Indonesian Journal of Electrical Engineering and Computer Science
Vol. 39, No. 2, August 2025, pp. 1337~1349
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v39.i2.pp1337-1349

A curvilinear-based approach for sign-to-text conversion of Kannada deaf sign language

Shantappa G Gollagi[1], Mahantesh Laddi[2], Suhas G K[3], Kalyan Devappa Bamane[4], Sulbha Yadav[5]
[1]Department of Computer Science and Engineering, S G Balekundri Institute of Technology, Belagavi, India
[2]Department of Computer Science and Engineering, Bharatesh Institute of Technology, Belagavi, India
[3]Department of Information Science and Engineering, Akshaya Institute of Technology, Tumakuru, India
[4]Department of computer Engineering, D Y Patil College of Engineering, Pune, India
[5]Depatment of Computer, Lokmanya Tilak College of Engineering, Navi Mumbai, India

Article Info

Article history:

Received Dec 17, 2024
Revised Mar 26, 2025
Accepted Jul 2, 2025

Keywords:

Contour detection
Curvilinear coding
Edge detection
Gaussian function
Kannada
Sign language

ABSTRACT

This research addresses the challenge of translating Kannada sign language into text to improve communication for the deaf community. Existing methods, primarily shape-based approaches, often fail to accurately imprisonment the complexity of hand gestures, leading to reduced translation accuracy. This study proposes a curvilinear-based approach that leverages peak curvature features and contour evolution techniques to overcome these limitations. This method enhances the recognition and interpretation of sign language gestures while reducing processing overhead. Experimental results demonstrate that the proposed system significantly outperforms traditional methods, achieving higher precision and recall rates. The enhanced system provides a reliable solution for improving accessibility and communication for the deaf community. This research represents a significant step toward developing more inclusive digital communication tools, with future work focused on real-time processing and extending the system to other regional sign languages.

Corresponding Author:

Shantappa G Gollagi
Department of Computer Science and Engineering, S G Balekundri Institute of Technology
Belagavi -590010, India
Email: shantesh1973@rediffmail.com

1. INTRODUCTION

This article presents a groundbreaking approach for translating Kannada sign language into text, aimed at enhancing communication for the deaf community. The research question is: “Can a curvilinear-based approach using peak curvature features and contour evolution improve the accuracy and efficiency of Kannada sign language translation compared to traditional shape-based methods?”. The hypothesis is that the curvilinear-based approach will enhance recognition accuracy and processing efficiency by preserving critical gesture details. The system addresses the critical challenge of bridging the communication gap by converting the complex gestures of Kannada sign language into easily understandable text, thereby promoting greater inclusivity and accessibility. The primary objective of this research is to develop a sophisticated translation system that leverages advanced techniques such as curvilinear tracing and peak curvature features to accurately recognize and translate sign language gestures with minimal processing overhead. This approach enhances the efficiency of translation while improving the understanding of spatial semantics in sign language, empowering the deaf community to communicate more seamlessly in a predominantly text-based world. Furthermore, this paper explores the current landscape of sign language recognition research, highlighting key studies and their limitations. Several notable works have laid the foundation for sign language recognition.

Durdi *et al.* [1] developed a technology-driven method combining convolutional neural networks (CNNs) with stochastic gradient descent algorithms for accurate hand-gestured sign language recognition. Mohsin *et al.* [2] explored the use of transfer learning algorithms in identifying American sign language (ASL) using deep learning approaches. They found InceptionV3 as the best model with a 96% accuracy rate, demonstrating the resilience of transfer learning and deep learning approaches. Aziz and Othman [3] retrospective explores sign language avatar systems' development over four decades, highlighting historical and current trends. Avina *et al.* [4], a new framework has been developed to convert English to ASL using a rolling average prediction mechanism and deep learning model.

In the research work [5] have developed a novel model using deep learning and attention techniques to improve sign language recognition for people with disabilities, achieving impressive accuracies of 98.9% and 97.6% on OkkhorNama: BdSL and MU hand images ASL datasets. Alsharif *et al.* [6] used the deep learning models, particularly ResNet-50, to improve ASL identification and promote accessible communication for hearing-impaired individuals. Abdullahi and Chamnongthai [7] explains the IDF-sign model, a spatial-temporal depth model that addresses challenges in sign language recognition due to erratic hand and body traits. Skedsmo [8], examines repair receipt customs in Norwegian sign language informal group conversations, focusing on self-healing.

Podder *et al.* [9], a system for identifying Arabic sign language from red, green, and blue (RGB) videos, using MobileNetV2-LSTM-SelfMLP and CNN backbones, achieving an accuracy rate of 87.69% and high precision, recall, F1 score, and specificity. Ji *et al.* [10] suggests a cost-effective data glove using inertial sensors to enhance communication between deaf and able-bodied individuals. The glove accurately recognizes 20 dynamic sign language gestures using four machine learning models, with 98.85% recognition accuracy compared to cutting-edge algorithms. Svendsen and Kadry [11] aims to bridge the communication gap in the deaf community by using image categorization techniques. Using machine learning models, it analyzes 24,300 photos of 27 NSL alphabet signs, achieving 99.9% accuracy and good computational efficiency. Paul *et al.* [12], explores sign language detection using deep learning and computer vision.

Ruiz *et al.* [13], deaf community is aiming to enhance their communication skills through sign language recognition. A system using transformer and bidirectional long-term memory models, with a 94.33% accuracy rate, is being developed, focusing on non-manual features. Dutta *et al.* [14], presents two-stage architecture for a patient aid system in medical settings for hand gesture recognition, addressing challenges like occlusion and illumination disparities. Kothadiya *et al.* [15], propose an approach to improve sign language recognition accuracy using explainable artificial intelligence (XAI) research. They propose an attention-based ensemble learning strategy, combining a self-attention model with ResNet50, achieving an accuracy of 98.20%. Woods and Rana [16], have explored the modeling of ASL using encoder-only transformers and key point data from human posture estimation. It demonstrates the accuracy of models with less than 100k learnable parameters, setting new benchmarks for ASL recognition with limited resources. Sreemathy *et al.* [17], Python-based system classifies 80 sign language terms using support vector machine (SVM) and you only live once v4 (YOLOv4), achieving 98.62% accuracy when integrated with media pipe.

Siddique *et al.* [18], creates an automatic detection system for Bangla sign language (BSL) to integrate the deaf community into mainstream society. It uses deep learning techniques, an edge device like Jetson Nano. Qahtan *et al.* [19] analyzes three Likert scales in a single fuzzy setting, focusing on real-time sign language recognition systems (SLRSs). The results show that the five-point scale outperforms the seven- and ten-point versions, providing better accuracy, use, and versatility. Shin *et al.* [20] addresses the novel multi-branch network combining transformers and convolutions that have been developed for Korean sign language (KSL) classification. The model achieved 89% accuracy for a 77-label KSL dataset and 98.30% accuracy for the lab dataset. Bora *et al.* [21], develops a machine-learning framework for recognizing assamese sign language, one of India's 22 modern languages, using a dataset of images. A feed-forward neural network achieves 99% accuracy, promoting inclusivity and accessibility. Liu *et al.* [22] discussed the DETR (detection transformer) approach aims to enhance sign language recognition accuracy for the deaf community using deep learning algorithms, specifically vision transformer. Hdioud and Tirari, [23], focuses on developing an Arabic sign language recognition (ArSLR) system using deep CNN architecture to improve communication for the deaf and mute community. The system achieved a 97% accuracy rate in ArSLR.

The novelty of our work lies in the use of a curvilinear-based approach, which effectively preserves intricate gesture details through peak curvature features and contour evolution. By improving the accuracy and efficiency of Kannada sign language translation, this research contributes to developing more effective communication tools and fostering greater inclusivity for the deaf community.

The paper is organized as follows: section 2 presents in detail, material and methods adopted. Section 3 details implementation considerations. In section 4 presents result and discussion. Finally, the paper concludes with a summary of its findings.

2. MATERIALS AND METHOD

The proposed system architecture follows a defined pipeline consisting of data acquisition, preprocessing, feature extraction, classification, and output generation.

2.1. System architecture

The system follows a structured process from input to output, ensuring precise recognition and translation. The overall process is illustrated in Figure 1.

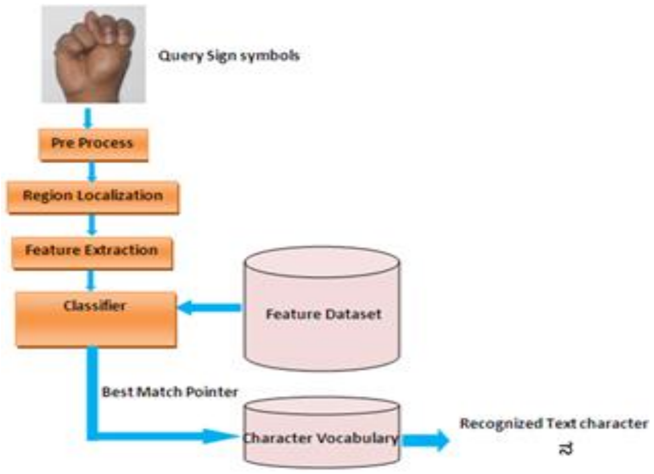


Figure 1. Model for translating sign language into text

The suggested model processes gesture inputs by standardizing dimensionality and data type. It uses a two-level binarization method for hand region localization, followed by feature extraction to capture key details like shape and movement. These features are matched against a database to classify and recognize the sign, which is then mapped to its corresponding Kannada word or character. For instance, a gesture for "Hello" is translated into "ಹಲೋ," facilitating accurate gesture-to-text conversion for Kannada communication.

2.2. Curvilinear feature representation

A leaping region-based approach is introduced for hand region representation, following three key phases. First, the input image undergoes edge detection and prediction using a descriptor, with the Canny operator applied to identify peak points based on maximum gradient magnitude after Gaussian filtering. Next, a forward-marching algorithm extracts the closed boundary contour from the detected edges. Finally, the curvature of this contour is analyzed, and significant peak-bound features are selected. The detected edge regions serve as a reference for contour delineation, ensuring precise boundary definition. Accurate contour detection relies on identifying true corner points while minimizing false positives. To maintain smooth boundary continuity, an effective corner localization method is necessary. A robust contour estimator helps approximate true corner points even in the presence of noise, ensuring a seamless and accurate contour representation for cue symbol analysis.

Leap forward tracing algorithm: the leap-forward tracing algorithm is explained step-by-step, highlighting its ability to detect edge regions accurately. Here's a detailed explanation of each step in the algorithm:

– Find initial edge pixel: the first step involves locating the starting point for contour tracing. We have to scan an image horizontally or vertically based on the provided edge information (which could indicate where the edge of an object starts). Identification of the initial edge pixel that marks the beginning of a contour as a result.

- Set seed pixel: to establish a reference point for tracing the contour. The initial edge pixel discovered in the first step is designated as the seed pixel. This pixel will serve as the reference point from which the contour tracing will begin. The seed pixel is set, and ready for the tracing process to start.
- Locate neighboring pixels: to identify the pixels that form the contour of the object. Starting from the seed pixel, the algorithm identifies the possible eight neighboring pixels. These neighbors are located in a 3×3 grid centered on the seed pixel. The tracing proceeds in an anti-clockwise direction, which helps in maintaining a consistent tracing path.
- Tracing order: to define the sequence in which the neighboring pixels are examined. The neighbors are checked in a specific order to ensure a continuous and coherent tracing of the contour. If '(x, y)' represents the coordinates of the seed pixel, the neighbors are examined in the following sequence: i) (x+1, y) - right, ii) (x+1, y+1) - bottom right, iii) (x, y+1) - below, iv) (x-1, y+1) - bottom left, v) (x-1, y) - left, vi) (x-1, y-1) - top left, vii) (x, y-1) - above, and viii) (x+1, y-1) - top right
- Update seed pixel: to advance the tracing process along the contour. As each neighboring pixel is examined according to the tracing order, any pixel found to be part of the contour becomes the new seed pixel. The steps (repetition) of locating neighboring pixels, following the tracing order, and updating the seed pixel are repeated iteratively. The algorithm continues until it returns to the initial seed pixel, completing the contour trace (Figure 2). The algorithm may involve additional parameters such as weights or orientation values, which help in determining the direction of tracing based on the image's content or specific application needs.

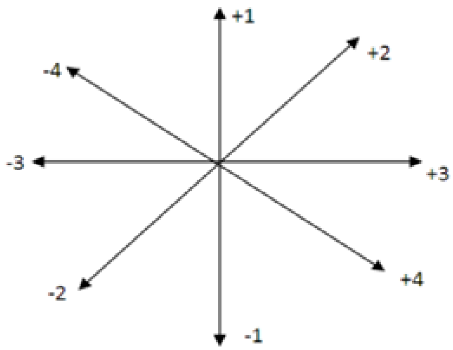


Figure 2. Orientation weight distribution for contour tracing

2.3. Peak curvature feature representation:

The peak curvature feature representation optimizes contour evolution tracing by reducing the high feature count, focusing on key curvature points for a more compact and efficient representation. This method consists of several essential steps. First, a convolution operation is applied between a standard Gaussian function and the curve, which is parameterized along a path. Next, the representation is designed to be resilient to transformations such as uniform scaling, rotation, and translation, ensuring robustness. Additionally, the curve is described at progressively abstract levels through a process known as "curve evolution," where convolution with a Gaussian function introduces smoothing, minimizing noise while preserving critical information. This refined approach maintains essential curve details while significantly lowering feature complexity, making it more effective and manageable for further analysis.

The convolution procedure between a conventional Gaussian function and the curve represented in a path-based parameter yields the new features. The standard deviation values of the Gaussian function fall between too small and too large, therefore when a curve is extracted, the resulting curves have planar noise. The translation, rotation, and uniform scaling of a curve do not affect this new representation. This trait, along with a few other significant ones, makes this new representation ideal for extracting a noise curve from an orientation. The following approach is utilized to derive a curve plane from the contour extracted from a given image. For a given set of contour coordinates (x(u), y(u)), the corresponding curve plane can be determined as follows:

$$k(u) = [\dot{x}(u) * \ddot{y}(u) - \dot{y}(u) * \ddot{x}(u)] \,/\, [(\dot{x}(u))^2 + (\dot{y}(u))^2]\text{^}(3/2) \tag{1}$$

This curvature is used from its representation in the curve plane of a planar curve. Formulas are simplified when they pertain to special parameterization scenarios. If w be the parameter denoting the length of the normalized arc, then,

$$k(w) = \dot{x}(w) * \ddot{y}(w) - \dot{y}(w) * \ddot{x}(w) \tag{2}$$

Given a planar curve,

$$\Gamma = \{ (x(w), y(w)) \mid w \in [0,1] \} \tag{3}$$

For an evolved form of that curve, where w is the parameter denoting the normalized arc length, the definition is,

$$\Gamma_\sigma = \{(X(u,\sigma), Y(u,\sigma)) | w\ \varepsilon[0,1]\} \tag{4}$$

Where,

$$X(u,\sigma) = x(u) \otimes g(u,\sigma) \qquad Y(u,\sigma) = y(u) \otimes g(u,\sigma) \tag{5}$$

$g(u,\sigma)$ describes the Gaussian width σ and defined as:

$$g(u,\sigma) = (1 / (\sigma\sqrt{(2\pi)})) * exp(-u^2 / (2\sigma^2)) \tag{6}$$

$g(u,\sigma)$ and $y(u,\sigma)$ are explicitly defined as follows:

$$X(u,\sigma) = \int from -\infty\ to\ \infty\ [x(v) * (1 / (\sigma\sqrt{(2\pi)})) * exp(-((u - v)^2) / (2\sigma^2))]\, dv \tag{7}$$

$$Y(u,\sigma) = \int from -\infty\ to\ \infty\ [y(v) * (1 / (\sigma\sqrt{(2\pi)})) * exp(-((u - v)^2) / (2\sigma^2))]\, dv \tag{8}$$

The curve plane of Γ σ is given by,

$$k(u,\sigma) = [Xu(u,\sigma) * Yuu(u,\sigma) - Xuu(u,\sigma) * Yu(u,\sigma)] / [(Xu(u,\sigma))^2 + (Yu(u,\sigma))^2]\hat{}(3/2) \tag{9}$$

The formula you've provided appears to be related to curvature computation in the context of curvilinear feature representation. Where, $k(u,\sigma)$: this represents the curvature at a point u with a smoothing parameter σ.
$x_u(u,\sigma)$ and $y_u(u,\sigma)$y: these denote the first derivatives of the x and y coordinates with respect to u.
$x_{uu}(u,\sigma)$ and $y_{uu}(u,\sigma)$: These denote the second derivatives of the x and y coordinates with respect to u.
The denominator $(x_u(u,\sigma)^2 + y_u(u,\sigma)^2)^{3/2}$ normalizes the curvature value.

This formula is used in this study curvilinear-based approach to detect and represent the curvature of gestures in sign language recognition. The critical points of curvature help preserve the intricate nuances of the gestures, which are vital for accurate translation from sign language to text, where:

$$Xu(u,\sigma) = \partial/\partial u\, [x(u) \otimes g(u,\sigma)] = x(u) \otimes gu(u,\sigma) \tag{10}$$

$$Xuu(u,\sigma) = \partial^2/\partial u^2\, [x(u) \otimes g(u,\sigma)] = x(u) \otimes guu(u,\sigma) \tag{11}$$

$$Yu(u,\sigma) = y(u) \otimes gu(u,\sigma$$

The following is an overview of the suggested region border curvature representation algorithm:
Step-1: open an image file.
Step-2: conversion to 2-level (1/0).
Step-3: extraction of contours via the forward march region.
Step-4: assessing curvature using,

$$K(u,\sigma) = [Xu(u,\sigma)Yuu(u,\sigma)] - Xuu(u,\sigma)Yu(u,\sigma)] / (Xu(u,\sigma)2 + Yu(u,\sigma)2)3/2$$

Step-5: proceed to evaluate curvature by smoothing it with a Gaussian factor (σ) variation.
Step-6: continue until all curvatures are smoothed.

Step-7: calculate a cutoff as,

$$k(u,\sigma) = \frac{X_u(u,\sigma)Y_{uu}(u,\sigma) - X_{uu}(u,\sigma)Y_u(u,\sigma)}{(X_u(u,\sigma)^2 + Y(u,\sigma)^2)^{3/2}}$$

T_h = (Max (peak)*k)/255, k is a factor in smoothening.
Step-8: if P (i,j) is less then Th, then set P(i,j) to zero; the choice is made based on the obtained threshold. Otherwise retain P (i,j) as it is , where P(i,j) is the obtained curvature element.

2.4. Enclosed region area characteristic

The proposed method presents a distinctive feature for the region-bound illustration of a cue symbol. In addressing spatial semanticity concerns, an additional feature is defined, known as the "closed bound area region." This feature characterizes the locked area under a curve [24], [25]. In this suggested approach, for the contour of the bounded region, areas with values classified as 'High' are considered as the region area. The density of the area filling, coupled with the derived curvature peaks, is utilized as a learning feature for an SVM.

2.5. Region bound area feature

This feature encapsulates information about the closed area under the curve, contributing to a more comprehensive representation of the spatial characteristics of the cue symbol [26], [27]. It serves as a valuable learning feature for machine learning models, particularly in the context of support vector machines.

$$A = \sum_{i=1}^{c} x(i,j)\ \forall x(i,j) \in C \tag{12}$$

The enclosed boundary contour is symbolized by C. If a coefficient is restricted within this contour, the area beneath the curve is determined as the total sum of the enclosed region. The classification model utilizes two key feature sets, (K, A), for training purposes. These features act as learning parameters for making classifications. The training dataset is represented as D ∈ (K_i,A_i), where K_i corresponds to the curvature characteristic of the region boundary, and A_i signifies the area-based attribute of the region boundary. This applies to each character from A–Z and numerical values from 0 to 9. These characteristics are regarded by the SVM system as learning knowledge parameters.

3. IMPLEMENTATION CONSIDERATION

The experimental setup involves a high-resolution camera system to capture Kannada deaf sign language gestures. A curvilinear algorithm processes the captured video data to recognize and interpret the sign patterns. This system is tested in a controlled environment with multiple lighting conditions to ensure robustness. The output is then converted into corresponding Kannada text using a specially developed software interface.

3.1. Data collection

The proposed dataset, essential for translating deaf sign language to Kannada, includes numerals (0-9) and 36 Vayanjanas, capturing the diversity of signing styles across different regions. Each element is represented with variations in handshapes, movements, and expressions.

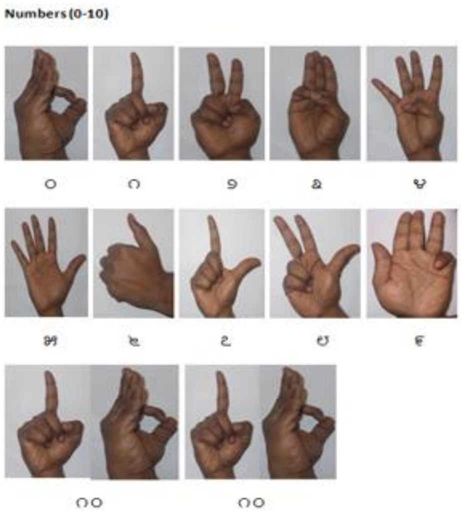
3.2. Training process

The training process utilizes a deep learning architecture tailored for sign language recognition, incorporating transfer learning to enhance gesture interpretation. The research setup includes high-resolution cameras and a motion capture system to capture detailed hand and body movements. A dedicated computing infrastructure with specialized graphics processing units (GPUs) supports efficient model training and real-time recognition.

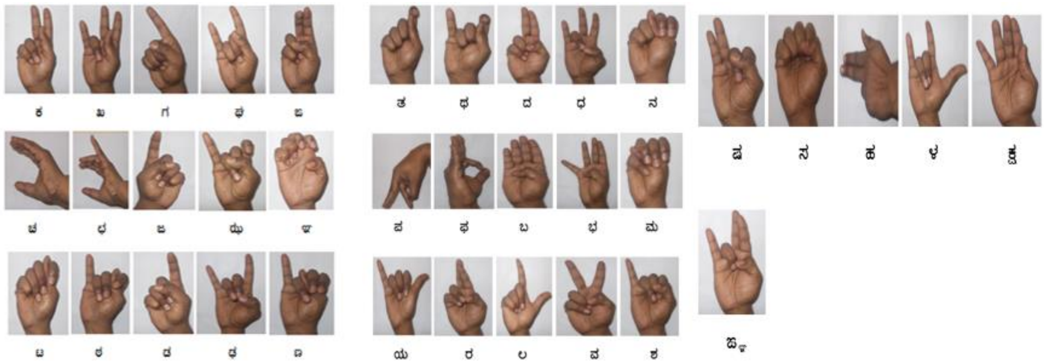
3.3. Evaluation metrics

Performance is measured using accuracy, precision, and recall, with cross-validation to ensure reliability. The experimental setup involved testing the system's ability to categorize signs using a dataset depicted in Figure 3, which includes numerals 0-9 in Figure 3(a) and 36 Vayanjanas in Figure 3(b).

The categorization performance, illustrated in Figures 4 to 7, showcases the system's effectiveness in image mapping and region prediction via thresholding logic for two-level realization. The transformation of the test sample into detected curvilinear regions demonstrates the advanced capability of the proposed system.



(a)



(b)

Figure 3. The dataset employed in system training (a) 0-9 Numerals and (b) 36-Vayanjanas in the Kannada language

Figure 4. Given test sample -1

Figure 5. processed 2-level images for the given test sample-1

Figure 6. Identification of curvilinear regions in the sample

Figure 7. Classification outcomes using the developed method

4. RESULTS AND DISCUSSION

The curvilinear feature representation performs better than other methods in converting Kannada deaf sign language into text in respect of precision, recall, efficiency, and adaptation to various styles of signing. For feature representation, computation time, and processing overhead detailed comparisons show that it significantly enhanced them. The authors intend to extend the dataset and develop tools for real-time translation and to help create an inclusive deaf-communicative environment. Quantitatively, the system's retrieval efficiency is measured by precision and recall rates:

$$Precision = \frac{Total\ relevant\ information}{Total\ images\ obtained} \tag{13}$$

$$Recall = \frac{Total\ relevant\ information}{Total\ relevant\ images\ available} \tag{14}$$

These metrics underscore the system's robust classification performance, further detailed in Table 1, which compares the curvilinear feature representation against traditional edge-based methods. Figure 8 presents a linear graph comparing true match counts across five test cue symbols (T1 to T5) for edge-shape, curvilinear, and ratio feature representations. Curvilinear features show consistently higher match counts, while edge-based features are lower, and the ratio remains stable with slight variations.

Table 1. Analysis of various test cue identifiers

Sr. No.	Test cue identifier	Recognition score		Exact match ratio
		Edge-based features	Curvilinear feature	
1	T1	2	4	4/5
2	T2	3	4	3/5
3	T3	2	3	4/5
4	T4	3	4	4/5
5	T5	3	3	3/5

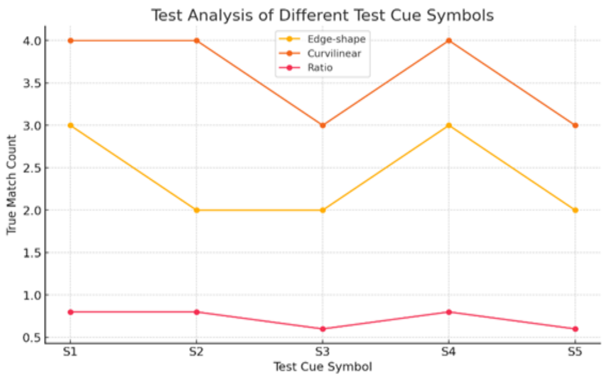


Figure 8. Analysis of various test cue identifiers

Table 2 provides the feature counts for test samples. Figure 9 compares feature counts for edge-shape and curvilinear representations. Edge-shape consistently identifies more features, as shown by the blue line, while curvilinear counts are lower (blue line), with significant disparities highlighted by the red line. The analysis includes a sign language-to-written character conversion using extracted features evaluating the approach using computation time, receiver operating characteristic (ROC) analysis, and processing overhead.

Table 2. Comparison of the two developed approaches' feature counts

Sr. No.	Test cue identifier	Feature count		Difference
		Edge-based feature count	Curvilinear feature count	
1	T1	685	74	611
2	T2	745	82	663
3	T3	571	65	506
4	T4	704	70	634
5	T5	667	67	600

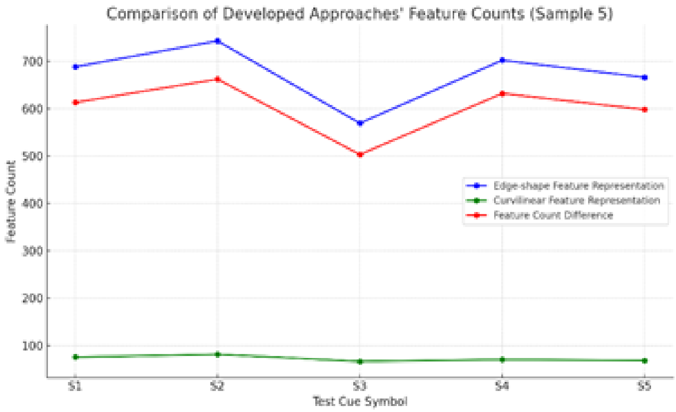


Figure 9. Comparison of developed approaches features counts

The curvilinear approach demonstrates superior recognition rates, supported by comparative data from Tables 3 and 4. Figure 10 shows that the computation time is reduced by 30 seconds compared to traditional shape-oriented coding. The shorter search time is due to fewer processing coefficients, while the extra time for feature extraction during coding minimally impacts recognition, as it is primarily designed for the training phase.

Table 3. Comparison of processing time analysis

Sr. No.	Observation	Computation time (sec)	
		Edge-based	Curvilinear
1	Obser-1	23.1267	16.6689
2	Obser-2	19.5259	14.8822
3	Obser-3	25.6455	17.0086

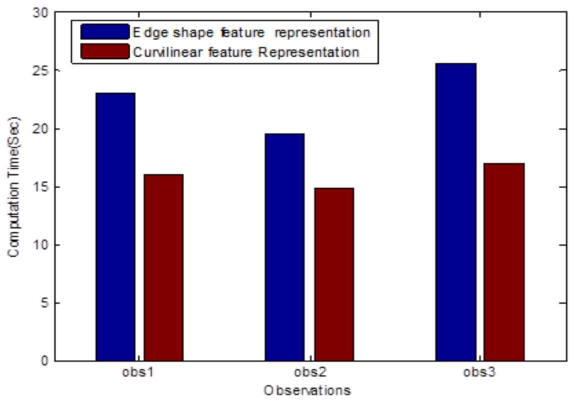


Figure 10. Evaluation of computation time for the two proposed approaches

Table 4. Comparative analysis of processing overhead

Sr. No.	Observation	Processing overhead (%)	
		Edge-based method	Curvilinear method
1	Obser-1	4.4488	3.0072
2	Obser-2	2.2516	1.2569
3	Obser-3	3.2018	1.3279

The ROC curve in Figure 11 shows a 0.12-unit increase in true positive rate, reflecting improved accuracy from the curvilinear approach, which efficiently captures dynamic movements and reduces computational overhead. Figure 12 compares processing overhead for edge shape and curvilinear feature representations, showcasing the system's efficiency.

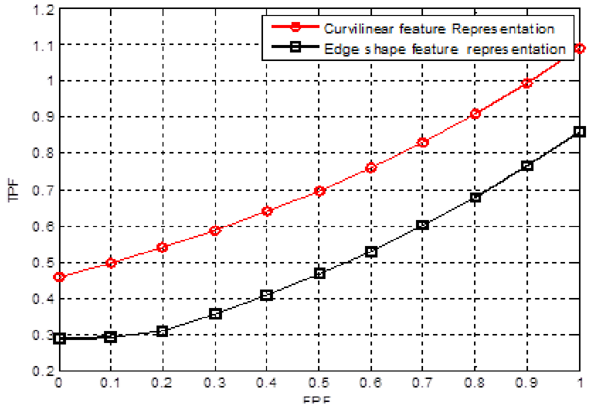


Figure 11. TPF v/s FPF for the proposed system

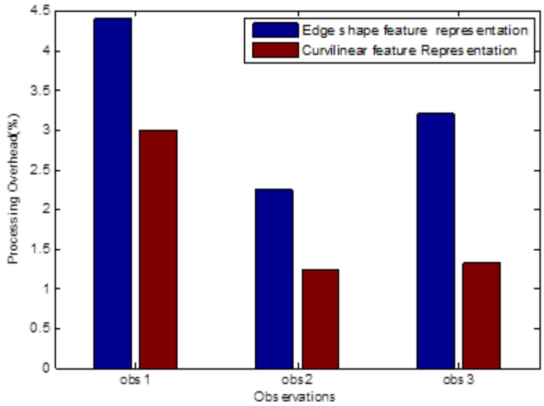


Figure 12. Processing overhead for the developed system

The study highlights the importance of preserving intricate gesture details for improved recognition. While the system showed high recognition rates, future work will focus on expanding the dataset, enhancing real-time processing, and adapting the system for other regional sign languages. These findings represent a significant step toward more inclusive communication tools for the deaf community.

5. CONCLUSION

The key discovery that the proposed curvilinear-based approach is significant improvement in translating Kannada sign language into text through curvilinear feature representation, outperforming traditional shape-based methods in both precision and recall. Advanced techniques like peak curvature features and contour evolution make the system robust, adaptable to gesture variations, and resistant to environmental noise. Its high accuracy preserves sign language nuances, offering valuable applications in everyday communication and educational settings where it could enhance learning for deaf students. However, the study's limitations include challenges in handling environmental conditions, which may affect recognition accuracy. Future directions include expanding the dataset, enhancing real-time processing, and adapting the system for different environmental conditions, gesture variations & other regional sign languages.

ACKNOWLEDGMENTS

The authors sincerely appreciate the reviewers for their insightful and constructive feedback, which has enhanced the clarity and presentation of the findings.

FUNDING INFORMATION

This research received no external funding.

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
Shantappa G Gollagi	✓	✓	✓	✓	✓	✓		✓	✓	✓		✓	✓	
Mahantesh Laddi	✓	✓				✓		✓	✓	✓	✓	✓		
Suhas G K			✓	✓			✓				✓		✓	
Kalyan Devappa Bamane					✓		✓			✓				
Sulbha Yadav					✓		✓			✓				

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

The authors confirm that there are no conflicts of interest associated with this work.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] V. B. Durdi, A. Kiran, A. Rao, S. S. Bhat, B. Santhosh, and N. Mahesh Kumar, "A novel method for recognizing hand gestured sign language using the stochastic gradient descent algorithm and convolutional neural network techniques," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 7s, pp. 529–538, 2024.

[2] S. Mohsin, B. W. Salim, A. K. Mohamedsaeed, B. F. Ibrahim, and S. R. M. Zeebaree, "American sign language recognition based on transfer learning algorithms," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 5s, pp. 390–399, 2024.

[3] M. Aziz and A. Othman, "Evolution and trends in sign language avatar systems: unveiling a 40-year journey via systematic review," *Multimodal Technologies and Interaction*, vol. 7, no. 10, 2023, doi: 10.3390/mti7100097.

[4] V. D. Avina, M. Amiruzzaman, S. Amiruzzaman, L. B. Ngo, and M. A. A. Dewan, "An AI-based framework for translating American sign language to English and Vice Versa," *Information (Switzerland)*, vol. 14, no. 10, 2023, doi: 10.3390/info14100569.

[5] N. F. Attia, M. T. F. S. Ahmed, and M. A. M. Alshewimy, "Efficient deep learning models based on tension techniques for sign language recognition," *Intelligent Systems with Applications*, vol. 20, 2023, doi: 10.1016/j.iswa.2023.200284.

[6] B. Alsharif, A. S. Altaher, A. Altaher, M. Ilyas, and E. Alalwany, "Deep learning technology to recognize American sign language alphabet," *Sensors*, vol. 23, no. 18, 2023, doi: 10.3390/s23187970.

[7] S. B. Abdullahi and K. Chamnongthai, "IDF-sign: addressing inconsistent depth features for dynamic sign word recognition," *IEEE Access*, vol. 11, pp. 88511–88526, 2023, doi: 10.1109/ACCESS.2023.3305255.

[8] K. Skedsmo, "Repair receipts in Norwegian sign language multiperson conversation," *Journal of Pragmatics*, vol. 215, pp. 189–212, 2023, doi: 10.1016/j.pragma.2023.07.015.

[9] K. K. Podder *et al.*, "Signer-independent Arabic sign language recognition system using deep learning model," *Sensors*, vol. 23, no. 16, 2023, doi: 10.3390/s23167156.

[10] A. Ji *et al.*, "Dataglove for sign language recognition of people with hearing and speech impairment via wearable inertial sensors," *Sensors*, vol. 23, no. 15, p. 6693, Jul. 2023, doi: 10.3390/s23156693.

[11] B. Svendsen and S. Kadry, "Comparative analysis of image classification models for Norwegian sign language recognition," *Technologies*, vol. 11, no. 4, p. 99, Jul. 2023, doi: 10.3390/technologies11040099.

[12] S. K. Paul *et al.*, "An Adam based CNN and LSTM approach for sign language recognition in real time for deaf people," *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 1, pp. 499–509, Feb. 2024, doi: 10.11591/eei.v13i1.6059.

[13] D. Sanchez Ruiz, J. A. Olvera-Lopez, and I. Olmos-Pineda, "Word level sign language recognition via handcrafted features," *IEEE Latin America Transactions*, vol. 21, no. 7, pp. 839–848, Jul. 2023, doi: 10.1109/TLA.2023.10244183.

[14] H. Pallab Jyoti Dutta, M. K. Bhuyan, D. R. Neog, K. F. MacDorman, and R. H. Laskar, "Patient assistance system based on hand gesture recognition," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–13, 2023, doi: 10.1109/TIM.2023.3282655.

[15] D. R. Kothadiya, C. M. Bhatt, A. Rehman, F. S. Alamri, and T. Saba, "SignExplainer: an explainable AI-enabled framework for sign language recognition with ensemble learning," *IEEE Access*, vol. 11, pp. 47410–47419, 2023, doi: 10.1109/ACCESS.2023.3274851.

[16] L. T. Woods and Z. A. Rana, "Modelling sign language with encoder-only transformers and human pose estimation keypoint data," *Mathematics*, vol. 11, no. 9, p. 2129, May 2023, doi: 10.3390/math11092129.

[17] R. Sreemathy, M. P. Turuk, S. Chaudhary, K. Lavate, A. Ushire, and S. Khurana, "Continuous word level sign language recognition using an expert system based on machine learning," *International Journal of Cognitive Computing in Engineering*, vol. 4, pp. 170–178, Jun. 2023, doi: 10.1016/j.ijcce.2023.04.002.

[18] S. Siddique, S. Islam, E. E. Neon, T. Sabbir, I. T. Naheen, and R. Khan, "Deep learning-based Bangla sign language detection with an edge device," *Intelligent Systems with Applications*, vol. 18, p. 200224, May 2023, doi: 10.1016/j.iswa.2023.200224.

[19] S. Qahtan, H. A. Alsattar, A. A. Zaidan, M. Deveci, D. Pamucar, and L. Martinez, "A comparative study of evaluating and benchmarking sign language recognition system-based wearable sensory devices using a single fuzzy set," *Knowledge-Based Systems*, vol. 269, p. 110519, Jun. 2023, doi: 10.1016/j.knosys.2023.110519.

[20] J. Shin *et al.*, "Korean sign language recognition using transformer-based deep neural network," *Applied Sciences (Switzerland)*, vol. 13, no. 5, p. 3029, Feb. 2023, doi: 10.3390/app13053029.

[21] J. Bora, S. Dehingia, A. Boruah, A. A. Chetia, and D. Gogoi, "Real-time assamese sign language recognition using MediaPipe and deep learning," *Procedia Computer Science*, vol. 218, pp. 1384–1393, 2022, doi: 10.1016/j.procs.2023.01.117.

[22] Y. Liu, P. Nand, M. A. Hossain, M. Nguyen, and W. Q. Yan, "Sign language recognition from digital videos using feature pyramid network with detection transformer," *Multimedia Tools and Applications*, vol. 82, no. 14, pp. 21673–21685, Jun. 2023, doi: 10.1007/s11042-023-14646-0.

[23] B. Hdioud and M. El Haj Tirari, "A deep learning based approach for recognition of Arabic sign language letters," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 4, pp. 424–429, 2023, doi: 10.14569/IJACSA.2023.0140447.

[24] E. K. Elsayed and D. R. Fathy, "Sign language semantic translation system using ontology and deep learning," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 1, pp. 141–147, 2020, doi: 10.14569/ijacsa.2020.0110118.

[25] H. Organization, "Deafness and hearing loss," 2021. https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing loss%0A.

[26] B. Hdioud and M. E. H. Tirari, "Facial expression recognition of masked faces using deep learning," *IAES International Journal of Artificial Intelligence*, vol. 12, no. 2, pp. 921–930, 2023, doi: 10.11591/ijai.v12.i2.pp921-930.

[27] K. Kumar, "DEAF-BSL: Deep lEArning framework for British sign language recognition," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 21, no. 5, pp. 1–14, Sep. 2022, doi: 10.1145/3513004.

BIOGRAPHIES OF AUTHORS

Dr. Shantappa G Gollagi is a Professor in the Department of Computer Science and Engineering at S G Balekundri Institute of Technology, Belagavi, India with 26 years of teaching and 8 years of research experience, he brings extensive expertise to the field. He earned his Bachelor's degree in computer science and engineering from BVB College of Engineering and Technology (now KLE Technological University), Hubli, his M.Tech. in computer engineering from College of Engineering Pune (COEP), Pune, and his Ph.D. in computer and information science from VT University, Belagavi. His research interests span software security, image processing, and pervasive computing. He can be contacted at email: shantesh1973@rediffmail.com.

Prof. Mahantesh Laddi is an Assistant Professor in the Department of Computer Science and Engineering at Bharatesh Institute of Technology, Belagavi, India with 12 years of teaching and 3 years of research experience, he brings extensive expertise to the field. He earned his Bachelor's degree in Computer Science and Engineering and Master in Computer Network Engineering from Bellary Institute of Technology, Bellary, and pursuing his Ph.D. in computer and information science from VT University, Belagavi. His research interests span cyber security, software security, and cloud computing. He can be contacted at email: mahantesh18689@gmail.com.

Dr. Suhas G K is a Professor and Head of Department of Information Science and Engineering at Akshaya Institute of Technology, Tumakuru, India. With 10 years of teaching and 5 years of research experience, he brings extensive expertise to the field. He earned his Bachelor of Engineering, Master of Technology and Ph.D. degree in computer science and engineering from Visvesvaraya Technological University. His research interests are compilers, bigdata analytics, computer networks, cyber security, and machine learning. He can be contacted at email: suhask300@gmail.com.

Dr. Kalyan Devappa Bamane is currently working as an Associate Professor in the Department of Computer Engineering at D Y Patil College of Engineering, Akurdi. He has completed his M.E. in Computer Engineering (2011) from SPPU Pune. He has completed his PhD in CSE from VTU Belagavi in 2021. His areas of interest include artificial intelligence, cyber security, distributed systems, and computer networks. He has 19 years of experience in academia and has published about 50 papers in international journals and conferences. He can be contacted at email: kdbamane@dypcoeakurdi.ac.in.

Prof. Sulbha Yadav is currently working as an Assistant Professor in the Computer Department at Lokmanya Tilak College of Engineering, Koparkhairane Navi Mumbai. She has completed his M.E. in computer engineering (2011). Her areas of interest include artificial intelligence, internet of things, big data analytics, blockchain technology, and computer networks. She has 20 years of experience in academia and has published about 22 papers in international journals and conferences. She can be contacted at email: sulbha.yadav@gmail.com.