Multilingual signs recognition using recurrent neural network

Thouseef Ulla Khan[1], Dileep Marichi Ramachandra[2]
[1]Department of Master of Computer Applications, Vidya Vikas Institute of Engineering and Technology, Visvesvaraya Technological University, Belagavi, India
[2]Department of Master of Computer Applications, Nitte Meenakshi Institute of Technology, Visvesvaraya Technological University, Belagavi, India

Article Info

Article history:

Received Jan 5, 2025
Revised Apr 23, 2026
Accepted May 11, 2026

Keywords:

Hand gestures
Hand landmark detection
Image processing
Long short-term memory recognition
Multilingual gesture recognition
Sign alphabets

ABSTRACT

Recognition of sign language is a crucial step towards providing individuals with hearing and speech impairment meaningful communication, but the fact that there are a number of distinct sign languages and gestures remain complex makes it a challenge to the current automated systems. The present paper describes a real-time multilingual sign language recognition system that is based on a recurrent neural network with long short-term memory (RNN-LSTM) with hand landmark MediaPipe-based hand landmark detection to successfully receive spatial and temporal gesture features. The proposed system was trained and tested over a self-collected set of alphabet gestures of the Chinese, American, and Indian sign language, including one-hand and two-hand gestures, and was run with Keras with extensive performance evaluation metrics. The strength and generalization abilities of the suggested approach as part of different gesture patterns and variations in users are confirmed by experimental outcomes that indicate high recognition rates of 99.58%, 99.62%, and 99.63% of the Chinese, American, and Indian sign languages, respectively. These results demonstrate the promise of the given framework as a dedicated assistive system of communication and give it a solid base to continue its development to the point of the system of the continuous sign language recognition (CSLR) and multimodal translators.

Corresponding Author:

Thouseef Ulla Khan
Department of Master of Computer Applications, Vidya Vikas Institute of Engineering and Technology
Visvesvaraya Technological University
Jnana Sangama, Machhe, Belagavi, Karanataka -590018, India
Email: thouseef588@gmail.com

1. INTRODUCTION

Artificial intelligence (AI) is an emerging technology in the present world. It uses the machine learning and deep learning to enhance technologies. AI could also be used to develop advanced systems so that hearing and speech impaired people can communicate with other members of society. There are various sign languages that are exercise in different parts of the world and some of them include American, Indian, Chinese, Korean sign language, and many more. These sign languages cannot be identified by an average person and this makes them be neglected in the society. The system that acknowledges sign language enables the normal people to learn the language of the hearing and speech impaired people by translating the hand gestures to text message that will enhance the communication between them. Sign language will be composed of letters, numbers, words, and sentences. Signs are captured with different hand gestures through camera, pre-processing of the captured images are performed, signs are recognized with recurrent neural network (RNN) algorithm and display the suitable characters on output screen as shown in Figure 1.

In our system, we have incorporated one-handed sign languages Chinese and American, and two-handed Indian sign language as seen in Figure 2. It shows the one-handed and two-handed sign languages included in this study, emphasising the variation in gesture patterns among different sign language systems. Figure 2(a) shows the Chinese sign language alphabets performed using single hand gestures which are compact and efficient hand movements. The American sign language alphabets are also represented using one-hand gestures as shown in Figure 2(b). The Indian sign language alphabets mainly involve two-handed gestures, as shown in Figure 2(c), highlighting the structural complexity handled by the proposed recognition model.

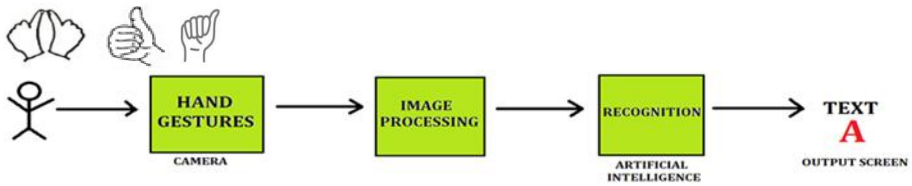
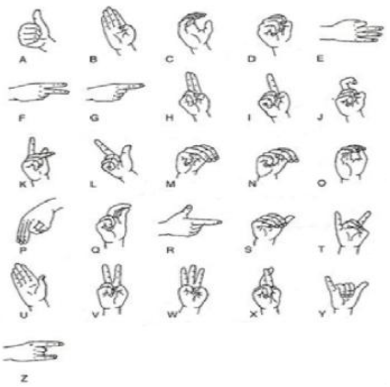
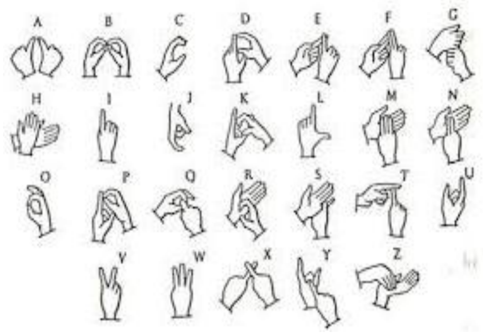


Figure 1. System architecture



(a)

(b)



(c)

Figure 2. One-handed and two-handed sign languages considered in this study of (a) Chinese sign language alphabets performed using single-hand gestures, (b) American sign language alphabets also represented with one-hand gestures, and (c) Indian sign language alphabets, which primarily involve two-handed gestures

We go over relevant research on sign language recognition as a literature survey. Kasapbaşi *et al*. [1], proposed convolutional neural network (CNN) model consists of three convolutional layers, as this configuration yielded the highest accuracy during their empirical tests. The efficiency of 99.38% with loss 0.025 has obtained on American sign language. Ghotkar *et al*. [2], model recognizes static Indian sign language digits, characters using Fourier descriptor to extract the features and nearest neighbourhood classifier to achieve efficiency of 94.15% with Euclidian distance. Dabwan *et al.* [3], approaches DenseNet-121 model with American sign language aplhabet dataset, comprising 24 classes corresponding to sign letters, achieving an accuracy rate of 97%. Tan *et al.* [4], developed hand gesture recognition with vision transformer (HGR-ViT) model on National University of Singapore (NUS) hand gestures datasets, American sign language numbers and American sign language characters, and obtained an efficiency of 99.85%, 99.36%, and 99.98% respectively. Laines *et al*. [5], model on Mexican sign language using sign language-tree structure skeleton image (SL-TSSI) DenseNet, where images are processed by DenseNet-121, which is converted from skeleton sequence using TSSI with efficiency of 98.0%. Kumari and Anand [6], hybrid architecture of CNN-long short-term memory (LSTM) has achieved an accuracy of 84.65% for American sign language tested on 100 classes of word-level American sign language (WLASL) dataset. Hybrid efficient convolution model [7], with EfficientNet-B3 and a few modified layers, obtains accuracy of 93.17% to recognize hundred isolated dynamic Bangla sign language on the 600 video clips dataset BdSL_OPA_23_ GESTURES [7].

Gangrade and Bharti [8], used Microsoft Kinect sensor to identify and separate hand portion from a depth picture. CNN used to construct features from Indian sign language signals, and up to 99.3% of motions are correctly recognized. Tripathi *et al*. [9], model converts gestures to text using a bag of visual words (BoVW) concept. A collection of 42,000 photos including alphabets and digits, split into 35 classes with 1,200 images each. Skin masking is a step in the pre-processing stage that involves converting the RGB to HSV, using canny edge detection to find keen edges. The mini-batch K-means clustering algorithm is used to aggregate the speeded-up robust features (SURF) elements and translate them into a visual language. Odartey *et al*. [10], Ghanaian sign language was classified and recognized using the suggested deep convolutional neural network (DCNN), which achieved 96.0% accuracy. Tolentino *et al*. [11], CNN model, where images were trained using Keras. American sign language letter recognition accounted for 90.04% of the system's average testing accuracy of 93.67%. Lomas *et al.* [12], suggested CNN method to utilize a compound coefficient to measure all the depth, breadth, and resolution parameters equally when utilizing the methodology on EfficientNet, and identified the method by selecting B0 model as architecture for testing with a dataset of Kaggle. Rinalduzzi *et al*. [13], model using a magnetic positioning system, which has better generalization features and achieves a classification efficiency of roughly 97% on 24 characters. Alaftekin *et al*. [14], Turkish sign language digits dataset is making use to training the you only look once version 4-cross stage partial (YOLOv4-CSP) models, and their outcomes are compared in terms of hand signal identification, which yields the results with precision 98.95%, recall 98.15%, F1-score 98.55, and mean average precision (mAP) 99.49% in 9.8 ms.

Using a path aggregation feature pyramid network with the ODMamba backbone network, YOLO-Mamba model produced recognition accuracy of 93.3% on unique dataset [15]. Sasidharan *et al*. [16], used variety of spatiotemporal features gathered from video sequences of sign language movements, which enable to identify signals by capturing both the temporal dynamics of the movements and the spatial arrangement. The accuracy was 90.87% for Chinese sign language and 89.46% for Arabic sign language character signs. Gated recurrent unit (GRU) model used for extraction of temporal and spatial characteristics, attained accuracy of 92.13% and 93.98% on the datasets DHG14/28 and SHREC'17, respectively [17]. Sharma and Singh [18], CNN model on Indian sign language gesture feature extraction and categorization. Both a publicly accessible American sign language dataset and a self-collected Indian sign language dataset are used to assess this method's performance. Three datasets were used in total, and the accuracy obtained was 92.43%, 88.01%, and 99.52%. In order to identify gestures, Tian *et al.* [19] developed the DAMR_3DNet model, which yielded accuracy of 55.3% and 92.93% on the HMDB51 and UCF101 datasets, respectively. Sreemathy *et al*. [20], illustrates how to identify the alphabets in Indian sign language by image processing techniques and machine learning. Additionally, AlexNet, GoogleNet, VGG-16, and VGG-19 were used in machine learning technique, yielding accuracies of 99.11%, 95.84%, 98.42%, and 99.11%, respectively. Chong and Lee [21], created a prototype for recognizing sign language using the leap motion controller with goal of recognizing American sign language. The results of the experiment showed that the 26 letters could be recognized using deep neural network (DNN) and support vector machine (SVM) at 80.30% and 93.81%, respectively.

Perumal *et al*. [22], suggest LSTM models using rectified linear unit (ReLU) activation functions, which have been shown to be 96.55% with appropriate training, with L2 regularization and CNN. Chung *et al*. [23], created and assessed a framework for Chinese sign language identification that makes use of spatiotemporal characteristics. The hand landmarks features are fed into a bidirectional long short-term memory (Bi-LSTM) system for recognizing sign. A continuous Chinese sign language dataset including 1,200 sample films encompassing 100 signals, and 8 participants, the recognition rate is 98.75%. Li *et al*. [24], technique for vocabulary segmentation based on the acceleration signal and the volume state of the surface electromyography (sEMG) signal. The average recognition rate of sign language vocabulary is 90.41% when the multi-sensor decision fusion approach of hidden Markov model is employed. Jiang *et al*. [25], give an extensive review of Chinese sign language recognition explaining how it developed through the years beginning with traditional feature-based techniques to modern AI and deep learning-based technology, which contextualizes and underlines the methodology used in this work. Liu *et al*. [26], used CNN to recognize hand positions and movement trajectories by combining stretchy strain sensors with body-mounted inertial measurement devices, 48 Chinese sign language terms used for training and achieved an efficiency of 95.85%.

The research studies have proven that sign language recognition has advanced considerably with the incorporation of deep learning, keypoint tracking, and smart system design. Alsharif *et al*. [27], suggested a real-time system of American sign language interpretation with the use of deep learning and keypoint-based tracking of the hands, which is highly accurate and real-time. Maashi *et al*. [28] proposed an IoT-based smart assistive communication system built on hybrid deep learning sign language recognition models, with a focus on scalability and real-world application to hearing-impaired users. Al-Latief *et al.* [29] presented a descriptive comparative review of deep learning models in the recognition of sign language and systematically analyzed them in terms of architecture, data sets, and challenges, which will contribute to important findings on the existing trends and research gaps. Moreover, Baihan *et al*. [30] have suggested an improved deep learning model with improved hybrid optimization algorithm, which shows better recognition result and stability, further confirming the efficiency of advanced optimization algorithm in sign language recognition algorithm.

In parallel, the research community is increasingly adopting transformer-based and multi-scale temporal modeling approaches for continuous sign language recognition (CSLR), effectively addressing variable sign durations and long-range temporal dependencies [31], while integrated CSLR-to-speech pipelines are being explored to move beyond gesture classification toward end-to-end assistive communication systems [32]. Recent works are also focused on the robustness, scalability, and practical deployment of sign language recognition (SLR) systems. For CSLR, attentive spatio-temporal networks have been proposed to effectively capture temporal dependencies over long gesture sequences [33]. In addition, pose-estimation-driven deep learning frameworks have been investigated to enhance recognition accuracy under variations in signer posture and viewpoint [34]. These advances are summarised in extensive review studies, which outline the present challenges and research avenues for SLR, highlighting the necessity of universal, effective, and multimodal SLR systems [35].

The organization of this document is as follows. Section 1 outlines relevant studies in the field of sign language recognition. Section 2 includes proposed methodology that explains dataset, preprocessing, feature extraction, model training, and sign recognition. Section 3 include performance metrics, accuracy, loss, and comparative analysis. Finally, section 4 of this document concludes with a summary.

The following are three strengthened contribution points that clearly emphasize the novelty and significance of our proposed model:

i) Unified multilingual recognition with signer-independent evaluation: the proposed RNN-LSTM framework is one of the few systems which recognise Chinese, American, and Indian sign languages including one-handed and two-handed gestures in a single architecture. We evaluate the model under a strict signer-independent protocol, showing strong generalisation to unseen users, and addressing a major limitation in existing sign language recognition studies, in comparison to many previous works.
ii) Effective temporal modeling using MediaPipe landmarks and LSTM: the proposed model combines the hand landmark extraction from MediaPipe with the deep temporal modelling based on stacked LSTM layers. This approach can capture spatial hand configurations and temporal gestures dynamics more effectively than the frame-based CNN models. Such a design results in consistently better performance on all three sign languages. Novelty lies in the combination of lightweight landmark features and sequence learning for real-time multilingual recognition.
iii) High accuracy with statistical robustness and overfitting control: the model attains state-of-the-art accuracy above 99.5% on Chinese, American, and Indian sign languages, and reports mean accuracy with standard deviations over multiple experimental runs. The contribution is further enhanced by the inclusion of early stopping, controlled training procedures and statistical analysis, which show not only high performance, but also stability, reproducibility and resistance to overfitting, often lacking in comparable studies.

2. METHOD

The real-time multilingual sign language recognition system was developed through a comprehensive approach. It begins with data collection and continues with pre-processing using hand tracking powered by Mediapipe. Finally, RNN are employed to recognize hand gestures, as illustrated in Figure 3.

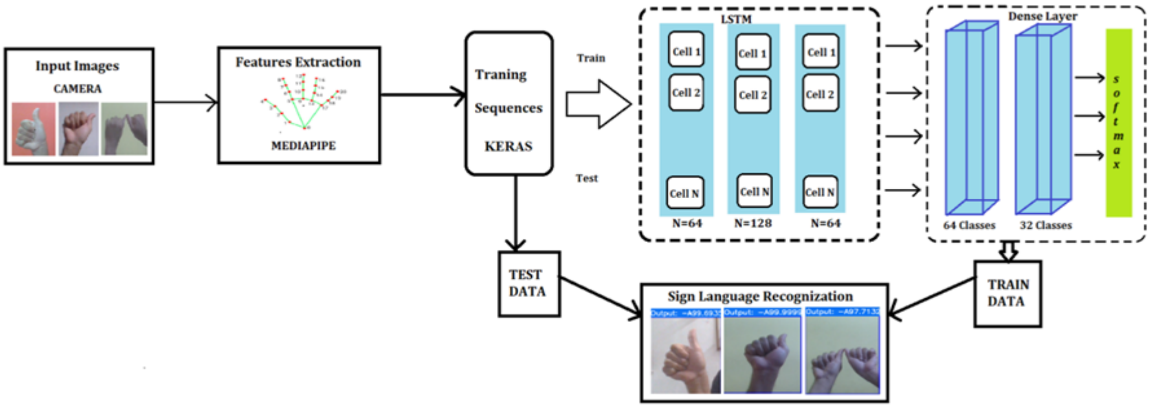


Figure 3. Proposed model of multilingual sign language recognition

2.1. Dataset

As shown in Table 1, the dataset used to evaluate the proposed multilingual sign language recognition system was self-collected and consists of isolated alphabet gestures from Chinese sign language, American sign language, and Indian sign language. A total of 78 alphabet classes (A–Z for each language) were recorded using a standard web camera under varying background conditions. Data were collected from multiple participants spanning different age groups, ensuring inter-user variability in hand shape, posture, and execution style. For each alphabet class, 30 gesture sequences were recorded per participant, where each sequence contains 30 frames and resulting in 900 samples per alphabet. Overall, the dataset comprises 70,200 samples per language, yielding a total of 210,600 gesture samples across all three sign languages.

Table 1. Dataset summary table

Dataset attribute	Chinese sign language	American sign language	Indian sign language
Number of participants	10	10	10
Alphabet classes	26 (A-Z)	26 (A-Z)	26 (A-Z)
Samples per class per participant	30 sequences×30 frames	30 sequences×30 frames	30 sequences×30 frames
Samples per class (total)	900	900	900
Total samples per language	70,200	70,200	70,200
Data representation	Mediapipe hand landmarks (NumPy)	Mediapipe hand landmarks (NumPy)	Mediapipe hand landmarks (NumPy)
Train/validation/test split	70%/15%/15%	70%/15%/15%	70%/15%/15%
Signer overlap between train and test	No (signer-independent split)	No (signer-independent split)	No (signer-independent split)

To mitigate identity bias, the dataset was split such that signers in the training set do not appear in the test set, allowing for signer-independent and fair evaluation of generalisation. The extracted sequences of hand landmarks were divided into training, testing, and validation subsets with a fixed split ratio. This strict design of the data set helps prevent user-specific overfitting and allows a fair comparison of the proposed RNN-LSTM model across multiple sign languages.

It uses OpenCV with frames to capture the hand gestures through camera and stored in the shapes array in PNG format with cv2.rectangle(frame,(0,40),(300,400),(255,255,255),2) having any color background using "cv2.rectangle(image, start_point, end_point, color, thickness)". Minimum 90 images of size 300×360 pixels with 24-bit depth with different postures from 10 different people was captured for each sign alphabet from A–Z in three different sign languages. The samples of captured hand gestures images for dataset creation are shown in Figure 4.

Figure 4(a) shows representative samples of Chinese sign language alphabets captured from multiple users, demonstrating common variations across individuals. Figure 4(b) illustrates sample American sign language alphabet gestures, including differences in finger configuration and movement. Figure 4(c) presents sample Indian sign language alphabets, highlighting variations in hand posture and orientation that affect visual appearance.

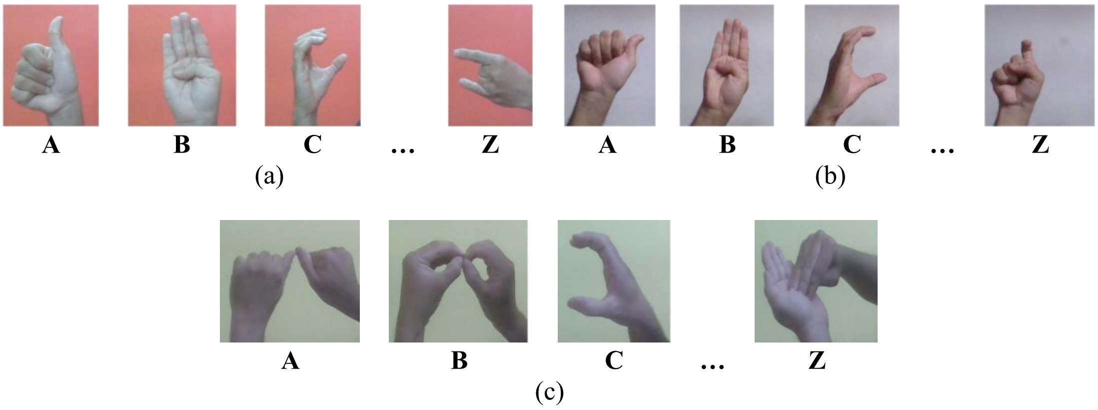


Figure 4. Captured images of hand gestures used for dataset creation of (a) Chinese sign language alphabets gestures, (b) American sign language alphabets gestures, and (c) Indian sign language alphabets gestures

2.2. Pre-processing using MediaPipe for hand landmarks detection

MediaPipe is used for hands keypoints tracking as shown in Figure 5, with model complexity of 0 and a minimum detection and tracking confidence of 0.5. Action will be taken on the range of sequence with frame numbering of sequence length of 90. OpenCV interpretes an image as BGR format by default; but while processing an image, it will be converted to RGB format as cvtColor(image, cv2.COLOR_BGR2RGB). The different colour formats of sign language image is used in recognition of sign language as seen in Figure 6. In this model, we have used RGB and BGR formats for the image processing.

Figure 5. MediaPipe's portrayal of hand landmarks

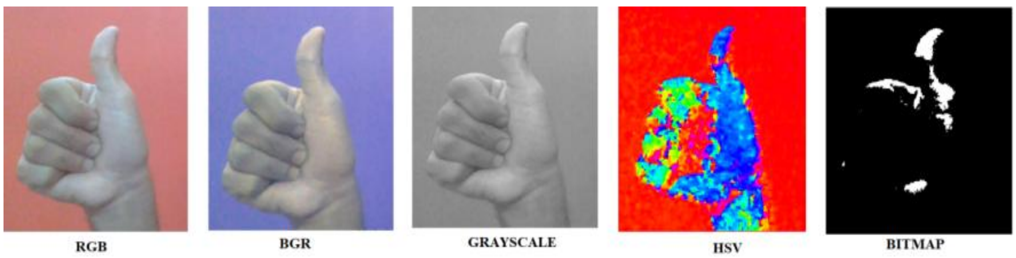


Figure 6. Different formats for image processing

Frames are created with OpenCV as seen in Figure 7, by reading PNG formatted images, and MediaPipe detection will be done on frames with hands gestures. Keypoints are extracted and stored in the numpy array. Figure 7 shows the extraction of keypoints through hand landmark detection from the captured images using MediaPipe for the sign alphabets in Chinese sign language, American sign language, and Indian sign language respectively. Figure 7(a) shows the detected hand landmarks for Chinese sign language gestures, Figure 7(b) presents landmark extraction for American sign language gestures, and Figure 7(c) illustrates the detected landmarks for Indian sign language gestures, clearly indicating key joint positions used for feature extraction.

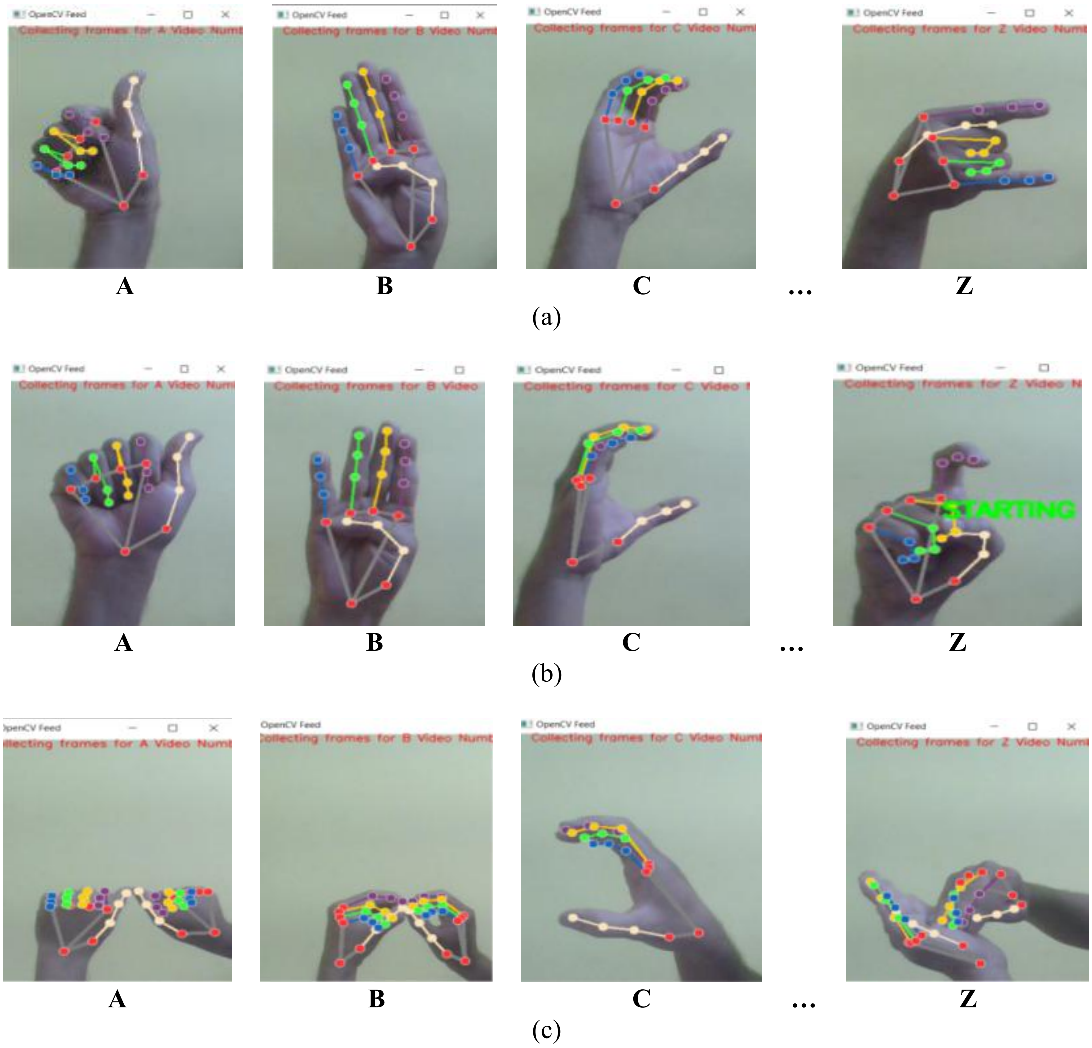


Figure 7. Hand landmark detection process using MediaPipe on different sign languages of (a) Chinese sign language gestures, (b) American sign language gestures, and (c) Indian sign language gestures

2.3. Feature extraction

Keras is used for model and layers. Initially we load numpy array in the data path and append that with window, sequence, and labels, whose architecture seen in Figure 8. X-coordinate is represented as numpy array and Y-coordinate is represented as categorizing the labels. Then we carry out splitting of x_Train, y_Train, and x_Test, y_Test with respect to x and y coordinates. Now Keras model is used to implement three layers of LSTM algorithm with ReLU activation and 64, 128, and 64 respectively.

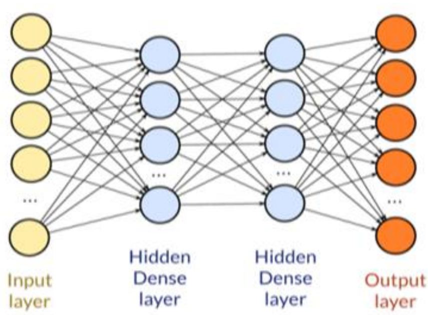


Figure 8. Architecture of Keras

Next Keras model is used to implement two layers of dense algorithm with ReLU and one layer with softmax activation. Softmax converts vector values to probabilities. Its derivative is monotonic, as is the ReLU function itself, Figure 9 illustrates this. The function returns 0 if it gets any negative input, but returns a positive number z if it receives any positive value. Consequently, the output's range is 0 to infinite. There is a mathematical formula as in (1) that represents ReLU. Next Keras model fit with x_Train, y_Train. Finally, model will be saved as JSON file with the name model.h5, which store records of model, layers, weights and trackables.

$$R(z) = MAX(0, \underline{z}) \tag{1}$$

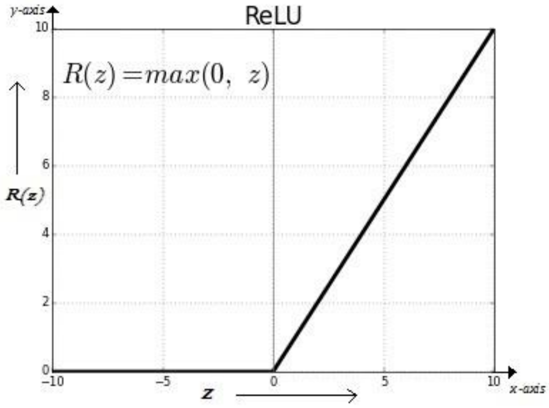


Figure 9. ReLU functionality

2.4. Model training

Training of model contains Keras, TensorBoard, labelling, train-test splitting, and LSTM and Dense algorithms with 200 epochs. By using an optimization technique like gradient descent in combination with backpropagation over time to determine the gradients needed for the optimization process, an RNN employing LSTM units, as illustrated in Figure 10, can be trained under supervision using a set of training sequences. This will alter each weight of the LSTM network proportionately to the derivative of the error (at the LSTM network's output layer) with respect to the corresponding weight.

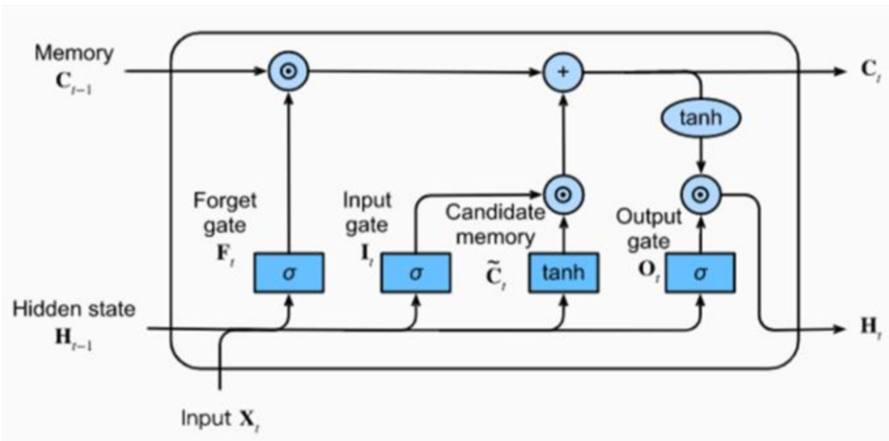


Figure 10. Architecture of LSTM

We employ LSTM in the suggested system, which has three gates: input, output, and forget gates. LSTM gates are sigmoid activation functions that produce a value between 0 and 1, which is often either 0 or 1. Figure 11 illustrates how the sigmoid function represents LSTM. The LSTM gates are illustrated by (2).

$$
\begin{aligned}
f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
\tilde{c}_t &= \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \\
h_t &= o_t \odot \sigma_h(c_t)
\end{aligned} \tag{2}
$$

Where i_t stands for input gate, f_t stands for forget gate, o_t stands for output gate, σ describes sigmoid function, w_x weight for the respective gate (x) neurons, h_{t-1} output of the previous LSTM block (at timestamp $t-1$), x_t input at current timestamp, b_x biases for the respective gates (x), c_t cell state (memory) at timestamp (t), and $\tilde{c}_t$ candidate for cell state at timestamp (t).

Error gradients disappear exponentially with the length of time between significant events, which are a drawback of gradient descent for conventional RNNs. But with LSTM units, the error stays in the cell of the LSTM unit even after error values are back-propagated from the output layer. Until the LSTM unit learns to shut off the value, this “error carousel” keeps feeding errors back to each of its gates.

The softmax function is used in the final layer of a neural network model for classification tasks, which takes the exponential of each output and normalizes them by dividing with total of all the exponentials. This process turns raw output scores into probabilities. This procedure guarantees that the output values may be interpreted as probabilities as they fall between 0 and 1 and add up to 1. The network's predictions may be interpreted probabilistically using the softmax function (3).

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{3}$$

Where σ is softmax, $\vec{z}$ is input vector, e^{z_i} is standard exponential function for input vector, K is number of classes in the multi-class classifier, e^{z_j} is standard exponential function for output vector, and e^{z_j} is standard exponential function for output vector.

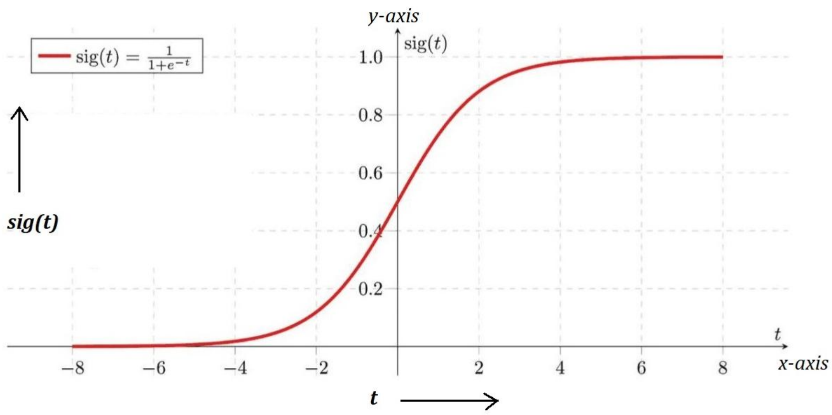


Figure 11. LSTM sigmoid function

2.5. Sign recognition

MediaPipe is used to predict hand landmarks from a numpy array supplied in a JSON file with a threshold of 0.8. The predict() function, which yields predictions in the format supplied by the network's output layer, will be used by the model to make predictions based on these hand keypoints. The argmax() NumPy function must be used to translate the multiclass classification results, which are an array of probabilities, into a single class output prediction. Finally, accuracy will be calcualted and displayed.

The accuracy representation of the proposed sign language recognition system for multiple sign languages is shown in Figure 12. Particularly, the recognition accuracy obtained for Chinese sign language is shown in Figure 12(a), which shows the efficiency of the model in recognising one-handed gesture patterns. The results of accuracy for American sign language are given in Figure 12(b). The accuracy performance of Indian sign language is given in Figure 12(c). The results show the robustness of the proposed approach for both one-hand and two-HGR scenarios.

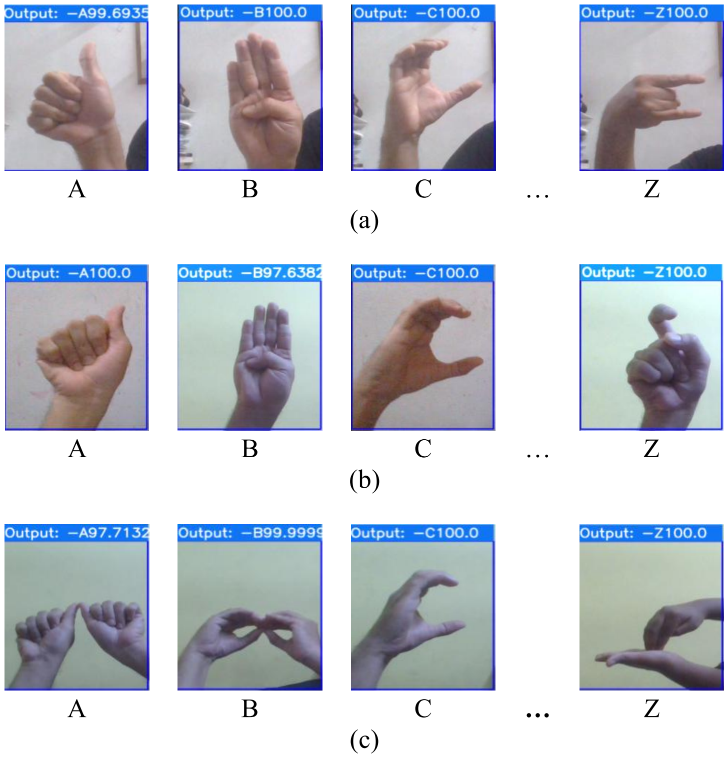


Figure 12. Recognition accuracy achieved by the proposed system across different sign languages of (a) Chinese sign language alphabets, (b) American sign language alphabets, and (c) Indian sign language alphabets

3. RESULTS AND DISCUSSION

Table 2 shows the experiments were repeated five times with different random initialisations while keeping the same signer-independent train/validation/test split to evaluate the statistical robustness and stability of the proposed multilingual sign language recognition system. The mean recognition accuracy and its standard deviation for each sign language are shown in Table 2. The small standard deviation values indicate that the proposed RNN-LSTM model has consistent and reliable performance across multiple runs, which implies effective overfitting control and robust generalisation capability among different signers.

Table 2. Accuracy stability across multiple experimental runs

Sign language	Mean accuracy (%)	Standard deviation (%)
Chinese sign language	99.58	±0.21
American sign language	99.62	±0.18
Indian sign language	99.63	±0.17

Note: results are averaged over five independent runs using the same signer-independent dataset split with different random initializations.

The standard F1-score, which calculates the number of times the model produced accurate predictions, has been used to assess the model accuracy performance. False positive (FP), false negative (FN), true positive (TP), and true negative (TN) are the measures that are employed. It uses (4) to (6) to determine the model's precision, recall, and F1-score.

$$Precission = \frac{TP}{TP+FP} \tag{4}$$

$$Recall = \frac{TP}{TP+FN} \tag{5}$$

$$F1-score = \frac{TP}{TP+\frac{1}{2}(FP+FN)} \tag{6}$$

Weighted average of model is calculated with (7).

$$Score_{weighted-avg} = 0.998 \cdot Score_{class_0} + 0.002 \cdot Score_{class_1} \quad (7)$$

Macro average of model is calculated with (8).

$$Score_{macro-avg} = 0.5 \cdot Score_{class_0} + 0.5 \cdot Score_{class_1} \quad (8)$$

The performance of the model is summarized in Table 3, which represents the total accuracy of the model to be 99.58% for recognizing Chinese sign language, 99.62% for recognizing American sign language, 99.63% for recognizing Indian sign language.

3.1. Comparative analysis with state of art methods

Table 3 illustrates the performance of the proposed RNN-LSTM-based multilingual sign language recognition system is compared with existing approaches reported in the literature to provide indicative benchmarking rather than direct performance comparison. Most of the existing works evaluate their models on different datasets, signer populations, gesture vocabularies, sensor configurations, and evaluation protocols, which makes strict one-to-one comparison infeasible. Therefore, the comparative results reported here are to provide context for the effectiveness of the proposed approach, not to claim absolute superiority.

Table 3. Comparative analysis of sign language recognition methods

Method	Model type	Dataset used	Sign languages	Evaluation nature	Accuracy (%)
Kasapbaşi *et al.* [1]	CNN	Public American sign language dataset	American sign language	Indicative (literature)	99.38
Ghotkar *et al.* [2]	Hand-crafted + NN	Indian sign language dataset	Indian sign language	Indicative (literature)	94.15
Dabwan *et al.* [3]	DenseNet-121	American sign language alphabet dataset	American sign language	Indicative (literature)	97.00
Chung *et al.* [23]	ResNet + Bi-LSTM	Chinese sign language continuous dataset	Chinese sign language	Indicative (literature)	98.75
Baseline CNN (reimplemented)	CNN	Self-collected dataset	Chinese sign language/American sign language/Indian sign language	Direct (same dataset)	96.12/96.34 /96.08
Proposed method	RNN-LSTM + MediaPipe	Self-collected dataset	Chinese sign language/American sign language/Indian sign language	Direct (same dataset)	99.58/99.62 /99.63

To guarantee an equitable and regulated comparison, a baseline CNN model was re-implemented and assessed using the identical self-collected dataset utilized in this study. The baseline CNN utilized the same preprocessing steps, MediaPipe-based hand landmark extraction, and the identical training-testing split strategy as the proposed method. The baseline CNN had average recognition accuracies of 96.12% for Chinese sign language, 96.34% for American sign language, and 96.08% for Indian sign language. These numbers are always lower than the accuracies of the proposed RNN-LSTM model (99.58%, 99.62%, and 99.63%, respectively). This performance gap demonstrates the advantage of incorporating temporal modeling through LSTM layers, which effectively capture sequential dependencies in hand landmark trajectories that are not fully exploited by frame-based CNN models.

Overall, while literature-based comparisons are indicative due to dataset and protocol differences, the internal baseline evaluation on the same dataset provides strong experimental evidence supporting the effectiveness and robustness of the proposed RNN-LSTM framework for multilingual sign language recognition. Table 4 represents precision, recall, and F1-score achieved for the sign alphabets A–Z with the support number of tests performed in Chinese sign language, which helps to compute accuracy, macro average, and weighted average of the model on Chinese sign language. Table 5 represents precision, recall, and F1-score achieved for the sign alphabets A–Z with the support number of tests performed in American sign language, which helps to compute accuracy, macro average, and weighted average of the model on American sign language. Table 6 represents precision, recall, and F1-score achieved for the sign alphabets A–Z with the support number of tests performed in Indian sign language, which helps to compute accuracy, macro average, and weighted average of the model on Indian sign language. The above performance of the model is computed using confusion matrix as seen in Figure 13, where Figure 13(a) shows Chinese, Figure 13(b) shows American, and Figure 13(c) shows Indian sign language.

Table 4. Performance of model on Chinese sign language

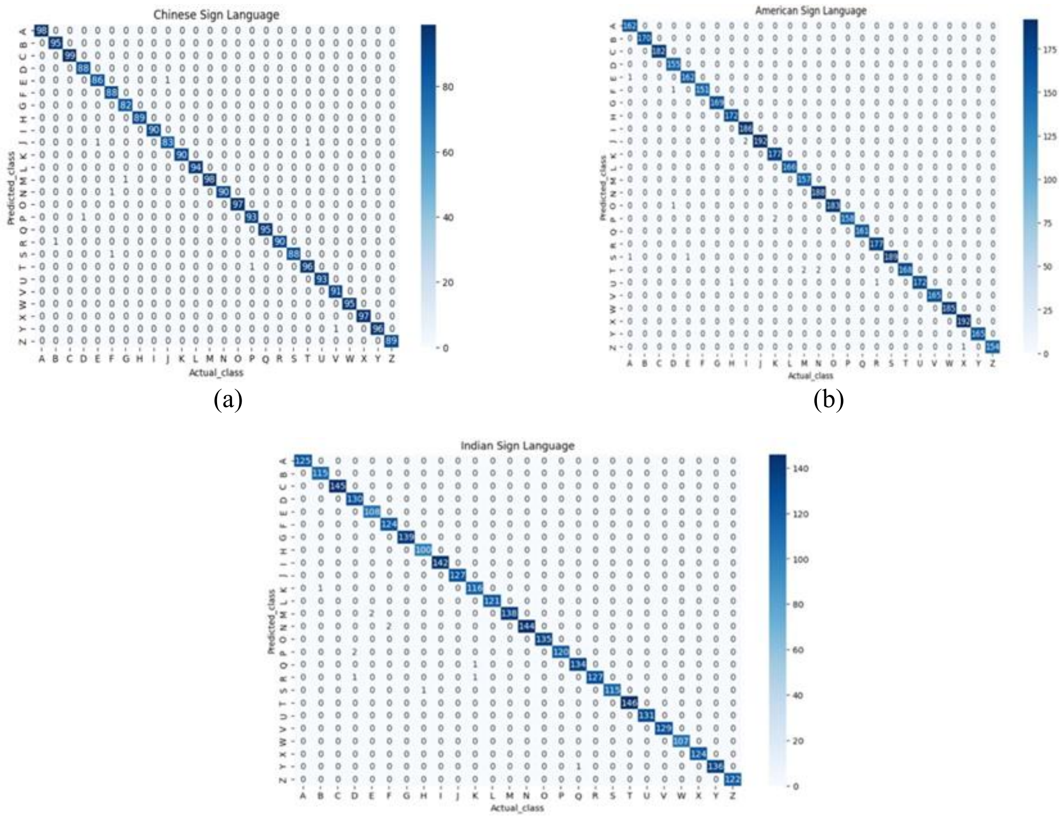
Labels	Precision	Recall	F1-score	Support
A	1	1	1	98
B	0.990	1	0.995	96
C	1	1	1	99
D	0.989	1	0.994	89
E	0.989	1	0.994	87
F	0.978	1	0.989	90
G	0.988	1	0.994	83
H	1	1	1	89
I	0.989	1	0.994	90
J	0.988	0.988	0.988	85
K	0.989	1	0.994	90
L	1	1	1	94
M	1	0.990	0.995	99
N	1	0.989	0.994	91
O	1	1	1	97
P	0.989	0.989	0.989	95
Q	1	1	1	95
R	1	0.989	0.994	91
S	1	0.989	0.994	89
T	1	0.990	0.995	98
U	1	0.989	0.995	94
V	0.989	1	0.995	92
W	1	1	1	95
X	1	1	1	98
Y	1	0.990	0.995	97
Z	1	1	1	89
Accuracy			0.9958	2410
Macro average	0.995	0.996	0.996	2410
Weighted average	0.995	0.996	0.996	2410

Table 5. Performance of model on American sign language

Labels	Precision	Recall	F1-score	Support
A	0.988	1	0.994	164
B	1	1	1	170
C	1	1	1	182
D	0.987	1	0.994	157
E	0.994	0.994	0.994	164
F	1.000	0.993	0.997	152
G	1.000	1	1	169
H	0.994	1	0.997	173
I	0.989	1	0.995	188
J	1	0.990	0.995	194
K	0.989	1	0.994	179
L	1	1	1	166
M	0.988	1	0.994	159
N	0.990	1	0.995	190
O	1	0.995	0.997	184
P	1	0.988	0.994	160
Q	1	1	1	161
R	0.994	1.000	0.997	178
S	1	0.990	0.995	191
T	1	0.977	0.988	172
U	1	0.989	0.994	174
V	1	1	1	165
W	1	1	1	185
X	0.990	1	0.995	193
Y	1	1	1	165
Z	1	0.987	0.994	156
Accuracy			0.9962	4491
Macro Average	0.996	0.996	0.996	4491
Weighted Average	0.995	0.996	0.996	4491

Table 6. Performance of model on Indian sign language

Labels	Precision	Recall	F1-score	Support
A	1	1	1	125
B	0.991	1	0.996	116
C	1	1	1	145
D	0.978	1	0.989	133
E	0.982	1	0.991	110
F	0.984	1	0.992	126
G	1	1	1.000	139
H	0.990	1	0.995	100
I	1	1	1	142
J	1	1	1	127
K	0.983	0.992	0.987	119
L	1	1	1	121
M	1	0.986	0.993	140
N	1	0.986	0.993	146
O	1	1	1	135
P	1	0.984	0.992	122
Q	0.993	0.993	0.993	136
R	1	0.984	0.992	129
S	1	0.991	0.996	116
T	1	1	1	146
U	1	1	1	131
V	1	1	1	129
W	1	1	1	107
X	1	1	1	124
Y	1	0.993	0.996	137
Z	1	1	1	122
Accuracy			0.9963	3323
Macro average	0.996	0.996	0.996	3323
Weighted average	0.995	0.996	0.996	3323



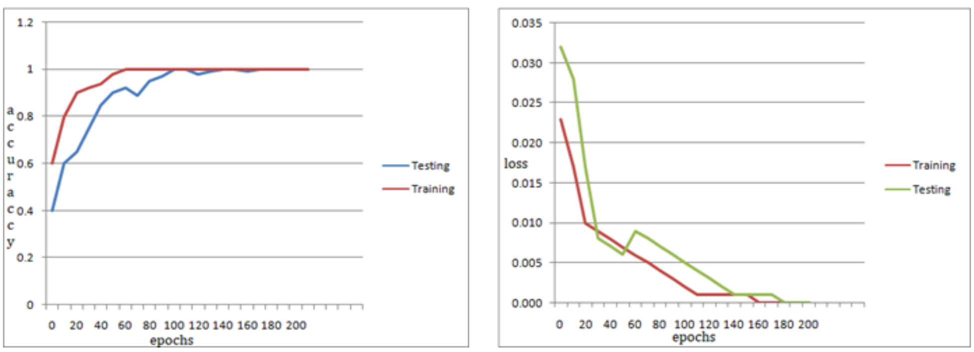
(a) (b)

(c)

Figure 13. Confusion matrix of different sign languages of (a) Chinese, (b) American, and (c) Indian

3.2. Accuracy and loss

We trained the model with 200 epochs and used early function to improve over fitting and generalization. Figure 14 shows the results of specific testing and training sessions based on accuracy (Figure 14(a)) and loss values (Figure 14(b)). Furthermore, overfitting was controlled through early stopping during training and by enforcing a strict signer-independent evaluation protocol, ensuring that no subject appears in both the training and test sets.

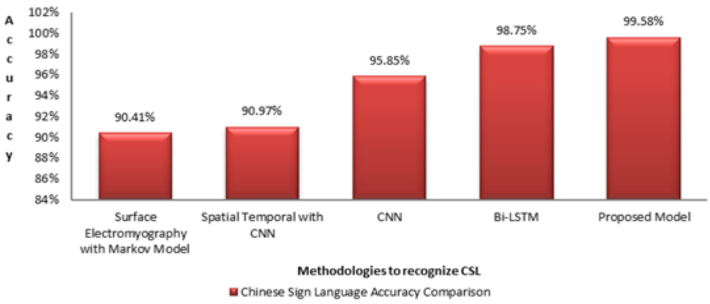


(a) (b)

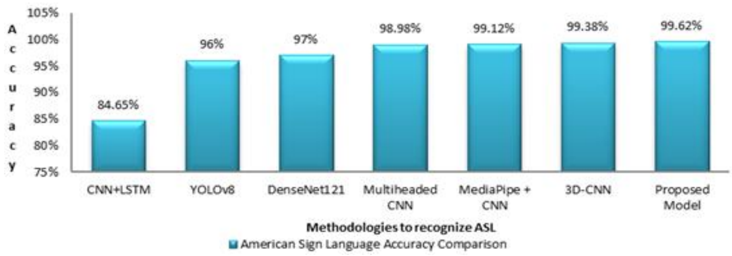
Figure 14. Metrics during the model training and testing process over 200 epochs for (a) accuracy and (b) loss

3.2. Comparative analysis

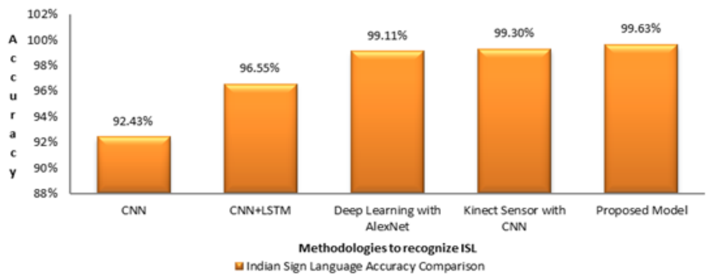
The accuracy of suggested model and current models, which were created with various technologies, has been compared [1]. The efficiency of CNN model for recognizing the sign language of America is 99.38% [2]. Fourier descriptor model designed for recognizing the Indian sign language with efficiency of 94.15% [3]. DenseNet-121 model alongside a dataset representing the American sign language alphabet with accuracy 97% [4]. HGR-ViT model that obtains an accuracy of 99.98% for American sign language [5]. SL-TSSI DenseNet model has 98.0% test accuracy on Mexican sign language [6]. CNN-LSTM hybrid framework model with average accuracy of 84.65% for American sign language [7]. CNN with the MediaPipe model on American sign language datasets has achieved accuracy 99.12% [8]. CNN are applied to recognize Indian sign language with accuracy up to 99.3% [10]. Ghanaian sign language was recognized with the efficiency of 96.0% using DCNN [11]. CNN model average testing accuracy of 93.67% for American sign language alphabet recognition [13]. Magnetic positioning system for recognizing American sign language alphabets with accuracy of 97% [14]. YOLOv4-CSP model recognised Turkish sign language 98.55 accuracy [15]. Multi-headed CNN model uses American sign language dataset with test accuracy 98.981% [16]. Combination of spatio-temporal characteristics with CNN to recognize Chinese sign language characters with an efficiency of 90.87% [17]. MediaPipe library and a CNN for American sign language with an efficiency of 99.95% [18]. CNN for Indian sign language gestures achieved accuracy is 92.43% [19]. detects American sign language using YOLOv8 with accuracy 96% [20]. The efficiency of recognizing Indian sign language was 99.11% by using approach of deep learning with AlexNet [21]. Leap motion controller for American sign language recognition using a DNN is 93.81% [22]. Integrates LSTM models with CNN for 96.55% efficiency in Indian sign language [23]. CSL recognition by Bi-LSTM model with accuracy 98.75% [24]. sEMG signal with hidden Markov model recognises Chinese sign language with accuracy 90.41% [26]. CNN has a 95.85% success rate in recognizing Chinese sign language. Figure 15 shows a comparative analysis of the accuracy of various technologies on various sign languages. Figure 15(a) shows the performance comparison of various models on Chinese sign language, Figure 15(b) illustrates the comparative accuracy results for American sign language, and Figure 15(c) shows the performance comparison for Indian sign language, highlighting the effectiveness of the proposed RNN-LSTM-based approach.



(a)



(b)



(c)

Figure 15. Comparative analysis of recognition accuracy achieved using different technologies across multiple sign languages for (a) Chinese sign language, (b) American sign language, and (c) Indian sign language

3.3. Computational complexity and inference speed analysis

To further improve methodological transparency, we report the computational complexity and inference performance of the proposed RNN-LSTM-based multilingual sign language recognition system. All tests were done on a standard workstation with an Intel i7 CPU, 16 GB of RAM, and an NVIDIA GTX-series GPU. The model works with hand landmark sequences that have been extracted from MediaPipe. This cuts down on the number of inputs needed compared to methods that use raw images.

Table 7 summarizes the model size, number of trainable parameters, average inference time per gesture sequence, and real-time throughput in frames per second (FPS). The results confirm that the proposed system is computationally efficient and suitable for real-time deployment while maintaining high recognition accuracy. The lightweight landmark-based input combined with sequence modeling enables the system to achieve low-latency inference without sacrificing recognition performance. Compared to frame-based CNN models, which typically require higher computational resources due to image-level processing, the proposed approach demonstrates a favorable trade-off between accuracy and efficiency, making it well suited for assistive and real-time applications.

Table 7. Computational complexity and inference performance of the proposed model

Metric	Value
Input representation	21 hand landmarks × (x, y, z)
LSTM layers	3 (64–128–64 units)
Total trainable parameters	~1.12 million
Model size	~5.4 MB
Average inference time (per sequence)	18.6 ms
Throughput	~53 FPS
Real-time capability	Yes

4. CONCLUSION

This work presented a robust and computationally efficient multilingual sign language recognition framework, which combines MediaPipe-based hand landmark extraction with deep temporal modeling using a stacked RNN-LSTM architecture to address both one-handed and two-handed gesture dynamics across Chinese, American, and Indian sign languages within a unified pipeline. The proposed system uses lightweight landmark representations and sequence learning to effectively capture spatiotemporal gesture characteristics without the high computational overhead of image-based models. A strict signer-independent evaluation protocol and repeated experimental runs confirmed strong generalization and statistical stability, yielding high recognition accuracies of 99.58%, 99.62%, and 99.63% for Chinese sign language, American sign language, and Indian sign language, respectively. Comprehensive performance analysis using precision, recall, F1-score, and confusion matrices as well as controlled baseline comparisons demonstrated the clear advantage of temporal modeling over frame-based CNN approaches. Furthermore, computational complexity analysis verified real-time inference capability with low latency and modest model size, supporting practical deployment. Overall, the proposed framework provides a scalable and generalisable basis for multilingual sign language recognition and provides a promising path towards future extensions for continuous sign recognition and multimodal assistive communication systems for the hearing and speech-impaired community.

ACKNOWLEDGEMENTS

The authors thankful to the management of Vidya Vikas Institute of Engineerig and Technology, Mysuru, Nitte Meenakshi Institute of Technology, Bengaluru and Visvesvaraya Technological University, Belagavi for their continuous support and guidance.

FUNDING INFORMATION

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. No funding was raised or provided for the conduct of this study.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Thouseef Ulla Khan	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓
Dileep Marichi Ramachandra	✓			✓			✓			✓		✓	✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors confirm that there is no conflict of interest with anyone involved.

DATA AVAILABILITY

The data that support the findings of this study will be available in the given link at https://drive.google.com/drive/folders/1-ONUK2OOhoZ_9nJsIbFHegvU2-vDZMDf?usp=drive_link

REFERENCES

[1] A. Kasapbaşi, A. E. A. Elbushra, O. Al-Hardanee, and A. Yilmaz, “DeepASLR: a CNN based human computer interface for American sign language recognition for hearing-impaired individuals,” *Computer Methods and Programs in Biomedicine Update*, vol. 2, 2022, doi: 10.1016/j.cmpbup.2021.100048.

[2] A. Ghotkar, P. Vidap, and S. Ghotkar, “Vision based hand gesture recognition using fourier descriptor for Indian sign language,” *Signal & Image Processing : An International Journal*, vol. 7, no. 6, pp. 29–38, Dec. 2016, doi: 10.5121/sipij.2016.7603.

[3] B. A. Dabwan *et al.*, “Hand gesture classification for individuals with disabilities using the DenseNet121 model,” in *2024 International Conference on Advancements in Power, Communication and Intelligent Systems (APCI)*, Jun. 2024, pp. 1–5, doi: 10.1109/APCI61480.2024.10616504.

[4] C. K. Tan, K. M. Lim, R. K. Y. Chang, C. P. Lee, and A. Alqahtani, “HGR-ViT: hand gesture recognition with vision transformer,” *Sensors*, vol. 23, no. 12, Jun. 2023, doi: 10.3390/s23125555.

[5] D. Laines, M. G. -Mendoza, G. O. -Ruiz, and G. Bejarano, “Isolated sign language recognition based on tree structure skeleton images,” in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2023, pp. 276–284, doi: 10.1109/CVPRW59228.2023.00033.

[6] D. Kumari and R. S. Anand, “Isolated video-based sign language recognition using a hybrid CNN-LSTM framework based on attention mechanism,” *Electronics*, vol. 13, no. 7, Mar. 2024, doi: 10.3390/electronics13071229.

[7] P. K. Chowdhury, K. U. Oyshe, M. A. Rahaman, T. Debnath, A. Rahman, and N. Kumar, “Computer vision-based hybrid efficient convolution for isolated dynamic sign language recognition,” *Neural Computing and Applications*, vol. 36, no. 32, pp. 19951–19966, Nov. 2024, doi: 10.1007/s00521-024-10258-3.

[8] J. Gangrade and J. Bharti, “Vision-based hand gesture recognition for Indian sign language using convolution neural network,” *IETE Journal of Research*, vol. 69, no. 2, pp. 723–732, Feb. 2023, doi: 10.1080/03772063.2020.1838342.

[9] K. M. Tripathi, P. Kamat, S. Patil, R. Jayaswal, S. Ahirrao, and K. Kotecha, “Gesture-to-text translation using SURF for Indian sign language,” *Applied System Innovation*, vol. 6, no. 2, Mar. 2023, doi: 10.3390/asi6020035.

[10] L. K. Odartey, Y. Huang, E. E. Asantewaa, and P. R. Agbedanu, “Ghanaian sign language recognition using deep learning,” in *Proceedings of the 2019 the International Conference on Pattern Recognition and Artificial Intelligence*, Aug. 2019, pp. 81–86, doi: 10.1145/3357777.3357784.

[11] L. K. S. Tolentino, R. O. S. Juan, A. C. Thio-ac, M. A. B. Pamahoy, J. R. R. Forteza, and X. J. O. Garcia, “Static sign language recognition using deep learning,” *International Journal of Machine Learning and Computing*, vol. 9, no. 6, pp. 821–827, Dec. 2019, doi: 10.18178/ijmlc.2019.9.6.879.

[12] M. S. Lomas, A. Quelal, and M. E. M. -Cayamcela, “Implementation of a lightweight CNN for American sign language classification,” in *Doctoral Symposium on Information and Communication Technologies*, Springer, Cham, 2022, pp. 197–207, doi: 10.1007/978-3-031-18347-8_16.

[13] M. Rinalduzzi *et al.*, “Gesture recognition of sign language alphabet using a magnetic positioning system,” *Applied Sciences*, vol. 11, no. 12, Jun. 2021, doi: 10.3390/app11125594.

[14] M. Alaftekin, I. Pacal, and K. Cicek, “Real-time sign language recognition based on YOLO algorithm,” *Neural Computing and Applications*, vol. 36, no. 14, pp. 7609–7624, May 2024, doi: 10.1007/s00521-024-09503-6.

[15] J. Wu, T. Yang, and J. Meng, “Research and application of sign language recognition and target tracking model based on YOLO-Mamba,” *Multimedia Systems*, vol. 31, no. 6, Dec. 2025, doi: 10.1007/s00530-025-02003-0.

[16] R. Sasidharan, M. Rashmi, and S. M. S. Suresh, “Sign language recognition by using spatio-temporal features,” *Procedia Computer Science*, vol. 233, pp. 353–362, 2024, doi: 10.1016/j.procs.2024.03.225.

[17] J. Ni, Y. Wang, G. Tang, W. Cao, and S. X. Yang, “A lightweight GRU-based gesture recognition model for skeleton dynamic graphs,” *Multimedia Tools and Applications*, vol. 83, no. 27, pp. 70545–70570, Jan. 2024, doi: 10.1007/s11042-024-18313-w.

[18] S. Sharma and S. Singh, “Recognition of indian sign language (ISL) using deep learning model,” *Wireless Personal Communications*, vol. 123, no. 1, pp. 671–692, 2022, doi: 10.1007/s11277-021-09152-1.

[19] Q. Tian, S. Li, Y. Zhang, H. Lu, and H. Pan, “Action recognition method based on a novel keyframe extraction method and enhanced 3D convolutional neural network,” *International Journal of Machine Learning and Cybernetics*, vol. 16, no. 1, pp. 475–491, Jan. 2025, doi: 10.1007/s13042-024-02235-y.

[20] R. Sreemathy, M. Turuk, I. Kulkarni, and S. Khurana, "Sign language recognition using artificial intelligence," *Education and Information Technologies*, vol. 28, no. 5, pp. 5259–5278, May 2023, doi: 10.1007/s10639-022-11391-z.

[21] T.-W. Chong and B.-G. Lee, "American sign language recognition using leap motion controller with machine learning approach," *Sensors*, vol. 18, no. 10, Oct. 2018, doi: 10.3390/s18103554.

[22] V. Perumal, H. Kar, S. Gautam, M. Mekala, M. Rahimi, and A. H. Gandomi, "Sign language translator for dumb and deaf," in *2023 10th International Conference on Soft Computing & Machine Intelligence (ISCMI)*, Nov. 2023, pp. 198–202, doi: 10.1109/ISCMI59957.2023.10458471.

[23] W.-Y. Chung, H. Xu, and B. G. Lee, "Chinese sign language recognition with batch sampling ResNet-Bi-LSTM," *SN Computer Science*, vol. 3, no. 5, Aug. 2022, doi: 10.1007/s42979-022-01341-4.

[24] W. Li, Z. Luo, W. Li, and X. Xi, "Chinese sign language recognition based on surface electromyography and motion information," *PLOS ONE*, vol. 18, no. 12, Dec. 2023, doi: 10.1371/journal.pone.0295398.

[25] X. Jiang, Y. Zhang, J. Lei, and Y. Zhang, "A survey on Chinese sign language recognition: from traditional methods to artificial intelligence," *Computer Modeling in Engineering & Sciences*, vol. 140, no. 1, pp. 1–40, 2024, doi: 10.32604/cmes.2024.047649.

[26] Y. Liu *et al.*, "A wearable system for sign language recognition enabled by a convolutional neural network," *Nano Energy*, vol. 116, Nov. 2023, doi: 10.1016/j.nanoen.2023.108767.

[27] B. Alsharif, E. Alalwany, A. Ibrahim, I. Mahgoub, and M. Ilyas, "Real-time American sign language interpretation using deep learning and keypoint tracking," *Sensors*, vol. 25, no. 7, Mar. 2025, doi: 10.3390/s25072138.

[28] M. Maashi, H. G. Iskandar, and M. Rizwanullah, "IoT-driven smart assistive communication system for the hearing impaired with hybrid deep learning models for sign language recognition," *Scientific Reports*, vol. 15, no. 1, Feb. 2025, doi: 10.1038/s41598-025-89975-1.

[29] S. T. A. Al-Latief, S. Yussof, A. Ahmad, and S. Khadim, "Deep learning for sign language recognition: a comparative review," *Journal of Smart Internet of Things*, vol. 2024, no. 1, pp. 77–116, Jun. 2024, doi: 10.2478/jsiot-2024-0006.

[30] A. Baihan, A. I. Alutaibi, M. Alshehri, and S. K. Sharma, "Sign language recognition using modified deep learning network and hybrid optimization: a hybrid optimizer (HO) based optimized CNNSa-LSTM approach," *Scientific Reports*, vol. 14, no. 1, Oct. 2024, doi: 10.1038/s41598-024-76174-7.

[31] S. Xue, L. Gao, L. Wan, and W. Feng, "Multi-scale context-aware network for continuous sign language recognition," *Virtual Reality & Intelligent Hardware*, vol. 6, no. 4, pp. 323–337, Aug. 2024, doi: 10.1016/j.vrih.2023.06.011.

[32] N. Amangeldy, A. Ukenova, G. Bekmanova, B. Razakhova, M. Milosz, and S. Kudubayeva, "Continuous sign language recognition and its translation into intonation-colored speech," *Sensors*, vol. 23, no. 14, Jul. 2023, doi: 10.3390/s23146383.

[33] W. Aditya *et al.*, "Novel spatio-temporal continuous sign language recognition using an attentive multi-feature network," *Sensors*, vol. 22, no. 17, Aug. 2022, doi: 10.3390/s22176452.

[34] K. Myagila, D. G. Nyambo, and M. A. Dida, "Framework for detecting and recognizing sign language using absolute pose estimation difference and deep learning," *Machine Learning with Applications*, vol. 21, Sep. 2025, doi: 10.1016/j.mlwa.2025.100723.

[35] L. Trabelsi, H. Harrouch, S. Mohamed, M. Jebali, and A. Sharma, "Advancements and challenges in vision-based sign language recognition: a comprehensive review," *Information Fusion*, vol. 126, Feb. 2026, doi: 10.1016/j.inffus.2025.103626.

BIOGRAPHIES OF AUTHORS

Thouseef Ulla Khan pursuing doctor of philosophy in the filed of AI and ML under Visvesvaraya Technioloigcal University, Belagavi, Karanataka, India. He holds master degree in Computer Applications from Visvesvaraya Technioloigcal University, Belagavi, Karnataka, India in the year 2012. He also received his B.Sc. in Physics, Mathematics, and Computer Science from Univeristy of Mysore, India in the year 2008. He has 15 years of teaching experience at different levels. He is currently working as an assistant professor at Department of Master of Computer Applications in Vidya Vikas Institute of Engineering and Technology, Mysore, Visvesvaraya Technioloigcal University, Belagavi, Karnataka, India. His research includes image processing, AI, and machine learning. He can be contacted at email: thouseef588@gmail.com.

Dileep Marichi Ramachandra holds doctor of philosophy and M.Phil. in Computer Science from Bharathiar University, Coimbatore in 2019 and 2015 respectively. He also received master degree in Computer Applications from Visvesvaraya Technioloigcal University, Belagavi, Karnataka, India in the year 2013. He has 12 years of teaching experience. His research includes digital image processing, AI, machine learning, and pattern matching. He has published over 42 papers in international journals and conferences from 2014 to 2024. He is life time member of International Research Developers and Publishers (IRDP). He also editorial board member of Artificial Intelligence Research Journal (AIRJ), Bio-Byword publishers, Sydney, Australia, Journal of Computer Science Research, Bilingual Publishing Co, Singapore, American Journal of Computer Science and Technology, Science Publishing Group (SciencePG), NewYork, USA and IEEE senior member. He can be contacted at email: dileep.kurunimakki@gmail.com.