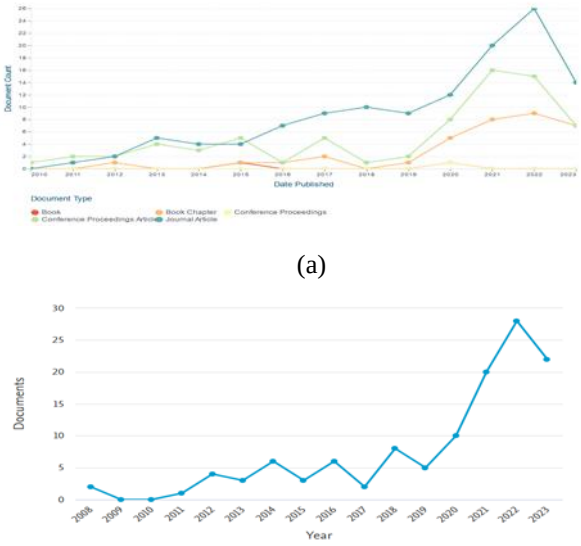
In 2010, [17], created RGB video dataset of 23 different ISL word gestures at 30 frames per second (fps) under various background and lightning conditions. The access of this dataset has been restricted and only given to educational institutions based on agreement through their website by the creators. In 2020, [18] presented Indian lexicon sign language dataset (INCLUDE) with the help of 7 experienced signers. This word level dataset consists of 263 classes of 15 categories, 4287 videos with 1920x1080 resolution and 25 fps. A subset of the above dataset having 50 signs across 15 categories was also proposed with same specifications called INCLUDE 50. Both the datasets include 15 words in total. In 2021, several other researchers came up with their own ISL datasets available freely like [19] Elakkiya et al developed first sentence level Indian sign language dataset for continuous Sign language translation and recognition i.e., ISL-CSLTR. The dataset contained 700 videos of 100 sentences made up with the help of 7 signers. Secondly, another ISL dataset for Alphanumeric (ISLAN) signs was developed by [20] comprising of 350 unique sign images and 12 unique videos compassing 24 alphabets of English language (except J, Z) and numbers totaling to 700 images and 24 videos by 6 signers. Another sign language dataset for emergency domain has been developed by [21]. It included 824 videos of 8 words by 12 males and 14 females. Indian Sign Language Video (INSIGNVID) dataset was developed by [22] for efficient recognition of 55 words of ISL. The dataset was created by 4 right-handed

persons and consists of videos with 30fps, 1920*1088 resolution and common background conditions. In 2022, Kothadiya et al [23] proposed a permission based Isolated ISL dataset (IISL2020) made up of 11 words from 16 persons and 1100 videos and average 28fps.

3.3 RQ3- What are the number of research paper published per year on ISLT/RS?

Error! Reference source not found.. elucidates an overview of ISLTS studies that are published annually from Lensorg and Scopus source in (a) and (b) parts. It has been observed from the figure that highest articles are published in the year 2023 till date i.e., maximum of 26 journal articles has already been published in the year 2023. The topic of ISLTS/ISLRS has garnered a lot of research attention in the last few years however, the work done in recent two years outshines the previous works quantitatively and qualitatively.

Three different categories of articles i.e., book chapter, conference articles and journal articles are contemplated for this research article as represented with figure 3(a) and 3(b). It has been observed that majority of articles published in the field of ISL recognition/translation domain are from journals i.e., 124 out of total 239 are research articles published in journals followed by numbers of articles in conference. This data has been taken from Lensorg website [24] with the constraint to include only selected articles related to the domain of ISLTS/ISLRS from the period 2010 to 2023.



(a)

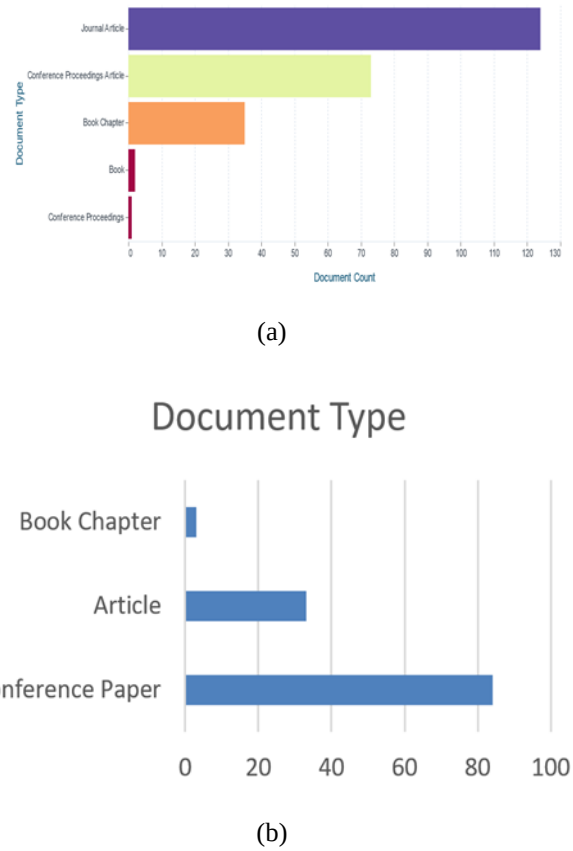
(b)

Fig. 2. Example of a Publications per year (a)LensOrg (b) Scopus

RQ4- What are existing techniques for translating ISL gestures and their performance?

Machine learning has garnered a lot of attention in the field of sign language processing over the last few years. Majumdar et al [25] in 2011 proposed Indian sign language recognition system with YCbCr segmentation, wavelet packet decomposition, principal curvature-based region as feature extraction, dynamic time warping (DTW) to classify alphabets with an accuracy of 91.3%. [26] concluded that Multi SVM classifier can classify static ISL gestures with recognition rate of 92.6 on a self-made dataset.[27]

proposed a ISL recognition system to classify 24 alphabet level gestures with 97 recognition accuracy using novel Eigen value weighted Euclidean distance. [28] proposed a framework for recognition of two-handed gestures of ISL by employing HOG feature extraction method and four popular pretrained models ALEXNET, VGG-16, VGG-19 and GoogleNet. The model attained highest accuracy of 99.11% with ALEXNET and VGG-19 pretrained transfer learning models to classify alphabets of ISL. [29] proposed model for recognizing alphabets of ISL using extreme learning with an average accuracy of 80.76% on self-made dataset. [30] developed a ISLRS for alphabets using CNN with



(a)

(b)

Fig. 3. Publication types (a) LensOrg (b) Scopus

3.4 RQ4- What are existing techniques for translating ISL gestures and their performance?

Machine learning has garnered a lot of attention in the field of sign language processing over the last few years. Majumdar et al [25] in 2011 proposed Indian sign language recognition system with YCbCr segmentation, wavelet packet decomposition, principal curvature-based region as

feature extraction, dynamic time warping (DTW) to classify alphabets with an accuracy of 91.3%. [26] concluded that Multi SVM classifier can classify static ISL gestures with recognition rate of 92.6 on a self-made dataset.[27] proposed a ISL recognition system to classify 24 alphabet level gestures with 97 recognition accuracy using novel Eigen value weighted Euclidean distance. [28] proposed a framework for recognition of two-handed gestures of ISL by employing HOG feature extraction method and four popular pretrained models ALEXNET, VGG-16, VGG-19 and GoogleNet. The model attained highest accuracy of 99.11% with ALEXNET and VGG-19 pretrained transfer learning models to classify alphabets of ISL. [29] proposed model for recognizing alphabets of ISL using extreme learning with an average accuracy of 80.76% on self-made dataset. [30] developed a ISLRS for alphabets using CNN with diffGrad optimizer and stochastic pooling to achieve validation accuracy of 99.64%.[31] proposed a framework for recognition of alphabets of ISL using correlation coefficient feature extraction and neurofuzzy algorithm as classifier to achieve an average accuracy of 92.3%.[32] proposed transfer learning based recognition of ISL alphabets with an accuracy of 95%. The VGG16 pretrained model consists of 13 convolution layers, average, max pooling, dropout layer for controlling overfitting, Adam optimizer and softmax as classifier layer.

In numeric ISLTS, [33] in 2014, proposed ISL numeric digit (0-9) recognition system on a self-made ISL dataset using KNN classifier and an accuracy of 97.1%. However, [34] proposed Kinect sensor based ISLRS using scale, rotation, and background lightning invariant ORB feature extraction method and KNN machine learning algorithm to classify (0-9) digits of ISL on a self-made dataset with an accuracy of 93.26% outperforming standard feature extraction techniques like SIFT and SURF.

In the domain of alphanumeric level recognition, [35] proposed ISLRS framework using fingertip algorithm and PCA to obtain 94% accuracy. In 2013, [36] used Fourier descriptors, distance transform and artificial neural network with four layers to classify 36 alphanumeric gestures of ISL with an average accuracy of 91.11 %. [37] Geetha et al suggested alphanumeric ISL sign recognition system with B-spline approximation and SVM classification algorithm.[38] proposed novel fusion descriptor for classification of ISL numeric signs with Nearest Mean classifier and an accuracy of 100%. The novel fusion descriptor comprises of two contour (Boundary, Fourier descriptor) and one region based(7Hu) descriptors.[39] classified gestures of ISL using SVM machine learning algorithm.[40],[41],[42] used Kinect sensor to classify gestures at alphanumeric and word level along with popular classification algorithms such as PCA, SVM to attain remarkable accuracies.[43] suggested translation of word level ISL gestures by extensive training of humanoid robot HOAP-2 along with direction histogram feature extraction, Euclidean distance metric has been used to attain an average accuracy of 90%. [23] Kothadiya et al. in 2022 classified 11 words of ISL using sequential combination of LSTM and GRU with accuracy of 97% on their dataset IISL2020. [44] developed a model for classifying 24 dynamic word gestures of ISL using novel dynamic time warping recognition technique along with accuracy of around 90%. 20 different gestures were classified by [45] using 3D CNN and attaining 88% validation accuracy in 100 epochs. The model

comprises of 3 convolution layers, max pooling, dropout and softmax activation function. [46] Subramaniam suggested integrated model of Media pipe with optimized GRU model for recognition of 13 ISL gestures to attain average accuracy of 95%. The proposed system has been compared with RNN, LSTM, standard GRU, BiGRU, and BiLSTM models.

Hybrid ISLT paradigm comprising of combination of word and alphanumeric and sentence level. In this, [22] suggested transfer learning approach using MobileNetV2 to transcribe clips of ISL into English language. The proposed system was analyzed using other pretrained models such as MobileNet, VGG16 and ResNet50 using 25 epochs with 9 trainable layers to attain testing accuracy of 93.89%. Although proposed system achieves better accuracy but time to train the system was comparative high i.e., more than 12 hours.

[47] classified alphanumeric, word gestures of ISL using Fourier descriptor feature extractor and distance metric to attain overall accuracy of 92.91%. The proposed system [48] recognized 26 alphabets, 10 numbers and 10 distinct phrases on self-made skinpixel segmentation, Moment based feature extraction and SVM algorithm to classify dynamic gestures. The system obtained an accuracy of 97.5% recognition rate in classifying 4 signs (3 alphabet and one word). [49] presents a signer independent communication model for real time using YCbCr segmentation, Zernike moments feature vector and SVM as classifier.[50] Deshpande et al. classified 56 signs real time using CNN into text and audio with 98% accuracy with the constraint of plain background. The proposed model had 5 convolution layers, ReLU activation function, max pooling, dropout layer, single valued stride function and softmax layer to classify different signs. [51] recognized gestures of 7 days of week using Kinect sensor and random forest classifier algorithm to give an accuracy of 74.29% with focus on low cost and maximum efficiency.[52] Nareshkumar attained an accuracy of 98.77% in translating alphanumeric gestures of ISL and American Sign language using novel pretrained model for mobile MobileNetV2 consisting of pointwise convolution layers, separable depthwise convolution, ReLU activation function, swish

activation function, batch normalization, dropout layer and softmax as final classifier layer. [53] developed ISLRS using HSV segmentation, PCA with OH 18 and 36 bins feature extraction mechanism to classify 10 ISL sentences having 2,3 or 4 gestures using six different distance metrics. Euclidean distance topped the performance chart with 93% recognition rate (RR) on 36 bins of orientation histogram. [54] proposed a lightweight framework for translating sentence level ISL gestures into text and audio, LiST, and used pretrained model InceptionV3, two layered LSTM architecture on open access dataset with a translation accuracy of 91.2%. [55] proposed a framework for translation of 10 signs of ISL using HMM and DWT to achieve lowest accuracy of 80 and highest accuracy of 100%. [56] proposed a Leap motion sensor based ISLTS for 35 words and 942 sentences using four gated cell LSTM with 2 dimensional CNN to attain average accuracy of 89.5% and 72.3% respectively.[57] proposed gesture recognition mechanism for 42 signs of ISL using KNN and SVM machine learning classification techniques. HSV, Otsu thresholding for segmentation and novel MFCC feature extraction method has been used along with wavelet descriptor to translate 42 static and dynamic gestures of ISL. The authors concluded that SVM with MFCC feature extraction mechanism

Table 4. Work done in the domain of ISLTS

Ref	*Year*	*Type*	*Specifications*	*Features*	*Results*
[25]	2011	ALPHABET	26Alphabet	Support Vector Machine (SVM), K-Nearest Neighbour (KNN) and Dynamic Time Warping (DTW)	91.3 accuracy
[26]	2012		26Alphabet	Multi SVM classifier	92.6 Recognition Accuracy
[27]	2013		Alphabet (24)	Eigen value weighted Euclidean distance	97 Recognition Rate
[28]	2022		Alphabet	Histogram Oriented Gradient (HOG), AlexNet	99.11Accuracy
[29]	2020		-	Extreme learning	80.76Accuracy
[30]	2022		26 Alphabet	Convolution Neural Network (CNN) with diffGrad optimiser	99.64
[31]	2019		26 Alphabet	Neurofuzzy algorithm with correlation coefficient feature extractor	92.3
[32]	2022		26 Alphabet	VGG16	95
[33]	2014	NUMBER	10 Numbers	Neural network with KNN	97.1
[34]	2020		10 Numbers	Kinect and Bag of visual words with ORB, KNN	93.26
[35]	2012	ALPHANUMERIC	36Alphanumeric	Principal Component Analysis (PCA)	94 Recognition Accuracy
[36]	2013		36Alphanumeric	4-layer Artificial Neural Network (ANN)	91.11 Recognition Rate
[37]	2012		26Alphabet, 6Numbers	B-spline approximation	-
[38]	2016		36Alphanumeric	Novel fusion descriptor,	99.61
[39]	2022		Alphanumeric	Bag of Visual Words (BOVW), Speeded Up Robust Features (SURF), SVM, CNN	99.64Accuracy

[40]	2020		Alphanumeric	Kinect with SURF, HOG, Local binary pattern SVM	Average Accuracy-71.85
[23]	2022		11Words	Long short-Term Memory (LSTM), Gated Recurrent Unit (GRU)	97Accuracy
[41]	2015		37Words	Kinect with SVM	86.16%Validation Accuracy
[42]	2013	WORD	10Words	PCA with ALI, Microsoft Kinect (25 key points of each gesture)	Best-100A, Average-40, Worst-25
[43]	2010		22Words	Euclidean distance, KNN with 36 bins	Lowest-48.42, Highest-100Accuracy
[44]	2016		24Words	DTW	90Accuracy
[45]	2021		20Words	3D-CNN	88.24Average Accuracy
[46]	2022		13Words	MOPGRU with ELU activation and softsign activation function	99.92Accuracy, 0.21 Loss
[22]	2021		55Word, 15Sentences	MobileNetV2 pretrained model	93 Recognition Rate
[47]	2015		10Signs	Fourier Descriptor with Euclidean distance	Lowest-85, Highest-97Accuracy
[48]	2016		01Word, Alphabet	3SVM, Kinect sensor	97.5 Recognition Accuracy
[49]	2022		26Alphabet, 11Word	Co-articulation, Zernike moment, SVM	Alphabets-91A, W-89A
[50]	2023		36 Alphanumeric, 20Word	Region of Interest, CNN	98Accuracy
[51]	2022		7 Word	Kinect V2 sensor, Random Forest	74.28 Accuracy
[52]	2023		26Alphabet, 3Word	Transfer learning, Modified MobileNet V2	98.77 Accuracy
[53]	2015	HYBRID	10Sentences	PCA with OH	Lowest-85 , Highest -93 Accuracy
[54]	2023		15Sign	Inception v3 CNN with LSTM	95.90 Accuracy
[55]	2015		10Sign	DWT, Hidden Markov Model (HMM)	Lowest-80, Highest-100 Accuracy
[56]	2019		942Sentences, 35Word	CNN with LSTM and Leap motion sensor	Sentences-72.3Avg. Accuracy, Words-89.5 Avg. accuracy
[57]	2017		42Words	KNN, SVM with MFCC feature extraction	97 Accuracy
[58]	2019		80Word, 50Sentences	Fuzzy clustering algorithm	75 Accuracy
[59]	2018		33Alphanumeric, 12 signs	HMM, KNN	99.7 Static Sign

					97.23 Dynamic Sign
[60]	2012	80Words Sentences	andPCA feature vector, Fuzzy inference system		96 Accuracy

classified ISL gestures with better accuracy than KNN. [58] proposed Fuzzy c-means clustering algorithm for classifying 80 words and 50 sentences of ISL with an average accuracy of 75%. [59] proposed ISLTS for 45 sign (alphanumeric and word) using skin color segmentation, Hidden Markov Model, K-nearest neighbor recognition algorithm with an accuracy of 99.7% for static signs and 97.23% for dynamic signs. [60] proposed video gesture recognition of ISL using Gaussian filter, Canny edge detector, Fourier descriptor and Sugeno fuzzy inference system to attain higher accuracy of 100 and lowest accuracy of 60 among total signs. Although there are various researchers who have been working in the domain of Indian sign language translation/recognition to develop optimal framework but there is tradeoff between accuracy and time. The system is affected by so many parameters discussed in next section.

4 Conclusion and Future Scope

In this paper, we have conducted comprehensive literature review on Indian Sign Language Translation/Recognition System using PRISMA guidelines. After rigorous screening, more than 50 papers were selected for this review from four major research databases- Scopus, Google Scholar, Science Direct and LensOrg. There were four main criteria on which this survey was conducted- previous work done, datasets available, number of research articles published per year and summary of important work done. It has been concluded a lot of work has already been done in sign language processing systems, but ISLTS are still lagging in a lot of aspects.

- Lack of well annotated standard open access datasets
- Alphanumeric recognition
- Two-way communication system
- Domain specific translation system
- Lack of quality review papers
- Sensor based devices gives better accuracy but are not comfortable

As there are many challenges but exploitation of new emerging machine learning algorithms is need of the hour in ISLTS as compared to other sign language processing system. We hope that this research paper will help other future researchers in the field of ISL.

Author contributions

Seema Sabharwal: Conceptualization, Methodology, Software, Field study, Data curation, Writing-Original draft preparation, Software, Validation, Field study, Visualization **Priti Singla:** Investigation, Writing-Reviewing and Editing.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] A. Wadhawan and P. Kumar, “Sign Language Recognition Systems: A Decade Systematic Literature Review,” Arch. Comput. Methods Eng., vol. 28, no. 3, pp. 785–813, May 2021, doi: 10.1007/s11831-019-09384-2.

[2] S. M. Kamal, Y. Chen, S. Li, X. Shi, and J. Zheng, “Technical Approaches to Chinese Sign Language Processing: A Review,” IEEE Access, vol. 7, pp. 96926–96935, 2019, doi: 10.1109/ACCESS.2019.2929174.

[3] ISLRTC, “History | Indian Sign Language Research and Training Center (ISLRTC), Government of India,” Indian Sign Language Research and Training Center (ISLRTC). Accessed: Feb. 14, 2022. [Online]. Available: http://islrtc.nic.in/history-0

[4] S. Sabharwal and P. Singla, “Indian Sign Language Digit Translation Using CNN with Swish Activation Function,” in Key Digital Trends Shaping the Future of Information and Management Science, vol. 671, in Lecture Notes in Networks and Systems, vol. 671. , Cham: Springer International Publishing, 2023, pp. 245–253. doi: 10.1007/978-3-031-31153-6_21.

[5] I. A. Adeyanju, O. O. Bello, and M. A. Adegboye, “Machine learning methods for sign language recognition: A critical review and analysis,” Intell. Syst. Appl., vol. 12, p. 200056, Nov. 2021, doi: 10.1016/j.iswa.2021.200056.

[6] Seema and P. Singla, “A Comprehensive Review of CNN-Based Sign Language Translation System,” in Proceedings of Data Analytics and Management, vol. 572, A. Khanna, Z. Polkowski, and O. Castillo, Eds., in Lecture Notes in Networks and Systems, vol. 572. , Singapore: Springer Nature Singapore, 2023, pp. 347–362. doi: 10.1007/978-981-19-7615-5_31.

[7] A. Liberati et al., “The PRISMA statement for reporting systematic reviews and meta-analyses of studies that evaluate healthcare interventions: explanation and elaboration,” BMJ, vol. 339, no. jul21

1, pp. b2700–b2700, Dec. 2009, doi: 10.1136/bmj.b2700.

[8] Daleesha M Viswanathan and Sumam Mary Idicula, "Recent Developments in Indian Sign Language Recognition: An Analysis," International Journal of Computer Science and Information Technologies, vol. 6, no. 1, pp. 289–293, 2015.

[9] M. R. K, H. Kaur, S. K. Bedi, and M. A. Lekhana, "Image-based Indian Sign Language Recognition: A Practical Review using Deep Neural Networks." arXiv, Apr. 28, 2023. Accessed: Sep. 18, 2023. [Online]. Available: http://arxiv.org/abs/2304.14710

[10] Anuja V. Nair and Bindu V., "A Review on Indian Sign Language Recognition," ijca, vol. 73, no. 22, pp. 33–38, 2013.

[11] V. K. Verma, S. Srivastava, and N. Kumar, "A comprehensive review on automation of Indian sign language," International Conference on Advances in Computer Engineering and Applications, ICACEA 2015, 2015, pp. 138–142. doi: 10.1109/ICACEA.2015.7164682.

[12] A. Tyagi and S. Bansal, "Feature extraction technique for vision-based Indian sign language recognition system: A review," Advances in Intelligent Systems and Computing, 2021, pp. 39–53. doi: 10.1007/978-981-15-6876-3_4.

[13] B. Samal and M. Panda, "Integrative review on vision-based dynamic Indian sign language recognition systems," 1st Odisha International Conference on Electrical Power Engineering, Communication and Computing Technology, ODICON 2021, 2021. doi: 10.1109/ODICON50556.2021.9429002.

[14] S. Das, S. K. Biswas, M. Chakraborty, and B. Purkayastha, "Intelligent Indian Sign Language Recognition Systems: A Critical Review,"Lecture Notes in Networks and Systems, 2022, pp. 703–713. doi: 10.1007/978-981-16-5987-4_71.

[15] A. Singh, S. K. Singh, and A. Mittal, "A Review on Dataset Acquisition Techniques in Gesture Recognition from Indian Sign Language," in Lecture Notes on Data Engineering and Communications Technologies, vol. 106, 2022, pp. 305–313. doi: 10.1007/978-981-16-8403-6_27.

[16] Rakesh Savant and Dr. Jitendra Nasriwala, "INDIAN SIGN LANGUAGE RECOGNITION SYSTEM: APPROACHES AND CHALLENGES," International Journal of Advance and Innovative Research, vol. 6, no. 3(IV), pp. 76–84, 2019.

[17] A. Nandy, J. S. Prasad, S. Mondal, P. Chakraborty, and G. C. Nandi, "Recognition of Isolated Indian Sign Language Gesture in Real Time," in Information Processing and Management, vol. 70, V. V. Das, R. Vijayakumar, N. C. Debnath, J. Stephen, N. Meghanathan, S. Sankaranarayanan, P. M. Thankachan, F. L. Gaol, and N. Thankachan, Eds., in Communications in Computer and Information Science, vol. 70. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 102–107. doi: 10.1007/978-3-642-12214-9_18.

[18] A. Sridhar, R. G. Ganesan, P. Kumar, and M. Khapra, "INCLUDE: A Large Scale Dataset for Indian Sign Language Recognition," in Proceedings of the 28th ACM International Conference on Multimedia, Seattle WA USA: ACM, Oct. 2020, pp. 1366–1375. doi: 10.1145/3394171.3413528.

[19] E. R, "ISL-CSLTR: Indian Sign Language Dataset for Continuous Sign Language Translation and Recognition." Mendeley, Jan. 22, 2021. doi: 10.17632/KCMPDXKY7P.

[20] E. R, "ISLAN." Mendeley, Jan. 08, 2021. doi: 10.17632/RC349J45M5.1.

[21] Adithya Venugopalan, "A Video Dataset of the Hand Gestures of Indian Sign Language Words used in Emergency Situations." Mendeley, Aug. 27, 2021. doi: 10.17632/2VFDM42337.1.

[22] K. Mistree, D. Thakor, and B. Bhatt, "Towards Indian Sign Language Sentence Recognition using INSIGNVID: Indian Sign Language Video Dataset," Int. J. Adv. Comput. Sci. Appl., vol. 12, no. 8, 2021, doi: 10.14569/IJACSA.2021.0120881.

[23] D. Kothadiya, C. Bhatt, K. Sapariya, K. Patel, A.-B. Gil-González, and J. M. Corchado, "Deepsign: Sign Language Detection and Recognition Using Deep Learning," Electronics, vol. 11, no. 11, p. 1780, Jun. 2022, doi: 10.3390/electronics11111780.

[24] "Results The Lens - Free & Open Patent and Scholarly Search," The Lens - Free & Open Patent and Scholarly Search. Accessed: Sep. 18, 2023. [Online]. Available: https://www.lens.org/lens

[25] J. Rekha, J. Bhattacharya, and S. Majumder, "Shape, texture and local movement hand gesture features for Indian Sign Language recognition," in 3rd International Conference on Trendz in Information Sciences & Computing (TISC2011), Chennai, India: IEEE, Dec. 2011, pp. 30–35. doi: 10.1109/TISC.2011.6169079.

[26] S. C. Agrawal, A. S. Jalal, and C. Bhatnagar, "Recognition of Indian Sign Language using feature fusion," in 2012 4th International Conference on Intelligent Human Computer Interaction (IHCI), Kharagpur, India: IEEE, Dec. 2012, pp. 1–5. doi:

10.1109/IHCI.2012.6481841.

[27] J. Singha and K. Das, "Indian Sign Language Recognition Using Eigen Value Weighted Euclidean Distance Based Classification Technique," Int. J. Adv. Comput. Sci. Appl., vol. 4, no. 2, 2013, doi: 10.14569/IJACSA.2013.040228.

[28] R. Sreemathy, M. Turuk, I. Kulkarni, and S. Khurana, "Sign language recognition using artificial intelligence," Educ. Inf. Technol., Nov. 2022, doi: 10.1007/s10639-022-11391-z.

[29] A. Kumar and R. Kumar, "A novel approach for ISL alphabet recognition using Extreme Learning Machine," Int. J. Inf. Technol. Singap., vol. 13, no. 1, pp. 349–357, 2021, doi: 10.1007/s41870-020-00525-6.

[30] U. Nandi, A. Ghorai, M. M. Singh, C. Changdar, S. Bhakta, and R. Kumar Pal, "Indian sign language alphabet recognition system using CNN with diffGrad optimizer and stochastic pooling," Multimed. Tools Appl., pp. 1–22, Jan. 2022, doi: 10.1007/s11042-021-11595-4.

[31] H. Bhavsar and Dr. J. Trivedi, "Indian Sign Language Alphabets Recognition from Static Images Using Correlation-Coefficient Algorithm with Neuro-Fuzzy Approach," SSRN Electron. J., 2019, doi: 10.2139/ssrn.3421685.

[32] T. S. Abraham, S. P. Sachin Raj, A. Yaamini, and B. Divya, "Transfer learning approaches in deep learning for Indian sign language classification," Journal of Physics: Conference Series, 2022. doi: 10.1088/1742-6596/2318/1/012041.

[33] A. K. Sahoo, M. Sharma, and R. Pal, "INDIAN SIGN LANGUAGE RECOGNITION USING NEURAL NETWORKS AND KNN CLASSIFIERS," ARPN, vol. 9, no. 8, pp. 1255–1259, Aug. 2014.

[34] J. Gangrade, J. Bharti, and A. Mulye, "Recognition of Indian Sign Language Using ORB with Bag of Visual Words by Kinect Sensor," IETE J. Res., vol. 68, no. 4, pp. 2953–2967, Jul. 2022, doi: 10.1080/03772063.2020.1739569.

[35] D. Deora and N. Bajaj, "Indian sign language recognition," in 2012 1st International Conference on Emerging Technology Trends in Electronics, Communication & Networking, Surat, Gujarat, India: IEEE, Dec. 2012, pp. 1–5. doi: 10.1109/ET2ECN.2012.6470093.

[36] V. Adithya, P. R. Vinod, and U. Gopalakrishnan, "Artificial neural network based method for Indian sign language recognition," in 2013 IEEE CONFERENCE ON INFORMATION AND COMMUNICATION TECHNOLOGIES, Thuckalay, Tamil Nadu, India: IEEE, Apr. 2013, pp. 1080–1085. doi: 10.1109/CICT.2013.6558259.

[37] Geetha M and Manjusha U C, "A Vision Based Recognition of Indian Sign Language Alphabets and Numerals Using B-Spline Approximation," International Journal on Computer Science and Engineering (IJCSE), vol. 4, no. 3, pp. 406–415, 2012.

[38] G. K. Kharate and A. S. Ghotkar, "Vision based multi-feature hand gesture recognition for indian sign language manual signs," Int. J. Smart Sens. Intell. Syst., vol. 9, no. 1, pp. 124–147, 2016, doi: 10.21307/ijssis-2017-863.

[39] S. Katoch, V. Singh, and U. S. Tiwary, "Indian Sign Language recognition system using SURF with SVM and CNN," Array, vol. 14, p. 100141, Jul. 2022, doi: 10.1016/j.array.2022.100141.

[40] T. Raghuveera, R. Deepthi, R. Mangalashri, and R. Akshaya, "A depth-based Indian Sign Language recognition using Microsoft Kinect," Sādhanā, vol. 45, no. 1, p. 34, Dec. 2020, doi: 10.1007/s12046-019-1250-6.

[41] K. Mehrotra, A. Godbole, and S. Belhe, "Indian Sign Language Recognition Using Kinect Sensor," in Image Analysis and Recognition, vol. 9164, M. Kamel and A. Campilho, Eds., in Lecture Notes in Computer Science, vol. 9164. , Cham: Springer International Publishing, 2015, pp. 528–535. doi: 10.1007/978-3-319-20801-5_59.

[42] Geetha M, Manjusha C, Unnikrishnan P, and Harikrishnan R, "A vision based dynamic gesture recognition of Indian Sign Language on Kinect based depth images," in 2013 International Conference on Emerging Trends in Communication, Control, Signal Processing and Computing Applications (C2SPCA), Bangalore, India: IEEE, Oct. 2013, pp. 1–7. doi: 10.1109/C2SPCA.2013.6749448.

[43] A. Nandy, S. Mondal, J. S. Prasad, P. Chakraborty, and G. C. Nandi, "Recognizing & interpreting Indian Sign Language gesture for Human Robot Interaction," in 2010 International Conference on Computer and Communication Technology (ICCCT), Allahabad, Uttar Pradesh, India: IEEE, Sep. 2010, pp. 712–717. doi: 10.1109/ICCCT.2010.5640434.

[44] W. Ahmed, K. Chanda, and S. Mitra, "Vision based Hand Gesture Recognition using Dynamic Time Warping for Indian Sign Language," in 2016 International Conference on Information Science (ICIS), Kochi, India: IEEE, Aug. 2016, pp. 120–125. doi: 10.1109/INFOSCI.2016.7845312.

[45] D. K. Singh, "3D-CNN based Dynamic Gesture

Recognition for Indian Sign Language Modeling," in Procedia Computer Science, 2021, pp. 76–83. doi: 10.1016/j.procs.2021.05.071.

[46] B. Subramanian, B. Olimov, S. M. Naik, S. Kim, K.-H. Park, and J. Kim, "An integrated mediapipe-optimized GRU model for Indian sign language recognition," Sci. Rep., vol. 12, no. 1, p. 11964, Jul. 2022, doi: 10.1038/s41598-022-15998-7.

[47] P. C. Badhe and V. Kulkarni, "Indian sign language translator using gesture recognition algorithm," in IEEE International Conference on Computer Graphics, Vision and Information Security (CGVIS), Bhubaneswar, India: IEEE, 2016, pp. 195–200. doi: 10.1109/CGVIS.2015.7449921.

[48] J. L. Raheja, A. Mishra, and A. Chaudhary, "Indian sign language recognition using SVM," Pattern Recognit. Image Anal., vol. 26, no. 2, pp. 434–441, Apr. 2016, doi: 10.1134/S1054661816020164.

[49] P. K. Athira, C. J. Sruthi, and A. Lijiya, "A Signer Independent Sign Language Recognition with Co-articulation Elimination from Live Videos: An Indian Scenario," J. King Saud Univ. - Comput. Inf. Sci., vol. 34, no. 3, pp. 771–781, Mar. 2022, doi: 10.1016/j.jksuci.2019.05.002.

[50] A. M. Deshpande, G. Inamdar, R. Kankaria, and S. Katage, "A Deep Learning Framework for Real-Time Indian Sign Language Gesture Recognition and Translation to Text and Audio," in Proceedings of the 6th International Conference on Advance Computing and Intelligent Engineering, vol. 428, B. Pati, C. R. Panigrahi, P. Mohapatra, and K.-C. Li, Eds., in Lecture Notes in Networks and Systems, vol. 428. , Singapore: Springer Nature Singapore, 2023, pp. 287–300. doi: 10.1007/978-981-19-2225-1_26.

[51] S. Paul, M. Jajoo, A. Raj, A. F. Mollah, M. Nasipuri, and S. Basu, "Dynamic Hand Gesture Recognition of the Days of a Week in Indian Sign Language Using Low-Cost Depth Device," in Intelligent Data Engineering and Analytics, vol. 266, S. C. Satapathy, P. Peer, J. Tang, V. Bhateja, and A. Ghosh, Eds., in Smart Innovation, Systems and Technologies, vol. 266. , Singapore: Springer Nature Singapore, 2022, pp. 141–149. doi: 10.1007/978-981-16-6624-7_15.

[52] M. Daniel Nareshkumar and B. Jaison, "A Light-Weight Deep Learning-Based Architecture for Sign Language Classification," Intell. Autom. Soft Comput., vol. 35, no. 3, pp. 3501–3515, 2023, doi: 10.32604/iasc.2023.027848.

[53] K. Tripathi and N. B. G. C. Nandi, "Continuous Indian Sign Language Gesture Recognition and Sentence Formation," Procedia Comput. Sci., vol. 54, pp. 523–531, 2015, doi: 10.1016/j.procs.2015.06.060.

[54] A. K, P. P, and R. C. Poonia, "LiST: A Lightweight Framework for Continuous Indian Sign Language Translation," Information, vol. 14, no. 2, p. 79, Jan. 2023, doi: 10.3390/info14020079.

[55] K. Tripathi, N. Baranwal, and G. C. Nandi, "Continuous dynamic Indian Sign Language gesture recognition with invariant backgrounds," in 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Kochi, India: IEEE, Aug. 2015, pp. 2211–2216. doi: 10.1109/ICACCI.2015.7275945.

[56] A. Mittal, P. Kumar, P. P. Roy, R. Balasubramanian, and B. B. Chaudhuri, "A Modified LSTM Model for Continuous Sign Language Recognition Using Leap Motion," IEEE Sens. J., vol. 19, no. 16, pp. 7056–7063, Aug. 2019, doi: 10.1109/JSEN.2019.2909837.

[57] N. Baranwal and G. C. Nandi, "An efficient gesture based humanoid learning using wavelet descriptor and MFCC techniques," Int. J. Mach. Learn. Cybern., vol. 8, no. 4, pp. 1369–1388, 2017, doi: 10.1007/s13042-016-0512-4.

[58] H. Muthu Mariappan and V. Gomathi, "Real-time recognition of Indian sign language," ICCIDS 2019 - 2nd International Conference on Computational Intelligence in Data Science, Proceedings, 2019. doi: 10.1109/ICCIDS.2019.8862125.

[59] K. Shenoy, T. Dastane, V. Rao, and D. Vyavaharkar, "Real-time Indian Sign Language (ISL) Recognition," 9th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2018, 2018. doi: 10.1109/ICCCNT.2018.8493808.

[60] P. V. V. Kishore and P. Rajesh Kumar, "A Video Based Indian Sign Language Recognition System (INSLR) Using Wavelet Transform and Fuzzy Logic," Int. J. Eng. Technol., vol. 4, no. 5, pp. 537–542, 2012, doi: 10.7763/IJET.2012.V4.427.

Intelligent Automation & Soft Computing

Tech Science Press

DOI: 10.32604/iasc.2023.035497
Article

Alphabet-Level Indian Sign Language Translation to Text Using Hybrid-AO Thresholding with CNN

Seema Sabharwal[1,2,*] and Priti Singla[1]

[1]Department of Computer Science and Engineering, Baba Mastnath University, Rohtak, 124001, India
[2]Department of Computer Science, Government Post Graduate College for Women, Panchkula, 134109, India
*Corresponding Author: Seema Sabharwal. Email: sabharwalseema@gmail.com
Received: 23 August 2022; Accepted: 13 January 2023; Published: 11 September 2023

Abstract: Sign language is used as a communication medium in the field of trade, defence, and in deaf-mute communities worldwide. Over the last few decades, research in the domain of translation of sign language has grown and become more challenging. This necessitates the development of a Sign Language Translation System (SLTS) to provide effective communication in different research domains. In this paper, novel Hybrid Adaptive Gaussian Thresholding with Otsu Algorithm (Hybrid-AO) for image segmentation is proposed for the translation of alphabet-level Indian Sign Language (ISLTS) with a 5-layer Convolution Neural Network (CNN). The focus of this paper is to analyze various image segmentation (Canny Edge Detection, Simple Thresholding, and Hybrid-AO), pooling approaches (Max, Average, and Global Average Pooling), and activation functions (ReLU, Leaky ReLU, and ELU). 5-layer CNN with Max pooling, Leaky ReLU activation function, and Hybrid-AO (5MXLR-HAO) have outperformed other frameworks. An open-access dataset of ISL alphabets with approx. 31 K images of 26 classes have been used to train and test the model. The proposed framework has been developed for translating alphabet-level Indian Sign Language into text. The proposed framework attains 98.95% training accuracy, 98.05% validation accuracy, and 0.0721 training loss and 0.1021 validation loss and the performance of the proposed system outperforms other existing systems.

Keywords: Sign language translation; CNN; thresholding; Indian sign language

1 Introduction

In today's era, specially-abled people are critiqued on the scale of their impairment. Deaf people have difficulty integrating into society due to their communication challenges. Sign language is used as a means of communication by deaf people around the globe apart from its application in various other fields such as defence, trade, sea-diving, etc. Communication with deaf people is hindered majorly by non-deaf people's inability to understand sign language gestures. The syntax and semantics of sign languages vary from one region to another. There are numerous sign languages used across the

world for approximately 70 million deaf people, for instance, Indian Sign Language (ISL), American Sign Language (ASL), Chinese Sign Language (CSL), Japanese Sign Language (JSL), Italian Sign Language, Malaysian Sign Language, etc. ISL is used to communicate with deaf persons in India in addition to other regional sign languages. The building blocks of ISL are 26 alphabets (A–Z), 10 numbers (0–9), and 10,000 words [1].

Due to a huge gap in the vocabulary of the English language and the Indian Sign language, the fingerspelling approach came into existence. This method creates each locution letter by letter via distinct gestures for sign language letters. The gestures of ISL can be classified based on the number of hands, movement of hands, features taken into consideration, and the way they are recognized (Static/Dynamic) [2]. As a consequence of technological innovation, there is a need to craft a sign language processing framework to facilitate deaf people. Currently, three types of systems are available for processing Sign language. Firstly, Sign Language Generation System (SLGS), decodes given text into apt gestures of sign language. Secondly, Sign Language Validation System (SLVS), verifies the legitimacy of a given sign language pose. Finally, Sign Language Translation System (SLTS), construes given sign language pose to typescript. SLTS employs two approaches -Hardware-based SLTS uses gloves (smart glove, data glove, and cyber glove), and sensors (LMC, ACC, EMG) [3].

On the other hand, vision-based SLTS employs a camera and various techniques such as machine learning and deep learning to recognize and construes sign language gestures [4]. The output can be either audio or text (alphabet, word, or sentence). In deep learning, CNN has outperformed several machine learning algorithms in sign language translation [5,6]. It has become a popular choice of researchers because of its architecture, and optimal performance in image classification tasks [7]. Image pre-processing is also one of the important steps in SLTS along with the CNN. The challenging part of an efficient SLTS is to have the finest feature extraction method and an efficient translation mechanism. In recent years, researchers have either focus on premium feature extraction techniques through image segmentation/thresholding or fine-tuning architectural parameters of CNN to have optimal accuracy. This paper aims to explore the role of image thresholding, pooling, and activation functions in an efficient CNN-based sign language translation framework. It focuses on an automatic sign language translation framework for deaf-mute persons. The following are key contributions of this work:

- A novel Hybrid-AO thresholding (based on Adaptive Gaussian thresholding and Otsu Algorithm) approach for segmentation is used on the open access dataset of ISL.
- A 5MXLR-HAO framework is proposed based on Hybrid-AO thresholding and deep learning. It is chosen by comparison among 9 models based on three variations of pooling (Max, Average, Global Average Pooling) and activation functions (ReLU, Leaky ReLU, and ELU).
- Evaluate the performance of the proposed 5MXLR-HAO using Training Accuracy and Validation accuracy metrics.

The paper is organized as follows: Section 2 discusses the related works. Section 3 explains the methodology followed by experimental results and a discussion in Section 4. Finally, the conclusion and future work are presented in Section 5.

2 Related Work

Gesture recognition and sign language translation have been well-studied topics in American Sign Language (ASL), but few studies on Indian Sign Language (ISL) have been published. In contrast to

ASL, two-handed signs in ISL have little duality, making them difficult to discern. A deep CNN-based SLRS was developed by [2] to identify the gestures of ISL using a model comprising 8 layers with Stochastic pooling, DiffGrad optimizer, and SoftMax as a classifier. Using transfer learning, a Bengali Sign Language recognition system has been proposed by [8] to classify 3 sets of 37 different symbols to attain a validation accuracy of 84.68%. Alphabet-based Arabic SLTS was developed by [9] which gives a speech as output with an accuracy of 90% employing CNN. Reference [10] proposed an American Sign Language Recognition system using Otsu thresholding for image segmentation and 2 channelled CNN on a self-made dataset with a recognition accuracy of 96.59%.

Reference [11] proposed SLTM for Arabic Sign Language using two datasets using the Synthetic Minority oversampling technique (SMOTE) to attain a test accuracy of 97.29%. The impact of skewed data on system accuracy is also investigated. Seven convolution layers with max pooling, ReLU activation, Batch normalisation and dropout layers are used for classification. Reference [12] proposed a real-time two-way communication system for ISL using the Canny Edge detection technique and CNN to convert sign language gestures to voice and vice versa. Reference [13] developed a hybrid framework called FiST_CNN as a combination of Fast Accelerated Segment Test (FAST), Scale Invariant Feature Transformation (SIFT), and CNN for the recognition of one-hand static ISL alphabets with an accuracy of 95.87%. Reference [14] proposed a model for multilabel classification using wearable sensors sEMG and IMU signals for the recognition of ISL.

Reference [15] proposed real-time translation of ISL using adaptive thresholding and hybridised SIFT with an accuracy of 92.78%. Reference [16] proposed a dynamic model for the recognition of alphanumeric gestures of ISL using Speeded Up Robust Features (SURF) with Support Vector Machine (SVM) and CNN. Canny Edge detection and Gaussian filtering are used for the pre-processing of images. Six Convolution layered CNN is used along with max Pooling, dropout, RELU activation function, and softmax as a classifier. The system uses a self-made dataset of 36000 images of 36 classes from 3 signers. Reference [17] proposed a model for the translation of static ASL gestures of alphanumeric data using fine-tuned CNN. The model comprises 8 layers with max max-pooling, dropout, ReLU activation and softmax as the classifier. Reference [18] used fifth Dimension Technology (5DT) gloves for data acquisition and k-Nearest Neighbour (k-NN) machine learning algorithm for Australian SLR with an accuracy of 97%. A real-time SLTS has been developed to translate the alphabet of Arabic Sign Language using AlexNet architecture [19]. A hybrid sign language recognition model based on CNN and LSTM was proposed by [20] to aid COVID-19 patients. The model was tested on two datasets and obtained a maximum average accuracy of 99.34%. Reference [21] reviewed 72 sensor-based sign language recognition systems and concluded that although they achieve accuracy up to 99.75%, these should be more user-friendly and have minimal circuitry without drawing attention. Further, a sensor-based system works better in experimental setup than in real-world situations. Aparna et al. [22] suggested CNN and stacked long short-term memory (LSTM) models for the recognition of isolated words of ISL. The model achieved a training accuracy of 94% on a self-made video dataset of 6 isolated words. For continuous word recognition in sign language, the different variations of the transformer have been used extensively by researchers to exploit spatial-temporal features [23,24]. Few works have also been done to explore the role of contextual information in sign language recognition with Generative adversarial network (GAN) [25].

The cost and obligation to wear hardware-based SLTS outweigh their accuracy. So, instead of using high-end equipment, we intend to overcome this challenge using cutting-edge computer vision and machine learning methods. It has been analysed that major work of alphabet-level SLTS has been performed in deep learning, which is a reflection of the machine learning (ML) paradigm and expedites learning by simulating human brain activity. Further, computer vision technology has become widely

popular in Sign Language processing as a result of innovations in deep learning. It incorporates the use of numerous layers of neural networks for complicated processing. This study aims to identify alphabets in Indian Sign Language based on the input images of gestures using deep learning. In this research, we have introduced novel hybrid-AO thresholding to translate the alphabets of ISL using state of art CNN framework.

3 Methodology

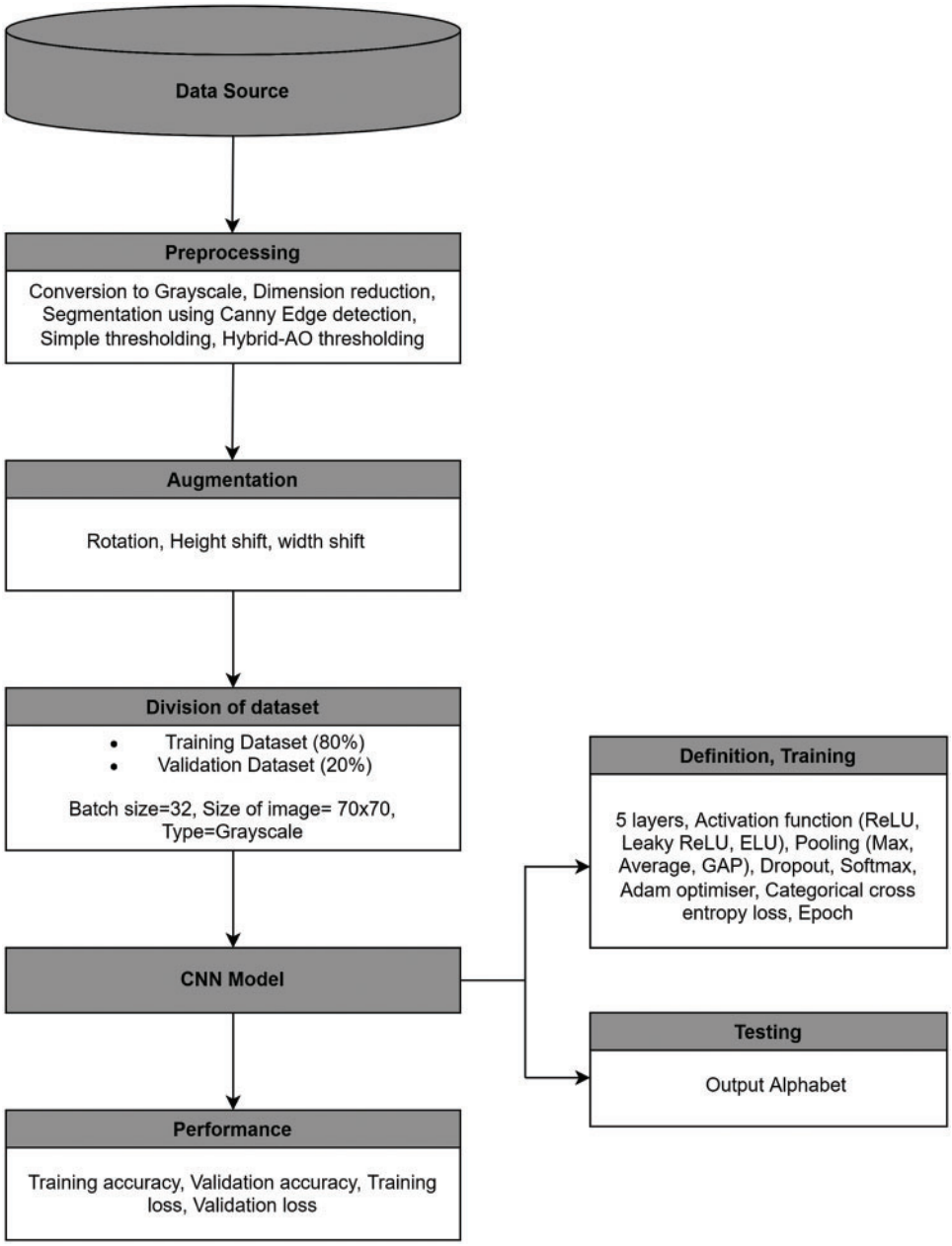


Figure 1: Architecture of the proposed ISLTS

The proposed model for the translation of the ISL alphabet to text is rendered using Fig. 1. The SLTS can be divided into the following steps:

Step 1-Alphabet level dataset of 26 classes has been taken from open access data source Kaggle [26].

Step 2-Data pre-processing has been performed by changing the coloured image to grayscale, dimension has been reduced to 70 × 70. A novel image segmentation called Hybrid-AO thresholding (Section 3.2.3) has been used by applying Adaptive Gaussian thresholding followed by Otsu Algorithm. Canny edge detection algorithm and Simple thresholding have also been performed on the input dataset to check the performance of Hybrid-AO thresholding.

Step 3-Three augmentation operations (Rotation, Height shift and Width shift) have been applied to segmented images to increase the number of images and reduce overfitting leading to a total of approx. 31 K grayscale samples.

Step 4-Dataset has been further divided into 80:20 for the training and testing phase of the CNN model of deep learning. The value of batch size has been taken as 32.

Step 5-CNN model has been applied to the pre-processed images for classification. Nine CNN models have been proposed based on three types of pooling and activation functions. The value of different parameters such as the number of layers, dropout, epoch, number of filters, size, stride, etc., has been defined.

Step 6-The performance of these models has been compared using training and validation accuracy.

3.1 Dataset

An open-access alphanumeric dataset for ISL is taken from Kaggle [26] as a data source. The dataset contains 31200 RGB images of 26 classes of alphabets from A to Z numbered from 0 to 25. Each class has 1200 images with 128 × 128 size. Fig. 2 displays the images of the alphabet used for translation in our proposed framework.

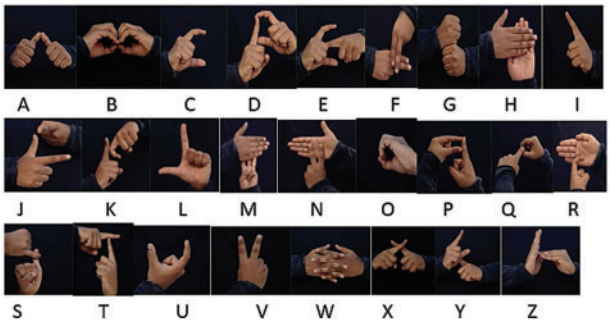


Figure 2: Alphabets of ISL dataset

3.2 Image Segmentation

The first and most important step in building SLTS is data pre-processing in which raw input data is prepared for the model. Initially, all the images are labelled and sorted into respective alphabet categories. To facilitate processing, the dimensions of each image in the dataset are reduced to 70 × 70

size and transformed to grayscale. Further, a gaussian filter is used to reduce the noise and smoothen the image.

Hand gesture recognition is an important phase of SLTS, so the foreground of the image must be differentiated from the background. Image segmentation is the mechanism of divvying a digital image into different regions for efficient analysis. The following image segmentation techniques are employed in this research article.

3.2.1 Canny Edge Detection Algorithm

It is a technique of image segmentation used for edge detection in images [27]. In this, the first Gaussian filter G (x, y) is applied to grayscale image f (x, y) for smoothness and removing noise using Eq. (1), where σ is the space scale coefficient or standard deviation specifying the amount of smoothing.

$$G(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\pi\sigma^2}\right) \tag{1}$$

Secondly, gradient magnitude G (x, y) and gradient direction Θ (i, j) is calculated using Eqs. (2) and (3) respectively where G_x and G_y are gradient in two classes.

$$G(x,y) = \sqrt{G_x^{\,2}(i,j) + G_y^{\,2}(i,j)} \tag{2}$$

$$\theta(i,j) = acrctan\left[\frac{G_y(i,j)}{G_x(i,j)}\right] \tag{3}$$

Thirdly, the algorithm aims to look for pixels of maximum edge direction value by traversing all the values of the gradient intensity matrix. This step is called Non-maximum Suppression and the output is binary images with thin edges.

Finally, double thresholding considers only pixels with a high and low threshold value, otherwise, they are discarded leading to final edges. This step is called Hysteresis Thresholding.

3.2.2 Simple Thresholding

Thresholding is another technique of image segmentation that segregates contrasting regions by comparing pixel intensity. Binary Inverse Thresholding is applied to calculate the value of threshold T on a given input image I (x, y) using Eq. (4).

$$I_d(x,y) = \begin{cases} 0, & I_s(x,y) < T \\ max, & otherwise \end{cases} \tag{4}$$

where I_s (x, y) is the source image, I_d (x, y) is the destination image and T is the threshold value.

3.2.3 Hybrid-AO Thresholding

The third technique is the hybrid-AO thresholding method of image segmentation based on a combination of Adaptive Gaussian thresholding and the Otsu algorithm is used. In Adaptive Gaussian Thresholding, local areas of the image are evaluated analytically to determine the optimal threshold value T using Eq. (5).

$$T = mean(I_L - C) \tag{5}$$

where I_L is the local subarea of the image, and C is the constant value to optimize threshold T. There are two ways to calculate mean-Arithmetic and Gaussian Mean. Since we have used the Gaussian mean, the method is used to calculate threshold value in Adaptive thresholding, hence it is called Adaptive Gaussian Thresholding. The weighted sum of neighbouring values in a Gaussian window is the threshold value. The images in the dataset consist of a black background, so Binary Inverse Adaptive Thresholding is used.

Otsu's algorithm automatically evaluates an optimum threshold based on pixel values' observed distribution [28]. It is used to separate the foreground from a background of an image by finding the global thresholding value. It maximizes between class variance values; for this, image f (x, y) is scanned for all the possible values of the threshold. f (x, y) is a grayscale image having threshold value (t) from $(0 \leq t \leq L-1)$.

ω_f (t), ω_b (t) are the probabilities of two classes divided by threshold t denoting foreground and background, the value of threshold ranges from 0 to 255 given by Eqs. (6) and (7).

$$\omega_f(t) = \sum_{i=0}^{t-1} p(i) \tag{6}$$

$$\omega_b(t) = \sum_{i=t}^{L-1} p(i) \tag{7}$$

$$p_i = \frac{n_i}{n} \tag{8}$$

where p is the probability of gray levels, t is a threshold, L is bins of histogram and n_i is the number of pixels in the gray level and n is several pixels in the overall image. σ_f, σ_b is the variance of background and foreground and σ is the total variance in threshold t, given by Eq. (9).

$$\sigma_b^2 t = \sigma^2 - \sigma_f^2 t = \omega_f(t)\,\omega_b(t)\left[\mu_f(t) - \mu_b(t)\right]^2 \tag{9}$$

Foreground and background Pixel intensity values for two classes C_1 (0 to t−1) and C_2 (t to L−1) are given by Eqs. (10) and (11).

$$\mu_f = \sum_{i=0}^{t-1} \frac{ip_i}{\omega_f(t)} \tag{10}$$

$$\mu_b = \sum_{i=t}^{L-1} \frac{ip_i}{\omega_b(t)} \tag{11}$$

From the above equations, we can say that sum of probabilities of foreground and background ω_f, ω_b will be 1 given by Eq. (12).

$$\omega_f + \omega_b = 1 \tag{12}$$

From Eq. (12), we can have Eq. (13).

$$\omega_f \mu_f + \omega_b\, \mu_b = \mu_T \tag{13}$$

where μ_T which is the average gray mean of the entire image can be given by Eq. (14).

$$\mu_T = \sum_{i=0}^{L-1} ip_i \tag{14}$$

There is the variance of foreground and background class defined by σ_f^2, σ_b^2 in Eqs. (15) and (16).

$$\sigma_f^2 = \sum_{i=0}^{t} \frac{\left(i - \mu_f\right)^2 p_i}{\omega_f} \tag{15}$$

$$\sigma_b^2 = \sum_{i=t+1}^{L-1} \frac{(i - \mu_b)^2 p_i}{\omega_b} \tag{16}$$

There are three types of variances defined for these classes i.e., total variance σ_T^2 given by Eq. (17), the variance within the class σ_w^2 given by Eq. (18), and variance between the class σ_{bet}^2 given by Eq. (19).

$$\sigma_T^2 = \sum_{i=0}^{L-1} (i - \mu_T)^2 p_i \tag{17}$$

$$\sigma_w^2 = \omega_f \sigma_f^2 + \omega_b \sigma_b^2 \tag{18}$$

$$\sigma_{bet}^2 = \omega_f \left(\mu_f - \mu_T\right)^2 + \omega_b \left(\mu_b - \mu_T\right)^2 \tag{19}$$

From Eqs. (17)–(19) we can infer Eq. (20).

$$\sigma_T^2 = \sigma_{bet}^2 + \sigma_w^2 \tag{20}$$

Optimal thresholding after performing discriminant analysis on the gray level can be given using Eq. (21).

$$t = arg_{0 \le t \le L-1} max \left\{\sigma_{otsu}^2 (t)\right\} \tag{21}$$

where $\sigma_{otsu}^2 (t)$ can be given finally using Eq. (22).

$$\sigma_{otsu}^2 (t) = \frac{\left[\mu_T \omega_f (t) - \mu_f\right]^2}{\omega_f (t) \left[1 - \omega_f (t)\right]} \tag{22}$$

Data augmentation includes a set of operations to increase data in the dataset. Rotation, height shift, and width shift operations are performed in data augmentation leading to a total of more than 31 K samples. The dataset is divided into two classes training and validation in the ratio of 80:20. Segmented grayscale images of 70 × 70 size are grouped into a batch size of 32.

3.3 Convolution Neural Network

CNN is popular for its performance in image classification and therefore has a vast span of applications in various fields [29] such as emotion detection [30], the medical field, the agriculture field [2], Natural Language Translation [31–33] and sign language translation. The CNN model is a feedforward artificial neural network used in image recognition tasks because of its optimal performance. In this, connectivity between neurons is similar to the organization of the human visual cortex. A 5-layer CNN model is used for the translation of pre-processed and segmented ISL datasets.

The model consists of three Convolution layers in which convolution operation is performed in 2-dimension using Eq. (23).

$$S (i,j) = (I * K) (i,j) = \sum_m \sum_n I (m,n) K (i - m, j - n) \tag{23}$$

$$Z_i = \sum_j W_{i,j} X_j + B_i \tag{24}$$

$$Y = softmax (Z) \tag{25}$$

$$softmax\,(Z_i) = \frac{\exp\,(Z_i)}{\sum_j \exp\,(Z_j)} \tag{26}$$

where S is the feature map; I is the input image; K is the kernel; m, n are dimensions; i, j are variables;

In Eq. (24), Z is the output of the neuron; W is weight; X is input; Y is output. SoftMax is the activation function used to classify multiple classes in the proposed CNN model, given by Eqs. (25) and (26).

Window-size learnable filters make up the convolution layer. The stride size is taken to be 1. We use a small window size [length $5 * 5$] for the convolution layer that spans the depth of the input matrix. The layer is made up of window-sized learnable filters. We moved the window by a certain amount (usually one stride size) during each iteration to compute the dot product of the filter entries and the input values at a specific location. As we proceed, a 2-Dimensional activation matrix will be created, which will respond to that matrix at every spatial place. In other words, the network will learn filters that turn on when it encounters certain visual features, such as an edge of a certain orientation or a splotch of a certain colour.

The pooling layer is used to reduce the computations and decrease the size of the activation matrix. In Max Pooling, the maximum value of activation is chosen out of window size using the formula [15] shown in Eq. (27).

$$s_j = \max_{i \in R_j} a_i \tag{27}$$

where S_j output function for max pooling and R_j is the set of all activation functions a_i.

In Average Pooling average value of activation is chosen out of window size using the formula shown in Eq. (28).

$$s_j = \frac{1}{|R_J|} \sum_{i \in R_j} a_i \tag{28}$$

In Global Average (Max) pooling, the GAP layer is used at the outer connected layer instead of the flattening layer and max pooling is used in the interior layers.

A comparison of three activation functions Rectified Linear Unit (ReLU), Leaky ReLU, and Exponential Linear Unit (ELU) is performed with the above-mentioned types of pooling.

A summary of the proposed work is shown in Fig. 3. It depicts 3 convolution layers of 32 kernels of window size $5 * 5$ with pooling (Max, Global average and average), dropout for regularisation, activation function (ReLU, Leaky ReLU and ELU) and SoftMax activation function as the classifier. The first phase of the model consists of 3 convolution layers with a first layer containing 32 kernels of 3×3 size and stride value as 1. There is a single hidden dense layer with 128 neurons in our proposed model. In addition to this, early stopping with patience is used to deal with the problem of overfitting. A dropout layer with a probability of 0.25 has been used. Fig. 3a displays the summary of the proposed CNN model with average pooling and further three combinations of the activation function. 4096 neurons have been used in the first input layer of the Artificial Neural Network (ANN). Flatten layer has been used to convert the feature map into a single dimension. Similarly, in Fig. 3b, Global average pooling in the outer layer and max pooling has been used in the inner layers. Instead of flattening the layer, global average pooling has been used for a better representation of the output of the convolution layer. Fig. 3c display the max pooling with three combinations of the activation function. In this, the

first convolution layer has 32 filters, the second and third layer has 64 filters. There are 4096 nodes in the input layer of ANN. There are 26 nodes in the final output layer to classify the alphabets of ISL.

Layer (type)	Output Shape	Param #
conv2d_12 (Conv2D)	(None, 68, 68, 32)	320
average_pooling2d_6 (AveragePooling2D)	(None, 34, 34, 32)	0
dropout_12 (Dropout)	(None, 34, 34, 32)	0
conv2d_13 (Conv2D)	(None, 34, 34, 64)	18496
average_pooling2d_7 (AveragePooling2D)	(None, 17, 17, 64)	0
dropout_13 (Dropout)	(None, 17, 17, 64)	0
conv2d_14 (Conv2D)	(None, 17, 17, 64)	36928
average_pooling2d_8 (AveragePooling2D)	(None, 8, 8, 64)	0
dropout_14 (Dropout)	(None, 8, 8, 64)	0
flatten_4 (Flatten)	(None, 4096)	0
dense_8 (Dense)	(None, 128)	524416
dense_9 (Dense)	(None, 26)	3354

Total params: 583,514
Trainable params: 583,514
Non-trainable params: 0

(a)

Layer (type)	Output Shape	Param #
conv2d_27 (Conv2D)	(None, 68, 68, 32)	320
max_pooling2d_18 (MaxPooling2D)	(None, 34, 34, 32)	0
dropout_27 (Dropout)	(None, 34, 34, 32)	0
conv2d_28 (Conv2D)	(None, 34, 34, 64)	18496
max_pooling2d_19 (MaxPooling2D)	(None, 17, 17, 64)	0
dropout_28 (Dropout)	(None, 17, 17, 64)	0
conv2d_29 (Conv2D)	(None, 17, 17, 64)	36928
max_pooling2d_20 (MaxPooling2D)	(None, 8, 8, 64)	0
dropout_29 (Dropout)	(None, 8, 8, 64)	0
global_average_pooling2d_2 (GlobalAveragePooling2D)	(None, 64)	0
dense_18 (Dense)	(None, 128)	8320
dense_19 (Dense)	(None, 26)	3354

Total params: 67,418
Trainable params: 67,418
Non-trainable params: 0

(b)

Layer (type)	Output Shape	Param #
conv2d_15 (Conv2D)	(None, 68, 68, 32)	320
max_pooling2d_6 (MaxPooling2D)	(None, 34, 34, 32)	0
dropout_15 (Dropout)	(None, 34, 34, 32)	0
conv2d_16 (Conv2D)	(None, 34, 34, 64)	18496
max_pooling2d_7 (MaxPooling2D)	(None, 17, 17, 64)	0
dropout_16 (Dropout)	(None, 17, 17, 64)	0
conv2d_17 (Conv2D)	(None, 17, 17, 64)	36928
max_pooling2d_8 (MaxPooling2D)	(None, 8, 8, 64)	0
dropout_17 (Dropout)	(None, 8, 8, 64)	0
flatten_5 (Flatten)	(None, 4096)	0
dense_10 (Dense)	(None, 128)	524416
dense_11 (Dense)	(None, 26)	3354

Total params: 583,514
Trainable params: 583,514
Non-trainable params: 0

(c)

Figure 3: Summary of proposed CNN model (a) Average Pooling with ReLU, Leaky ReLU, ELU (b) GA (Max) Pooling with ReLU, Leaky ReLU, ELU (c) Max Pooling with ReLU, Leaky ReLU, ELU

4 Results and Discussion

Python programming language is used to implement this entire experiment on Google Colab Pro with TensorFlow, Keras, NumPy, Matplotlib, OpenCV python packages, categorical cross entropy and Adam optimiser.

In the image segmentation phase, the output image of proposed Hybrid-AO thresholding is compared with Canny Edge detection, and Binary Inverse thresholding and the same is shown in Fig. 4. It has been observed that Hybrid-AO thresholding produces better results than Canny Edge Detection and Binary Inverse Thresholding due to automatic calculation of threshold value. It uses peripheral pixel values to analyze and process the input image and produces minimal noise segmented output image.

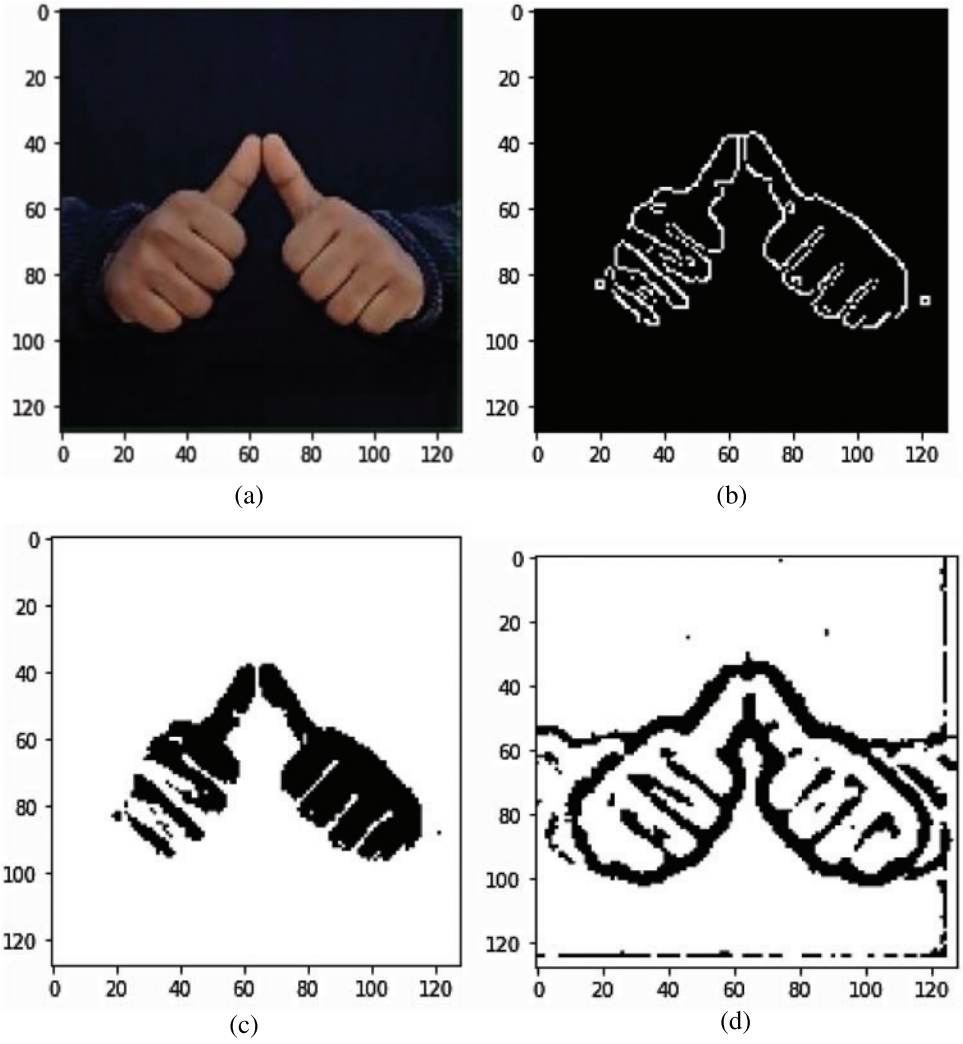


Figure 4: Image segmentation results (a) Original RGB image (b) Canny edge detection (c) Simple thresholding (d) Hybrid-AO thresholding

The segmented image is fed to 5 layered CNN sign language translation framework. And to optimise the performance of the proposed framework, three different types of pooling and three

different types of activation functions were considered leading to a total of nine models. The segmentation results obtained from one sign language gesture are shown in Fig. 4d, which verifies the denoising performance of HAO thresholding. In the architectural aspect, 5-layer CNN has been used with leaky ReLU activation function and max pooling. The comparison between three commonly used activation functions and pooling techniques has been performed to obtain the superlative performance of the proposed model.

Table 1 displays the performance metrics of 9 models, i.e., 5AVEL-HAO, 5AVLR-HAO, 5AVRL-HAO, 5GAEL-HAO, 5GALR-HAO, 5GARL-HAO, 5MXEL-HAO, 5MXL-HAO, and 5MXRL-HAO in terms of training accuracy, validation accuracy, training loss and validation loss. The highest and lowest training accuracy has been attained by 5MXLR-HAO and 5GALR-HAO Indian sign language translation frameworks respectively. Similarly, the highest and lowest validation accuracy has been obtained by 5MXLR-HAO and 5GAEL-HAO Indian sign language translation frameworks, respectively.

Table 1: Performance metrics of proposed models

Nomenclature	Pooling	Activation function	Training accuracy	Validation accuracy	Training loss	Validation loss
5AVEL-HAO	Average	ELU	92.97	92.76	0.1911	0.2045
5AVLR-HAO	Average	Leaky ReLU	94.90	92.97	0.1152	0.1914
5AVRL-HAO	Average	ReLU	95.31	94.90	0.1102	0.1189
5GAEL-HAO	GAP	ELU	91.78	55.47	0.2727	1.6617
5GALR-HAO	GAP	Leaky ReLU	90.62	77.34	0.2544	0.5279
5GARL-HAO	GAP	ReLU	92.27	90.62	0.2358	0.4337
5MXEL-HAO	Max	ELU	97.66	94.90	0.0731	0.1138
5MXLR-HAO	**Max**	**Leaky ReLU**	**98.95**	**98.05**	**0.0725**	**0.1021**
5MXRL-HAO	Max	ReLU	96.38	94.53	0.1027	0.1490

Nine models are proposed based on a comparison between types of pooling and activation functions. The graph of all these nine models is depicted in Fig. 5. The X-axis denotes the number of epochs and the y-axis is used to depict training accuracy, validation accuracy, training loss and validation loss. After 25 epochs, 5MXLR-HAO topped the chart with 98.95% training accuracy and 98.05% validation accuracy and minimum training and validation loss when compared with other proposed ones.

After choosing the best framework for ISLTs, it is important to compare it with existing ones. A comparison of the proposed framework with other existing ones is shown in Table 2. Adaptive thresholding is used in [15] with an accuracy of 92.78% which is lesser than our proposed model. In [34] Modified Canny Edge Detection technique for segmentation is used to achieve a validation accuracy of 95% while [35] accomplishes 90.43% validation accuracy with YOLOv3 with background subtraction and edge detection for translation of Sign language. It has been observed that the accuracy of our proposed framework is better than the three existing systems.

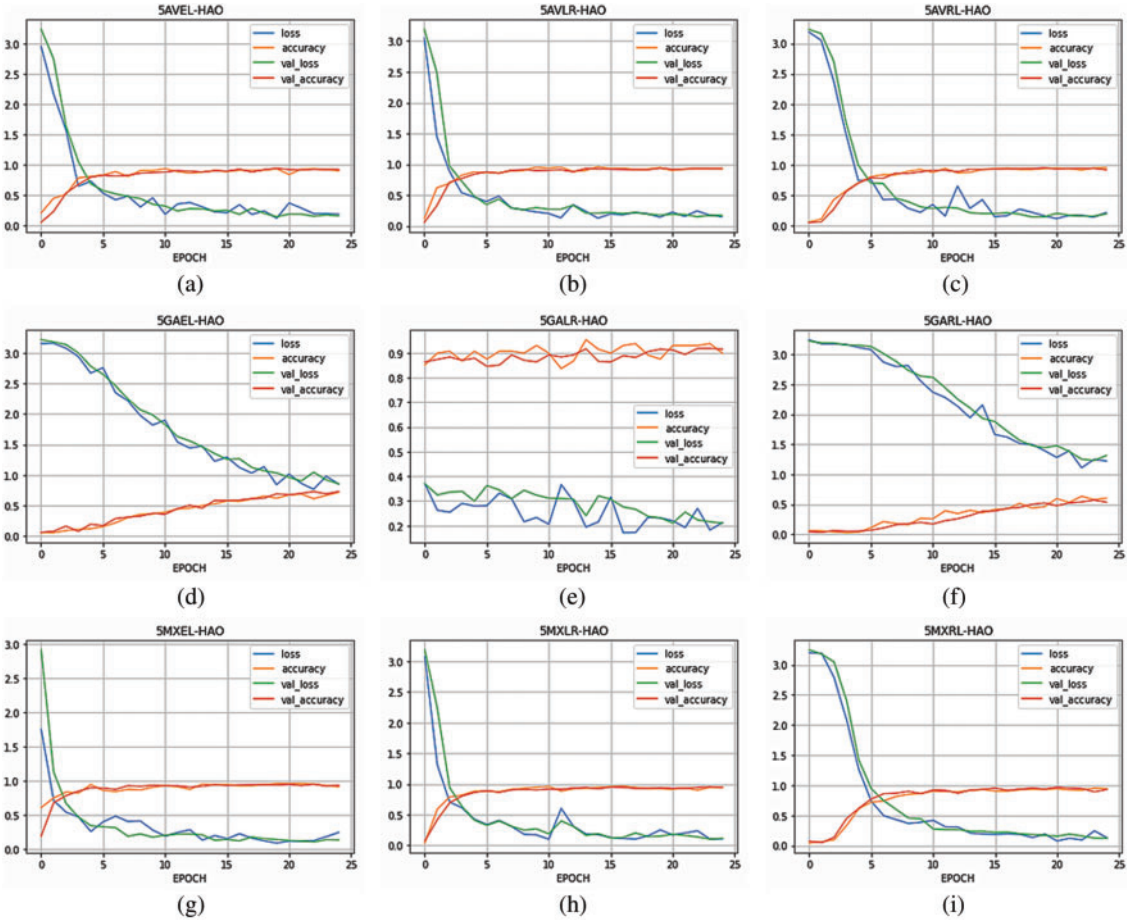


Figure 5: Training and validation accuracy and training and validation loss of proposed SLTM (a) 5AVEL-HAO (b) 5AVLR-HAO (c) 5AVRL-HAO (d) 5GAEL-HAO (e) 5GALR-HAO (f) 5GARL-HAO (g) 5MXEL-HAO (h) 5MXLR-HAO (i) 5MXRL-HAO

Table 2: Comparison of the proposed ISLTS approach with the existing framework

Reference	Technique	Validation accuracy (%)
[15]	Adaptive thresholding	92.78
[34]	Modified canny edge segmentation	95
[35]	YOLOv3 with background subtraction and edge Detection	90.43
Proposed work	**5MXLR-HAO**	**98.05**

In this study, we have presented and applied the proposed 5MXLR-HAO framework for the translation of ISL. The framework has two important phases- firstly segmentation part, and secondly architectural aspect of CNN. In the segmentation part, we have calculated optimal thresholding by applying adaptive Gaussian thresholding, the Otsu algorithm and binary inverse thresholding. Adaptive Gaussian thresholding reduces the noise and increases the sharpness of the image and calculates the regional threshold value. Binary inverse thresholding has been used to separate the background black colour from the foreground gesture. Finally, the Otsu algorithm has been used to calculate the global threshold value of the input image. It has been analyzed that max-pooling gave better results than the average and the global average with max-pooling and leaky ReLU activation function performed better than ReLU and ELU. So, the highest loss in terms of training and validation has been observed in the case of GA pooling and ELU activation functions.

5 Conclusion

In this paper, an alphabet-level framework for the translation of Indian Sign Language into text has been proposed based on hybrid-AO thresholding and deep learning. 5-layer CNN framework i.e., 5MXLR-HAO has been selected as the best performing framework among other models based on variations in image segmentation (Canny Edge Detection, Simple thresholding and Hybrid-AO thresholding), pooling (Max, Average, GAP) and activation function (ReLU, Leaky ReLU, ELU). The proposed framework has shown improved performance by attaining a training accuracy of 98.95% and a validation accuracy of 98.05%. Hybrid-AO segmentation can also be applied in various fields such as medical imaging, sign language processing, and other image processing applications. The biggest limitation of our work is the fact that it is tested on a single dataset, so it can be tested on various datasets. Future work would be to contemplate various factors such as variation in the number of layers of CNN, optimization function, and inclusion of non-manual features in the dataset.

Funding Statement: The authors received no specific funding for this study.

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

References

[1] M. Chaitoo, "International Day of Sign Languages," United Nations, 2015. [Online]. Available: https://www.un.org/en/observances/sign-languages-day

[2] U. Nandi, A. Ghorai, M. M. Singh, C. Changdar, S. Bhakta *et al.,* "Indian sign language alphabet recognition system using CNN with diffGrad optimizer and stochastic pooling," *Multimedia Tools and Applications*, pp. 1–22, 2022. https://doi.org/10.1007/s11042-021-11595-48

[3] M. S. Amin, S. T. H. Rizvi and M. M. Hossain, "A comparative review on applications of different sensors for sign language recognition," *Journal of Imaging*, vol. 8, no. 4, pp. 98, 2022.

[4] M. Al-Qurishi, T. Khalid and R. Souissi, "Deep learning for sign language recognition: Current techniques, benchmarks, and open issues," *IEEE Access*, vol. 9, pp. 126917–126951, 2021.

[5] K. Divya Lakshmi and S. R. Balasundaram, "Evaluation of machine learning models for sign language digit recognition," in *Proc. of Int. Conf. on Artificial Intelligence: Advances and Applications*, Jaipur, India, pp. 491–499, 2022.

[6] K. Myagila and H. Kilavo, "A comparative study on performance of SVM and CNN in Tanzania sign language translation using image recognition," *Applied Artificial Intelligence*, vol. 36, no. 1, pp. 2005297, 2022.

[7] R. Rastgoo, K. Kiani and S. Escalera, "Sign language recognition: A deep survey," *Expert Systems with Applications*, vol. 164, pp. 113794, 2021.
[8] M. A. Hossen, A. Govindaiah, S. Sultana and A. Bhuiyan, "Bengali sign language recognition using deep convolutional neural network," in *Proc. of Joint 7th Int. Conf. on Informatics, Electronics & Vision (ICIEV) and 2nd Int. Conf. on Imaging, Vision & Pattern Recognition (icIVPR)*, Kitakyushu, Japan, pp. 369–373, 2018.
[9] M. M. Kamruzzaman, "Arabic sign language recognition and generating arabic speech using convolutional neural network," *Wireless Communications and Mobile Computing*, vol. 2020, pp. 1–9, 2020.
[10] M. A. Rahim, J. Shin and K. S. Yun, "Hand gesture-based sign alphabet recognition and sentence interpretation using a convolutional neural network," *Annals of Emerging Technologies in Computing (AETiC)*, vol. 4, no. 4, pp. 20–27, 2020.
[11] A. A. Alani and G. Cosma, "ArSL-CNN a convolutional neural network for arabic sign language gesture recognition," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 22, no. 2, pp. 1096, 2021.
[12] V. Brahmankar, N. Sharma, S. Agrawal, S. Ansari, P. Borse *et al.,* "Indian sign language recognition using canny edge detection," *International Journal of Advanced Trends in Computer Science & Engineering*, vol. 10, no. 3, pp. 1576–1583, 2021.
[13] A. Tyagi and S. Bansal, "Feature extraction technique for vision-based Indian sign language recognition system: A review," in *Proc. of Computational Methods and Data Engineering, ICMDE 2020*, Singapore, Springer, vol. 1227, pp. 39–53, 2021.
[14] R. Gupta and A. Kumar, "Indian sign language recognition using wearable sensors and multi-label classification," *Computers & Electrical Engineering*, vol. 90, pp. 106898, 2021.
[15] S. Rajarajeswari, N. M. Renji, P. Kumari, M. Keshavamurthy and K. Kruthika, "Real-time translation of Indian sign language to assist the hearing and speech impaired," in *Proc. of Innovations in Computational Intelligence and Computer Vision*, Singapore, Springer, vol. 1424, pp. 303–322, 2022.
[16] S. Katoch, V. Singh and U. S. Tiwary, "Indian sign language recognition system using SURF with SVM and CNN," *Array*, vol. 14, pp. 100141, 2022.
[17] A. Mannan, A. Abbasi, A. R. Javed, A. Ahsan, T. R. Gadekallu *et al.,* "Hypertuned deep convolutional neural network for sign language recognition," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–10, 2022.
[18] S. Johnny and S. J. Nirmala, "Sign language translator using machine learning," *SN Computer Science*, vol. 3, no. 1, pp. 36, 2022.
[19] Z. Alsaadi, E. Alshamani, M. Alrehaili, A. A. D. Alrashdi, S. Albelwi *et al.,* "A real-time arabic sign language alphabets (ArSLA) recognition model using deep learning architecture," *Computers*, vol. 11, no. 5, pp. 78, 2022.
[20] A. Venugopalan and R. Reghunadhan, "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*, pp. 1–14, 2022.
[21] K. Kudrinko, E. Flavin, X. Zhu and Q. Li, "Wearable sensor-based sign language recognition: A comprehensive review," *IEEE Reviews in Biomedical Engineering*, vol. 14, pp. 82–97, 2021.
[22] C. Aparna and M. Geetha, "CNN and stacked LSTM model for Indian sign language recognition," in *Proc. of Symp. on Machine Learning and Metaheuristics Algorithms, and Applications*, Trivandrum, India, vol. 1203, pp. 126–134, 2020.
[23] W. Aditya, T. K. Shih, T. Thaipisuthikul, A. S. Fitriajie, M. Gochoo *et al.,* "Novel spatio-temporal continuous sign language recognition using an attentive multi-feature network," *Sensors*, vol. 22, no. 17, pp. 6452, 2022.
[24] P. Xie, M. Zhao and X. Hu, "PiSLTRc: Position-informed sign language transformer with content-aware convolution," *IEEE Transactions on Multimedia*, vol. 24, pp. 3908–3919, 2022.
[25] I. Papastratis, K. Dimitropoulos and P. Daras, "Continuous sign language recognition through a context-aware generative adversarial network," *Sensors*, vol. 21, no. 7, pp. 2437, 2021.

[26] P. Arikeri, "Indian sign language," 2021. [Online]. Available: https://www.kaggle.com/datasets/prathumarikeri/indian-sign-language-isl
[27] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed., India: Pearson Education, pp. 711–783, 2009.
[28] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems Man Cybernetics-Systems*, vol. 9, no. 1, pp. 62–66, 1979.
[29] H. Varun Chand and J. Karthikeyan, "CNN based driver drowsiness detection system using emotion analysis," *Intelligent Automation & Soft Computing*, vol. 31, no. 2, pp. 717–728, 2022.
[30] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan *et al.,* "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, pp. 53–74, 2021.
[31] S. Bawa, "Sanskrit to universal networking language EnConverter system based on deep learning and context-free grammar," *Multimedia Systems*, pp. 1–17, 2020.
[32] S. Bawa, "SANSUNL: A sanskrit to UNL enconverter system," *IETE Journal of Research*, vol. 67, no. 1, pp. 117–128, 2021.
[33] S. Bawa, S. Sitender, M. Kumar and S. Sangeeta, "A comprehensive survey on machine translation for English, Hindi and Sanskrit languages," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–34, 2021.
[34] N. N. Alleema, S. Babeetha, P. S. Kumar, S. Chandrasekaran, S. Pandiaraj *et al.,* "Recognition of American sign language using modified deep residual CNN with modified canny edge segmentation," *SSRN Journal*, 2022. https://doi.org/10.2139/ssrn.4052252
[35] Y. V. Kuriakose and M. Jangid, "Translation of American sign language to text: Using YOLOv3 with background subtraction and edge detection smart innovation, systems and technologies," in *Proc. of Smart Systems: Innovations in Computing*, Singapore, Springer, vol. 235, pp. 21–30, 2022.

Received: April 30, 2023. Revised: June 2, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Optimised Machine Learning-based Translation of Indian Sign Language to Text

Seema Sabharwal[1]* **Priti Singla[1]**

[1]*Department of Computer Science and Engineering, Baba Mastnath University, Rohtak, Haryana, 124001, India*
* Corresponding author's Email: sabharwalseema@gmail.com

Abstract: Sign language translation through deep learning is a popular topic among researchers nowadays. It opens the doors of communication for deaf and mute people by translating sign language gestures. The translation of input sign language gestures into text is called the sign language translation system (SLTS). In this paper, optimised machine learning-based SLTS for Indian sign language (ISL) has been proposed to facilitate deaf-mute persons. Further, this paper presents a simulation analysis of the impact of the number of convolution layers, size of stride function, epochs, and activation function on the accuracy of translation of ISL gestures. An optimised ISL translation system (ISLTS) for fingerspelled alphanumeric data of 36 classes using a convolution neural network (CNN) with a novel RADAM_NORM optimiser has been proposed. The proposed system has been implemented using two datasets- the first customised ISL alphanumeric dataset has been taken from Kaggle and the second dataset has been prepared by the author consisting of 36 classes and nearly 50K images. The accuracy of the proposed ISLTS on the first dataset is 99.446 % and on the second dataset is 97.889%.

Keywords: Indian sign language, Radam_norm optimiser, ISL dataset, Sign language translation system.

1. Introduction

Machine learning is a subfield of artificial intelligence that entails the development of algorithms that allows machines to learn from data and enhance their efficacy on particular tasks instead of being explicit coding[1]. Contrary to conventional supervised algorithms, machine learning algorithms are non-parametric supervised techniques that don't presume the statistical dissemination of input feature sets. Convolution neural network (CNN) is a type of machine learning paradigm that has been suggested in the literature for the effective translation or recognition of sign language gestures. It revolutionized the computer vision domain and provided state of art results in this arena. CNNs exceptional performance is largely attributed to the brilliant design ideas by researchers and the meticulous hyperparameter value choices. Recommendation systems, Natural language processing, are the domains in which CNN is employed [2, 3]. CNN is broadly acknowledged by a multitude of parameters and requires exhaustive tuning of hyperparameters like the layer count, batch size, activation function, optimisers, pooling and epoch size[4]–[6]. Various algorithms have been used to finetune the parameters of CNN[7]. To solve this, we have proposed hypertuned machine learning-based translation for gestures of ISL into text. The in-depth contributions of this article are as follows-

- A hypertuned machine learning-based translation system for alphanumeric ISL gestures into text has been proposed.
- An alphanumeric image dataset of ISL has been made with 36 classes and nearly 50k images.
- The role of various hyperparameters such as count of convolution layers, epochs, stride and activation function has been analysed on the efficacy of the proposed system.
- A novel optimisation function RADAM_NORM has been employed to optimise the efficiency of the Indian sign language translation system (ISLTS).

The remainder of this paper is arranged as follows: section 2 sums up the literature review, section 3 elaborates on the dataset and the methodology used, section 4 reports the results and discussion and section 5 concludes and points out some future research directions.

2. Literature review

SLTS can be categorised broadly into three parts based on the modalities- hardware-based, vision-based and hybrid SLTS. In hardware-based SLTS, wearable and device based such as cameras, webcams, data gloves, Kinect, and leap motion controllers are used to input sign language gestures[8-9]. Despite their precision, they are quite expensive, bulky and cumbersome to wear in public places. Vision-based SLTS employs image-processing practices to process sign language gestures. It is more versatile than hardware-based SLTS [10]. Hybrid SLTS is the combination of hardware and vision-based techniques. Preliminary studies of SLTS focus on using various machine learning algorithms to classify sign language gestures. Several studies have been carried out in recent years to solve the issues of automatic SLTS with the use of deep learning techniques such as CNN [11].

In 2023, [12] proposed a sign language recognition system (SLRS) for the Bangla sign language using hybrid transfer learning and a random forest (RF) classifier to classify fingerspelled alphanumeric sign language gestures to text. Adaptive thresholding is applied for pre-processing of the dataset and the proposed system achieved an average accuracy of 97.33% in the case of digits and 91.67% in the case of alphabets. Open access dataset of 2080 images was used to evaluate the model. The model works better for smaller datasets and its performance is optimised by lowering the learning rate. A multitask sign language recognition framework built on CNN and K-nearest neighbour (KNN) module was proposed by [13]. The novel concept of Wireless sensing has been used for sign language recognition, although the model achieve an accuracy of 99.9% on smaller datasets but it took higher time to train the model. In 2022, [14] proposed an integrated Media Pipe optimised gated recurrent unit (MPOGRU) for the classification of 13 dynamic ISL words. Average prediction accuracy of 95% has been attained on a real-time dataset of 30 videos for each class. Further, the word-level American and Argentinian dataset of sign language was also used to validate the proposed model. However, the model had a limited dataset and recognised only 13 words, further, it has been optimised by using ELU and softsign classifier. In 2022, [15] created an original arabic alphabet phonetics dataset (AAPD) using sound recordings of 1420 persons. Further, Mel-frequency cepstral coefficient (MFCC) with Mel-bands number 20 is used for feature extraction in VGG based Arabic speech recognition model to achieve an accuracy of 95.68%. The model has been hypertuned by focussing on feature extraction techniques and the type of neural network. In addition to this, [16] proposed a transfer learning-based alphabet-level Arabic sign language recognition system with training and testing accuracy of 98% and 95% respectively. Open access dataset has been used to train and test the gestures of Arabic sign language using EfficientNetB4. Several pre-processing techniques and pre-trained architectures have been analysed in this paper to achieve optimal performance on single-handed gestures of Arabic sign language. In 2022, [17] developed a DeepCNN deep learning-based CNN model for the recognition of 24 alphabets of American sign language (ASL) using the MNIST dataset. It attains an accuracy of 99.67% with CNN having three convolution layers in 20 epochs. The training and validation loss was 7.7924 and 7.0703 respectively. The model has been hypertuned by the varying number of convolution layers and works only for single-handed American sign language recognition. In 2021, [18] proposed transfer learning based on Faster R-CNN for the recognition of gestures in Turkish sign language. A self-made alphabet-level dataset of 29 classes has been used to attain an accuracy of 99.82%. A similar background has been used for all the signers while constructing the dataset. Batch size and region based-CNN models have been analysed to improve accuracy. In 2019, [19] developed a Korean sign language translation model using Korean electronics technology institute (KETI) video dataset. 2D coordinates of human key points are estimated from the video dataset to translate sentences into text along with the sequence-to-sequence (seq2seq) model and attention mechanism. The model was related to the emergency domain and covered 419 words, and 105 sentences with a translation accuracy of 93.28%. The insufficient size of training data and a limited number of key points have been constraints on the accuracy of the proposed model. The lack of availability of standard datasets is a major hindrance in the domain of sign language translation [20]. Further, feedforward neural networks are considered as blackbox and researchers have focussed on individual or multiple hyper-tuning of parameters (such as number of layers, number of filters, size of stride, choice of pooling, choice of activation

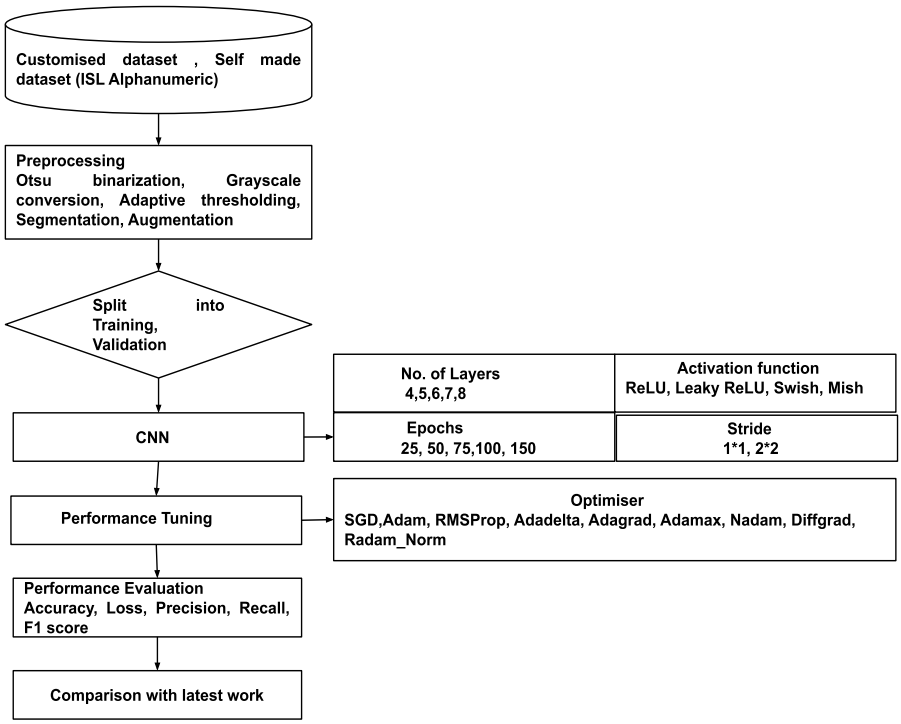


Figure. 1 Flowchart of the proposed system

function, learning rate, choice of optimiser) to have optimal performance [21]. In this paper, we have focussed on building an improvised machine learning-based Indian sign language translation system by selecting the optimal value of hyperparameters.

3. Methodology

The flowchart of our translation system has been shown in Fig. 1. The steps of the same are explained below-

3.1 Dataset

Due to the limited availability of publicly available and standard datasets, the following datasets have been employed to validate the performance of the proposed ISLTS. Firstly, customised open-access datasets from Kaggle are used for the alphabets and numbers of ISL. The dataset consists of alphanumeric data i.e., 26 alphabets (A-Z), 10 numbers (0-9). The dataset contains 36 classes and each class has 1200 images. It is referred to as dataset I. Secondly; a new image dataset is created for 26 classes of alphabets (A-Z) and 10 classes of numbers (0-9). All the images were captured by 40 persons. Different pictures per class have been taken with the signer's hand in various positions and angles. The dataset consists of nearly

Figure. 2 Proposed ISL dataset II

50K images. A sample of our dataset has been shown in Fig. 2. This dataset has been referred to as dataset II. Different backgrounds have been considered while constructing the dataset to ensure diversity and involve nonmanual features.

3.2 Data pre-processing and feature extraction

The image is transformed to a grayscale and binarized. Gaussian blur has been applied to each

input image along with Canny edge detection and adaptive thresholding. The dimensions of the input image have been reduced to 98*98 after hand detection, thresholding and segmentation. The dataset is divided into 80:20 ratios for the training and validation dataset. Data augmentation operations such as rotation, height shift, and width shift have been applied to prevent overfitting and increase the dataset.

3.3 Classification

CNNs are a particular class of machine learning algorithms that are feedforward neural networks employed in the recognition and processing of images and videos [22]. It consists of mainly three tiers-convolution, pooling and a fully connected layer. These layers are further made up of several hidden layers, a fully connected layer and a final output layer. Convolution and pooling layers are utilised for feature extraction and size reduction. The stride parameter is the extent of the shift among the application of the filter to the source image and has a default value of 1. It is uniform in height and width aspects. The role of the size of the stride function has also been analysed in this paper with the variation in several epochs. In padding, extra zeroes are added to the input matrix to maintain uniformity with output dimensions. In this paper, we have analysed the impact of the number of layers, activation function, epoch, size of stride function and optimiser on the performance of ISLTS.

The choice of hidden layers plays a vital part in the efficacy of ISLTS, as poor design may lead to overfitting and underfitting. As the count of the number of layers increases, it will lead to complexity, and few layers can't process huge image and video data. So, it is a very difficult job to decide the exact number of layers for optimal performance. In this paper, the count of convolution layers is varied to obtain maximum performance.

The term activation function symbolises the attributes of stimulated neurons that can be preserved and mapped out by a nonlinear function. It is used to address nonlinear issues. In this paper, we have compared the working of four standard activation functions ReLU, Leaky ReLU, Swish and Mish.

3.4 Finetuning

In CNN different optimisers are used to enhance the precision of the overall system. In this section, we will discuss nine states of art optimisers. Table 1. describes various notations and their description which are used for the proposed system.

Table 1. Notation list

Symbol	Description
Θ_{t+1}	Parameters (weight/bias) at time t+1
α	Learning rate
L	Loss function
$(\frac{\partial L}{\partial \Theta_t})$ or g	Gradient
ε	Small positive number
T	Iteration count
β	Decay rate
E	exponentially decaying weighted average
υ_t	Variance of the gradient at time t
m_t	Mean of the gradient at time t
$\widehat{\upsilon_{t,\iota}}$	Variance with corrected bias at time t for i^{th} parameter
$\widehat{m_{t,\iota}}$	Mean with bias correction at time t for i^{th} parameter
$\xi_{t,i}$	diffGrad friction coefficient for the i^{th} parameter at the t^{th} iteration
g_{norm}	Normalised gradient value
ρ_t	Degree of freedom (Length of approximated simple moving average)

3.4.1. Stochastic gradient descent (SGD)

It is also known as online training because all the parameters are updated as per each training image and with the same learning rate. It has slow convergence because a single record is updated in one iteration of forward and backward propagation. The training images are selected randomly to update trainable parameters i.e., weights and biases. Due to significant variance in the selected images, these modifications cause major perturbations in loss function thereby generating noise in the training phase. The weights and biases are updated based on Eq. (1).

$$\Theta_{t+1} = \Theta_t - \alpha(\frac{\partial L}{\partial \Theta_t}) \quad (1)$$

Θ_t is the value of the parameter (weight/bias) at time t, α is the learning rate, ∂L is the gradient of the loss function, $\partial\Theta_t$ is the gradient of the parameter for the selected image. The drawbacks of SGD include slow gradient, careful initialisation, sensitivity to learning rate, noisy gradient, gradient-dependent update rule and underfitting in deep architectures [1]. To deal with noise, smoothening is performed and SGD with momentum is used.

3.4.2. Adagrad

It adjusts the learning rate of each weight automatically based on previous gradient data as

opposed to previous optimisation algorithms in which the learning rate was fixed and deals with the problem of exploding gradients. Initially, the learning rate is high and then it keeps on decreasing leading to faster convergence. The equations of the parameter are shown using Eq. (2) and (3).

$$\theta_{t+1}^{i} = \theta_{t}^{i} - \alpha_t(\frac{\partial L}{\partial \theta_t}) \tag{2}$$

$$\alpha_t = \frac{\alpha}{\sqrt{\sum_{i=1}^{t}(\frac{\delta L}{\delta \theta_i})^2 + \epsilon}} \tag{3}$$

where α_t is the dynamic learning rate, t is the iteration count and ε is a small positive number.

It shows better performance with sparse data but creates problems while dealing with very deep networks or with non-convex loss functions and has premature convergence.

3.4.3. Adadelta

It is the modified version of the adagrad optimiser that resolves the problem of a relatively small learning rate in the subsequent training phases by customising it to past gradient changes and less memory requirement. In this type of optimisation algorithm, the sum of the square gradients is replaced with the exponential decaying average of the squared gradients. The parameters are updated as per Eqs. (4), (5) and (6).

$$\theta_{t+1}^{i} = \theta_{t}^{i} - \alpha_t(\frac{\partial L}{\partial \theta_t}) \tag{4}$$

$$\alpha_t = \frac{\alpha}{\sqrt{v_t + \epsilon}}, v_t = E[g^2]_t \tag{5}$$

$$v_t = \beta v_{t-1} + (1-\beta)(\frac{\partial L}{\partial \theta_t}) \tag{6}$$

where E is the exponentially decaying weighted average initialised to zero, β is the decay rate, and g is the gradient. It calculates per parameter adaptive learning rate. It requires much memory to store the additional moving average of updated learning rates It is similar to RMSProp except for the exponentially decaying average of squared parameter updates. It suffers from the problem of non-convex optimisation, slow convergence rate near minimum and computationally expensive.

3.4.4. Root mean square propagation (RMSProp)

It is a type of adaptive learning rate optimisation algorithm which is used when gradients of various parameters change widely in magnitude. It requires less memory as compared to Adadelta because it stores the moving average of squared gradients. These parameters are modified with the help of the moving average of the squared gradient, as shown in Eqs. (7) to (9).

$$\theta_{t+1} = \theta_t - \alpha_t(\frac{\partial L}{\partial \theta_t}) \tag{7}$$

$$\alpha_t = \frac{\alpha}{\sqrt{v_t + \epsilon}}, v_t = E[g^2]_t \tag{8}$$

$$v_t = \beta v_{t-1} + (1-\beta)(\frac{\partial L}{\partial \theta_t})^2 \tag{9}$$

3.4.5. Adaptive moment estimation (Adam)

It is the most commonly used optimiser in deep learning frameworks because it incorporates the merits of Nesterov momentum, AdaGrad and RMSProp optimisers. In this, features of momentum with dynamic learning rate are combined. It is an SGD approach based on the adaptive assessment of first-order and second-order moments. It is well suited for non-stationary objectives and sparse gradients. The equation of this is given by Eqs. (10) to (12).

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\alpha_t \times \widehat{m_{t,i}}}{\sqrt{\widehat{v_{t,i}} + \epsilon}} \tag{10}$$

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)(\frac{\partial L}{\partial \theta_t}) \tag{11}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2)(\frac{\partial L}{\partial \theta_t})^2 \tag{12}$$

where m_t , v_t are the mean and variance of the gradient to include the features of momentum and dynamic learning rate[23]. β_1 and β_2 are decay rates for the first and second moments. To solve the problem of large step size, bias correction is introduced in the first and second moments. So, Eqs (11) and (12) with mean and variance with bias correction can be written as Eq. (13).

$$\widehat{m_{t,i}} = \frac{m_{t,i}}{(1-\beta_1^t)}, \quad \widehat{v_{t,i}} = \frac{v_{t,i}}{(1-\beta_2^t)} \tag{13}$$

3.4.6. Adamax

It is a modified version of the Adam optimiser which deals with highly fluctuating gradient values. In this optimisation algorithm, the maximum of the past gradient is used to calculate the value of weight and bias parameters. The parameters are evaluated using Eq. (14).

$$\theta_t^i = \theta_{t-1}^i - \frac{\alpha}{\upsilon_t+\epsilon}.\widehat{m_t} \quad (14)$$

where $\upsilon_t = max(\beta_2.\upsilon_{t-1}, \left|\frac{\partial L}{\partial \theta_t}\right|)$

$$\widehat{m_t} = \frac{m_t}{1-\beta_1^t}\ ,\ \ m_t = \beta_1 m_{t-1} + (1-\beta_1)(\frac{\partial L}{\partial \theta_t})$$

3.4.7. Nesterov accelerated adaptive moment estimation (Nadam)

Another variation of Adam, where Nesterov momentum is used to have faster convergence and using the gradient at the intended next place serves as the foundation for an update. The parameter update rule is given using Eq. (15).

$$\theta_t = \theta_{t-1} - \frac{\alpha}{\sqrt{\widehat{\upsilon_t}+\epsilon}}\left(\beta_1\widehat{m}_t + \frac{(1-\beta_t)\left(\frac{\partial L}{\partial \theta_t}\right)}{1-\beta_1^t}\right) \quad (15)$$

3.4.8. DiffGrad

It is a type of gradient descent optimisation method in which the learning rate for each weight is calculated dynamically by considering the first and second moments of the gradient[24]. In this, parameters are updated using Eq. (16).

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\alpha_t \times \xi_{t,i} \times \widehat{m_{t,i}}}{\sqrt{\widehat{\upsilon_{t,i}}}+\in} \quad (16)$$

where $\xi_{t,i} = \frac{1}{\left(1+e^{-(|g_{t-1}-g_t|)}\right)}$

with the range [0.5,1]. $\xi_{t,i}$ is the diffGrad friction coefficient for the i^{th} parameter at t^{th} iteration to control the oscillations and slow convergence[24]. It has gained popularity in recent years due to its effective performance in the domain of deep learning.

3.4.9. Radam_norm

It is another type of adaptive SGD optimisation algorithm where gradient norm is performed using L2-norm to improve the limitations of the Adam optimisation algorithm[23]. It has been observed that normalisation will further increase the accuracy of existing optimisation algorithms and leads to fast convergence. The gradient of radam_norm is calculated using the following equations.

$$g_{norm} = L_2 Norm(g_t) \quad (17)$$

$$\theta_t = \begin{cases} \theta_{t-1} - \frac{\alpha_1 m_t}{\sqrt{\upsilon_t+\epsilon}} & if\ \rho_t >= 5 \\ \theta_{t-1} - \alpha_2 m_t & else \end{cases} \quad (18)$$

where $\alpha_1 = \frac{\alpha\sqrt{(1-\beta_2)\rho_u/\rho_d}}{(1-\beta_1^t)}$, $g_t = (\frac{\partial L}{\partial \theta_t})$

And g_{norm} is the normalised gradient value. In this method, the norms of each gradient are corrected in every iteration based on adaptive training history. It is an improved version of Rectified Adam where L2-normaliser is added. Further, this overcomes the limitation of the standard optimisation algorithm by adding normalisation.

4. Results and discussion

4.1 Dataset

To check the efficacy of the proposed system, a customised alphanumeric open-access dataset from Kaggle and a self-made dataset have been utilised.

Firstly, a customised dataset of 36 classes, approximately 1200 images per class has been sourced from the Kaggle [25], [26]. Initially, the dataset has 35 classes i.e., alphabets(A-Z) and numbers (1-9) but a class with 1200 images has been added. The resultant customised dataset has more than 45K images. This dataset has been referred to as Dataset I.

Secondly, a new dataset has been created with 40 signers. The images were captured using mobile phone cameras of varying resolutions to introduce nonlinearity and variation in the background. There were 40 signers, and 36 classes composed of 26 alphabets (A-Z) and 10 numbers (0-9) used in the creation of the dataset. Our dataset has nearly 50K images with 26 classes for the alphabet and 10 classes for number images. Other details of the dataset have been given in Table 1. This dataset has been referred to as Dataset II. The dataset has been divided into 80:20 ratios for training and validation datasets

4.2 Experimental setting

All the experiments were performed on the cloud using Google Colaboratory (Colab) Pro with 25GB RAM and 200 GB storage and Windows operating system. Pytorch, a high-level general-purpose programming language is used along with TensorFlow, Keras, Open CV, Pandas, NumPy, matplotlib etc. During the training phase of the proposed system, the following values of hyperparameters have been used as shown in Table 2.

Table 2- Initial values of the hyperparameters

Parameter	Value	Parameter	Value
α	0.001 to 0.0001	γ	0.90
β_1	0.9	Batch size	32
β_2	0.99	Pooling	Max

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 98, 98, 32)	320
max pooling2d (MaxPooling2D)	(None, 49, 49, 32)	0
dropout (Dropout)	(None, 49, 49, 32)	0
conv2d_1 (Conv2D)	(None, 49, 49, 64)	18496
max pooling2d 1 (MaxPooling 2D)	(None, 24, 24, 64)	0
dropout_1 (Dropout)	(None, 24, 24, 64)	0
conv2d_2 (Conv2D)	(None, 24, 24, 64)	36928
max pooling2d_2 (MaxPooling 2D)	(None, 12, 12, 64)	0
dropout_2 (Dropout)	(None, 12, 12, 64)	0
conv2d_3 (Conv2D)	(None, 12, 12, 128)	73856
max pooling2d_3 (MaxPooling 2D)	(None, 6, 6, 128)	0
conv2d 4 (Conv2D)	(None, 6, 6, 128)	147584
max pooling2d 4 (MaxPooling 2D)	(None, 3, 3, 128)	0
dropout 3 (Dropout)	(None, 3, 3, 128)	0
flatten (Flatten)	(None, 1152)	0
dense (Dense)	(None, 128)	147584
dense 1 (Dense)	(None, 36)	4644

Total params: 429,412
Trainable params: 429,412
Non-trainable params: 0

Figure. 3 Architecture of the proposed system

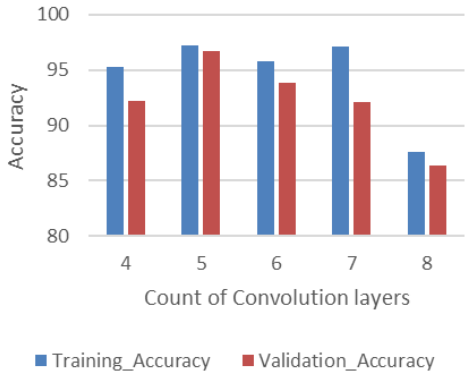
4.3 Quantitative analysis

After a comparative analysis of the count of layers, activation function, stride size, number of epochs and optimisers, CNN with 5 convolution layer, max pooling, swish activation function, 100 epochs, 1*1 stride, 32 batch size, softmax classifier and novel Radam_Norm has been selected as the optimal model. The architectural details of the proposed CNN have been shown in Fig. 3. It has 429412 parameters with 98x98 input image and L2 kernel regulariser for faster convergence, overfitting, underfitting issues and better performance.

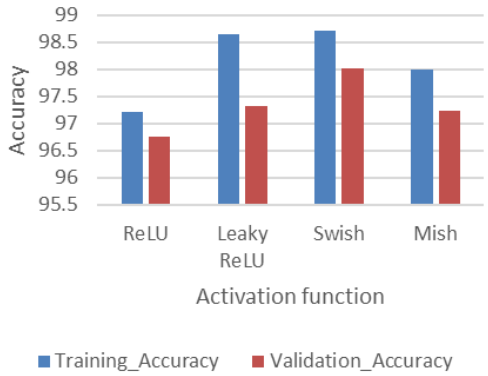
The impact of the count of convolution layers has been shown in Fig. 4 (a), 7-layer CNN with 5 convolution layers exhibited validation accuracy of 96.75% and loss of 1.2255%, which was better than all other models. The impact of the activation function has been displayed using Fig. 4 (b), and the validation accuracy rate of 7-layer CNN with ReLU, Leaky ReLU, Swish and Mish activation functions are 96.75%, 97.32%, 98.02% and 97.23% respectively. The swish activation function performed better than the remaining three activation functions. Further, the effect of the number of epochs (25,50, 75,100, 150) and stride size(1*1, 2*2) has been analysed on a 7-layer CNN with a swish activation function. The results of the same have been presented using Fig. 4 (c). The values of accuracies and loss concerning epochs and stride size have been depicted in Table 3. After finetuning of hyperparameters, the performance of the system has been enhanced using comparative analysis of the optimisers as shown in Fig. 4 (d). A novel optimiser

Table 3. Effect of Epochs, Stride Size on the proposed system

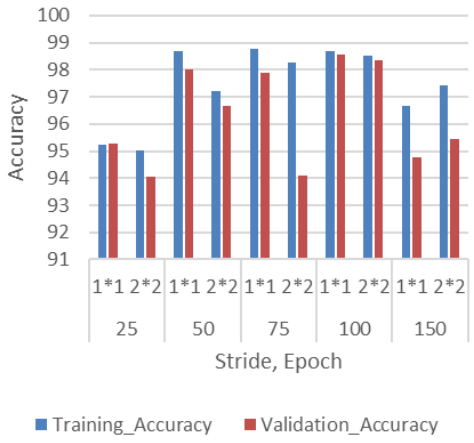
Efficacy	**Epochs**									
	25		**50**		**75**		**100**		**150**	
	1*1	**2*2**	**1*1**	**2*2**	**1*1**	**2*2**	**1*1**	**2*2**	**1*1**	**2*2**
TA	95.23	95.02	98.71	97.23	**98.77**	**98.27**	98.67	98.54	96.66	97.42
VA	95.265	94.07	98.026	96.66	**97.887**	**94.114**	98.543	98.347	94.759	95.455
TL	0.801	0.894	1.101	0.483	**0.117**	**0.136**	0.193	0.242	0.286	0.243
VL	0.9887	0.9144	1.1117	0.1878	**0.3788**	**0.3945**	0.1062	0.6095	0.2785	0.0941



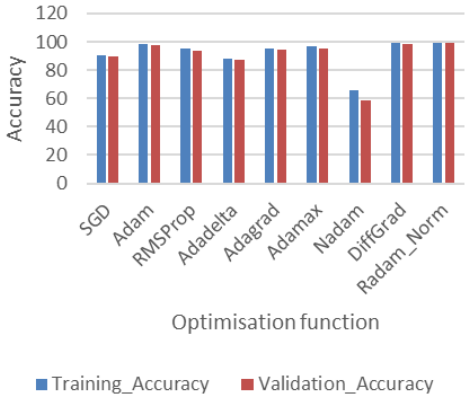
(a)



(b)



(c)



(d)

Figure. 4 Simulation analysis of proposed system: (a) Impact of layer count on accuracy, (b) Impact of activation function on accuracy, (c) Impact of epochs, stride size on accuracy, and (d) Impact of optimiser on accuracy

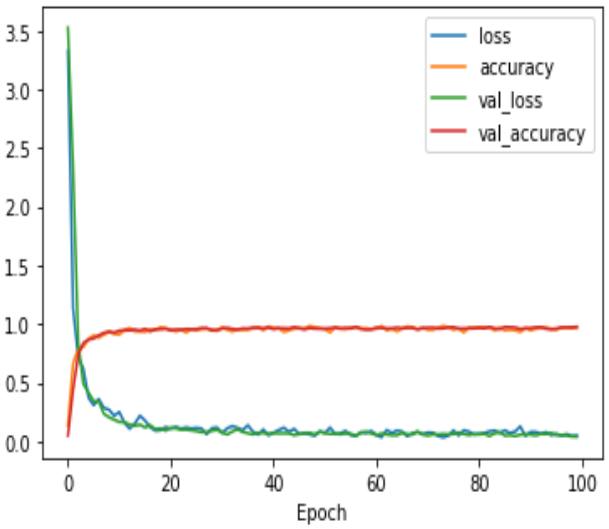


Figure. 5 Accuracy and loss curve of the proposed system

Radam_Norm has been used to achieve accuracy and loss of 98.96%, and 0.099% and its performance have been equated with other state of art deep learning

Table 4. Effect of optimiser on the proposed system

Efficacy	Optimiser								
	SGD	**Adam**	**RMSProp**	**Adadelta**	**Adagrad**	**Adamax**	**Nadam**	**DiffGrad**	**Radam_Norm**
TA	90.73	98.77	95.2	87.89	95.45	96.73	65.47	99.23	**99.446**
VA	89.99	97.88	93.75	87.56	94.21	94.85	58.34	98.76	**98.96**
TL	1.523	0.117	1.177	3.418	1.161	1.135	1.529	0.542	**0.0411**
VL	1.788	0.378	1.9987	1.4579	1.2678	1.1368	1.6354	1.6095	**0.099**

Table 5. Classification results

Dataset	Technique	Accuracy	Loss	Precision	Recall	F1-score
I	Proposed	97.889	0.0316	97.43	97.55	97.44
II		99.446	0.0411	98.87	98.01	98.43

Table 6. Performance comparison with other sign language translation models

Ref	Dataset	Technique	Modality	Accuracy
[27]	I	Multiple deep CNN models	Alphanumeric	92.85%
Ours		CNN with Radam_Norm optimiser		**97.44%**
Comparison II				
Ours	**II**	CNN with Radam_Norm optimiser	26Alphabet, 10 Number	99.46%
[28]	Own	SURF		96%
[29]	Own	CNN		89.30%

optimisers such as SGD, Adam, RMSProp, Adadelta, Adagrad, Adamax, Nadam, diffGrad with accuracy 89.99%, 97.88%, 93.75%, 87.56%, 94.21%, 94.25%, 58.34% and 98.76% respectively. The details are shown using Table 4.

4.4 Comparative analysis

The performance of the proposed automatic alphanumeric ISLTS has been demonstrated using Fig. 5, in terms of accuracy and loss of training and validation phase using Dataset II. The x-axis denotes the number of epochs and the y-axis denotes the respective value of loss and accuracy of the proposed ISLTS. The potency of the system is evaluated by employing two datasets in terms of evaluation metrics such as Precision, Recall, F1-score, Accuracy and loss. Dataset I is a customised open-access dataset of ISL gestures from Kaggle and Dataset II is a self-made dataset with images of alphanumeric data. Our proposed ISLTS demonstrated better results in the case of both datasets as shown in Table 5.

The efficacy of the proposed approach is also demonstrated by validating its performance and juxtaposing it with recent works in Table 6 and our proposed system has shown more promising results than other recent works of ISL. Due to the lack of availability of standard datasets of ISL, we have made comparisons on two criteria. Firstly, our proposed method is compared based on Dataset I, i.e., open access Kaggle dataset, in which [27] attained an accuracy of 92.85% for translating ISL gestures using pre-trained CNN models while our proposed ISLTS achieved an accuracy of 97.44%. Secondly, the comparison has been made based on the modality, [28] and [27] created in their respective alphanumeric dataset of ISL having 36 classes. [28] proposed an ISLTS using SURF and attained an accuracy of 96% and [29] proposed CNN-based ISLTS on alphanumeric dataset of their own and achieved 89.3% accuracy.

5. Conclusion and future scope

The main purpose of this paper is to build an efficient machine learning-based system for the translation of ISL gestures into text. After creating the dataset and pre-processing, different models of deep learning framework 2DCNN were analysed based on hyperparameters such as the layers count, epochs, stride size, and activation function. Further, a comparative analysis of various optimisers has been performed to enhance the performance of the proposed system. It has been concluded that the proposed system promises better results when compared with recent works in the domain of ISL translation. This study can be used by researchers to

choose optimal hyperparameters and optimisers for static alphanumeric translation of ISL gestures into text.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization, Seema; methodology, Seema; software, Seema; validation, Seema; formal analysis, Seema; investigation, Seema; resources, Seema; data curation, Seema; writing—original draft preparation, Seema; writing—review and editing, Seema; supervision, Priti Singla.

References

[1] P. Sharma and R. S. Anand, "A comprehensive evaluation of deep models and optimizers for Indian sign language recognition", *Graphics and Visual Computing*, Vol. 5, p. 200032, 2021, doi: 10.1016/j.gvc.2021.200032.

[2] Sitender, and S. Bawa, "Sanskrit to universal networking language EnConverter system based on deep learning and context-free grammar", *Multimedia Systems*, pp. 1–17, 2020.

[3] H. V. Chand, and J. Karthikeyan, "CNN Based Driver Drowsiness Detection System Using Emotion Analysis", *Intelligent Automation and Soft Computing*, Vol. 31, No. 2, pp. 717–728, 2022.

[4] R. G. Rajan, P. S. Rajendran, S. Smys, R. Bestak, R. Palanisamy, and I. Kotuliak "Comparative Study of Optimization Algorithm in Deep CNN-Based Model for Sign Language Recognition", in *Computer Networks and Inventive Communication Technologies*, Eds., in Lecture Notes on Data Engineering and Communications Technologies, Singapore: Springer Singapore, Vol. 75, pp. 463–471, 2022.

[5] Y. Wang, Y. Li, Y. Song, and X. Rong, "The Influence of the Activation Function in a Convolution Neural Network Model of Facial Expression Recognition", *Applied Sciences*, Vol. 10, No. 5, p. 1897, 2020.

[6] R. Nirthika, S. Manivannan, A. Ramanan, and R. Wang, "Pooling in convolutional neural networks for medical image analysis: a survey and an empirical study", *Neural Computing and Applications*, Vol. 34, No. 7, pp. 5321–5347, 2022.

[7] Y. Wang, H. Zhang, and G. Zhang, "cPSO-CNN: An efficient PSO-based algorithm for fine-tuning hyper-parameters of convolutional neural networks", *Swarm and Evolutionary Computation*, Vol. 49, pp. 114–123, 2019.

[8] T. W. Chong and B. G. Lee, "American Sign Language Recognition Using Leap Motion Controller with Machine Learning Approach", *Sensors*, Vol. 18, No. 10, p. 3554, 2018.

[9] I. A. Adeyanju, O. O. Bello, and M. A. Adegboye, "Machine learning methods for sign language recognition: A critical review and analysis", *Intelligent Systems with Applications*, Vol. 12, p. 200056, 2021.

[10] A. A. Alani, and G. Cosma, "ArSL-CNN a convolutional neural network for Arabic sign language gesture recognition", *Indonesian Journal of Electrical Engineering and Computer Science*, Vol. 22, No. 2, p. 1096, 2021.

[11] Seema, and P. Singla, A. Khanna, Z. Polkowski, and O. Castillo, "A Comprehensive Review of CNN-Based Sign Language Translation System", In: *Proc. of Data Analytics and Management*, Eds. in Lecture Notes in Networks and Systems, Singapore: Springer Nature Singapore, Vol. 572, pp. 347–362, 2023.

[12] S. Das, M. S. Imtiaz, N. H. Neom, N. Siddique, and H. Wang, "A hybrid approach for Bangla sign language recognition using deep transfer learning model with random forest classifier", *Expert Systems with Applications*, Vol. 213, pp. 118914, 2023.

[13] Z. Gao, C.C. Lee, L. Zheng, R. Zhang, and X. Xu, "A Multitask Sign Language Recognition System Using Commodity Wi-Fi", *Mobile Information Systems*, Vol. 2023, pp. 1–11, 2023.

[14] B. Subramanian, B. Olimov, S. M. Naik, S. Kim, K. H. Park, and J. Kim, "An integrated mediapipe-optimized GRU model for Indian sign language recognition", *Scientific Reports*, Vol. 12, No. 1, p. 11964, 2022.

[15] E. Almekhlafi, M. A. L. Makhlafi, E. Zhang, J. Wang, and J. Peng, "A classification benchmark for Arabic alphabet phonemes with diacritics in deep neural networks", *Computer Speech & Language*, Vol. 71, p. 101274, 2022.

[16] M. Zakariah, Y. A. Alotaibi, D. Koundal, Y. Guo, and M. M. Elahi, "Sign Language Recognition for Arabic Alphabets Using Transfer Learning Technique", *Computational Intelligence and Neuroscience*, Vol. 2022, pp. 1–15, 2022.

[17] A. Mannan, A. Abbasi, A. R. Javed, A. Ahsan, T. R. Gadekallu, and Q. Xin, "Hypertuned Deep Convolutional Neural Network for Sign Language Recognition", *Computational Intelligence and Neuroscience*, Vol. 2022, pp. 1–10, 2022.

[18] T. Yirtici and K. Yurtkan, “Regional-CNN-based enhanced Turkish sign language recognition”, *Signal Image and Video Processing*, Vol. 16, No. 5, pp. 1305–1311, 2022.

[19] S. K. Ko, C. J. Kim, H. Jung, and C. Cho, “Neural Sign Language Translation Based on Human Keypoint Estimation”, *Applied Sciences*, Vol. 9, No. 13, p. 2683, 2019.

[20] U. Nandi, A. Ghorai, M. M. Singh, C. Changdar, S. Bhakta, and R. K. Pal, “Indian sign language alphabet recognition system using CNN with diffGrad optimizer and stochastic pooling”, *Multimedia Tools and Applications*, pp. 1–22, 2022.

[21] S. H. S. Basha, S. R. Dubey, V. Pulabaigari, and S. Mukherjee, "Impact of Fully Connected Layers on Performance of Convolutional Neural Networks for Image Classification", *Neurocomputing*, Vol. 378, pp. 112-119, 2020.

[22] P. Unkule, C. Shinde, P. Saurkar, S. Agarkar, and U. Verma, “CNN based Approach for Sign Recognition in the Indian Sign language”, In: *Proc. of International Conference on Augmented Intelligence and Sustainable Systems, ICAISS 2022*, pp. 92–97, 2022.

[23] S. R. Dubey, S. K. Singh, and B. B. Chaudhuri, "AdaNorm: Adaptive Gradient Norm Correction based Optimizer for CNNs", In: *Proc. of IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2023*, pp. 5284-5293, 2023.

[24] S. R. Dubey, S. Chakraborty, S. K. Roy, S. Mukherjee, S. K. Singh, and B. B. Chaudhuri, “diffGrad: An Optimization Method for Convolutional Neural Networks”, *IEEE Trans. Neural Netw. Learning Syst.*, Vol. 31, No. 11, pp. 4500–4511, 2020.

[25] P. Arikeri, “ISL dataset Kaggle”, Accessed: Jul. Vol. 12, 2022. [Online]. Available:https://www.kaggle.com/datasets/prathumarikeri/indian-sign-language-isl

[26] P. Arikeri, “American Sign Language (ASL) Dataset”, *Kaggle*. https://www.kaggle.com/datasets/prathumarikeri/american-sign-language-09az (accessed Sep. 18, 2022).

[27] T. Kujani and V. Dhilipkumar, “Multiple Deep CNN models for Indian Sign Language translation for Person with Verbal Impairment”, *International Journal of Intelligent Systems and Applications in Engineering*, Vol. 10, No. 3, pp. 382–389, 2022.

[28] K. M. Tripathi, P. Kamat, S. Patil, R. Jayaswal, S. Ahirrao, and K. Kotecha, “Gesture-to-Text Translation Using SURF for Indian Sign Language”, *ASI*, Vol. 6, No. 2, p. 35, 2023.

[29] P. Paul and G. N. Rathna, “Real-time Indian sign language recognition”, *Summer Research Fellowship Programme of India's Science Academies*, [Online] Available: reports.ias.ac.in.