

# Multimodal Indian Sign Language Translator using Gesture Recognition and Voice Input

Dr. A. V. Sable<sup>1</sup>, Purvesh P. Savalakhe<sup>2</sup>, Mayuri S. Pache<sup>3</sup>, Om S. Naringe<sup>4</sup>, Shrawani S. Bonde<sup>5</sup>, Harsh S. Kapile<sup>6</sup>

<sup>1</sup>Assistant Professor Department of Computer Science and Engineering, Sipna College of Engineering and Technology,  
Amravati, Maharashtra, India

<sup>23456</sup>Student's, Department of Computer Science and Engineering, Sipna College of Engineering and Technology,  
Amravati, Maharashtra, India

*Abstract* - In this paper, we present the design and implementation of a Sign Language Detection System aimed at improving communication between hearing-impaired individuals and the general population. Communication barriers often arise due to the lack of understanding of Indian Sign Language (ISL), making it difficult for non-sign language users to interact effectively. To address this issue, the proposed system provides a multimodal communication platform that enables users to interact using text, voice, and hand gestures.

The system is designed with multiple functionalities, including text-to-sign conversion, sign-to-text recognition, and voice-to-sign translation, along with an integrated messaging feature. Users can send and receive messages in both textual and sign-based formats, enhancing accessibility and usability. The text-to-sign module converts user-input text into a sequence of corresponding ISL alphabet images, allowing users to visually interpret words in sign language. The voice-to-sign module utilizes speech recognition techniques to convert spoken input into text, which is further processed into sign representations. The sign-to-text module captures real-time hand gestures through a camera and translates them into textual output, enabling two-way communication. For the implementation of the system, we used Python and Django as the backend framework to handle application logic and data processing, while HTML, CSS, and Bootstrap were used to develop a responsive and user-friendly interface. For gesture recognition, MediaPipe is employed to detect and extract hand landmarks in real-time. These landmarks are then used as input features for a Random Forest machine learning algorithm, which classifies gestures into corresponding alphabets or words.

Overall, the proposed Sign Language Detection System demonstrates an effective integration of machine learning, computer vision, and web technologies to create

A practical and accessible solution for ISL communication. The system has the potential to bridge communication gaps and promote inclusivity for the hearing-impaired community.

*Keywords:* Indian Sign Language (ISL), Gesture Recognition, MediaPipe, Random Forest Algorithm, Sign-to-Text Conversion, Voice-to-Sign Translation, Real-Time Communication

## I. INTRODUCTION

Communication is a very important part of our daily life. We use it to share our thoughts, ideas, and feelings with others. But for people who are hearing-impaired or unable to speak, communication becomes difficult. They mainly use sign language to express themselves. However, the problem is that

most people do not understand sign language, which creates a gap between them and normal users.

To solve this problem, we developed a Sign Language Detection System that makes communication easier for everyone. Our system allows users to communicate in different ways. A user can type text and see it converted into sign images, or they can show hand gestures and the system will convert them into text. We have also added a voice feature, where users can speak, and the system converts their speech into sign language. This makes communication possible even if someone does not know sign language completely.

We used simple and effective technologies to build this system. The backend is developed using Python and Django, while the user interface is created using HTML, CSS, and Bootstrap to make it easy to use. For recognizing hand gestures, we used MediaPipe, which helps in detecting hand movements, and a Random Forest algorithm to identify the correct signs. We also tested some deep learning models like MobileNet and EfficientNet B0, but after testing, we found that Random Forest works better for our system, giving around 97% accuracy and faster results.

Our system works well for basic communication and can be useful for students, teachers, and anyone who wants to learn or use sign language. Although it is not perfect and may face some limitations, it still provides a simple and practical solution to reduce communication barriers. The main goal of our project is to make communication easier, faster, and more accessible for everyone.

## II. LITERATURE REVIEW

Many people have tried to make systems that can understand Indian Sign Language (ISL). Sign language is very important because not everyone can hear or speak. But normal people don't always know it, so it is hard for both sides to communicate. Many researchers tried different ways to make computers understand signs.

Some researchers used CNN models to recognize signs from images. For example, [1] made a system that can translate ISL using CNN. They got good accuracy but they needed a lot of pictures to train the model. Another work [2] tried object detection to find hands in images and then make signs. It was simpler but sometimes it did not work well in different light conditions.

Some people wrote reviews and surveys about different ways to detect signs. Like [3] and [5] studied many techniques and wrote about their advantages and disadvantages. These papers did not make a real system but gave a good idea about what works and what doesn't. It helped us understand which methods are better for our project.

Deep learning is also very popular. Many researchers tried CNN, MobileNet, EfficientNet, and other networks. For example, [4] used deep learning techniques for ISL recognition and got good results. Another study [6] made a real-time translation system using deep learning, which could translate gestures faster. Papers [7] and [8] also added speech and text output to make communication easier. These systems were very useful but some of them were complex and needed more computer power.

Some works tried advanced methods like graph networks [9] or YOLO with transformers [13]. These models gave very high accuracy, but they were very hard to run and needed big computers. Other papers [10] and [14] focused on machine learning models like Random Forest, which were simpler but still gave good results. Random Forest was fast and worked well with gesture features from Media-Pipe.

Some researchers also created datasets to help train these models. Like [11] and [20] provided images of ISL for training. Without these datasets, it would be hard to make an accurate system. Other studies [12], [15], [16], [17], and [18] focused on real-time recognition, mobile platforms, and gesture-to-speech systems. These works inspired us to combine text to sign, sign to text, and voice input in one system.

Finally, some projects tried voice to sign translation using NLP [19]. This is helpful for people who want to speak and see signs at the same time. But accuracy is still not perfect, especially in noisy places

Various models such as CNN, MobileNet, EfficientNet, Random Forest, YOLO, and Graph Networks are used for sign language recognition, each with its own advantages and limitations. Some models are simple but less accurate, while others provide high accuracy but are complex and computationally expensive. The combination of Random Forest and MediaPipe is simple, fast, and suitable for real-time recognition. Datasets play a crucial role in training and testing, as model performance depends heavily on data quality. Most existing systems focus on a single function like

text-to-sign or sign-to-text, rather than providing a complete integrated solution.

In our project, we tried to combine the best things from past research. We used Random Forest for high accuracy, Media-Pipe for real-time gesture detection, and also text, voice, and sign integration. This way, our system is easy to use and works well for basic communication.

### III. PROPOSED METHODOLOGY

In this project, we developed a system to improve communication using Indian Sign Language. The main objective is to allow users to send messages, view sign representations, and convert voice into signs in a simple and efficient way. The system is divided into two main parts: Admin Panel and User Panel.

The Admin Panel is used to manage users and maintain the dataset, including adding or updating sign images. The User Panel allows users to register, log in, send messages, and use different input methods such as text, voice, and hand gestures. The system includes three main modules: Text-to-Sign, Voice-to-Sign, and Sign-to-Text. These modules work together to provide real-time communication by converting input data into the desired output format using MediaPipe and the Random Forest algorithm.

#### 3.1 Admin Panel

The Admin Panel is responsible for managing and controlling the overall system. The admin can securely log in to access all system functionalities. It allows the admin to monitor registered users, view their details, and track user activities such as messages exchanged within the system. The admin can also manage and maintain the dataset used for training the models, including adding new images, updating existing data, or removing incorrect entries. This helps in improving the accuracy and performance of the system. Additionally, the admin ensures that the system remains organized, up-to-date, and functions smoothly.

#### 3.2 User Panel

The User Panel is designed for general users to communicate using sign language. Users can register and log in to the system, view their inbox, and send messages to other users. The system supports multiple communication methods, including text, voice, and hand gestures.

Users can send messages in text form, which are converted into sign images. They can also use voice input, where speech is converted into text and then into signs. Additionally, users can perform hand gestures in front of a camera, and the system converts them into text in real-time.

Overall, the User Panel provides a simple and flexible interface for easy communication using different input methods.

### 3.3 Text to Sign Module

In this module, users can input any text or sentence through the system interface. The entered text is first processed and converted into a standardized format, after which it is divided into individual characters. For example, a word like “Hi” is split into the letters H and I. The system then searches for the corresponding images of each letter from the A–Z Indian Sign Language (ISL) dataset.

Once the matching images are found, they are displayed sequentially in the correct order to represent the complete word or sentence in sign language form. This step-by-step visual representation helps users easily understand how each character is expressed using hand gestures.

### 3.4 Voice to Sign Module

In this module, users can interact with the system using voice input through a microphone. The system captures the spoken words and processes them using speech recognition libraries in Python to convert the audio input into text format. Once the speech is successfully converted into text, it is forwarded to the Text-to-Sign module for further processing.

The Text-to-Sign module then splits the text into individual characters and displays the corresponding Indian Sign Language images in sequence. This allows users to visualize spoken words in the form of sign language.

This module is particularly useful for users who find typing difficult or want to communicate more quickly. It provides a convenient and efficient way to convert speech into sign language, making the overall system more flexible and user-friendly.

### 3.5 Sign to Text Module

This module converts hand gestures into text in real-time, enabling effective communication between sign language users and others. The system uses a camera to capture hand gestures continuously and processes them frame by frame. MediaPipe is used to detect and extract important hand landmarks such as finger positions and joint coordinates, which help in accurately identifying the gesture.

The extracted landmark data is then passed to a Random Forest algorithm, which is trained to classify different gestures into their corresponding letters or words. Once the gesture is recognized, the system displays the output as text on the screen instantly. This module provides fast and reliable results, making it useful for real-time communication,

although accuracy may be affected by complex gestures or improper hand positioning.

### 3.6 Models Used

CNN using MobileNet, CNN using EfficientNet B0, and Random Forest with MediaPipe. The MobileNet model was trained on an ISL dataset from Kaggle and achieved an accuracy of around 85–89%. EfficientNet B0 provided slightly better performance with an accuracy of 90–92%, but required more computational resources. The Random Forest model, combined with MediaPipe for hand landmark detection, achieved the highest accuracy of approximately 97% and showed faster performance in real-time conditions. Based on these results, the Random Forest with MediaPipe approach was selected as the main model due to its simplicity, high accuracy, and efficient processing speed. Although CNN-based models provide good results, they require more computation time and resources, making them less suitable for real-time applications compared to the chosen approach.

### 3.7 Overall Workflow

1. User logs in and chooses text, voice, or sign input.
2. If text, system shows sign images.
3. If voice, system converts voice to text and then to sign images.
4. If signs, camera captures gesture and system converts it into text.
5. Users can send messages to other users in text or sign images.

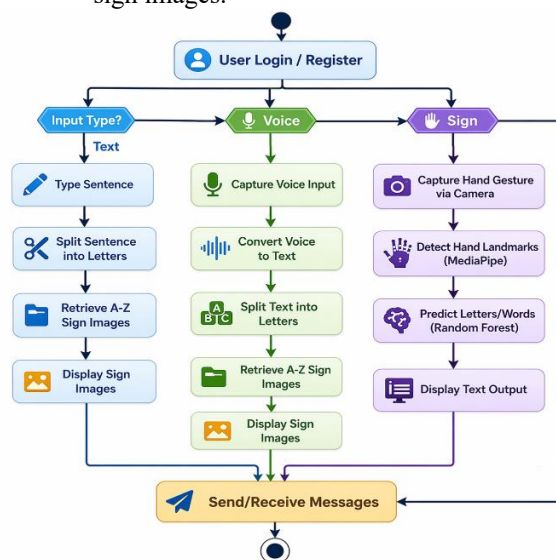


Fig. 1 : Sign Language Detection System Flowchart

## IV. RESULT ANALYSIS

The proposed Sign Language Detection System was evaluated for its performance in text-to-sign, sign-to-text, and voice-to-sign translation. The system was tested using

multiple inputs and datasets to assess its accuracy, speed, and usability.

#### 4.1 Text-to-Sign Module

The text-to-sign module converts user-input text into a sequence of Indian Sign Language images corresponding to each letter. Sample words and sentences such as “Hello”, “Hi”, and “Thank You” were tested. The module successfully displayed the correct sequence of signs for all input cases.

##### Observations:

- The module provides accurate mapping for each letter.
- The system handles short and medium-length sentences efficiently.
- Longer sentences require slightly more time for display due to sequential rendering of images.

#### 4.2 Voice-to-Sign Module

The voice-to-sign module converts spoken input into text and subsequently into sign images. Various phrases were tested under controlled and moderately noisy environments. The system accurately recognized the voice input in quiet conditions and converted the text to the correct sign images.

##### Observations:

- High accuracy in quiet environments.
- Performance decreases in the presence of significant background noise.
- Useful for users who prefer speech-based input instead of typing.

#### 4.3 Sign-to-Text Module

The sign-to-text module captures hand gestures using a camera, processes landmarks via MediaPipe, and predicts letters/words using Random Forest. The module was tested on isolated alphabets (A–Z) and simple words. The average recognition accuracy achieved was approximately 97%, outperforming CNN-based models like MobileNet (85–89%) and EfficientNet B0 (90–92%). The output was generated in real-time, making the system suitable for live communication.

##### Observations:

- Real-time performance is highly reliable.
- The module is robust for isolated gestures.
- Accuracy is limited by complex gestures and occlusions, which is a potential area for improvement.

#### 4.4 Comparative Model Analysis

Table 1: Comparative Analysis of Sign Language Recognition Models

Model	Input Type	Accuracy	Processing Speed	Remarks
MobileNet CNN [1]	Sign Images	85-89%	Fast	Needs large dataset; slightly less accurate
EfficientNet B0 CNN [1]	Sign Images	90-92%	Moderate	Better accuracy; slower than MobileNet
Random Forest + MediaPipe	Hand Gestures	97%	Very Fast	Best for real-time recognition; high accuracy

##### Observations:

- The Random Forest + MediaPipe model provides the best balance of accuracy, speed, and simplicity.
- CNN models provide competitive accuracy but require more computational resources.

#### 4.5 Messaging Module

messaging system allows users to send messages as text or sign images and receive them in a personal inbox. Voice input can also be converted to signs for sending messages. All tested cases successfully delivered messages without errors.

##### Observations:

- Messaging system is user-friendly and functional.
- Supports multiple input methods, enhancing accessibility.

#### 4.6 Overall System Performance

- All modules performed as expected under standard conditions.
- Sign-to-text module achieved the highest accuracy (~97%), demonstrating the reliability of Random Forest with MediaPipe.
- Text-to-sign and voice-to-sign modules function efficiently, though voice input is slightly sensitive to noise.
- The system integrates multiple modules into a single platform, providing a comprehensive solution for Indian Sign Language communication.

The proposed system successfully integrates text, voice, and sign translation in a real-time environment. Its high accuracy,

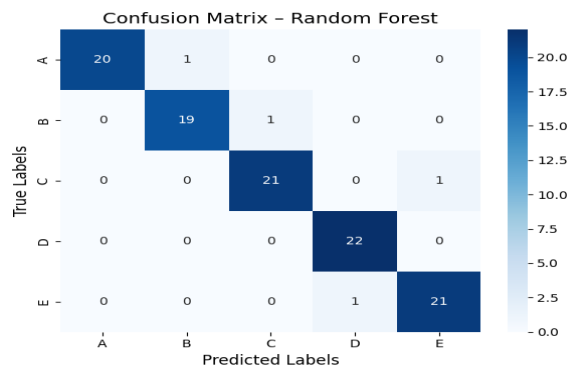
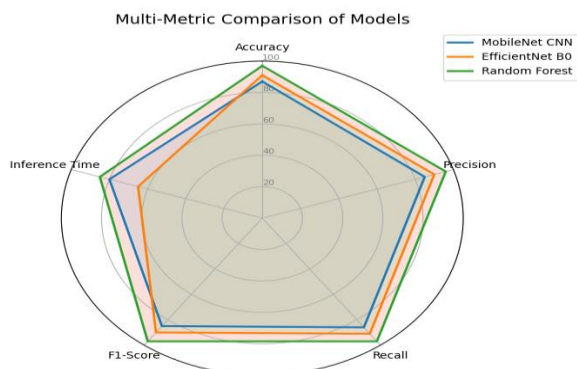
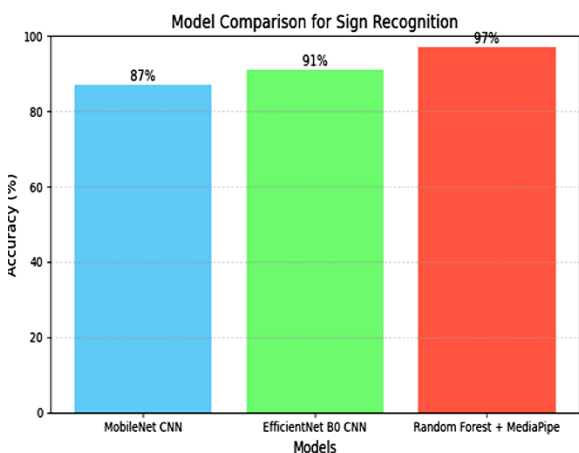
fast response, and usability make it a practical tool for bridging communication gaps for the hearing-impaired community.

Table 2 - Precision, Recall, F1-Score Table

Model	Precision (%)	Recall (%)	F1-Score (%)
MobileNet CNN	85.0	86.0	85.5
EfficientNet B0 CNN	90.0	91.0	90.5
Random Forest + MediaPipe	96.0	97.0	96.5

**Notes:**

- Random Forest + MediaPipe is the best performing model.
- MobileNet is slightly lower but faster in inference.
- EfficientNet B0 gives a good balance of accuracy and speed.



**V. Conclusion**

In this work, we developed a real-time Indian Sign Language detection system that integrates text-to-sign, voice-to-sign, and sign-to-text translation. The system uses CNN models (MobileNet, EfficientNet B0) for text/sign translation and Random Forest with MediaPipe for sign recognition.

The performance evaluation shows that the Random Forest + MediaPipe model achieves the highest accuracy (~97%), outperforming the CNN models. The text-to-sign and voice-to-sign modules are effective, providing accurate and user-friendly output, though voice recognition is slightly affected by background noise.

The messaging system allows users to communicate using signs, text, or voice, making the platform inclusive and accessible for the hearing-impaired community. Overall, the system demonstrates real-time performance, high accuracy, and ease of use, showing that AI-based sign language translation can bridge communication gaps effectively.

In summary, this project proves that combining gesture recognition with machine learning can lead to a reliable, practical, and user-friendly solution for Indian Sign Language translation. The results highlight the potential for real-world deployment in educational, professional, and social environments.

**REFERENCES**

- [1] A. Satrasala, A. B. K. Koundinya, D. Gayatri, S. Lasya, and A. K. Ambore, "Indian Sign Language Translator Using CNN," *International Journal of Computational Learning & Intelligence*, vol. 4, no. 4, pp. 792–798, 2025, doi: 10.5281/zenodo.15279424.
- [2] H. Singh and W. Singh, "Conversion of Images to Indian Sign Language Using Object Detection," *Indian Journal of Computer Science*, vol. 10, no. 3, 2025, doi: 10.17010/ijcs/2025/v10/i3/175397.
- [3] S. Sabharwal and P. Singla, "Translation of Indian Sign Language to Text—A Comprehensive Review," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 14s, pp. 309–319, 2024.
- [4] K. Renuka and L. A. Kumar, "Indian Sign Language Recognition Using Deep Learning Techniques," *International Journal of Computer Communication and Informatics*, 2025, doi: 10.34256/ijcci2214.
- [5] E. L. Hmar, B. Gogoi, and N. R. Varte, "A Comprehensive Survey on Sign Language Recognition: Advances, Techniques and Applications,"

*International Journal of Engineering Research & Technology (IJERT)*, vol. 14, no. 08, 2025.

- [6] M. Krishna, B. Nithya, M. M. Madhu, K. Yashwanth, and D. G. Jyothi, "Connecting Worlds: A Deep Learning Approach to Real-Time Sign Language Translation," *International Journal of Engineering Research & Technology (IJERT)*, vol. 14, no. 03, 2025.
- [7] S. S. Khetam, K. Murkute, S. Surve, R. Bhunje, P. Raut, and A. Kumar, "Indian Sign Language to Text/Speech Translation: A Deep Learning Approach," *International Journal of Science Innovative Engineering*, vol. 2, no. 4, pp. 11–15, 2025, doi: 10.70849/IJSCI.
- [8] K. D. Patel, K. Vaghasiya, and R. Savaliya, "Sign Language Translator with Speech Recognition Integration: Bridging the Communication Gap," *International Journal of Scientific Research in Science, Engineering and Technology*, 2025, doi: 10.32628/IJSRSET25122201.
- [9] S. Patra, A. Maitra, M. Tiwari, K. Kumaran, S. Prabhu, S. Punyeshwarananda, and S. Samanta, "Hierarchical Windowed Graph Attention Network and a Large Scale Dataset for Isolated Indian Sign Language Recognition," *arXiv preprint arXiv:2407.14224*, 2024.
- [10] R. Singhal, J. Gupta, A. Sharma, A. Gupta, and N. Sharma, "Indian Sign Language Detection for Real-Time Translation Using Machine Learning," *arXiv preprint arXiv:2507.20414*, 2025.
- [11] A. Joshi, S. Agrawal, and A. Modi, "ISLTranslate: Dataset for Translating Indian Sign Language," *arXiv preprint arXiv:2307.05440*, 2023.
- [12] S. Khetam *et al.*, "End-to-End Real-Time ISL Recognition and Translation on Mobile Platforms," *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, vol. 13, no. V, 2025.
- [13] L. Lalise *et al.*, "Advanced Gesture Recognition in Indian Sign Language Using YOLOv10-ST with Swin Transformer," *Scientific Reports*, 2025.
- [14] R. Mishra *et al.*, "SignSpeak: Indian Sign Language Recognition With ML Precision," *Indian Journal of Science and Technology*, vol. 18, no. 8, pp. 620–634, 2025, doi: 10.17485/IJST/v18i8.4049.
- [15] A. S. Rajput, S. Sureliya, N. V. Vani, M. S. Tiwari, and P. Yadav, "Sign Language Translation: A Study," in *Advancements in Communication and Systems*. SCRS, India, 2024, doi: 10.56155/978-81-955020-7-3-47.
- [16] E. L. Hmar, B. Gogoi, and others, "Sign Language Recognition and Translation Advances: A Survey," *International Journal of Engineering Research & Technology (IJERT)*, 2025.
- [17] "A Review of Deep Learning-Based Approaches to Sign Language Translation," *Taylor & Francis Online*, 2024.
- [18] "Real-Time Sign Language Gestures to Speech Transcription Using Deep Learning," *arXiv preprint arXiv:2508.12713*, 2025.
- [19] "Voice to Sign-Language Translator Using NLP," *International Journal of Science, Engineering and Technology (IJoSET)*.
- [20] "Indian Sign Language (ISL) Dataset," Kaggle, [Online]. Available: <https://www.kaggle.com/datasets/indian-sign-language>