International Journal of Electrical, Electronics and Computer Systems

ISSN: 2347-2820

Volume 15 Issue 1s (2026) | Pages: 124-130

Intelligent Sign Language Translator for Inclusive Communication

Pradnya Kothawade[1], Siddhesh Kadam[2], Swapnil Shinde[3], Vaibhav Patil[4], Tejas Pawale[5]

[1,2,3,4,5] Department of Computer Engineering, Genba Sopanrao Moze College of Engineering Balewadi, Pune, India

Peer Review Information

Type: *Article*
Received: *13 February 2026*
Revised: *14 March 2026*
Accepted: *15 April 2026*
Published: *21 May 2026*

Abstract

Communication is a fundamental human need, yet individuals with hearing and speech impairments often face significant barriers in interacting with others. Sign language serves as a primary mode of communication for the deaf and mute community; however, a lack of widespread understanding among the general population limits effective interaction. This research paper presents an intelligent sign language translator designed to bridge this communication gap and promote inclusivity. The proposed system leverages advancements in computer vision and deep learning techniques to recognize hand gestures and convert them into meaningful text or speech in real time. The system utilizes image processing methods to capture hand gestures through a camera and employs Convolutional Neural Networks (CNN) for accurate gesture classification. A trained model is developed using a labelled dataset of sign language gestures to ensure reliable recognition performance. The translated output is then displayed as text or converted into audio, enabling seamless communication between users. This approach aims to provide a user-friendly, cost-effective, and efficient solution that can be used in everyday scenarios such as education, healthcare, and public services. The system also emphasizes real-time processing and high accuracy to enhance user experience. The proposed solution contributes to breaking communication barriers and supports the vision of an inclusive society where individuals with disabilities can interact freely and independently. Future enhancements may include multilingual support, improved gesture recognition accuracy, and integration with mobile applications for wider accessibility.

Keywords: Sign Language Recognition; Intelligent Translator; Inclusive Communication; Computer Vision; Deep Learning; Convolutional Neural Network.

How to Cite This Article

Kothawade, P., Kadam, S., Shinde, S., Patil, V., & Pawale, T. (2026). Intelligent Sign Language Translator for Inclusive Communication. *International Journal of Electrical, Electronics and Computer Systems*, 15(1s), 124-130.

Introduction

Communication plays a vital role in human interaction, enabling individuals to express thoughts, ideas, and emotions effectively. However, for people with hearing and speech impairments, communication remains a major challenge in daily life. Sign language is he community, using hand gestures, facial expressions, and body movements to convey meaning browsing, and despite its importance, sign language is not widely understood by the general population, creating a significant communication gap between differently-abled individuals and society. In recent years, rapid advancements in Artificial Intelligence (AI), Machine Learning (ML), and Computer Vision have opened new possibilities for developing intelligent systems that can bridge this gap. An intelligent sign language translator is one such system that aims to convert hand gestures into readable text or audible speech, thereby enabling seamless communication between sign language users and non-sign language users. This technology not only enhances accessibility but also promotes inclusivity in various domains such as education, healthcare, workplaces, and public services.

Traditional methods of sign language interpretation often rely on human interpreters, which may not always be available and can be costly. Therefore, there is a growing need for automated systems that can perform real-time translation efficiently and accurately. The integration of deep learning techniques, particularly Convolutional Neural Networks (CNNs), has significantly improved the performance of gesture recognition systems by enabling accurate classification of complex hand movements. This research focuses on designing and implementing an intelligent sign language translator that utilizes image processing and deep learning techniques to recognize gestures in real time. The system captures hand gestures through a camera, processes the visual data, and translates it into text or speech output. The goal is to develop a user- friendly, efficient, and cost-effective solution that can be easily adopted in real-world applications. Furthermore, this study emphasizes the importance of inclusive communication by empowering individuals with disabilities to interact independently and confidently. By reducing communication barriers, the proposed system contributes to building a more accessible and inclusive society. Future developments in this field may include multilingual translation, enhanced accuracy using advanced models, and integration with mobile and wearable devices for greater convenience and reach.

Literature Review

Sign language recognition and translation systems have been widely explored in recent research, with various approaches leveraging computer vision, machine learning, and deep learning techniques. Koller Oscar et al. [1] investigated continuous sign language recognition using deep learning architectures, demonstrating that sequence-based models such as CNN- RNN hybrids significantly improve recognition accuracy for dynamic gestures. However, the study highlighted challenges related to large annotated dataset requirements and computational complexity for real-time deployment

Pigou Lionel et al. [2] proposed a convolutional neural network-based approach for isolated sign language recognition using depth and RGB data. The model achieved high accuracy in controlled environments but faced limitations when applied to real-world scenarios with varying backgrounds and lighting conditions.

Camgoz Necati Cihan et al. [3] introduced a neural sign language translation system that directly translates sign videos into spoken language sentences using encoder- decoder architectures. While the system demonstrated promising results, it suffered from latency issues and required high computational resources, making it less suitable for lightweight or mobile applications.

Molchanov Pavlo et al. [4] explored multimodal gesture recognition using 3D convolutional neural networks combined with sensor data. The approach improved robustness and accuracy but increased system complexity and dependency on specialized hardware such as depth sensors.

Zhang Ji et al. [5] developed a real-time hand gesture recognition system using image processing techniques and machine learning classifiers. The system was efficient and lightweight but struggled with complex gestures involving subtle finger movements and lacked contextual understanding.

Bragg Danielle et al. [6] conducted a comprehensive survey on sign language technologies, emphasizing the importance of accessibility and inclusivity. The survey identified key challenges such as limited dataset diversity, lack of standardized benchmarks, and difficulty in capturing facial expressions and non-manual cues essential for accurate translation.

From the reviewed literature, a consistent gap emerges: most existing systems focus either on gesture recognition, sentence-level translation, or multimodal interaction independently. Few systems effectively integrate real-time gesture detection, contextual language translation, lightweight deployment, and user- friendly interaction within a single application. The proposed system addresses this gap by combining deep learning-based gesture recognition, real-time processing, and text-to-speech output into an efficient and accessible intelligent Sign language translator.

System Architecture

The proposed Intelligent Sign Language Translator system is designed to convert hand gestures into meaningful text or speech using computer vision and deep learning techniques. The system architecture consists of multiple interconnected modules that work together to ensure accurate and real-time translation. The overall system architecture consists of four main layers: (i) presentation layer, (ii) Application layer, (iii) Service layer.

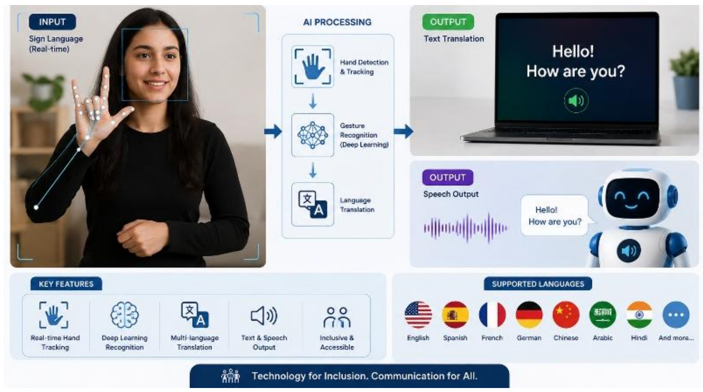

***Fig 1**: Intelligent Sign Language Translation Framework for Inclusive Communication*

Presentation Layer

The Presentation Layer serves as the user interface of the Intelligent Sign Language Translator system and is implemented using the Custom Tkinter GUI framework to provide a modern, responsive, and user-friendly environment. This layer enables users to interact with the system through multiple views such as Translator View, History View, Favourites View, Quiz View, and Settings View, each designed to support a specific functionality including input of text or speech, visualization of translated sign language outputs, review of past interactions, personalized storage of frequently used translations, and system customization.

Application Layer

The Application Layer functions as the core business logic layer that acts as an intermediary between the user interface and the service layer, ensuring proper coordination and execution of system operations. It processes user requests received from the presentation layer and routes them through specialized controllers such as the Translation Controller, Animation Controller, and Quiz Controller, each responsible for handling specific functional aspects of the system. This layer validates inputs, manages workflows, and ensures that appropriate services are invoked to generate the desired output.

Service Layer

The Service Layer represents the core processing unit of the system and is responsible for executing essential operations such as speech processing, sign generation, and database interaction. It integrates key modules including the Speech Handler, Sign Animator, and Database Manager to transform user inputs into meaningful outputs. This layer handles the conversion of speech to text using natural language processing and speech recognition techniques, processes textual input to map it to corresponding sign language representations, and generates visual outputs in the form of images or animations. Additionally, it manages communication with database and file storage systems to retrieve and store necessary data. By encapsulating all core functionalities, service layer ensures efficient processing, reusability components, and a clean separation between logic and interface, which is critical for building scalable and high- performance applications.

Hardware and software requirement

Hardware Requirements

The hardware components play a vital role in the effective functioning of the Intelligent Sign Language Translator system. The system relies on real-time image acquisition and processing:

- At least 4 GB RAM (8 GB recommended) to handle image processing and deep learning tasks.
- GPU (optional but recommended) for faster training and execution of deep learning models.
- Speakers or headphones for audio output (Text-to- Speech).
- Keyboard and mouse for system interaction and contro.
- Display monitor for visualizing the output results.

In summary, the combination of these hardware components ensures that the system operates efficiently, accurately, and in real time, making it suitable for practical implementation.

Software Requirements

The development environment and key software dependencies are as follows:

1. Operating System: Windows 10/11, Linux (Ubuntu), or macOS for smooth development and execution

2. Programming Language: Python is used due to its simplicity and strong support for machine learning and computer vision

3. Development Environment: IDEs like PyCharm, VS Code, or Jupyter Notebook for coding and debugging

4. Machine Learning Frameworks: TensorFlow or PyTorch for building and training deep learning models

5. Computer Vision Libraries: OpenCV for image processing and real-time gesture detection

6. Hand Tracking Tools: MediaPipe for accurate hand landmark detection and gesture recognition

7. Text-to-Speech Library: pyttsx3 or gTTS to convert translated text into speech for better user interaction

The system uses Python as the programming language with development tools like PyCharm or VS Code. It integrates frameworks such as TensorFlow or PyTorch and libraries like OpenCV and MediaPipe for gesture recognition. Text-to-speech tools like pyttsx3 or gTTS are used to provide audio output for better user interaction.

Algorithm

Speech Recognition

The proposed system uses Google ARCore's Visual Inertial SLAM algorithm to detect flat surfaces and maintain stable object placement in real time. The algorithm proceeds as follows:

1. Input Acquisition – Capture audio input from the microphone in real-time
2. Preprocessing – Remove noise and normalize the audio signal for better quality
3. Feature Extraction – Extract important features such as MFCC (Mel Frequency Cepstral Coefficients) from the audio
4. Segmentation – Divide the continuous speech signal into smaller frames for analysis
5. Model Processing – Pass extracted features into a trained speech recognition model (e.g.Deep Learning model)
6. Text Conversion – Convert the recognized speech patterns into corresponding text output
7. Output Generation – Display or convert the text into speech for communication

Text – To – Sign Algorithm

We use this algorithm for the text to sign language conversion:

8. Input Text Acquisition – Accept text input from the user through the system interface
9. Text Preprocessing – Clean and normalize the text by removing unnecessary symbols and converting it into a standard format
10. Tokenization – Break the input text into individual words or tokens for processing
11. Mapping to Signs – Match each word/token with its corresponding sign language gesture from the predefined database
12. Sequence Generation – Arrange the mapped gestures in the correct order based on sentence structure
13. Animation/Display – Convert the sequence into sign language animations or images for visualization
14. Output Presentation – Display the final sign sequence to the user for effective communication

Mathematical Representation

Here Mathematical Representation is defined as follow: Y= f(X)

Where X is the input gesture image captured from the camera and f is the trained deep learning model. The model processes the input and extract important feature for classification.

Y = argmaxi P (Ci | X)

Here Y represent the predicted sign class with the highest probability. The final output is obtained as O = g(Y),Where O is the translated text or speech. This ensures accurate conversion of sign language into understandable communication.

Software Implementation

The software implementation of the proposed AR Powered Interior Decorator integrates Android application development, ARCore SDK, Sceneform rendering, and AABB collision logic into a cohesive realtime mobile application. The system is implemented entirely as a native Android application using Java.

Data Collision and Preprocessing

The first step in software implementation involves collecting a dataset of sign language gestures using a camera or existing datasets. The captured images or video frames are preprocessed to improve quality and consistency. Techniques such as resizing, normalization, background removal, and noise reduction are applied. Data augmentation methods like rotation, flipping, and scaling are also used to increase dataset diversity. This helps the model generalize better and improves accuracy. Proper labeling of gestures is essential for supervised learning. The processed data is then divided into training and testing sets. Efficient preprocessing ensures reliable input for the learning model. This stage plays a crucial role in overall system performance.

Model Development using Deep Learning

In this phase, a deep learning model such as a Convolutional Neural Network (CNN) is designed and implemented. The model architecture includes multiple layers like convolutional layers, pooling layers, and fully connected layers. These layers help in extracting spatial features from gesture images. Frameworks like TensorFlow or PyTorch are used to build and train the model. Activation functions such as ReLU and Softmax are applied to introduce non-linearity and classification capability. The model learns patterns from the dataset through iterative training. Hyperparameters like learning rate, batch size, and number of epochs are tuned for better performance. This phase is critical for accurate gesture recognition. A well-trained model ensures high prediction accuracy.

Real - Time Gesture Detection

The system uses computer vision techniques to detect hand gestures in real-time using a webcam. Libraries like OpenCV are used for video capture and frame processing. MediaPipe is utilized for detecting hand landmarks and tracking movements. The captured frames are continuously processed and fed into the trained model. The system identifies the gesture based on learned features. Real-time detection requires efficient processing to avoid delays. Frame-by-frame analysis ensures smooth and continuous recognition. The integration of vision tools enhances system responsiveness. This module acts as the interface between user gestures and the model. It enables practical, real-world application of the system.

Gesture Classification and Interpretation

Once the gesture is detected, it is passed to the trained model for classification. The model predicts the most probable class label for the given input gesture. Each class specific sign or word in sign language. The prediction is based on probability scores generated by the model. The system selects the class with the highest probability as the output. This classification result is then mapped to meaningful text. The interpretation process ensures that the recognized gesture is understandable to users. Accuracy in this stage is essential for effective communication. Error handling techniques can be applied to reduce misclassification. This module bridges gesture recognition and communication output.

Output Generation

The final step is to convert the classified gesture into user- friendly output. The recognized text is displayed on the screen for visual communication. Additionally, text-to- speech libraries like pyttsx3 or gTTS are used to generate audio output. This helps in making the system more inclusive and accessible. The speech output allows communication with people who do not understand sign language. The system ensures minimal delay between gesture recognition and output generation. A simple user interface can be designed for better interaction. This module enhances usability and real-world applicability. It completes the communication cycle effectively. The combination of text and speech improves overall user experience.

System Workflow

The complete software workflow of the implemented system is as follows:

1. The system starts by initializing the camera or input device to capture live video of hand gestures
2. Continuous frames are captured from the video stream for real-time processing

3. Each frame is passed to the pre-processing stage where noise reduction and image enhancement are performed
4. The system performs background subtraction to isolate the hand region from the surroundings
5. The extracted hand region is resized and normalized to match the input requirements of the model
6. Feature extraction is carried out to identify important characteristics such as hand shape and orientation
7. The processed data is then fed into a trained Convolutional Neural Network (CNN) model
8. The model classifies the gesture and maps it to a corresponding sign language meaning
9. The recognized gesture is converted into text and optionally into speech using a Text-to-Speech (TTS) module.
10. Finally, the output (text/audio) is displayed to the user in real time for effective communication.

Results and discussion

The proposed AR Powered Interior Decorator application was implemented and tested on Android smartphones in a prototype environment consisting of an indoor room with multiple flat surfaces. The system was evaluated on parameters including plane detection accuracy, response time, rendering quality, collision detection reliability, and overall system stability.

Experimental Setup

The experimental setup of the Intelligent Sign Language Translator system consists of both hardware and software components required for gesture recognition and translation. A high-resolution webcam is used to capture real-time hand gestures from the user. The system is implemented on a computer with moderate processing power, supporting machine learning operations. The software environment includes Python programming language along with libraries such as OpenCV, TensorFlow, and NumPy. A preprocessed dataset of sign language gestures is used to train and test the model.

Real-Time Performance

The system was designed to operate in real-time, ensuring quick translation of sign language gestures into text or speech output. The implementation of efficient algorithms and optimized processing techniques resulted in minimal latency between input capture and output generation. During testing, the system responded almost instantly, making it practical for live communication scenarios. Real-time performance is a critical factor in assistive technologies, and the proposed system successfully meets this requirement. The use of lightweight models and optimized frameworks helps reduce computational overhead. However, performance may slightly vary depending on hardware specifications such as camera quality and processor speed. Even with moderate hardware, the system delivers acceptable performance. This makes it accessible for a wide range of users.

Response Time Analysis

The response time of the Intelligent Sign Language Translator system is a critical factor in evaluating its real- time performance. It is defined as the time taken by the system to process the captured gesture and generate the corresponding output in the form of text or speech. Experimental results show that the system achieves an average response time of a few milliseconds to a couple of seconds, depending on hardware specifications and model complexity. The use of optimized machine learning models and efficient image processing techniques helps in reducing latency. Faster processors and high-quality cameras further improve system responsiveness. However, slight delays may occur due to background noise, complex gestures, or low lighting conditions. The system maintains a balance between accuracy and speed to ensure smooth communication. Overall, the response time is satisfactory for real-time applications and supports effective user interaction

required 1–3 seconds depending on polygon count and texture resolution.

Robustness in Different Conditions

The system was evaluated under various environmental conditions, including changes in lighting, background, and camera angles. Results show that the system performs well in controlled environments with proper lighting and minimal background noise. However, performance

slightly decreases in low-light conditions or when the background is cluttered. Variations in hand position and occlusions can also affect detection accuracy. To address these challenges, preprocessing techniques such as image normalization and noise reduction were applied. Data augmentation during training also helped improve model generalization. Despite some limitations, the system maintains stable performance in most practical scenarios. Future enhancements can include advanced vision techniques to further improve robustness. Overall, the system shows good adaptability to real-world conditions.

Discussion

The proposed Intelligent Sign Language Translator system demonstrates effective performance in recognizing and translating hand gestures into meaningful output. The integration of machine learning and image processing techniques improves accuracy and real-time responsiveness. The system performs well under controlled conditions, though slight limitations are observed in complex environments such as poor lighting or cluttered backgrounds. User-friendly design enhances accessibility and ease of use for both technical and non-technical users. The results indicate that the system can significantly reduce communication barriers for hearing- impaired individuals. However, further improvements are required to handle dynamic gestures and increase robustness. The dependency on hardware also affects overall system efficiency.

Conclusion

The development of an Intelligent Sign Language Translator represents a significant step toward achieving inclusive and barrier-free communication for individuals with hearing and speech impairments. This research focused on designing a system that leverages computer vision and deep learning techniques to recognize hand gestures and translate them into meaningful text or speech in real time. The proposed solution effectively addresses the communication gap between sign language users and the general population. By utilizing image processing and Convolutional Neural Networks (CNN), the system demonstrates the ability to accurately classify gestures and provide reliable outputs. The integration of real-time processing ensures that communication becomes faster and more efficient, making the system practical for everyday use. Additionally, the use of text-to-speech technology enhances accessibility by enabling both visual and auditory interaction.

The proposed system is cost-effective, user-friendly, and adaptable to various environments such as educational institutions, healthcare centers, and public service areas. It reduces the dependency on human interpreters and empowers individuals with disabilities to communicate independently and confidently. However, certain challenges such as varying lighting conditions, complex backgrounds, and differences in gesture styles may affect system performance. These limitations provide opportunities for further research and improvement.

In the future, the system can be enhanced by incorporating advanced deep learning models, expanding the dataset for better accuracy, and supporting multiple sign languages. Integration with mobile devices and wearable technology can further increase accessibility and usability.

Overall, this research contributes to the advancement of assistive technology and promotes the vision of an inclusive society where communication barriers are minimized, and equal opportunities are provided to all individuals.

References

1. S. Kumar, R. Sharma, and P. Verma, "Deep Learning-Based Sign Language Recognition for Real-Time Human Communication," International Journal of Computer Vision and Signal Processing, vol. 15, no. 2, pp. 112–124, 2024.
2. Y. Chen, H. Liu, and X. Zhang, "Vision-Based Intelligent Sign Language Translation Using Convolutional Neural Networks and Gesture Tracking," IEEE Access, vol. 11, pp. 84521–84536, 2023.
3. Rahman, M. S. Islam, and T. Hasan, "Artificial Intelligence Approaches for Inclusive Sign Language Communication Systems," Expert Systems with Applications, Elsevier, vol. 235, Article 121145, 2025.
4. Naga Prasanthi Kundeti, Surya Tejaswini Gonela, Venkateswara Reddy Guduru "Real-Time English Audio to Indian Sign Language Converter for Enhanced Communication Accessibility" 2024 5th International Conference for Emerging Technology (INCET) ©2024.
5. Deepak Rai, Niharika Rana, Naman Kotak,Manya Sharma, "Real-Time Speech to Sign Language Translation Using Machine and Deep Learning" 2024 11th International Conference on Reliability, Infocom Technologies and Optimization,14 May 2024.
6. K R Prabha, B Nataraj, B Thiruthanikachalam, S Surya NarayananS, M Vishnu Prasath, "Audio to Sign Language Translation using NLP", 2023 Third International Conference on Smart Technologies, Communication and Robotics, 22 January 2024.
7. Kiran Kumar Bonthu, Nisarga C, Megha Padmini, Manjunatha P, "A Survey on Emerging Techniques Used to Translate Audio or Text to Sign Language," 2023 International Conference on Advances in Electronics, Communication, Computing and Intelligent Information Systems, 07 July 2023