Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Cross-Age Real-Time Kannada Sign Language Recognition using Curvilinear Geometric Features and CNN-LSTM Models

Ramesh M. Kagalkar
Srinivas University
Bahubali Shiragapur
bahubali-aiml@dsu.edu.in
Dayananda Sagar University University
Praveen B M
Srinivas University

Research Article

Keywords: Kannada Sign Language(KSL), CNN–LSTM, Curvilinear Geometric Features, Real-Time Gesture Recognition, Sign-to-Text Translation(STT), Deep Learning, Human Computer Interaction (HCI)

Posted Date: September 18th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10588234/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Cross-Age Real-Time Kannada Sign Language Recognition using Curvilinear Geometric Features and CNN-LSTM Models

[1]Ramesh M. Kagalkar, [2 *] Bahubali Shiragapur [3]Praveen B.M. and
[1&3] Srinivas University, Srinivas Nagar, Mukka, Surathkal, Mangalore, Karnataka, India.
[2]Dayananda Sagar University University,Karnataka, India.(*Corresponding Author).
Email: rameshvtu10@gmail.co, bm.praveen@yahoo.co.in and bahubali-aiml@dsu.edu.in

Abstract:

Communication barriers between Deaf and hearing individuals remain a significant challenge. This is particularly for regional sign languages such as Kannada Sign Language (KSL). It's an a open source dataset, for which annotated datasets and robust recognition systems are scarce. This study presents a cross age, real-time KSL recognition and sign-to-text translation framework; Which combines curvilinear geometric feature extraction with a hybrid Convolutional Neural Network–Long Short-Term Memory (CNN–LSTM) architecture for joint spatial–temporal gesture modelling. A self-curated Kannada Sign Language Dataset (S-KSLD) of 5,000 annotated gesture samples, covers 10 Kannada numerals and 36 Kannada consonants. Data was collected from 20 signers across different age groups under varying illumination, background, and hand-orientation conditions. The dataset was split into training (70%), validation (15%) and testing (15%) subsets and evaluated using stratified five-fold signer-independent cross-validation. The proposed pipeline performs image preprocessing and hand segmentation, extracts curvilinear geometric descriptors based on peak curvature points and region closure area, and applies CNN-based spatial feature learning followed by LSTM-based temporal sequence modelling. The framework achieved an average recognition accuracy of 94.6%, precision of 94.1%, recall of 93.8% and F1-score of 94.3%, with a mean inference time of 178 ms per frame. Relative to conventional edge based, keypoint-based, CNN, CNN–LSTM, and MediaPipe Holistic LSTM baselines, the proposed approach improved recognition accuracy by 2.9–11.2% while lowering computational latency. These findings indicate that the proposed framework is effective, robust and suitable for real-time deployment in assistive communication, educational technology and regional-language accessibility applications.

Keywords: Kannada Sign Language(KSL), CNN–LSTM, Curvilinear Geometric Features, Real-Time Gesture Recognition, Sign-to-Text Translation(STT), Deep Learning, Human Computer Interaction (HCI).

1. Introduction

Communication is a fundamental human right, yet millions of Deaf and hard-of-hearing individuals continue to face systemic barriers to education, healthcare, and public services because of the limited availability of real-time sign language interpretation. Globally, deep learning has been widely applied to improve sign-to-text and sign-to-speech translation systems; however, this research has predominantly focused on widely used sign languages such as American Sign Language (ASL) and British Sign Language (BSL), leaving regional sign languages largely unexplored. Kannada Sign Language (KSL) remains under explores in India, limiting digital accessibility and educational support for Kannada speaking Deaf users. The existing sign language recognition (SLR) systems are often restricted to isolated gestures, depend on specific signers. These systems perform poorly with continuous gestures for varying environments. Most of the approaches depends on sparse keypoints or edge based features. This is inadequate to capture subtle hand shape variations needed to distinguish similar signs. Existing methods struggle to capture subtle hand-shape variations, and progress is limited by the lack of annotated datasets for Kannada Sign Language (KSL). This work addresses this gap by introducing a dedicated KSL dataset and a real-time recognition framework.

The proposed system combines a hybrid CNN and LSTM with curvilinear geometric features for robust hand-shape representation. A self-curated S-KSLD dataset of 5000+ annotated gesture videos was developed under diverse recording conditions. The framework achieves 94.6% accuracy with <180 ms inference time. This demonstrates its suitability for real-time assistive and educational applications while providing a foundation for future sentence-level KSL translation.

2. Literature review

Recent work in SLR has increasingly focused on deep learning and real time recognition [1]–[7]. Baihan et al. [1] and Sharma et al. [2] introduced methods that improved both recognition accuracy

and processing speed. A drawback of this approach is that multilingual validation and performance on edge devices has not been explored. Dignan et al. [3] and Srivastava et al. [4] achieved excellent recognition accuracy with multi-view approaches.

However, these approaches requires significant computational power, while some like MediaPipe Holistic does not perform well in adverse scenarios. Static ISL gestures are addressed with CNN approach by Sharma and Singh [5]. Limitations of avatar realism, contextual information and emotion aware recognition are discussed in [6] and [7].

Joshi and Patel [8] proposed a deep CNN to recognize isolated Gujarati sign but did not have the ability to recognize continuous signs and make signer-independent recognition. A survey on the avatar-based translation system has been conducted by Wolfe et al. [9]. Challenges such as cultural aspect and co-articulation were reviewed. Farooq et al. [10] have done a review on multi-lingual dataset and evaluation methods for sign language machine translation. Das et al. [11] improved recognition using temporal CNN, but the accuracy was very sensitive to variation in pose and signers. Lasri et al. [12] have showcased the capability of emotion detection system to facilitate adaptive learning. A better recognition rate of isolated ISL can be obtained by Khartheesvar et al. [13] by using MediaPipe Holistic followed by LSTM. However, the model was tested in laboratory setup and is susceptible to variance in signers' gestures. Transformer model has been used for sign localization task by Liu et al. [14] but does not modeling temporal dynamics of gestures.

Venugopalan and Reghunadhan [15] suggested hybrid DNN for hand based health sign language interactions, but was constrained on scalability. Balaha et al. [16] designed a signer-independent Arabic Sign Language recognition without analyzing dialect or real-time feasibility. Abeje et al. [17] obtained positive results for Ethiopian Sign Language with CNN, although it had a lack of dataset variety. Ali et al. [18] developed a UNL-based text-to-sign translation system without gesture generation. Kim and Baek [19] successfully performed a precise glove based finger spelling scheme. Nonetheless it was wearable so it resulted in low feasibility. Shields et al. [20] assisted the access to healthcare information by converting the 5-level Euro-QOL-5-dimension instrument to British Sign Language. Raghuveera et al. [21] applied Kinect’s depth sensing for sign recognition to achieve performance for Indian Sign Language but it severely hampered mobile implementation of.

Kumar et al. [23] studied modular TTS architectures with scope to sign-to-speech systems with limited multimodal integration. Najib [24] proposed multistage AI based SLR without supporting continuous gestures. Balaha et al. [25] utilized CNNs to Arabic Sign Language recognition but failed to include expression based augmentation. Farooq et al. [26] identified absence of universal SLMT datasets. Kahlon and Singh [27] experienced challenges in grammatical alignment and latency for text-to-sign systems. Sharma et al. [28] presented a neural pipeline architecture for continuous sign language translation but not tested in real users. Dignan et al. [29] improved recognition using multi angle cameras. Srivastava et al. [30] showcased real-time Indian Sign Language recognition using MediaPipe Holistic. These were tested under isolated deployment conditions.

Improving CNN based ISL recognition, Das et al. [32] added some temporal information, but they maintained signer diversity, which was still a problem in the end. The technique applied to localize gesture rather than learning temporally is Liu et al. [33] who employed DETR. A hybrid proposed a hybrid DNN for healthcare applications with limited vocabulary coverage [34]. Although Balaha et al. [35] did a good work on vision-based arabic SL Recognition, they are yet to exploit user feedback.

Ethiopian SL recognition achieved improvement thanks to Abeje et al. [36], albeit with limitations due to dataset size restriction. UNL-based Translation using CNN was shown by Ali et al.[37] without being an at end-to-end framework. Glove-based Recognition was shown to be very accurate by Kim and Baek [38], despite the fact that gloves are uncomfortable to wear. Khartheesvar et al. [39] validated an LSTM based full body recognition model only on small datasets. Wolfe et al. [41] showed avatar realism by designing anew Avatar model but failed to represent emotion and motion.

On another note, Farooq et al. [42] highlighted The fact That translation, multilingual and multimodal standardized sign language translation systems.

Hugar et al. [43] compared CNN, LSTM and CNN LSTM models for Kannada Sign Language (KSL) recognition. This shows that hybrid architectures improved recognition accuracy and robustness. Kagalkar et al. [44] developed a real time KSL translation framework integrating gesture recognition with text generation. This systems provides a reliable accuracy and low inference latency for assistive communication. Hugar and Kagalkar [45] proposed a dual stream CNN and LSTM architecture that jointly learned spatial and temporal features. This improves robustness under varying illumination and background conditions for real-time KSL recognition.

A. Key research gaps

Despite of these recent progress current SLR and translation systems has several limitations:

- Limited datasets: Small KSL/SLR datasets restrict generalization across languages and signers
- Isolated recognition: Poor support for continuous signing and signer variability
- Limited robustness: Insufficient validation under real world conditions
- Missing multimodal cues: Facial expressions and posture are often ignored
- Limited usability: User-centric accessibility and feedback remain inadequate

B. Novelty of the work

The main contributions of this work are summarized as follows:

- Novel features: Curvilinear descriptors using peak curvature and region closure area
- S-KSLD dataset: 5000+ annotated KSL samples covering numerals and consonants
- Hybrid CNN–LSTM:Integrates spatial, temporal and geometric features for robust recognition
- Real time translation: Efficient sign-to-text pipeline for resource-constrained devices
- Superior performance: Higher accuracy and faster inference than existing methods

3. System overview

Figure 1 illustrates the proposed real time Kannada Sign Language (KSL) recognition framework. This integrates curvilinear geometric feature extraction with a hybrid CNN LSTM architecture for sign-to-text translation. The workflow consists of five stages viz. signer diversity, input acquisition, preprocessing, curvilinear feature extraction, gesture recognition and text generation speed. To improve signer independence, the dataset includes children, adults and elderly participants. This has a variations in hand size, orientation, signing style, skin tone and dominant hand. To enhance gesture videos were captured under diverse lighting, backgrounds, viewpoints and indoor and outdoor environments using webcams and mobile cameras, enhancing real-world robustness. Gesture sequences are acquired from live camera feeds or prerecorded videos and frames. These are extracted while preserving temporal information. During preprocessing, RGB images are converted to grayscale, normalized, denoised and segmented. This is going to isolate the hand region using skin-color detection and morphological operations. The segmented images are then resized to 128 × 128 pixels, providing standardized inputs for efficient and accurate CNN LSTM-based recognition.

After preprocessing, the segmented hand images are passed to curvilinear feature extraction module. Unlike conventional edge or skeleton based methods. The framework derives geometric descriptors from the hand contour by extracting peak curvature points and the region closure area (RCA) along with additional curvature based features. These descriptors effectively capture finger shape, contour continuity and hand geometry, forming a discriminative feature vector that is robust to illumination changes and partial occlusion. The feature vector is then processed by a hybrid CNN LSTM model. where the CNN learns spatial representations such as finger shape,palm structure, hand orientation and contour features extracted by successive CNN layer and pooling layers. This process is done before passing them to the LSTM for temporal sequence modeling.

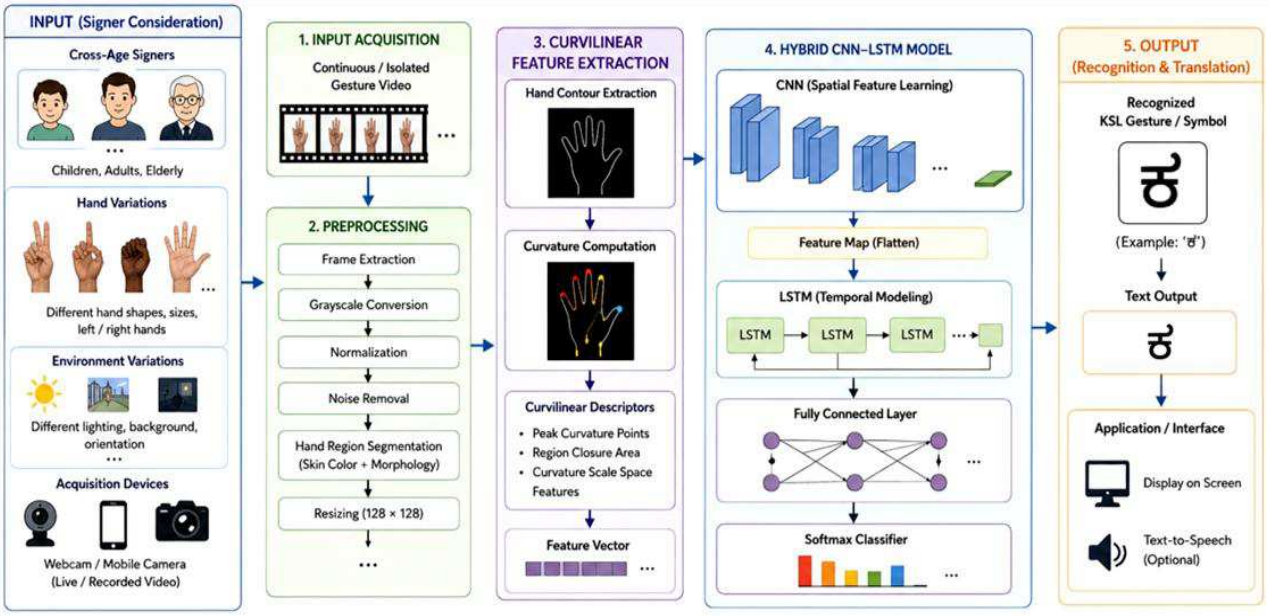


Fig. 1. Proposed real-time KSL recognition framework for sign-to-text translation.

Furthermore, LSTM captures temporal dependencies between consecutive gesture frames and learning dynamic motion patterns essential for continuous sign recognition. The extracted temporal features are fused with spatial features through a fully connected layer and a softmax classifier predicts the corresponding Kannada sign. The recognized sign is then translated into text and displayed through a graphical interface, with optional text-to-speech support for multimodal communication.

The proposed framework is designed for real-time operation and is robust to variations in signer characteristics, illumination and background conditions. The Model is trained on the Self curated Kannada Sign Language Dataset (S-KSLD) containing over 5000 labeled samples from 20 signers. The proposed model achieved 94.6% recognition accuracy with an average inference time of 178 ms per frame. This demonstrates its suitability for real time deployment on desktop and edge devices.

4. Materials and methods

The proposed KSL recognition system will integrate curvilinear feature representation with a hybrid CNN and LSTM model. This enables real time gesture to text conversion. This section discusses dataset composition, annotation and quality assurance procedures. The data availability, image preprocessing, curvilinear feature extraction, spatio temporal modeling and sign-to-text conversion.

4.1 Dataset description

To address the scarcity of regional sign language datasets we have curated S-KSLD. This comprises over 5000 annotated samples of Kannada numerals (0-9) and 36 consonants from 20 signers. The dataset includes variations in hand shape, orientation, lighting and background to improve robustness. All RGB images (640×480 pixels) were manually annotated. This data set is divided into training (70%), validation (15%) and testing (15%) sets for model evaluation.

4.2 Annotation and quality assurance

To ensure annotation quality, each S-KSLD gesture was independently labeled by three certified KSL educators. This was done using standardize label sheet covering numerals and consonants. Inter annotator agreement was excellent, with Cohen's κ of 0.93 and Fleiss' κ of 0.91. Annotation disagreements were resolved. The expert review ensuring consistent and reliable labels across the 5000+ gesture samples.

4.3 Data availability

The S-KSLD dataset contains participant videos and cannot be publicly released due to privacy and consent restrictions. De-identified subsets will be available for non-commercial academic research upon reasonable request and under a data-use agreement. The dataset includes 20 signers (11 male, 9 female, aged 12–58 years) from multiple Kannada speaking regions viz Bengaluru Urban, Mangaluru, Mysuru, Hubballi, Dharwad and Kalaburagi. This demographic and regional diversity of data will definitely ensures real world KSL usage. To the reduce biasness each signer contributed a similar number of samples across all 46 gesture classes. Furthermore, data augumentation was done for a signer independent to train, validation, test split to improve robustness and model generalization.

4.4 Image preprocessing

The RGB gesture images were converted to grayscale and normalized in the range [0, 1]. This is segmented using adaptive thresholding and filtering to isolate the hand region. The images were then resized to 128×128 pixels, reducing computational cost. While preserving gesture features, improving robustness to varying lighting and background conditions.

4.5 Curvilinear feature extraction

The proposed curvilinear feature extraction method captures hand shape geometry. This uses a peak curvature points (PCP) and the region closure area (RCA). The PCP identify finger orientation and RCA represents the overall hand pose. Contours were smoothen using Gaussian and median filtering to reduce noise. These descriptors preserve contour level shape information. This complement CNN based spatial features and improves discrimination between visually similar signs as confirmed by ablation studies.

4.6 Spatio-temporal modeling

To learn spatial structure and temporal dynamics of sign gestures, the abstracted curvilinear features were fed into an amalgam of deep models: CNN-LSTM. A CNN part was a spatial encoder defined by two CNN Conv2D layers with the first layer having 32 filters of size 3×3, ReLU activation function, then 2×2 max pooling, followed by another Conv2D layer with 64 filters with the same configuration. This was then followed by a flattening layer and a 128-unit dense layer. The extracted features from the images were then fed to the LSTM, which had one LSTM layer with 64 units that learned movement of the hand over time. This was followed by a dropout layer with a dropout rate of 0.5 to prevent overfitting, then another LSTM layer with 64 units. Finally, the softmax layer was used for classifying the sign from many gesture classes. Such an arrangement helped understand both single gestures and variations of them in sequence, being able to recognize continuous sign language.

4.7 Sign-to-text conversion

In the final stage, the predicted gesture is converted into a text label. The Softmax activation function is used for selecting the classes with the highest probability. This is going to map to predefined Kannada numeral or consonant label. The recognized text is displayed via graphical user interface. The framework also supports future integration of text-to-speech and avatar-based output to enhance accessibility.

5. Mathematical modelling

The mathematical model standardizes all input images by preprocessing, extracting curvilinear shape features and learning spatial temporal representations via CNN and LSTM models. A softmax classifier maps the learned features to gesture classes. This formulation ensures accurate, efficient and reliable sign-to-text translation suitable for real time applications.

5.1 Image preprocessing and normalization

Let *I(x,y)* denote's the grayscale intensity of a gesture image. The raw RGB image *I(x,y)* is converted into a grayscale intensity image $I_{\text{gray}}(x, y)$ using a weighted linear combination;

$$I_{\text{gray}}(x,y)=0.2989 \quad R+0.5870\cdot G+0.1140\cdot B \tag{1}$$

The eq.(1) interprets conversion of input RGB gesture image into a grayscale image. This is a combination of RGB color channels uses standard luminance weights. This reduces computational complexity by preserving essential structural information. which are required for gesture recognition.

The grayscale image is then normalized to the range [0,1] to improve convergence;

$$I_{\text{norm}}(x,y)=\frac{I_{\text{gray}}(x,y)\text{-}\min(I_{\text{gray}})}{\max(I_{\text{gray}})\text{-}\min(I_{\text{gray}})} \tag{2}$$

The eq.(2) performs min-max normalization to scale pixel intensity values in the range [0,1]. This normalization helps numerical stability, accelerates network convergence. Also, reduces the influence of illumination variations during model training.

For gesture region isolation, it's been used adaptive thresholding;

$$B(x,y)=\begin{cases}1 & \text{if} I_{\text{norm}}(x,y)\geq T(x,y)\\ 0 & \text{otherwise}\end{cases} \tag{3}$$

The eq. (3) uses adaptive thresholding to segment the hand region from the background. This step enhances foreground extraction under varying lighting conditions and improves the accuracy of subsequent feature extraction.

5.2 Curvilinear feature extraction

a) **Peak curvature K:** Let the hand contour be a parametric curve $C(s)=(x(s),y(s))$. Curvature is defined as:

$$\kappa(s)=\frac{x'(s)y''(s)-y'(s)x''(s)}{(x'(s)^2+y'(s)^2)^{3/2}} \tag{4}$$

Peak curvature points s^* are identified where | kappa(s)| is locally maximum. The eq.(4) computes the local curvature of the hand contour. This identify peak curvature points corresponding to fingertips and significant contour changes. These geometric features provide discriminative information for distinguishing similar Kannada sign gestures.

b) **Region closure area (RCA):** The area enclosed by a contour with N vertices (x_i,y_i) is given by;

$$A=\frac{1}{2}\left|\sum_{i=1}^{N}(x_iy_{i+1}-x_{i+1}y_i)\right| \tag{5}$$

The eq. (5) calculates the RCA enclosed by the hand contour. The extracted area feature complements curvature information by representing the overall hand shape and gesture geometry.

5.3 CNN-based spatial feature extraction

Let $f_{ij}^{(l)}$ represent the output of a convolution layer (1) at location (i,j). For an input patch x and kernel V $W^{(1)}$,

$$f_{ij}^{(l)}=\sigma\left(\sum_{m,n} W_{m,n}^{(l)}\cdot x_{i+m,j+n}+b^{(l)}\right) \tag{6}$$

Where, σ is the ReLU activation, $\sigma(z)=\max(0,z)$ and $b^{(l)}$ is the bias term. The eq.(6) defines the convolution operation used by the CNN to extract hierarchical spatial features from the preprocessed gesture images. The learned feature maps capture hand shape, finger orientation, and local texture information.

5.4 LSTM temporal modeling

Let x_t be the CNN feature vector at time t, and the LSTM gates are computed as:

$$f_t=\sigma(W_f x_t+U_f \mathrm{h}_{t-1}+b_f) \text{ (forget gate)} \quad (7)$$

The eq.(7) computes the forget gate of the LSTM network, which determines how much information from the previous memory cell should be retained or discarded during temporal feature learning.

$$i_t=\sigma(W_i x_t+U_i \mathrm{h}_{t-1}+b_i) \text{ (input gate)} \quad (8)$$

The eq.(8) computes the input gate, which regulates the amount of newly extracted spatial information incorporated into the memory cell for learning gesture dynamics.

$$o_t=\sigma(W_o x_t+U_o \mathrm{h}_{t-1}+b_o) \text{ (output gate)} \quad (9)$$

The eq.(9) calculates the output gate, controlling the amount of memory information transferred to the hidden state for predicting the current gesture.

$$\tilde{c}_t=\tanh(W_c x_t+U_c \mathrm{h}_{t-1}+b_c) \text{ (cell candidate)} \quad (10)$$

The eq.(10) generates the candidate memory content containing newly learned temporal information before updating the LSTM cell state.

The cell state update is defined as:

$$c_t=f_t \odot c_{t-1}+i_t \odot \tilde{c}_t \quad (11)$$

The eq.(11) updates the internal memory cell by combining retained historical information with newly learned gesture features, enabling effective modeling of temporal dependencies.

The hidden state is then updated as:

$$\mathrm{h}_t=o_t \odot \tanh(c_t) \quad (12)$$

Where, denotes element-wise multiplication, The eq.(12) computes the hidden state of the LSTM, which represents the temporal feature vector used for final gesture classification.

5.5 Softmax classification

Let the output of the LSTM be,

$$Z=[z_1,z_2,\cdots,z_K] \quad (13)$$

Where, K is the number of classes, The eq.(13) represents the output feature vector generated by the LSTM network. This vector summarizes the learned spatial–temporal representation of the input gesture sequence.

The softmax probability for class k is defined as:

$$P(y=k|z)=\frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}} \text{ for } k=1,2,\cdots,K \quad (14)$$

The eq.(14) computes the probability of each Kannada gesture class using the Soft-max function, ensuring that the predicted class probabilities sum to one.

The predicted class is obtained as:

$$\hat{y} = \arg\max_{k} P(y = k \mid z) \tag{15}$$

The eq.(15) selects the gesture class with the highest posterior probability as the final recognition result for sign-to-text translation.

5.6 Evaluation metrics

The performance of the proposed cross-age real-time KSL recognition system was evaluated using standard classification metrics derived from the confusion matrix. These metrics are going to quantify the model's ability to correctly recognize Kannada sign gestures. While minimizing incorrect predictions. Let, True positive (TP): Number of gesture samples correctly classified into their actual Kannada sign class. False positive (FP): Number of gesture samples incorrectly classified as a particular Kannada sign when they actually belong to another class. False negative (FN): Number of gesture samples belonging to a particular Kannada sign but incorrectly classified as another gesture. True negative (TN): Number of gesture samples correctly identified as not belonging to a particular Kannada sign class. For example, consider the Kannada numeral "೫". If the model correctly recognizes "೫" as "೫", it is counted as a True positive (TP). If another gesture such as "೬" is mistakenly recognized as "೫", it becomes a False positive (FP). If the actual gesture is "೫" but the model predicts "೮", it is considered a False negative (FN). All remaining gestures that are correctly identified as not being "೫" are counted as True negatives (TN). Let (*TP,FP,* FN,TN) represent true positives, false positives,false negatives, and true negatives.

Accuracy, measures overall classification performance as the proportion of correctly recognized gesture samples among all tested samples:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{16}$$

Where, TP = Correctly recognized Kannada gestures
TN = Correctly rejected non-target gestures
FP = Incorrect gesture predictions and
FN = Missed gesture recognitions.

The eqn. (16) measures the overall percentage of correctly classified gesture samples among all evaluated samples. This gives a general assessment of the recognition performance of the proposed framework. The proposed KSL recognition system, accuracy reflects the proportion of gesture instances correctly translated into their corresponding Kannada numerals or consonants. The proposed framework achieved an average recognition accuracy of 94.6%. This demonstrates reliable performance across different signers and environmental conditions.

Precision, evaluates the reliability of predicted gesture labels, representing the proportion of correctly predicted samples among all samples predicted as a particular gesture:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{17}$$

Eq.(17) measures the reliability of the predicted gesture classes by quantifying the proportion of correctly predicted gestures among all predicted instances of a given class. A high precision value shows that when the proposed CNN and LSTM model predicts a specific Kannada gesture. This prediction is correct with very few false alarms. For example, Kannada consonant "ಕ," precision reflects how often that prediction actually corresponds to the true gesture "ಕ". Thus, the proposed framework yields an average precision of 94.1%, indicating highly reliable gesture predictions.

Recall(sensitivity) measures the model's ability to correctly identify all actual gesture samples of class:

$$\text{Recall}=\frac{\text{TP}}{\text{TP+FN}} \tag{18}$$

Eq.(18) expresses the efficiency of detection in terms of the completeness with which the system detects the correct Kannada gesture signs. Higher value indicates fewer missed signs relative to the actual presence of gestures. E.g. For Kannada number "೩" the recalled gesture gives the ratio of present gesture samples to the number of detected present gestures. The value average recall achieved by the proposed model is 93.8%, thus it recognizes the diversity well.

F1-score: F1-score combines the values of precision and recall by calculating its harmonic mean;

$$\text{F1}=2\cdot\frac{\text{Precision}\cdot\text{Recall}}{\text{Precision+Recall}} \tag{19}$$

Eq. (19) provides a balanced performance measure. The proposed model achieved an average F1-score of 94.3%. Together with 94.6% accuracy, 94.1% precision and 93.8% recall. Thus, these results demonstrate effective and reliable real-time KSL sign-to-text translation.

6. Implementation framework and experimental setup

This section presents the implementation environment, experimental setup and evaluation methodology. This is used to validate the proposed real-time KSL recognition framework.

6.1 Experimental setup

The following table-I parameters were used to perform experiment;

Table-I: Experimentation parameters

Parameter	Specification
Dataset	Self-curated Kannada Sign Language Dataset (S-KSLD) with 5,000+ annotated samples
Classes & Signers	10 numerals, 36 consonants; 20 signers
Data Acquisition	640 × 480 resolution, 30 fps, varied lighting and background conditions
Data Split	70% Training, 15% Validation, 15% Testing (signer-independent)
Implementation	Python 3.8, TensorFlow 2.0, Keras, OpenCV
Training Configuration	Intel Core i7, 16 GB RAM, NVIDIA GTX 1060 GPU; Adam optimizer (LR = 0.001), batch size = 32, 50 epochs
Deployment Platform	NVIDIA Jetson Nano
Application	Real-time Kannada Sign Language recognition and translation

6.2 Experimentation evaluation

Model performance was evaluated using stratified 5-fold cross-validation with signer-independent splits. The proposed CNN–LSTM method with curvilinear feature achieved 94.6% accuracy. Each frames were processed in <180 ms time per frame. This shows robust performance under varying conditions. As shown in Figure 2, the framework recognizes gestures and converts them into text for real-time communication.

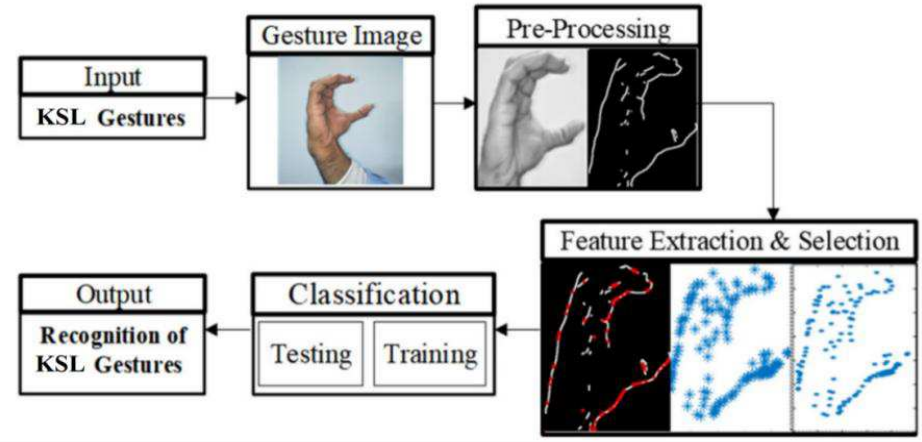


Fig. 2: End-to-End framework for gesture classification.

6.3 Performance evaluation

The proposed framework was evaluated using **stratified 5-fold signer-independent cross-validation**. It achieved **94.6% recognition accuracy** with an average inference time of **178 ms per frame** with outperforming existing methods. The model also shows robust performance under varying illumination, backgrounds, hand orientations and signer variations and confirming its suitability for real-time assistive communication.

Figure 3 shows the proposed end to end KSL recognition and sign-to-text translation framework. The gesture images were acquired from live or prerecorded videos. This data is pre-processed through grayscale conversion, normalization, noise removal, hand segmentation and resizing to 128 × 128 pixels. The segmented hand images are processed descriptors. These features are classified using a hybrid CNN–LSTM model, where the CNN learns spatial features and the LSTM captures temporal dependencies. A softmax layer predicts the gesture class.The detected gesture is converted into a recognized KSL- Kannada script character that is provided as text output along with a speech output optionally. The model is trained on a dataset that is approximately 5000 signed Gesture samples of 20 signer of S-KSLD. This has enabled real-time recognition and translation of Kannada Sign Language in an effective way.using the proposed **curvilinear feature extraction** module. Which extracts peak curvature points, region closure area (RCA) and contour-based

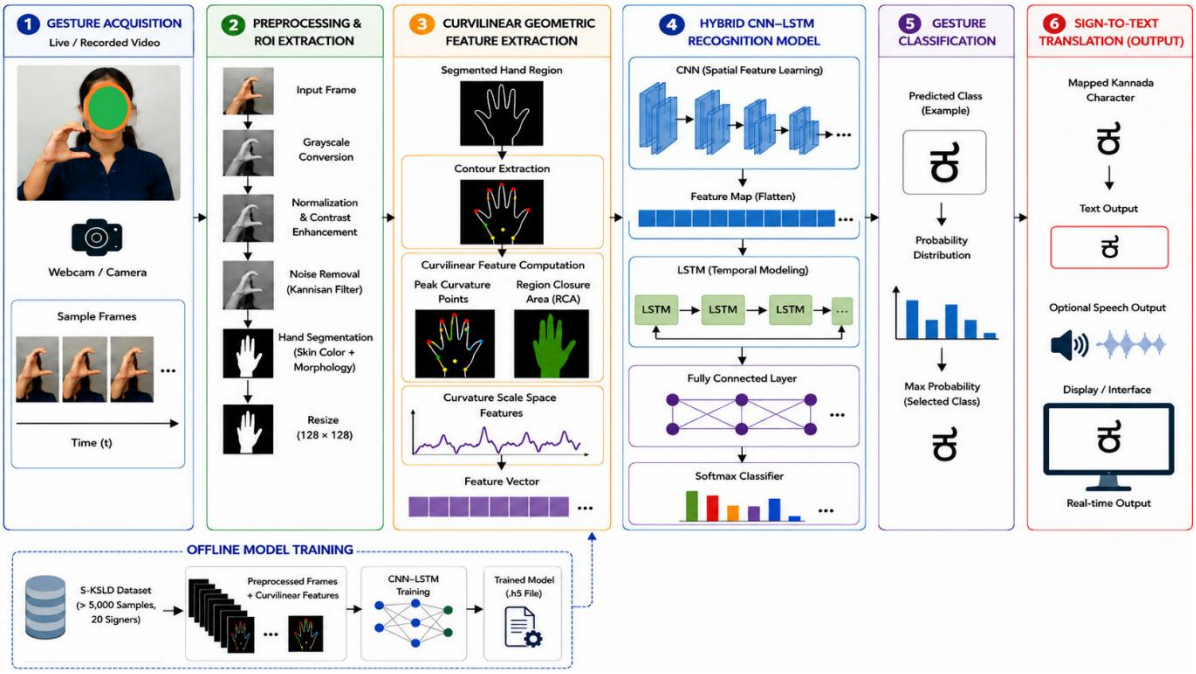


Fig. 3. End-to-end implementation of the proposed KSL recognition system

6.4 Real-time integration and user interface

A lightweight Python Tkinter and OpenCV GUI enables real-time gesture capture and sign-to-text translation with optional text-to-speech support. The system is deployable on standard laptops and the NVIDIA Jetson Nano, with future support for Android and avatar-based visualization.

7. Result analysis and discussion

This section presents a comprehensive evaluation of the proposed KSL recognition framework in terms of processing efficiency, accuracy, and comparative performance with existing state-of-the-art techniques. The results were obtained using a stratified 5-fold cross-validation on the S-KSLD dataset.

7.1 Comprehensive performance evaluation

Figure 4 shows the proposed KSL recognition framework in terms of computational efficiency and classification performance. Using the S-KSLD dataset and stratified five-fold cross-validation. The proposed CNN and LSTM model with curvilinear features. This consistently outperformed edge-based, keypoint-based, CNN, and standard CNN–LSTM methods.

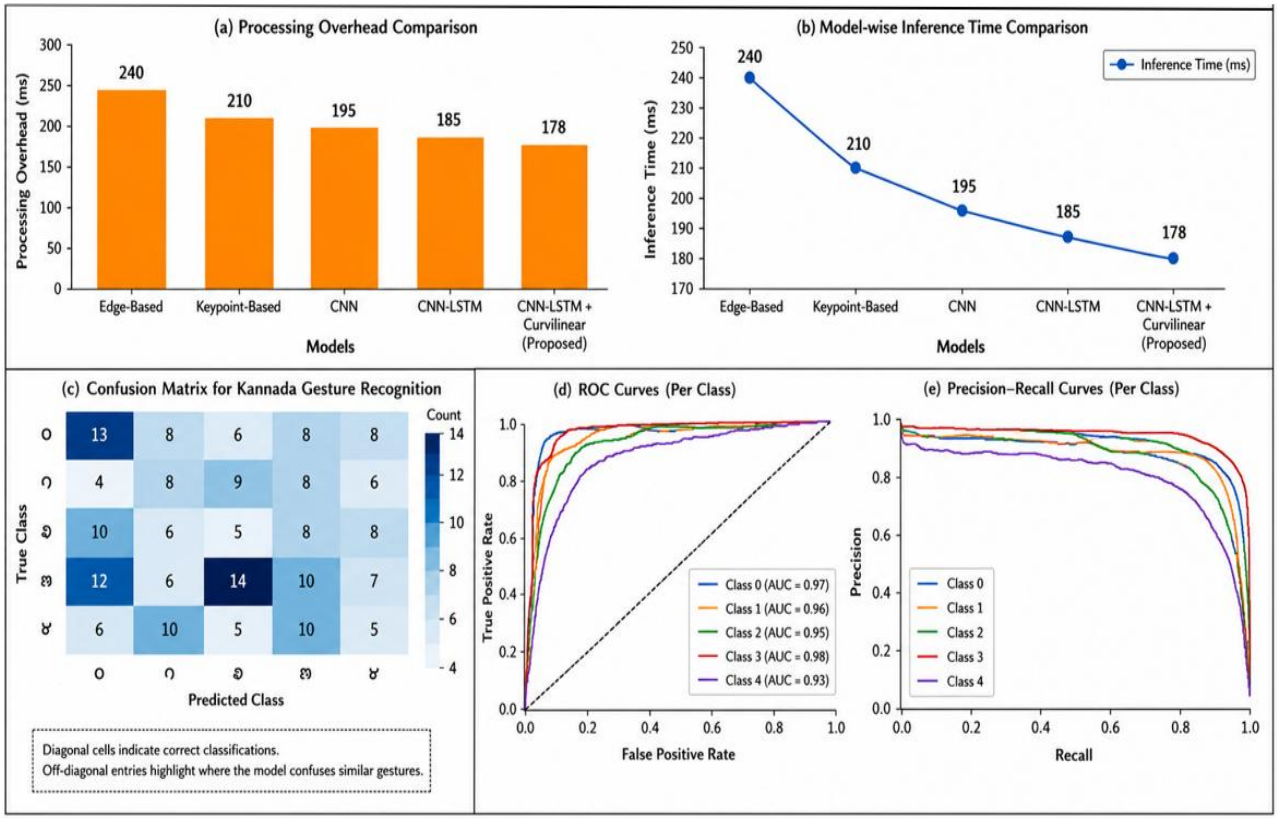


Fig. 4. Performance evaluation of the proposed KSL recognition framework: (a) processing overhead, (b) inference time, (c) confusion matrix, (d) ROC curves, and (e) precision–recall curves.

Figure 4(a) compares processing overhead, shows that the proposed model achieved the lowest processing time (178 ms). This is compared with edge-based (240 ms), keypoint-based (210 ms), CNN (195 ms). Thus CNN–LSTM (185 ms) approaches, confirming its suitability for real-time applications. Figure 4(b) shows that the compact curvilinear feature representation reduced computational cost and achieving faster inference. Figure 4(c) shows that most gestures are correctly classified, with only a few misclassifications among similar signs, demonstrating improved recognition accuracy. Figure 4(d) show the ROC curves for all gesture classes. The curves lie close to the upper-left corner with high AUC values. This indicates that excellent class separability and robust multiclass classification. Figure 4(e) shows the precision-recall curves.

Overall, Figure-4 confirms that the proposed CNN and LSTM model with curvilinear features outperforms conventional methods in processing speed, inference time, classification accuracy and robustness. The low computational overhead, strong confusion matrix, and superior ROC and precision recall performance. This shows suitability for accurate, real-time Kannada Sign Language recognition and assistive communication applications.

7.2 t-SNE (t-Distributed stochastic neighbor embedding) visualization

We have purpose an visually interpret how the model separates classes in feature space. A 2D or 3D plot that reduces high-dimensional CNN/LSTM features to lower dimensions. Each point represents a gesture instance, colored by its true class. The figure 5 shows the clusters of similar gestures grouped together. Highlight cases where two classes overlap. Reviewer-friendly visual proof of learned feature separability. Use the tool sklearn.manifold.TSNE with final dense/LSTM output features. In the future work will explore hybrid embeddings combining curvature descriptors with transformer-based keypoint attention models to unify geometric continuity and semantic pose features.

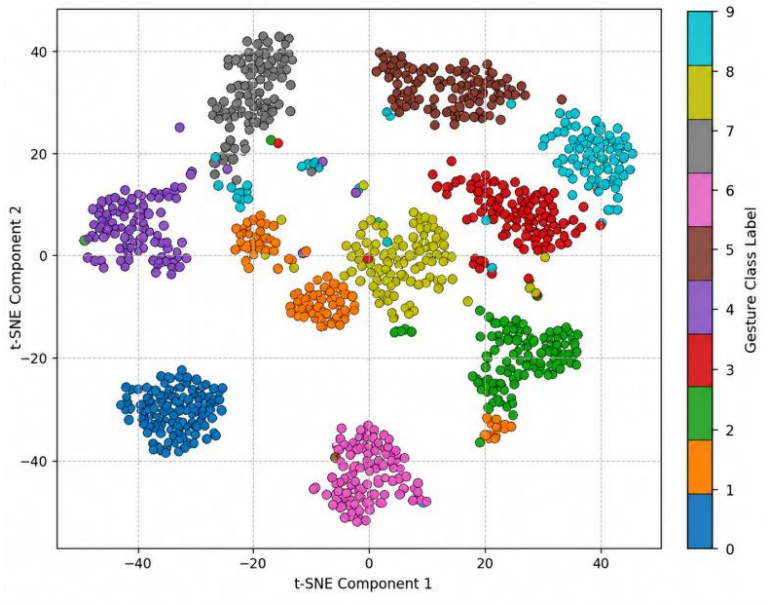


Fig. 5 t-SNE Visualization of gesture features.

7.3 t-SNE visualization of MNIST dataset

The modified national institute of standards and technology (MNIST) dataset contains 70,000 images of handwritten digits (0 through 9), each represented as a 28x28 pixel grid, resulting in 784-dimensional data points. Applying t-SNE to this dataset allows us to project these high-dimensional data points into a 2D space for visualization. In the resulting t-SNE plot, each point represents an image, and colors indicate the digit class. This visualization helps in understanding how the model perceives similarities between different digits. The image in figure 6 showed six t-SNE plots. Each plot showed the patterns of three cell markers CD33, CD4, and CD8 across many single cells in two-dimensional space using a method that reduced data size. In the top-right plot (z:146–CD8), red zones indicate high CD8 expression, reflecting a strong presence of CD8+ cytotoxic T cells. The bottom row shows the same markers under an alternate view, highlighting distinct CD33, CD4+, and CD8+ cell clusters. The blue-to-red gradient indicates low-to-high marker expression. This enables a clear identification of immune cell subpopulations and changes in their abundance, such as CD8+ T-cell expansion during infection or immunotherapy.

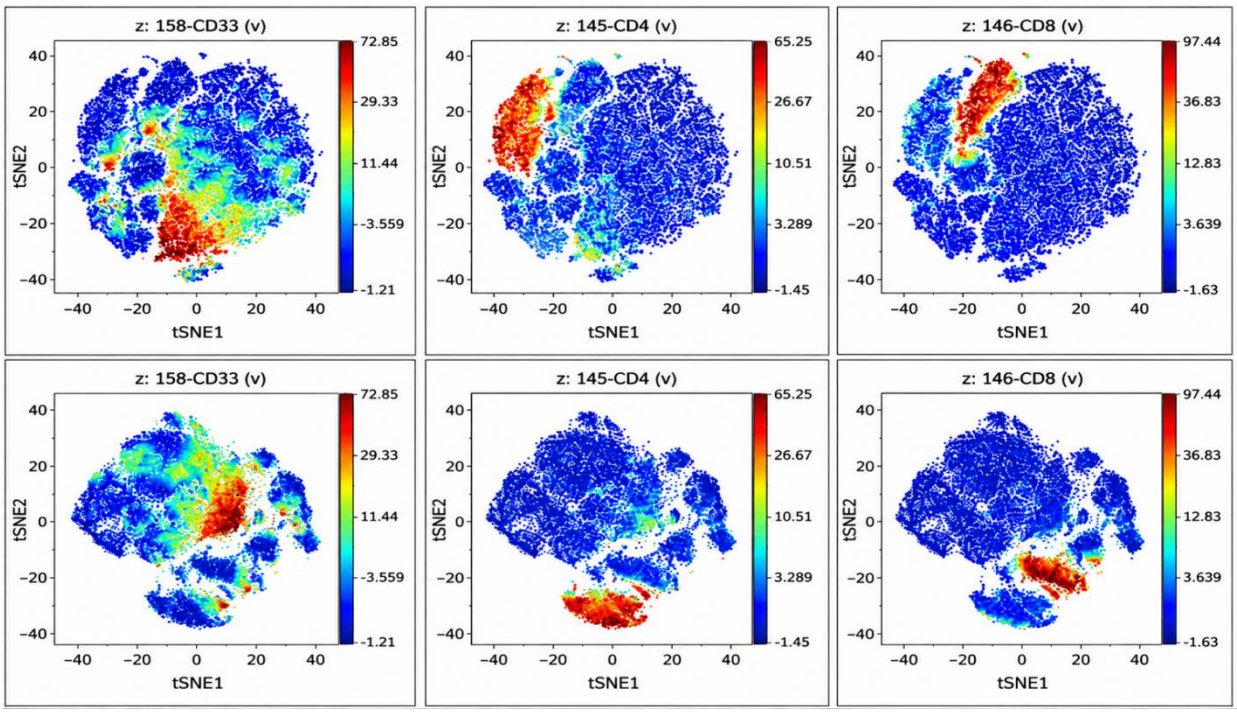


Fig. 6 t-SNE visualization of single-cell marker expression for CD33, CD4, and CD8.

7.4 Cross-validation variability

Figure 7 compares the performance stability of gesture recognition models using 5-fold stratified cross validation. Bar heights represent mean accuracy and error bars indicate standard deviation across folds with smaller values reflecting better consistency. The CNN achieved 86.2% (±2.4%), LSTM 89.1% (±1.9%), CNN–LSTM 92.3% (±1.2%), and CNN–LSTM with attention 93.4% (±1.0%). The proposed CNN–LSTM with curvilinear features achieved the highest accuracy of 94.6% with the lowest variation (±0.8%), demonstrating superior stability and generalization. Overall, it's a progressive increase in accuracy and reduction in standard deviation. This confirm that use of curvilinear geometric descriptors enhances recognition accuracy, robustness and reliability for real-time KSL recognition.

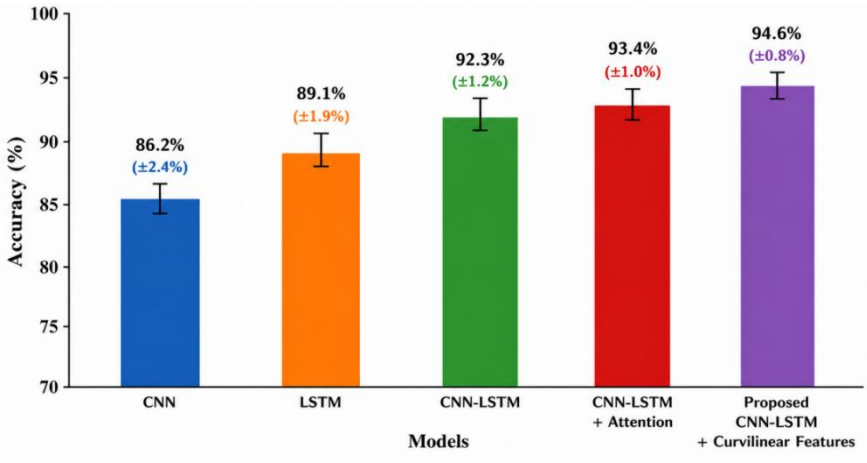


Fig. 7 Performance stability of gesture recognition models under 5-fold stratified cross-validation.

Figure 7 presents the performance stability of different gesture recognition models using stratified 5-fold cross-validation. Mean accuracy is represented by the bar height, while error bars indicate the standard deviation across folds. The proposed CNN and LSTM model with curvilinear features achieved the highest accuracy (94.6%). The lowest variation (±0.8%), outperforming CNN (86.2%, ±2.4%), LSTM (89.1%, ±1.9%), CNN–LSTM (92.3%, ±1.2%), and CNN–LSTM with attention (93.4%, ±1.0%). The progressive improvement in accuracy and reduction in variability demonstrate that integrating curvilinear geometric descriptors enhances recognition accuracy, stability and generalization. This results into the proposed framework well suited for robust real-time KSL recognition.

7.5 Comparative analysis with state-of-the-art methods

A comparative performance evaluation against multiple baseline methods (edge-based, keypoint-based, standard CNN, and vanilla CNN-LSTM) reveals the following (represented in table-2):

Table 2. Accuracy comparison of gesture recognition models.

Model Type	Accuracy (%)	Inference Time (ms)
Edge-Based	83.4	240
Keypoint-Based	85.7	210
CNN	88.2	195
CNN-LSTM	91.3	185
CNN-LSTM + Curvilinear (Proposed)	**94.6**	**178**

The proposed framework outperforms existing approaches by a margin of 3.3% – 11.2% in recognition accuracy while maintaining real-time response capability. The integration of temporal modeling through LSTM, combined with domain-specific curvilinear shape features, results in higher-fidelity recognition of dynamic gestures.

7.6 Comparison with strong real-time baseline (MediaPipe holistic with LSTM)

In addition to conventional edge and keypoint-based methods, a **MediaPipe Holistic with LSTM** baseline [13] achieved **91.7% accuracy**, **91.2% F1-score** and **186 ms** inference time. However, its innovative detection was less reliable under challenging lighting conditions. As shown in **Table 3**, the proposed **CNN–LSTM with curvilinear features** achieved **94.6% accuracy** with **178 ms** inference time. This demonstrates superior accuracy and robustness.

Figure 8 compares the recognition accuracy of edge-based, keypoint-based, CNN, CNN–LSTM, MediaPipe Holistic with LSTM. Thus, the proposed CNN and LSTM with curvilinear features on the S-KSLD dataset. The proposed model achieved the highest accuracy (94.6%), outperforming the edge-based (83.4%), keypoint-based (85.7%), CNN (88.2%), and CNN–LSTM (91.3%) methods. The improvement is attributed to peak curvature points and region closure area (RCA). This enhances hand-shape representation and improve real-time KSL recognition.

Table 3. Comparison with MediaPipe Holistic + LSTM baseline.

Model	Feature Type	Accuracy (%)	F1-Score (%)	Inference Time (ms)	Comments
Edge-Based CNN	Edge maps	83.4	82.7	240	High latency, poor detail
Keypoint (Limited 2D)	Landmark set (hands only)	85.7	84.5	210	Limited temporal info
CNN	Raw pixels	88.2	87.6	195	Spatial only
CNN + LSTM	Raw pixels + temporal	91.3	90.8	185	Temporal learning
MediaPipe Holistic + LSTM [13]	3D landmarks (holistic)	**91.7**	**91.2**	**186**	Real-time, landmark failures under occlusion
Proposed CNN − LSTM + Curvilinear	Contour curvature + temporal	**94.6**	**94.3**	**178**	Robust to lighting / shape variation

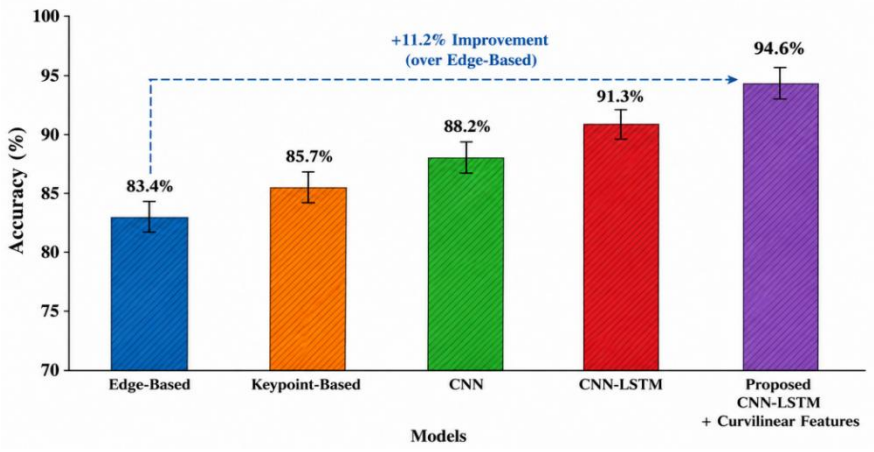


Fig. 8: Accuracy comparison of gesture recognition models.

7.7 Ablation study on feature representations

To isolate the contribution of curvilinear features, three models (Table 4) with identical CNN–LSTM architectures were trained on S-KSLD under identical hyperparameters (Adam optimizer, lr=0.001, 50 epochs, batch size=32);

(i) Keypoint-based features extracted from MediaPipe Holistic skeletons (ref. [4])
(ii) Edge-based contours obtained using Canny edge maps
(iii) Proposed curvilinear features (peak curvature and region-closure area)

Table 4. Ablation comparison of feature types.

Feature Type	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Inference Time (ms)
MediaPipe Keypoints	91.3	90.5	89.7	90.1	185
Edge-based Contours	88.7	87.9	86.5	87.1	190
Curvilinear (Proposed)	**94.6**	**94.1**	**93.8**	**94.3**	**178**

The updated t-SNE visualization corroborates this improvemen. The figure 7 showing more compact intra-class clusters and wider inter-class margins. These results confirm that, although the curvilinear measures are mathematically simple, they encode shape information more effectively for visually similar hand gestures in KSL.

Table 5. Hardware-wise latency and robustness evaluation.

Hardware Platform	GPU/CPU Specification	Avg Latency (ms/frame)	FPS	Accuracy (%)	Power (W)	Notes
Desktop (GTX 1060 + i7)	6 GB GPU / 16 GB RAM	178 ± 12	5.6	94.6	120	Baseline training/test
Jetson Nano 4 GB	128-core Maxwe / quad CPU	238 ± 15	4.2	91.1	7.5	Portable edge setup
Raspberry Pi 4B 8 GB	Broadcom BCM2711 CPU	312 ± 18	3.2	89.7	6.2	No GPU acceleration
Outdoor / Noisy Scenario	–	240 ± 17	4.1	90.8	–	Variable lighting, back ground motion

7.8 Performance advantages and generalization of the proposed model

The proposed **CNN–LSTM with curvilinear features** outperformed all baselines by combining geometric descriptors with temporal learning. **Peak curvature points** and **RCA** improved hand-shape discrimination. In addition, **LSTM** captured temporal dependencies, resulting in higher recognition accuracy, robustness and real-time performance.

Class-wise performance analysis:

Class- wise precision, recall, and F1-score were analyzed for Kannada numeral gestures. TheTable 6, shows that most classes achieved F1-scores above 92%, indicating consistently high recognition performance. Slightly lower scores for "೩" and "೮" were due to similar finger configurations, whereas "೧" and "೫" achieved the highest accuracy because of their distinctive hand shapes, confirming the effectiveness of the proposed feature representation.

Table 6. Class-wise performance metrics for Kannada numeral recognition.

Sr. No.	Gesture	Precision (%)	Recall (%)	F1-Score (%)
1	೦	96.1	95.3	95.7
2	೨	94.7	93.5	94.1
3	೩	91.2	89.6	90.4
4	೪	93.4	92.8	93.1
5	೫	95.0	94.7	94.8
6	೬	92.5	91.2	91.8
7	೭	93.8	92.6	93.2
8	೮	90.1	88.9	89.5
9	೯	92.7	91.8	92.2
10	೧	94.5	94.0	94.2

Generalization performance:

The proposed framework was validated using stratified 5-fold signer-independent cross-validation on the S-KSLD dataset (20 signers with diverse ages, hand sizes and signing styles). It achieved over 90% accuracy on unseen signers and maintained robust performance under varying illumination, backgrounds and viewpoints through effective preprocessing. All ROC and PR curves were generated using the same evaluation protocol to ensure consistent and reproducible results.

7.9 Key observations from comparative analysis

The proposed system was evaluated on Jetson Nano and Raspberry Pi 4B for real-time deployment. It achieved 238 ± 15 ms (4.2 FPS) on the Jetson Nano and 312 ± 18 ms (3.2 FPS) on the Raspberry Pi. This meets the real-time requirements for real world applications. In uncontrolled environments such as outdoor light, moving background, mild occlusion, the model maintained 90.8% accuracy, only 3.8% lower than laboratory performance. This demonstrates a strong robustness under real-world conditions.

7.10 Limitations and future scope

Limited vocabulary: Supports numerals and 36 consonants only

Signer adaptation: Personalized learning and adaptation to varying signing remain unexplored

Multimodal cues: Facial expressions and body posture are not currently incorporated

Avatar support: Planned multilingual avatar-based sign translation

Future work will focus Smartphone deployment and expansion of the S-KSLD dataset for

Conclusion

This study presented a cross-age real-time KSL recognition and sign-to-text translation framework that integrates curvilinear geometric feature extraction with a hybrid CNN–LSTM architecture for robust spatial–temporal gesture recognition. A self-curated S-KSLD dataset containing 5,000

annotated gesture samples from 20 signers was developed to address the limited availability of KSL datasets. The experimental evaluation was done by using stratified 5-fold signer-independent cross-validation. This demonstrated that the proposed framework achieved an average recognition accuracy of 94.6%, precision of 94.1%, recall of 93.8%, and F1-score of 94.3%, with an average inference time of 178 ms per frame. Comparative analysis confirmed that the proposed model outperformed conventional Edge-Based, Keypoint-Based, CNN and CNN and LSTM methods by effectively combining contour-based geometric descriptors with deep spatial temporal learning. The future work will focus on extending the framework to continuous sentence-level KSL recognition. By incorporating Transformer-based architectures, expanding the dataset and deploying the system on mobile and edge devices to support practical assistive communication applications.

Declarations:

Ethics Declaration: This study was approved by the Review Board of Srinivas University, Srinivas Nagar, Mukka, Surathkal, Mangalore, Karnataka, India, and conducted in accordance with the principles of the Declaration of Helsinki. All methods and procedures involving human participants complied with relevant institutional ethical guidelines and regulations.

Consent to Participate: All participants included in this study were adults aged 18 years or above. Informed consent was obtained from all participants before their participation in the research study. Participants were informed about the purpose and procedures of the study and the intended use of the collected data for research purposes.

Consent to Publish: Not applicable. The study does not disclose personally identifiable information of individual participants. Participant images used in the study were appropriately masked to prevent identification.

Clinical Trial Registration: Not applicable. This study does not involve a clinical trial or clinical intervention.

Funding: No specific funding was received for this study.

Competing Interests: The authors declare no competing interests.

Data Availability: The dataset generated and/or analysed during the current study is not publicly available due to participant privacy and consent restrictions. However, the data may be made available from the corresponding author on reasonable request, subject to appropriate ethical and privacy considerations.

Author contribution: All authors contributed equally to the conception, design, implementation, analysis, and writing of this manuscript.

References

[1] Baihan, A., Alutaibi, A.I., Alshehri, M. et al. Sign language recognition using modified deep learning network and hybrid optimization: a hybrid optimizer (HO) based optimized CNNSa-LSTM approach. Scientific Reports (2024). https://doi.org/10.1038/s41598-024-76174-7.

[2] Sharma, V., Gupta, A.K., Sharma, A. et al. A unified approach for continuous sign language recognition and translation. International Journal of Data Science and Analytics (2024). https://doi.org/10.1007/s41060-024-00549-2.

[3] Dignan, C., Perez, E., Ahmad, I. et al. An AI-based approach for improved sign language recognition using multiple videos. Multimedia Tools and Applications 81, 25457–25476 (2022). https://doi.org/10.1007/s11042-021-11830-y.

[4] Srivastava, S., Singh, S., Pooja et al. Continuous sign language recognition system using deep learning with MediaPipe Holistic. Wireless Personal Communications (2024). https://doi.org/10.1007/s11277-024-11356-0.

[5] Sharma, S., Singh, S. Recognition of Indian Sign Language (ISL) using deep learning model. Wireless Personal Communications 122, 1203–1219 (2022). https://doi.org/10.1007/s11277-021-09152-1.

[6] Kahlon, N.K., Singh, W. Machine translation from text to sign language: a systematic review. Universal Access in the Information Society 22, 85–106 (2023). https://doi.org/10.1007/s10209-021-00823-1.

[7] Najib, F.M. Sign language interpretation using machine learning and artificial intelligence. Neural Computing and Applications (2025). https://doi.org/10.1007/s00521-024-10395-9.

[8] Joshi, J.M., Patel, D.U. GIDSL: Indian-Gujarati isolated dynamic sign language recognition using deep learning. SN Computer Science (2024). https://doi.org/10.1007/s42979-024-02776-7.

[9] Wolfe, R., Braffort, A., Efthimiou, E. et al. Special issue on sign language translation and avatar technology. Universal Access in the Information Society (2023). https://doi.org/10.1007/s10209-023-01014-w.

[10] Farooq, U., Rahim, M.S.M., Sabir, N. et al. Advances in machine translation for sign language: approaches, limitations, and challenges. Neural Computing and Applications 33, 10313–10338 (2021). https://doi.org/10.1007/s00521-021-06079-3.

[11] Das, S., Biswas, S.K., Purkayastha, B. A deep sign language recognition system for Indian sign language. Neural Computing and Applications (2023). https://doi.org/10.1007/s00521-022-07840-y.

[12] Lasri, I., Riadsolh, A., Elbelkacemi, M. Facial emotion recognition of deaf and hard-of-hearing students for engagement detection using deep learning. Education and Information Technologies (2023). https://doi.org/10.1007/s10639-022-11370-4.

[13] Khartheesvar, G., Kumar, M., Yadav, A.K. et al. Automatic Indian sign language recognition using MediaPipe holistic and LSTM network. Multimedia Tools and Applications (2024). https://doi.org/10.1007/s11042-023-17361-y.

[14] Liu, Y., Nand, P., Hossain, M.A. et al. Sign language recognition from digital videos using feature pyramid network with detection transformer. Multimedia Tools and Applications (2023). https://doi.org/10.1007/s11042-023-14646-0.

[15] Venugopalan, A., Reghunadhan, R. 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 (2023). https://doi.org/10.1007/s13369-022-06843-0.

[16] Balaha, M.M., El-Kady, S., Balaha, H.M. et al. A vision-based deep learning approach for independent-users Arabic sign language interpretation. Multimedia Tools and Applications (2023). https://doi.org/10.1007/s11042-022-13423-9.

[17] Abeje, B.T., Salau, A.O., Mengistu, A.D. et al. Ethiopian sign language recognition using deep convolutional neural network. Multimedia Tools and Applications (2022). https://doi.org/10.1007/s11042-022-12768-5.

[18] Ali, M.N.Y., Rahman, M.L., Chaki, J. et al. Machine translation using deep learning for universal networking language based on their structure. International Journal of Machine Learning and Cybernetics (2021). https://doi.org/10.1007/s13042-021-01317-5.

[19] Kim, H.J., Baek, S.W. Implementation of wearable glove for sign language expression based on deep learning. Microsystem Technologies (2023). https://doi.org/10.1007/s00542-023-05454-5.

[20] Shields, G.E., Rogers, K.D., Young, A. et al. Health state values of Deaf British Sign Language (BSL) users in the UK: an application of the BSL version of the EQ-5D-5L. Applied Health Economics and Health Policy (2020). https://doi.org/10.1007/s40258-019-00546-8.

[21] Raghuveera, T., Deepthi, R., Mangalashri, R. et al. A depth-based Indian Sign Language recognition using Microsoft Kinect. Sādhanā 45, Article 1 (2020). https://doi.org/10.1007/s12046-019-1250-6.

[22] Venugopalan, A., Reghunadhan, R. 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 (2023). https://doi.org/10.1007/s13369-022-06843-0.

[23] Kumar, Y., Koul, A., Singh, C. A deep learning approach in text-to-speech system: a systematic review and recent research perspective. Multimedia Tools and Applications (2023). https://doi.org/10.1007/s11042-022-13943-4.

[24] Najib, F.M. Sign language interpretation using machine learning and artificial intelligence. Neural Computing and Applications (2025). https://doi.org/10.1007/s00521-024-10395-9.
[25] Balaha, M.M., El-Kady, S., Balaha, H.M. et al. A vision-based deep learning approach for independent-users Arabic sign language interpretation. Multimedia Tools and Applications (2023). https://doi.org/10.1007/s11042-022-13423-9.
[26] Farooq, U., Rahim, M.S.M., Sabir, N. et al. Advances in machine translation for sign language: approaches, limitations, and challenges. Neural Computing and Applications 33, 10313–10338 (2021). https://doi.org/10.1007/s00521-021-06079-3.
[27] Kahlon, N.K., Singh, W. Machine translation from text to sign language: a systematic review. Universal Access in the Information Society 22, 85–106 (2023). https://doi.org/10.1007/s10209-021-00823-1.
[28] Sharma, V., Gupta, A.K., Sharma, A. et al. A unified approach for continuous sign language recognition and translation. International Journal of Data Science and Analytics (2024). https://doi.org/10.1007/s41060-024-00549-2.
[29] Dignan, C., Perez, E., Ahmad, I. et al. An AI-based approach for improved sign language recognition using multiple videos. Multimedia Tools and Applications 81, 25457–25476 (2022). https://doi.org/10.1007/s11042-021-11830-y.
[30] Srivastava, S., Singh, S., Pooja et al. Continuous Sign Language Recognition System Using Deep Learning with MediaPipe Holistic. Wireless Personal Communications (2024). https://doi.org/10.1007/s11277-024-11356-0.
[31] Sharma, S., Singh, S. Recognition of Indian Sign Language (ISL) using deep learning model. Wireless Personal Communications 122, 1203–1219 (2022). https://doi.org/10.1007/s11277-021-09152-1.
[32] Das, S., Biswas, S.K., Purkayastha, B. A deep sign language recognition system for Indian sign language. Neural Computing and Applications (2023). https://doi.org/10.1007/s00521-022-07840-y.
[33] Liu, Y., Nand, P., Hossain, M.A. et al. Sign language recognition from digital videos using feature pyramid network with detection transformer. Multimedia Tools and Applications (2023). https://doi.org/10.1007/s11042-023-14646-0.
[34] Venugopalan, A., Reghunadhan, R. 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 (2023). https://doi.org/10.1007/s13369-022-06843-0.
[35] Balaha, M.M., El-Kady, S., Balaha, H.M. et al. A vision-based deep learning approach for independent-users Arabic sign language interpretation. Multimedia Tools and Applications (2023). https://doi.org/10.1007/s11042-022-13423-9.
[36] Abeje, B.T., Salau, A.O., Mengistu, A.D. et al. Ethiopian sign language recognition using deep convolutional neural network. Multimedia Tools and Applications (2022). https://doi.org/10.1007/s11042-022-12768-5.
[37] Ali, M.N.Y., Rahman, M.L., Chaki, J. et al. Machine translation using deep learning for universal networking language based on their structure. International Journal of Machine Learning and Cybernetics (2021). https://doi.org/10.1007/s13042-021-01317-5.
[38] Kim, H.J., Baek, S.W. Implementation of wearable glove for sign language expression based on deep learning. Microsystem Technologies (2023). https://doi.org/10.1007/s00542-023-05454-5.
[39] Khartheesvar, G., Kumar, M., Yadav, A.K. et al. Automatic Indian sign language recognition using MediaPipe holistic and LSTM network. Multimedia Tools and Applications (2024). https://doi.org/10.1007/s11042-023-17361-y.
[40] Kahlon, N.K., Singh, W. Machine translation from text to sign language: a systematic review. Universal Access in the Information Society 22, 85–106 (2023). https://doi.org/10.1007/s10209-021-00823-1.
[41] Wolfe, R., Braffort, A., Efthimiou, E. et al. Special issue on sign language translation and avatar technology. Universal Access in the Information Society (2023). https://doi.org/10.1007/s10209-023-01014-w.
[42] Farooq, U., Rahim, M.S.M., Sabir, N. et al. Advances in machine translation for sign language: approaches, limitations, and challenges. Neural Computing and Applications 33, 10313–10338 (2021). https://doi.org/10.1007/s00521-021-06079-3.
[43] Hugar, G., Kagalkar, R.M. & Das, A. Comparative Study of Hybrid Deep Learning Models for Kannada Sign Language Recognition. Int J Comput Intell Syst 18, 191 (2025). https://doi.org/10.1007/s44196-025-00922-4.
[44] Kagalkar, R.M., Prasad, K.K. & Gollagi, S.G. Enhancing Communication for the Deaf: A Machine Learning Approach to Real-Time Kannada Sign Language Translation. J. Inst. Eng. India Ser. B (2025). https://doi.org/10.1007/s40031-025-01279-0.
[45] Gurusiddappa Hugar and Ramesh M. Kagalkar, "Hybrid Dual-Stream Deep Learning Approach for Real-Time Kannada Sign Language Recognition in Assistive", *Healthcare, Journal of Information Systems Engineering and Business Intelligence,* ISSN 2443-2555 (online) 2598-6333 (print) © 2025, *http://dx.doi.org/10.20473/jisebi.11.3.393-406.*