INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD)
© 2026 IJNRD | Volume 11, Issue 3, MARCH 2026 | ISSN: 2456-4184 | IJNRD.ORG

Cross-Sign Language Translation System

[1]Jithin S, [2]Abhinav KP, [3]Saran Scaria, [4]Sayanth S, [5]Vishnu Vinodan

[1]Assistant Professor, [2]Student, [3]Student, [4]Student, [5]Student
[1]Department of Computer Science and Engineering,
[1]St. Thomas College of Engineering and Technology, Kannur, Kerala, India.

Abstract : The Cross-Sign Language Translation System is designed to overcome the communication challenges faced by the deaf and speech-impaired communities across different regions by enabling real-time translation between major sign languages such as Indian Sign Language (ISL), American Sign Language (ASL), and British Sign Language (BSL).Existing sign recognition systems typically focus on single-language translation and rely solely on hand gesture detection, neglecting crucial elements such as facial expressions and body posture that significantly influence meaning. To address these limitations, the proposed system employs MediaPipe and OpenCV for real-time landmark detection, while deep learning architectures such as Convolutional Neural Networks (CNNs) and Transformers are utilized for robust feature extraction and accurate gesture recognition. The recognized gestures are first converted into an intermediate gloss text, which is then cross-translated into the target sign language. The final output can be delivered as text, speech, or 3D avatar animation, ensuring accessibility for a wide range of users. Furthermore, the system is optimized for real-time performance and cloud-edge deployment, enabling low-latency operation across different platforms and devices. By combining multimodal recognition with multilingual translation, this project aims to foster inclusivity, cultural integration, and equal communication opportunities, making it a scalable solution applicable in educational institutions, healthcare settings, and global communication platforms.

IndexTerms **-** Indian Sign Language (ISL), American Sign Language (ASL), MediaPipe , OpenCV, Convolutional Neural Networks (CNNs), Transformers, Multimodal Gesture Recognition, Real-Time Translation.

I. INTRODUCTION

Sign language serves as the primary mode of communication for individuals with hearing and speech impairments. It enables the expression of thoughts, emotions, and intentions through structured hand gestures, facial expressions, and body movements. However, unlike spoken languages, sign languages are not universal. Different regions and cultures use distinct sign languages such as Indian Sign Language (ISL), American Sign Language (ASL), and British Sign Language (BSL), each having its own grammar, syntax, and vocabulary. This diversity creates significant communication barriers when individuals from different sign language communities interact.

With the rapid growth of digital communication platforms and globalization, the need for inclusive and accessible communication systems has become increasingly important. Although several sign language recognition systems have been developed in recent years, most existing solutions are limited to sign-to-text or sign-to-speech translation within a single sign language. These systems fail to address cross-lingual communication, where gestures from one sign language must be translated into another sign language in real time.

To overcome these limitations, this paper presents CROSS-SIGN, a Cross-Sign Language Translation System that enables real-time translation between different sign languages. The proposed system captures multimodal inputs—including hand gestures, facial expressions, and body movements—and processes them using a spatio-temporal transformer-based deep learning model. Recognized signs are converted into an intermediate gloss representation, translated into a target sign language, and rendered using a 3D avatar with synchronized speech output. By integrating real-time translation, visual signing, and text-to-speech synthesis, the system aims to bridge linguistic and geographic gaps within the deaf and hard-of-hearing community.

A. Problem Statement

Communication is still a major challenge for the deaf and speech-impaired community, especially across different regions. Sign languages like ISL, ASL, and BSL are not the same, which makes it difficult for users to communicate with each other globally.

Most existing systems focus only on hand gestures and ignore important factors like facial expressions and body posture, leading to inaccurate interpretations. In addition, many solutions support only a single sign language and are not suitable for real-time use.

Because of these limitations, people still depend on human interpreters, which are not always available. This creates barriers in everyday situations like education, healthcare, and public communication.
To solve this, there is a need for a system that can recognize full sign language expressions and translate between multiple sign languages in real time, making communication more accessible and inclusive.

II. LITERATURE SURVEY

A literature survey is a vital part of any research work as it provides insight into existing technologies, methodologies, and developments in the chosen domain. It helps identify strengths, limitations, and research gaps in prior studies, enabling researchers to build upon proven concepts while avoiding redundancy.

This survey provides a systematic review of existing research works related to a particular problem domain. In the context of sign language recognition and translation, numerous studies have explored gesture recognition, continuous sign language translation, and multimodal deep learning techniques. This chapter reviews key research contributions that form the foundation for the proposed Cross-Sign Language Translation System and highlights the limitations of existing approaches that motivate this work.

A. STFE-Net: A Spatial-Temporal Feature Extraction Network for Continuous Sign Language Translation [1].

STFE-Net introduces a spatial–temporal feature extraction framework designed for continuous sign language translation. The model focuses on extracting discriminative spatial features using pose estimation and temporal dependencies using Transformer-based architectures. To improve efficiency, the authors reduce the COCO-WholeBody keypoints from 133 to 53 by eliminating less relevant body parts such as legs and excessive facial landmarks, concentrating primarily on upper-body and hand movements that are crucial in sign language communication.

For temporal modeling, the extracted spatial features are passed through a Transformer with relative positional encoding, enabling the system to capture long-range dependencies between gestures. The model demonstrates strong performance across multiple benchmark datasets, achieving significantly higher BLEU scores compared to baseline models. The study highlights the effectiveness of combining spatial pose-based features with attention-based temporal modeling. However, the system is largely language-specific and does not support cross-sign language translation, limiting its applicability in multilingual environments.

B. Hand Gesture Recognition for Multi-Culture Sign Language Using Graph and General Deep Learning Network [2].

This work proposes the GmTC (Graph meets Transformer and CNN) architecture to address the challenges of recognizing multiple culturally distinct sign languages. The system integrates graph-based learning with deep neural networks to capture both spatial relationships and long-range dependencies in gesture data. Superpixels extracted from gesture images are modeled as graph nodes and processed using Graph Convolutional Networks (GCN), enabling the system to learn spatial correlations effectively.

In parallel, a CNN–Transformer hybrid stream extracts high-level semantic features using Multi-Head Self-Attention mechanisms. The features from both streams are fused to improve recognition accuracy. Experimental results show that the combined architecture significantly outperforms single-stream models across multiple sign language datasets such as ASL and KSL.

While the model demonstrates excellent recognition accuracy across different sign languages, it focuses primarily on classification rather than real-time translation. Additionally, it does not address continuous cross-language gesture-to-gesture translation, which is essential for real-world communication systems.

C. Toward Real-Time Recognition of Continuous Indian Sign Language Using a Multi-Modal Approach [3].

The SignFlow framework introduces a real-time continuous sign language recognition system specifically tailored for Indian Sign Language (ISL). The model adopts a dual-stream approach combining RGB video features with pose-based features extracted using MediaPipe. A 3D CNN is used for visual feature extraction, while a Transformer encoder models temporal dependencies without relying on explicit positional encoding.

One of the major contributions of this work is its focus on real-world data collection, incorporating variations in background, lighting, device type, and signer posture. The system achieves a low Word Error Rate (WER) and demonstrates strong generalization across benchmark datasets. The study also introduces a detection rate metric suitable for real-time applications.

Despite its effectiveness, the framework is limited to ISL recognition and does not provide cross-sign language translation. The absence of a language-agnostic intermediate representation restricts its scalability to multilingual communication scenarios.

III. METHODOLOGY

The proposed Cross-Sign Language Translation System is designed using a modular and layered system architecture to support real-time sign language communication across different linguistic systems. The methodology begins with capturing live video input through a camera during a video call, allowing continuous interaction between users. The system processes this input to extract multimodal features such as hand gestures, facial expressions, and upper-body movements using computer vision tools like MediaPipe and OpenCV. These extracted landmarks are normalized and converted into structured numerical representations, ensuring consistency across different users and environments. The processed feature vectors are then passed to a deep learning–based recognition module. A combination of Convolutional Neural Networks (CNNs) and Transformer-based spatial–temporal

models is employed to learn both frame-level spatial patterns and sequential temporal dependencies, enabling accurate recognition of continuous sign language gestures while preserving contextual information.

Once the gestures are recognized, the system converts them into an intermediate gloss representation, which acts as a language-neutral semantic layer and decouples gesture recognition from language translation. This gloss-based output is then translated into the target sign language using predefined translation mappings stored in the system database. The translated result is presented to the user through multiple output modes, including 3D avatar animation for visual representation, text display for readability, and speech output for broader accessibility. To meet real-time performance requirements and ensure smooth video communication, the recognition model is optimized using TensorFlow Lite (TFLite), significantly reducing latency and computational overhead. This methodology ensures an efficient, scalable, and user-centric system design capable of supporting cross-sign language communication in real-world video conferencing environments, while also allowing easy extension to additional sign languages in the future.

IV. PROPOSED SYSTEM

A. Objectives

The primary objective of the proposed system is to enable real-time cross-sign language communication between users belonging to different sign language communities. The system aims to accurately recognize continuous sign language gestures and translate them into an equivalent target sign language during live video communication. Another key objective is to incorporate multimodal inputs such as hand gestures, facial expressions, and upper-body movements to improve recognition accuracy. The system also focuses on achieving low latency and real-time performance using optimized deep learning models, ensuring accessibility, scalability, and ease of integration with video-calling platforms.

B. System Architecture

The proposed system follows a layered and modular architecture designed for real-time performance and scalability. It consists of three main layers: the User Interface layer, the Processing layer, and the Output layer. The User Interface layer handles video capture, user interaction, and language selection. The Processing layer performs gesture recognition, spatial–temporal modeling, and cross-sign language translation. The Output layer generates translated results in the form of 3D avatar animation, text, or speech output. An API gateway or backend service manages communication between these layers, ensuring secure and efficient data flow.

C. Functional Modules

The system is divided into multiple functional modules to simplify development, maintenance, and scalability. The User Interface and Video Communication Module is responsible for handling live video input, user interaction, and selection of source and target sign languages. The Gesture Capture and Frame Processing Module captures video frames from the camera and performs necessary preprocessing to prepare the data for analysis. The Feature Detection and Extraction Module utilizes MediaPipe and OpenCV to extract meaningful hand, facial, and upper-body landmarks from the processed frames. These extracted features are passed to the Sign Recognition Module, which employs CNN and Transformer-based deep learning models to recognize continuous sign language gestures. The recognized gestures are then processed by the Cross-Sign Language Translation Module, which converts them into an intermediate gloss representation and maps them to the corresponding target sign language. Finally, the Output Generation Module presents the translated result through multiple formats such as 3D avatar animation, text, or speech output, ensuring accessibility. In addition, the Admin and Database Management Module supports system administration by managing user accounts, trained models, datasets, and translation mappings required for accurate and efficient system operation.

D. Data Flow and Implementation

The data flow begins when the user initiates a video session and performs sign language gestures. The camera captures live video frames, which are forwarded to the frame processing module. Extracted multimodal features are passed to the sign recognition model for analysis. The recognized gestures are converted into an intermediate gloss representation and sent to the translation module. The translated output is then generated and rendered to the user in the selected output format. Administrative data such as model updates and translation mappings are handled through the backend database.

The system is implemented using computer vision libraries such as MediaPipe and OpenCV for real-time landmark detection. Deep learning models based on CNNs and Transformers are trained for continuous sign recognition. To enable real-time execution on low-resource devices, the trained model is optimized using TensorFlow Lite (TFLite). The system supports modular deployment, allowing independent updates to recognition models, translation databases, and output rendering components.

V. PROPOSED SYSTEM DESIGN

The proposed Cross-Sign Language Translation System is designed using a layered architecture to support real-time, accurate, and scalable sign language translation. The architecture clearly separates user interaction, data acquisition, processing, and output generation, ensuring modularity and ease of maintenance. Figure 1 illustrates the overall system architecture of the proposed system.

User Interface Layer

The User Interface layer serves as the interaction point between the user and the system. It provides a real-time video preview that allows users to perform sign language gestures during a video call. This layer also includes video call controls such as start, stop, and language selection options. Users can select the source and target sign languages before initiating communication. The interface ensures smooth video streaming and continuously captures user gestures for further processing.

API Gateway / Reverse Proxy

The API Gateway acts as an intermediary between the User Interface and the backend processing components. It manages incoming requests, routes data to the appropriate system modules, and ensures controlled access to backend services. This layer improves scalability, enables better traffic management, and enhances security by isolating the front-end from direct access to core processing components.

Input Layer

The Input Layer is responsible for collecting and organizing all necessary inputs required for sign language translation. It includes multimodal user input such as hand gestures, facial expressions, and body movements captured from the video stream. In addition, this layer accepts the source sign language specification (such as ASL or ISL) selected by the user. Optional metadata, including lighting conditions and user profile information, is also utilized to optimize recognition accuracy and system performance.

Processing Pipeline

The Processing Pipeline forms the core intelligence of the system. It consists of three major components: Gesture Recognition, Spatial–Temporal Transformer Model, and Target Sign Language Translator. The gesture recognition module analyzes the extracted landmarks to identify meaningful sign patterns. The recognized gestures are then passed to the spatial–temporal transformer model, which captures temporal dependencies across frames and interprets continuous sign sequences. The processed output is converted into an intermediate gloss representation and translated into the target sign language using the target sign language translator. This pipeline enables accurate, context-aware, and real-time translation.

Output Layer

The Output Layer is responsible for presenting the translated results to the user in accessible formats. The system supports output through a 3D avatar renderer that visually performs the translated sign language gestures. In addition, text-based transcript export and speech output synchronized with the avatar are provided to enhance accessibility. This multi-format output ensures inclusivity for users with different communication needs.

Design Summary

The proposed system design follows a modular and layered architecture that ensures efficient data flow from input capture to output generation. By separating responsibilities across layers and integrating optimized deep learning models within the processing pipeline, the system achieves low latency, scalability, and real-time performance. This architecture makes the proposed system suitable for deployment in video communication platforms and extensible for supporting additional sign languages in the future.

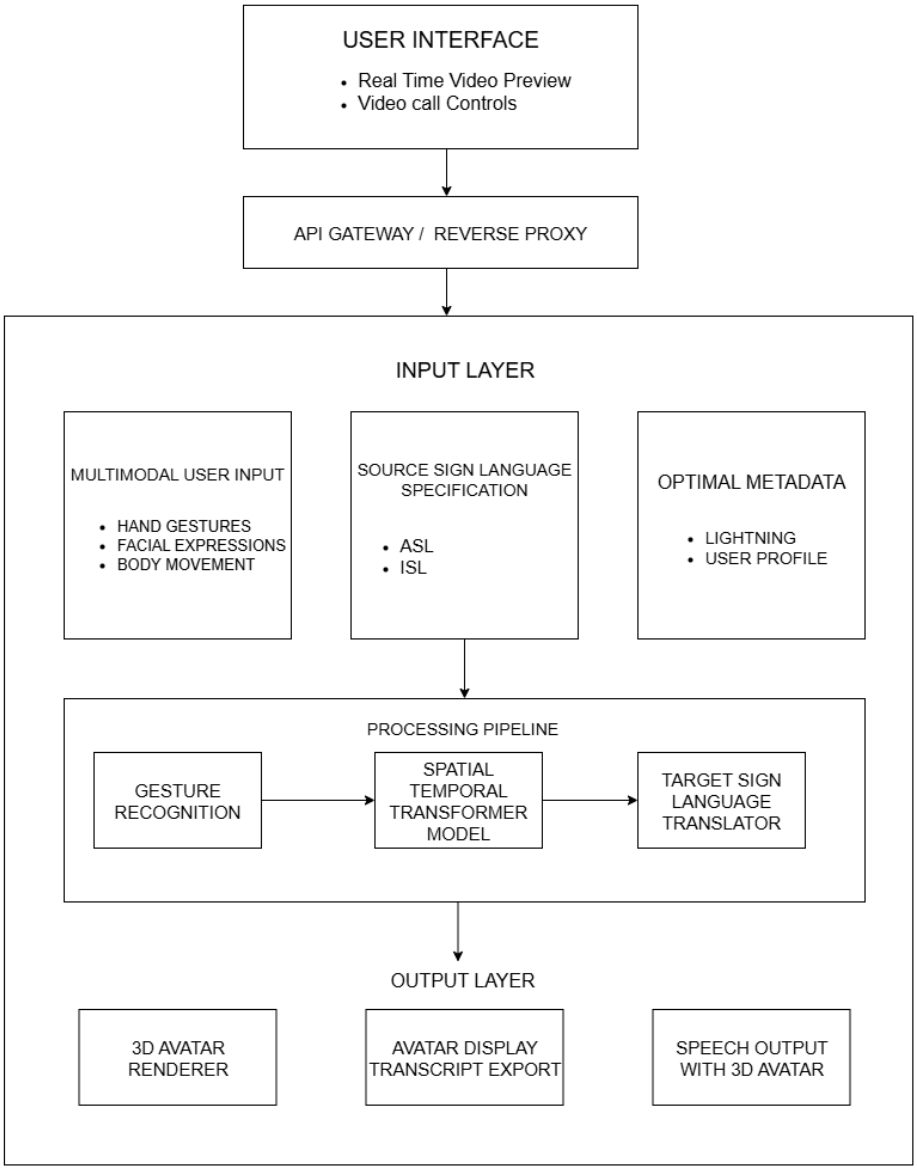

Figure 1. Architecture Diagram

The use case diagram represents the interaction between users, administrators, and the system in the Cross-Sign Language Translation System. It illustrates how different actors interact with various system functionalities to perform sign language translation.

Actors

- **User**: The primary actor who interacts with the system to perform sign language translation.
- **Admin**: Responsible for managing system settings, datasets, and models.
- **System**: Represents the internal processing unit that performs recognition and translation tasks.

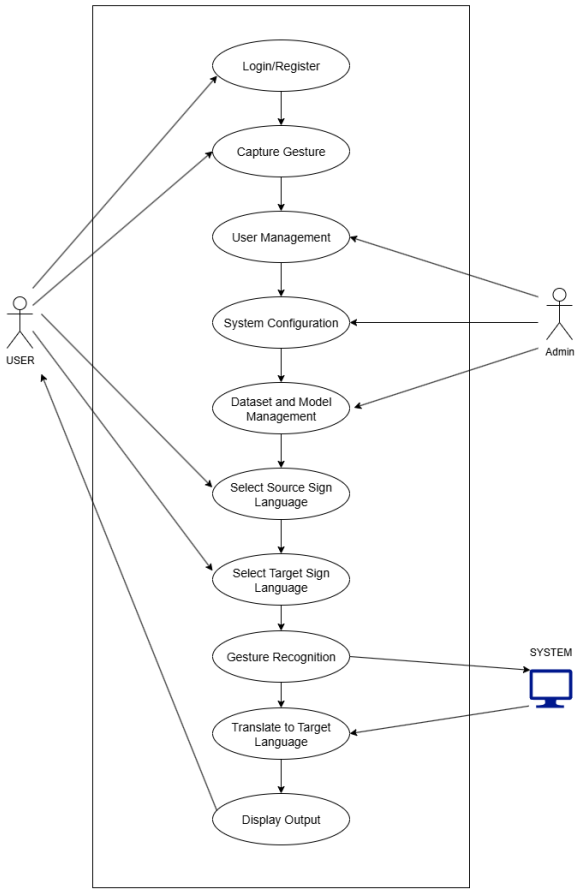

Figure 2. Use Case Diagram

1. Login/Register
The user begins by logging into the system or registering a new account to access the platform features.

2. Capture Gesture
The system captures real-time gestures from the user through a camera interface for further processing.

3. User Management
This function allows management of user-related data such as profiles and access control. The admin can also monitor and manage users.

4. System Configuration
The admin configures system settings such as parameters, preferences, and operational rules to ensure proper functioning.

5. Dataset and Model Management
The admin manages datasets and trained models used for gesture recognition and translation. This includes updating, training, and maintaining models.

6. Select Source Sign Language
The user selects the input sign language (e.g., ASL or ISL) that they are using.

7. Select Target Sign Language
The user selects the output sign language into which the gesture should be translated.

8. Gesture Recognition
The system processes the captured gestures using deep learning models to recognize the performed signs.

9. Translate to Target Language
The recognized gestures are translated into the selected target sign language using an intermediate representation.

10. Display Output

The final translated result is displayed to the user in the form of text, speech, or a 3D avatar.

VI. RESULTS

The proposed system was evaluated using both American Sign Language (ASL) and Indian Sign Language (ISL) datasets to assess its performance across different sign language representations. Key performance metrics, including training accuracy and validation accuracy, were monitored during the training process. The results are summarized in Table \ref{tab:results}. In addition to numerical results, graphical plots of accuracy and loss were analyzed to understand the model's learning behavior across epochs. These curves show that the model effectively learns relevant spatio-temporal gesture features while maintaining stable convergence and minimal overfitting.

A comparative analysis between ASL and ISL results indicates that the model adapts well to variations in hand shapes, motion patterns, and contextual differences present in different sign languages. The system also demonstrated efficient real-time inference with low latency and high responsiveness for both datasets.

Overall, the experimental results confirm the robustness, scalability, and effectiveness of the proposed system, making it suitable for real-world applications such as assistive communication, educational tools, and human-computer interaction systems.

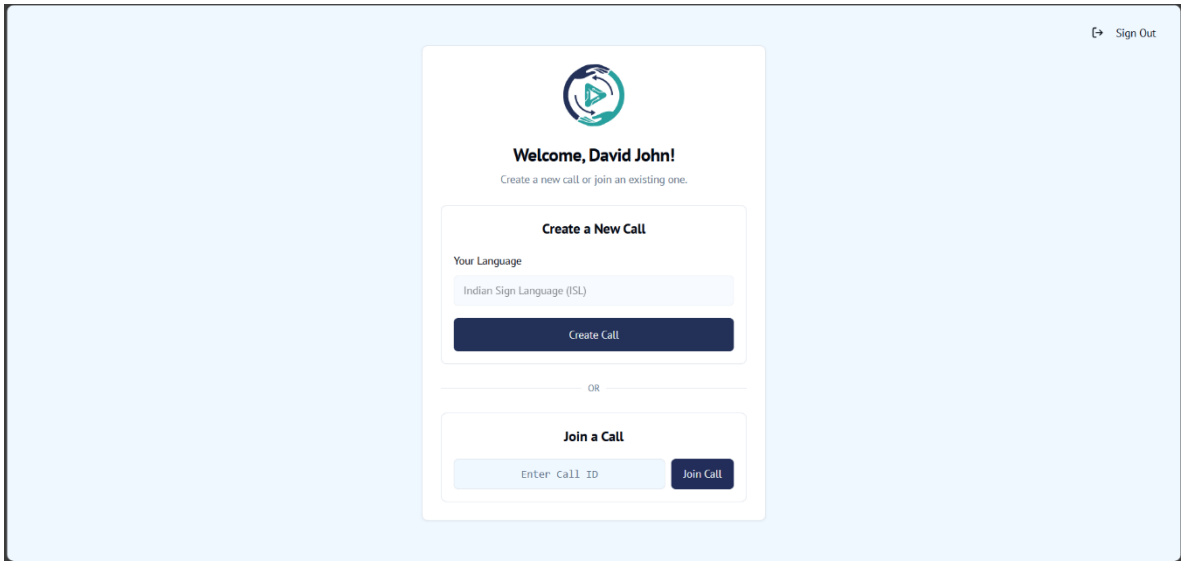

Fig 3. User Interface

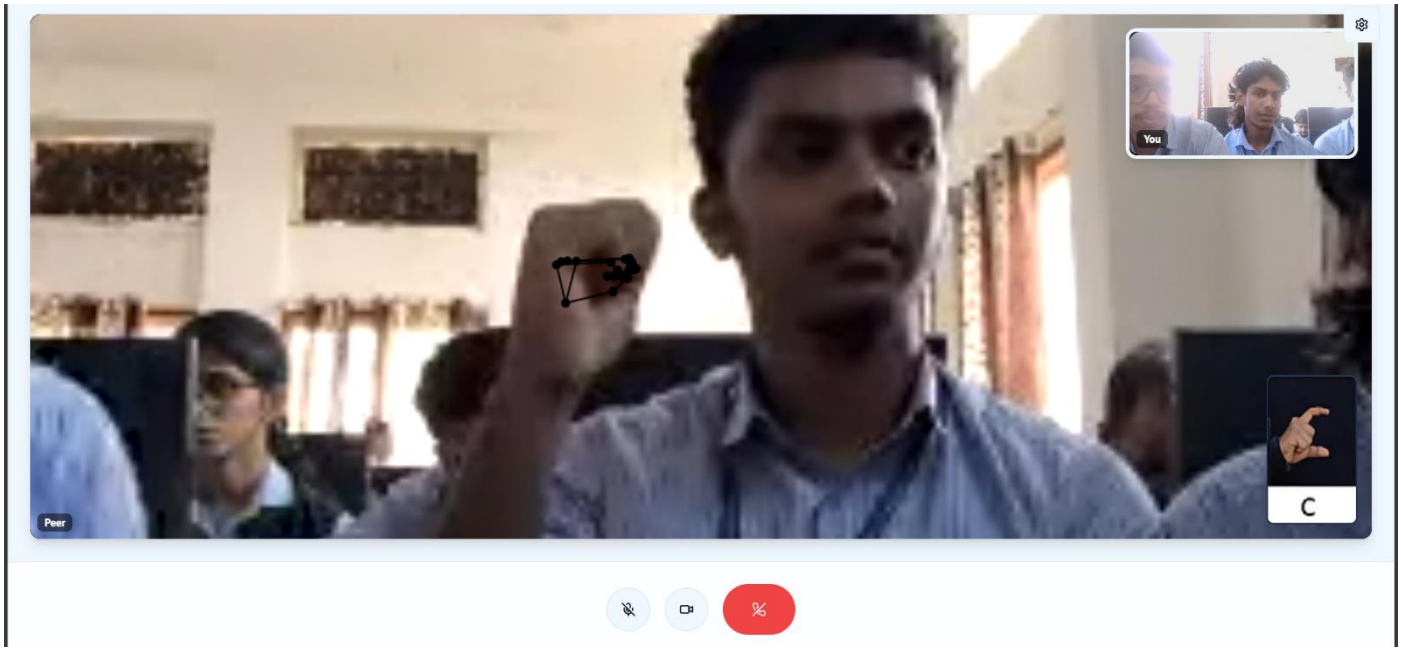

Fig 4 : Sign Translation

VII. CONCLUSION

This paper presented the design and implementation of a Cross-Sign Language Translation System aimed at enabling real-time communication between users belonging to different sign language communities. The proposed system integrates computer vision and deep learning techniques to accurately recognize continuous sign language gestures and translate them into a corresponding target sign language. Multimodal features such as hand gestures, facial expressions, and upper-body movements are captured using MediaPipe and OpenCV, ensuring that both manual and non-manual components of sign language are effectively represented. The use of a CNN-based gesture recognition model, combined with optimized training and preprocessing strategies, enables reliable extraction of discriminative visual features.

Experimental evaluation using ASL and ISL datasets sourced from Kaggle demonstrates that the proposed system achieves high recognition accuracy while maintaining real-time inference capability. The modular and layered system architecture allows efficient data flow from input capture to output generation and supports scalability and ease of maintenance. Overall, the proposed system addresses key limitations of existing single-language sign recognition approaches and provides a practical, efficient, and inclusive solution for cross-sign language communication in real-world video conferencing environments.

ACKNOWLEDGMENT

We would like to sincerely thank St. Thomas College of Engineering and Technology for their support. The resources provided through the Research and Development Lab were essential to the success of this project and gave us the environment we needed to turn this research into a working reality.

REFERENCES

[1] R. Yang, S. Sarkar, and B. L. Lo, "Sign language recognition with long short-term memory," *IEEE Access*, vol. 8, pp. 184386–184398, 2020.

[2] J. Pu, W. Zhou, and H. Li, "Iterative alignment network for continuous sign language recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4165–4174

[3] O. Koller, H. Ney, and R. Bowden, "Deep learning of mouth shapes for sign language," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 85–93.

[4] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2006, pp. 369–376.

[5] A. Vaswani et al., "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 5998–6008.

[6] S. Albanie, A. Vedaldi, and A. Zisserman, "Recognizing sign language using motion and appearance cues," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 5, pp. 1237–1250, May 2019.

[7] Z. Zhou, G. Zhao, and M. Pietikäinen, "Towards a practical lipreading system," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 7, pp. 1581–1596, Jul. 2019.

[8] T. Starner, J. Weaver, and A. Pentland, "Real-time American Sign Language recognition using desk and wearable computer-based video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1371–1375, Dec. 1998.

[9] S. Camgoz, O. Koller, S. Hadfield, and R. Bowden, "Neural sign language translation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7784–7793.

[10] J. Huang, W. Wang, and T. Tan, "Continuous sign language recognition: A review of recent advances," *Pattern Recognition*, vol. 115, p. 107870, 2021.

[11] J. Han, Z. Cao, and F. Li, "A comprehensive survey on sign language recognition using deep learning," *ACM Computing Surveys*, vol. 55, no. 2, pp. 1–34, 2023.

[12] L. Pigou, S. Dieleman, P. Kindermans, and B. Schrauwen, "Sign language recognition using convolutional neural networks," in *Proceedings of the European Conference on Computer Vision Workshops (ECCVW)*, 2018, pp. 572–578.

[13] S. Stoll, S. P. Camgoz, H. Koller, and R. Bowden, "Sign language recognition using subUNets," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 307–315.

[14] Y. Huang, Y. Guo, J. Pan, and X. Wang, "Hand gesture recognition for multi-culture sign language using graph and general deep learning network," *IEEE Transactions on Multimedia*, vol. 25, pp. 4021–4034, 2023.

[15] M. Zhang, Q. Wu, and Y. Liu, "Efficient fusion of spatio-temporal features for sign language recognition using 3D-CNN and GCN," *Pattern Recognition Letters*, vol. 170, pp. 73–80, 2023.

[16] A. Pandey, R. Kumar, and S. Srivastava, "GRASTA-Net: A lightweight graph attention sign language translation architecture for Indian sign language recognition," *Expert Systems with Applications*, vol. 214, p. 119083, 2023.

[17] S. Ghosh et al., "Indian sign language detection for real-time translation using machine learning," *arXiv preprint arXiv:2507.20414*, 2025.

[18] C. Lugaresi et al., "MediaPipe: A framework for building perception pipelines," *arXiv preprint arXiv:1906.08172*, 2019.

[19] M. Abadi et al., "TensorFlow: A system for large-scale machine learning," in *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2016, pp. 265–283.

[20] Google, "TensorFlow Lite: On-device machine learning." [Online]. Available: https://www.tensorflow.org/lite