International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 14, Issue 11, November 2025)

Sign Sync: Instant Sign Language to Multilingual Speech & Text

Dr. Saraswathi D[1], Prajwal G[2], Pranav D S[3], Keerthi Narayan[4], Darshan S[5]

[1]ProfessorInformation Science and Engineering, Maharaja Institute of Technology, Mysore, Karnataka, India
[2,3,4,5]Student Maharaja Institute of Technology, Mysore, Karnataka, India

***Abstract*—Communication barriers between the hearing-impaired community and the general population often lead to reduced accessibility and social inclusion. To address this challenge, our project presents a real-time system that converts sign language gestures into audible speech across multiple spoken languages. The proposed model integrates computer vision–based hand-gesture recognition with deep-learning classification methods to accurately interpret sign language signs. A translation module then maps recognized signs to text, followed by a multilingual text-to-speech engine that outputs speech in the user's selected language. The system supports dynamic interaction, high accuracy under varied lighting and background conditions, and seamless switching between languages. Experimental evaluations demonstrate robust performance and low latency, making the system suitable for education, public services, and assistive communication applications. This work contributes to inclusive human–computer interaction by providing an affordable and scalable solution that bridges communication gaps for individuals with hearing or speech impairments.**

***Keywords*—Sign Language Recognition, Gesture Recognition, Deep Learning, Computer Vision, Multilingual Speech Synthesis, Assistive Technology, Human–Computer Interaction, Accessibility, Real-Time Processing, Text-to-Speech (TTS), Machine Learning, Communication Aid.**

I. INTRODUCTION

A. ProblemStatement:

Despite significant advancements in communication technologies, individuals with hearing and speech impairments continue to face persistent barriers in interacting with the broader community. Sign language—though rich, expressive, and widely used within the deaf community—is not universally understood by the general population. This lack of common understanding results in communication gaps across essential domains such as education, healthcare, employment, and public services. Existing sign-language interpretation systems often suffer from major limitations: they are restricted to a single language, require expensive hardware such as sensor-based gloves, or fail to operate effectively in real time under varying environmental conditions.

Additionally, most available tools do not provide multilingual speech output, reducing their accessibility and global usability.

There is a significant communication barrier between sign language users and the hearing population, as most people cannot understand sign gestures in real time. Existing solutions lack accuracy, natural sentence formation, and multilingual speech support, creating the need for a fast, reliable, and accessible real-time sign language translation system.

Therefore, there is an urgent need for a robust, low-cost, and scalable solution that can accurately recognize sign-language gestures and convert them into intelligible speech in multiple languages. Addressing this gap will enable more inclusive communication, support independent living for people with hearing and speech impairments, and contribute to the development of universal assistive technologies.

B. ProposedSolution

SignConnect is a real-time, multilingual sign language translation system that uses webcam-based gesture recognition and machine learning to interpret hand landmarks extracted through MediaPipe. A RandomForest classifier predicts signs, which are then converted into meaningful sentences using dictionary-based mapping and autocorrection. The system also supports bidirectional communication through text-to-sign visualization, speech-to-text input, multilingual translation, and text-to-speech output. Built on a scalable Flask web framework, SignConnect provides an accessible, low-cost solution for bridging communication between signers and non-signers.

The solution supports bidirectional communication through text-to-sign visualization, speech-to-text recognition, and multilingual text-to-speech generation using gTTS. A translation module further enables conversion of text into major Indian languages, making the system accessible to a wider audience. Overall, Sign

Connect provides an efficient, low-cost, and scalable approach to sign language translation by integrating computer vision, machine learning, and NLP technologies into a unified application.

Contributions:

This work introduces SignConnect, a real-time multilingual sign language translation system with the following key contributions:

- A webcam-based sign recognition pipeline using MediaPipe landmarks and a RandomForest classifier.
- A stability buffer that ensures reliable, noise-free predictions during live signing.
- A sentence formation module with dictionary-based mapping and Levenshtein autocorrection for meaningful text output.

Bidirectional communication through text-to-sign, speech-to-text, and multilingual text-to-speech.A lightweight, fully web-based architecture enabling easy deployment and cross-platform accessibility.

The system introduces a real-time sign recognition pipeline with intelligent sentence formation, multilingual translation, and speech output to enable seamless communication. It further contributes a lightweight, modular framework that supports bidirectional interaction through speech-to-text and text-to-sign features.

C. PaperStructure:

"The remainder of this paper is organized as follows: SectionIIdetailsthe LiteratureSurvey.SectionIIIdescribes the System Architecture including implementation of the navigation.SectionIVpresentstheexperimentalresults, and Section V concludes the paper with a discussion on future work."

II. Literature Survey

Recent advancements in computer vision and deep learning have significantly transformed sign language recognition from manual, rule-based methods to automated, real-time, and multimodal systems. Toshpulatov et al. [1] presented a comprehensive survey of deep learning pathways for sign language processing, highlighting the use of 2D/3D CNNs, Transformers, and GCNs for recognition, translation, and avatar-based production.

Navendu and Sahula [2] proposed a Media Pipe-based ISL recognition framework using an LSTM–GRU hybrid network trained on the INCLUDE dataset, achieving 89.50% accuracy while addressing temporal dependencies in word-level gestures. Sujatha et al. [3] expanded regional sign language datasets and demonstrated the effectiveness of MediaPipe Holistic combined with Bi-LSTM, achieving 91.25% accuracy for Kannada mathematical signs. Swetha and R. [4] developed a real-time Kannada sign recognition system for healthcare using MobileNet-based CNNs and LSTM models, integrating MediaPipe and Google TTS for immediate speech output.Classical approaches remain relevant in limited contexts; Kagalkar and Nagaraj [5] employed edge detection and feedforward neural networks for static KSL recognition, demonstrating feasibility despite environmental constraints. Sensor-based systems, like those by Kagalkar and Nagaraj [6], achieved 96.66% accuracy using glove-integrated flex and IMU sensors with Random Forest classifiers, offering low-latency results but limited scalability for natural signing. Lightweight MediaPipe–SVM systems have also shown strong outcomes, with Halder and Tayade [7] reporting accuracies up to 99.29% across multiple sign languages.More advanced deep learning systems such as MediSign by Ihsan et al. [8] used MobileNetV2 with Attention-BiLSTM for medical sign recognition, achieving 95.83% accuracy and demonstrating the strength of attention mechanisms for temporal modeling. Fusion-based gesture recognition approaches, such as the enhanced YOLOv5 and Copula Bayesian classifier proposed by Han et al. [9], further emphasize the trend toward efficient edge-deployable models. Meanwhile, Adaloglou et al. [10] provided a large-scale comparative analysis of deep learning methods for continuous sign language recognition, noting the challenges of glossalignment, signer variability, and large annotated dataset requirements.S

III. System Design And Methodology

Hardware Requirements

A system running this application should have an Intel Core i3 processor (or an equivalent/higher model) along with a minimum of 8 GB RAM to ensure smooth real-time processing. At least 2 GB of free disk space is required to store application files, trained models, and datasets.

A functional built-in or external webcam with a minimum resolution of 640×480 is necessary for accurate hand-gesture capture, while a working microphone is needed to support speech-to-text operations. Additionally, speakers or headphones are required to enable audio output during text-to-speech functionality.

Software Components

The system supports Windows 10/11 (64-bit) or Linux distributions such as Ubuntu 20.04 and above, and it requires Python 3.11.0 as the primary programming environment. Essential libraries and dependencies include Flask, OpenCV, MediaPipe, scikit-learn, gTTS, SpeechRecognition, googletrans, PyAudio, PortAudio development libraries, and all additional packages specified in the requirements.txt file. For accessing the web interface and application features, a modern web browser such as Google Chrome, Mozilla Firefox.

Methodology

Model Training: A dataset of sign language hand landmarks is collected using webcam recordings, and MediaPipe is employed to extract 21 landmark coordinates from each frame. These coordinates are normalized to ensure consistency across users, and a Random Forest classifier is trained on the annotated features with optimized hyperparameters to achieve high-accuracy. Data Collection and Preprocessing: Multiple frames for each sign are captured under different lighting and background conditions. The extracted landmarks are cleaned, scaled, and balanced to remove noise and ensure uniform representation across-all-sign-classes. Real-Time Sign Detection: Live webcam frames are processed through MediaPipe to obtain hand landmarks, which are fed into the trained model for classification. A stability buffer is applied to prevent flickering and ensure reliablereal-time predictions.

Sentence Formation: Recognized signs are accumulated and transformed into meaningful words using a dictionary-based mapping system. Levenshtein distance is applied to auto-correct minor errors, and coherent sentences areconstructed-dynamically. Multilingual Translation and *Speech Output*: The generated sentences are translated into selected Indian languages using the translation module and converted into speech using gTTS, enabling seamless sign-to-speech communication.

Bidirectional Communication: Speech-to-text functionality allows non-signers to interact through voice input, while the text-to-sign module displays corresponding sign images, supporting reverse communication.

System Integration: All modules are incorporated into a Flask-based web interface, providing real-time video streaming, sign prediction, sentence display, translation, and speech output within a unified and user-friendly platform.

1. Working Principle:1. Image Acquisition (Live Video Capture): A built-in or external webcam continuously captures live video of the user's hand gestures, providing real-time frames for processing under different lighting and background.

2. Frame Extraction and Landmark Preprocessing: The live video stream is divided into individual frames using OpenCV. Each frame is processed with MediaPipe Hands to extract 21 hand landmarks, which are then normalized and scaled to ensure consistency across users and camera positions.

3. Sign Detection using Random Forest Classifier: The normalized landmark vectors are passed into a trained RandomForest classifier that identifies the corresponding sign. A stability buffer validates repeated consistent predictions to eliminate flickering and produce reliable real-time outputs.

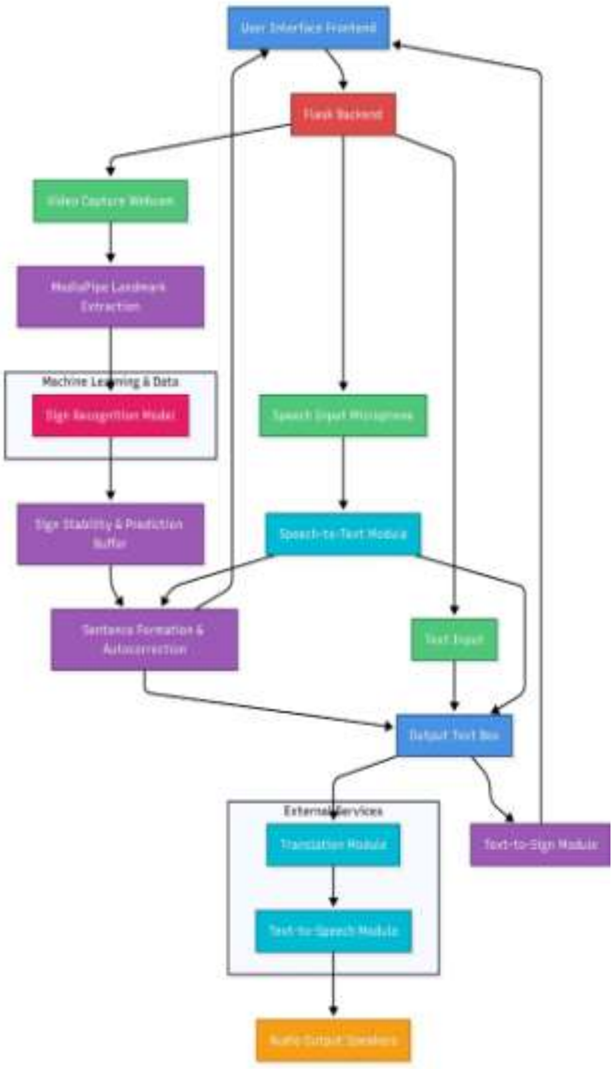


Fig 7.1: Overall System Architecture of SignConnect [Sign to Text]

4. *Word Formation and Sentence Generation:* Detected signs are accumulated and converted into meaningful words using a dictionary-based mapping system. Levenshtein auto correction corrects minor errors, and refined words are combined into coherent sentences displayed to the user.

5. *Multilingual Translation:* The generated sentences are translated into selected Indian languages such as Hindi, Kannada, Telugu, Tamil, or Malayalam using the translation module for regional communication support.

6. *Text-to-Speech Output:* The translated or original sentence is converted to audible speech using gTTS, enabling real-time sign-to-speech interaction through the system's-speakers.

7. *Reverse Communication (Speech-to-Text and Text-to-Sign):* The system supports speech-to-text for non-signers and displays corresponding sign images through the text-to-sign module, enabling two-way communication.

8. *Real-Time Feedback and User Interaction:* All processes run within a Flask-based web interface, delivering live video streaming, sign detection, translation, and speech output with minimal latency for smooth real-time operation.

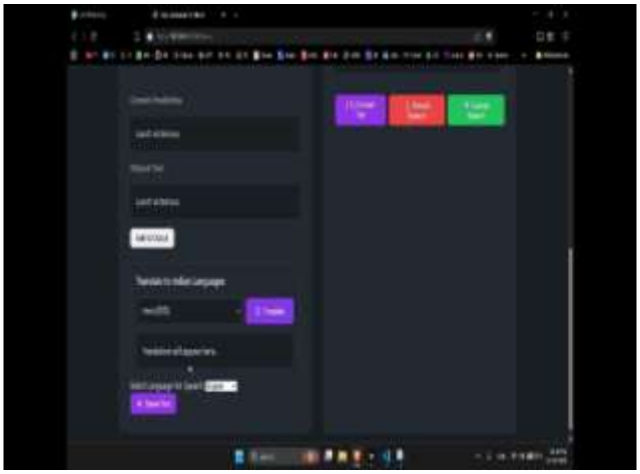
IV. RESULTS AND DISCUSSION

The Sign Connect system was evaluated based on recognition accuracy, real-time responsiveness, translation quality, and overall usability. The Random Forest Classifier trained on normalized hand landmark features demonstrated high accuracy for isolated alphabet signs and selected words. During live testing, the model consistently recognized static signs with minimal latency, maintaining real-time responsiveness under typical conditions. The stability buffer improved prediction consistency by reducing flickering and eliminating-frame-to-frame misclassifications.

Real-time system performance remained stable on standard hardware configurations, achieving 18–25 FPS without requiring GPU acceleration. MediaPipe landmark extraction combined with lightweight model inference allowed smooth execution, ensuring natural and uninterrupted communication during-live-sign-translation. The sentence formation module successfully transformed individual signs into coherent words and meaningful sentences. The Levenshtein-distance-based auto correction improved word accuracy, generating fluent text even when sign inputs contained noise or minor inaccuracies. This enhanced the overall quality and readability of the translated output.

Multilingual translation and text-to-speech modules functioned reliably across supported Indian languages. Translation latency remained low, typically between 1–3 seconds, while gTTS provided clear and intelligible speech output. This enabled seamless communication between signers and-non-signers-in-multilingual-settings.

Bidirectional communication features such as speech-to-text and text-to-sign operated effectively, allowing hearing users to communicate back to sign language users. The interface for displaying sign images proved useful for foundational learning andbasic-reverse-communication.



The Flask-based web interface offered an intuitive user experience, with real-time video feed, sign predictions, sentence output, translation, and speech functionalities integrated into a single accessible platform. User feedback indicated that the system was easy to navigate and responsive.

Overall, the results confirm that SignConnect provides efficient and practical real-time sign language translation. Its combination of accurate sign recognition, natural sentence formation, multilingual support, and low-latency performance makes it suitable for real-world use in educational environments, accessibility services, and communication assistance systems.

V. Conclusion

The "Sign Connect: Real-time Sign Language Translator" system successfully demonstrates an effective, accessible, and bidirectional communication platform that bridges the gap between deaf and hearing communities. By integrating MediaPipe-based landmark extraction, a trained Random Forest classifier, intelligent sentence formation, multilingual translation, and text-to-speech synthesis, the system provides accurate and real-time sign-to-speech conversion. Its reverse communication modules—speech-to-text and text-to-sign—enable seamless interaction in both directions. The system's modular architecture, multilingual support, and user-friendly web interface validate its practical applicability in educational environments, accessibility services, and public communication scenarios. Overall, Sign Connect showcases how real-time computer vision and machine learning can drive meaningful social impact by enhancing inclusivity and communication accessibility.

VI. Future Work

Future enhancements can significantly expand the system's capabilities. Increasing the sign vocabulary to include more words, phrases, and continuous sign language recognition (CSLR) will enable more natural and expressive communication. Incorporating non-manual features such as facial expressions and body posture will improve recognition accuracy and contextual understanding. Advanced deep learning models—such as CNN-LSTM hybrids or Transformer-based architectures—may further boost performance. Synthetic dataset generation and transfer learning can address data scarcity and improve generalization. Additional improvements include advanced NLP-based grammatical correction, 3D avatar-based sign visualization, expanded multilingual support, mobile application development, cloud deployment, offline capability, and user feedback–driven refinement. These advancements will move Sign Connect closer to becoming a fully comprehensive and widely deployable sign language communication assistant.

REFERENCES

[1] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *DokladyAkademiiNauk SSSR*, vol. 163, no. 4, pp. 845–848, 1965.

[2] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *Proc. NeurIPS*, 2015.

[4] K. Simonyan and A. Zisserman, "Two-Stream Convolutional Networks for Action Recognition in Videos," in *Proc. NeurIPS*, 2014.

[5] Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning Spatiotemporal Features with 3D Convolutional Networks (C3D)," in *Proc. ICCV*, 2015.

[6] J. Carreira and A. Zisserman, "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset (I3D)," in *Proc. CVPR*, 2017, pp. 6299–6308.

[7] G. Bertasius, H. Wang, and L. Torresani, "Is Space-Time Attention All You Need for Video Understanding? (TimeSformer)," in *Proc. ICML*, 2021, pp. 8132–8145.

[8] A Arnab et al., "ViViT: A Video Vision Transformer," in *Proc. ICCV*, 2021, pp. 6836–6846.

[9] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local Neural Networks," in *Proc. CVPR*, 2018, pp. 7794–7803.

[10] R. S. Saini and H. K. Bhadana, "A Survey on Indian Sign Language Recognition," *International Journal of Computer Applications*, vol. 129, no. 1, pp. 1–6, 2015.

[11] T. S. K. Singh and R. P. Singh, "Real-time Indian Sign Language Recognition using Deep Learning," in *Proc. ICCCSP*, 2017, pp. 1–5.

[12] Y. Wu et al., "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation," *arXiv:1609.08144*, 2016.

[13] Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[14] Bradski, "The OpenCV Library," *Dr. Dobb's Journal*, 2000.

[15] Python SpeechRecognition Library. Available: https://pypi.org/project/SpeechRecognition/