International Journal on Science and Technology (IJSAT)

E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

SignCom AI: A Real-Time Communication System for Indian Sign Language Using AI and NLP Techniques

G.Jemilda[1], S.Valthi Siva Ram[2], A.Vinil Britus[3], M. Williams Pearlson Jebus[4], Mohammed Muhsin Iliyas[5]

[1]Faculty, [2,3,4,5]UG Scholar
Computer Science and Engineering,
[1,2,3,4,5]Jayaraj Annapackiam CSI College of Engineering, Nazareth, India
[1]gjemilda@gmail.com, [2]valathy2002@gmail.com, [3]vinilrich46@gmail.com,
[4]williamsjebus2018@gmail.com, [5]muhsiniliyas03@gmail.com

Abstract - Communication barriers between the deaf and hearing communities remain a significant challenge in today's society. Our project aims to bridge this gap by developing a real-time Sign Language Translator App capable of converting Indian Sign Language (ISL) into both text and speech. The application uses the device's camera to capture hand gestures and leverages machine learning and computer vision techniques to recognize and interpret signs accurately. It supports translation into multiple Indian regional languages, making it accessible to a broader audience. Additionally, the app features a Learning Mode to teach ISL, and a Text-to-Sign avatar that displays user-input text in animated sign language. This comprehensive approach empowers the deaf community by facilitating inclusive communication, promoting ISL learning, and fostering societal awareness.

Keywords: Indian Sign Language (ISL), Machine Learning, Computer Vision, Learning Mode, Text-to-Sign.

1. INTRODUCTION

In a world driven by communication, individuals with hearing and speech impairments often face significant challenges in expressing themselves and engaging with society. According to the World Health Organization (WHO), over 430 million people worldwide live with disabling hearing loss, and this number is expected to rise to 700 million by 2050. In India, it is estimated that over 63 million people suffer from significant hearing loss, making it one of the largest populations affected by hearing impairments globally. Sign language, particularly ISL serves as a crucial mode of communication for this community. However, the lack of ISL awareness among the general public creates a serious communication barrier, impacting education, employment, healthcare access, and social interaction for the deaf and hard-of-hearing population. This often results in marginalization and social isolation. With the advancement of technologies such as artificial intelligence, machine learning, and computer vision, there is a powerful opportunity to design inclusive tools that help bridge this communication gap. Our proposed solution is a mobile application that functions as a real-time Indian Sign Language Translator, converting ISL gestures into text and speech in various Indian regional languages. The app uses a smartphone's camera to detect and interpret signs through trained machine learning models, providing accurate and accessible translations. Beyond translation, it features a Learning Mode to help users understand and practice ISL interactively, and a Text-to-Sign mode powered by a virtual avatar to animate any typed text in ISL format. This comprehensive approach promotes ISL learning, increases accessibility, and facilitates inclusive communication between hearing and non-hearing individuals. Ultimately, our solution aims to break barriers, foster societal inclusion, and empower millions through innovative technology.

2. LITERATURE SURVEY

Priti Jadhav et al. [1] developed a real-time sign language detection system using Convolutional Neural Networks (CNN). Their approach captured hand gestures using a webcam and classified them into corresponding alphabets. The system was able to recognize American Sign Language (ASL) alphabets with an accuracy of over 95%. Their work demonstrates how CNNs can effectively process gesture images for real-time sign interpretation.

Adithya R. et al. [2] proposed a deep learning-based sign language translator using CNN and Long Short-Term Memory (LSTM) models. The CNN extracted spatial features from the hand gesture images while the LSTM handled temporal dynamics in continuous sign recognition. Their hybrid model achieved improved accuracy compared to standalone CNN, showcasing the potential of combining temporal and spatial learning for fluent sign language translation.

Chandrasekar A et al. [3] implemented an ISL recognition model using image processing and neural networks. The model was trained on a dataset of Indian hand signs and achieved significant recognition accuracy for basic alphabets. They also addressed issues like background noise and lighting, emphasizing the need for robust models in real-world scenarios.

Swapnil S. et al. [4] introduced a mobile application to translate static and dynamic gestures into text using a lightweight CNN model optimized for Android devices. Their work focused on making sign translation accessible on smartphones without heavy computational requirements. The app supported gesture-to-text conversion with over 90% accuracy for a predefined set of gestures.

Ravi Kumar et al. [5] developed a virtual avatar system that converts text into sign language animations. The goal was to help hearing individuals communicate with the deaf by simply typing text, which the avatar would translate into sign language gestures. This study contributes to improving ISL literacy and awareness among the hearing population while reducing the communication gap.

3.RESEARCH METHODOLOGY

This work uses a design-based research approach to develop an translation system with three main features: Sign to Text/Audio, Text to Sign, and Learn Mode. The system uses machine learning and computer vision to recognize ISL gestures through a camera and converts them into text and audio. For text-to-sign, it maps input text to sign language animations using avatars or video clips. The Learn Mode offers interactive lessons and quizzes to help users learn ISL. The backend is built with Python (Flask), and the frontend uses Flutter. Public datasets and user feedback guide the development and improvement of the system. Figure 1 represents the architecture diagram.

International Journal on Science and Technology (IJSAT)

E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

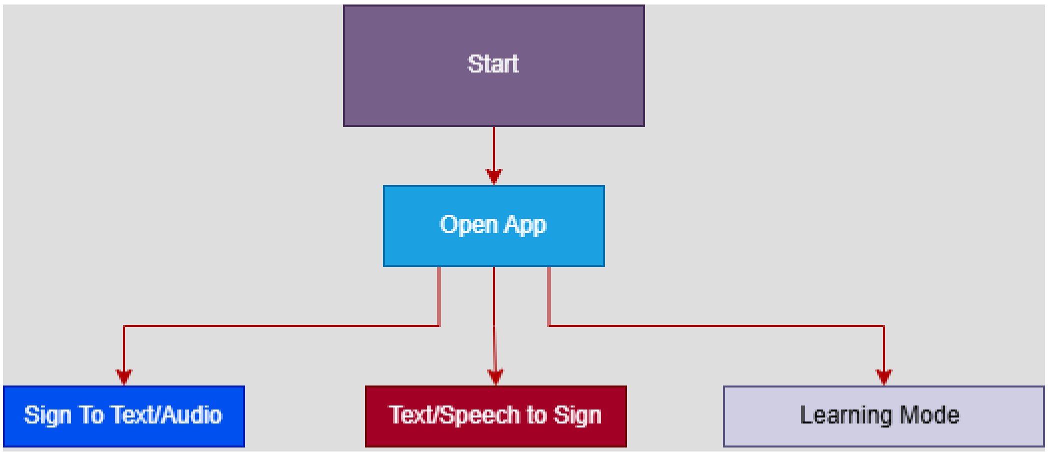

Figure.1 Architecture Diagram

3.1. Sign To Text/Audio

This feature captures hand gestures from the live video feed and converts them into text and speech outputs. The system processes each video frame using OpenCV, extracts gesture features with a lightweight CNN model, and classifies them in real-time. Recognized gestures are immediately translated into text, which is displayed on the screen. Simultaneously, the text is converted into audio output using a Text-to-Speech engine, ensuring smooth and accessible communication for users in multiple languages. Figure 2 represents the training process of the Sign to Text/Audio and Figure 3 represents the output of sign To Text/Audio.

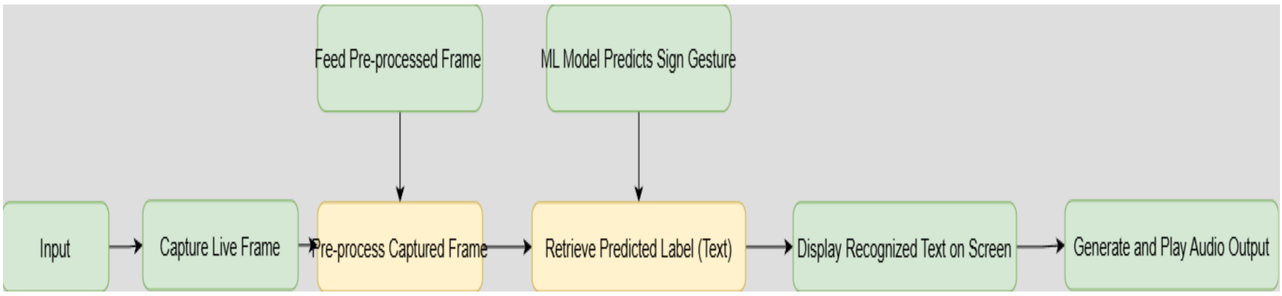

Figure. 2 Training

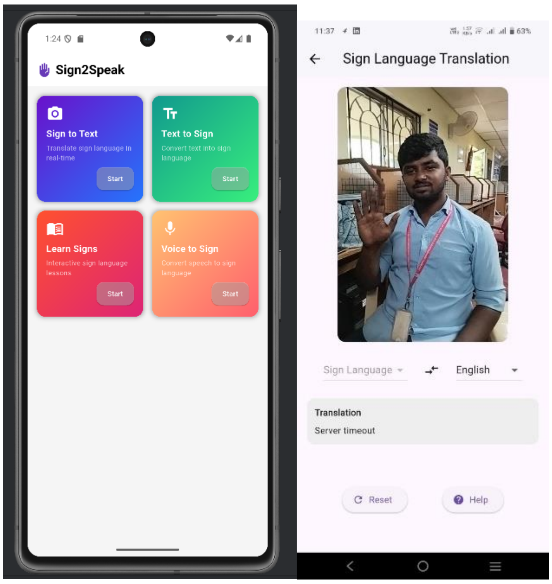

Figure.3 Output of sign To Text/Audio

3.1.1 Input Phase

The system begins with a user performing sign gestures in front of the device's camera. This live sign input is the foundation of the recognition pipeline. The application is designed to support both static gestures (like alphabets or numbers) and potentially dynamic gestures (like words or phrases) in future iterations.

3.1.2 Live Frame Capture

The application uses a built-in or external camera to continuously capture video frames. These real-time frames are the input to the vision processing pipeline. Efficient frame capture with minimal delay is essential for ensuring responsiveness, especially in interactive scenarios.

- Frame Rate: Typically set between 15-30 FPS for a balance between performance and computational load.
- Frame Resolution: Optimized to match the input dimension of the CNN (e.g., 64x64 or 128x128 pixels).

3.1.3 Frame Pre-processing

The captured frames undergo several pre-processing steps to enhance the input quality before being passed to the machine learning model:

- Cropping and Resizing: Ensures uniform image dimensions compatible with the CNN input.
- Grayscale or HSV Conversion: Reduces computational load and removes irrelevant color data.
- Noise Filtering: Uses Gaussian blur or median filtering to reduce background noise.

These steps improve feature extraction and classification accuracy.

3.1.4 Feeding the Processed Frame into the ML Model

After pre-processing, the image frame is normalized and passed into a pre-trained CNN model. The CNN performs feature extraction and classification by identifying spatial hierarchies in the input image.

- The CNN is trained on a custom or publicly available sign language dataset.
- It outputs the predicted label representing the gesture class.

3.1.5 Gesture Prediction by CNN

The trained CNN evaluates the processed frame and classifies it into one of the known sign language gestures. The classification result is returned as a numeric label or class ID, which maps to a specific alphabet, number, or word.The class with the highest confidence score is selected as the predicted gesture.

3.1.6 Label Retrieval and Mapping

The numerical output from the model is converted to a human-readable label. This label corresponds to the recognized sign. For example, an output of 5 might map to the label "E" in ISL or ASL.

3.1.7 Display of Recognized Text

The translated gesture is immediately displayed as text on the user interface. This real-time feedback allows the user and the listener to verify the translation.

- The text is shown in a clear, readable font on output box.
- Optional logging of recognized signs can be maintained for reference.

3.1.8 Audio Output Generation

To aid communication with non-signers, the system converts the recognized text into speech using a Text-to-Speech (TTS) engine. The spoken output helps in situations where textual display is not sufficient or accessible.

- Libraries like GTTS (Google Text-to-Speech) or pyttsx3 are commonly used.
- The audio output is played through speakers or headphones in near real-time.

3.2. Text/Speech to Sign

This feature converts user input (text or speech) into Indian Sign Language using a 3D animated avatar. The system checks if the full sentence exists in the database and plays the corresponding GL Transmission Format Binary file GLB animation. If not, it breaks the sentence into words and plays animations for each known word. If a word is missing, the system displays a message indicating that it's a planned feature, allowing future updates. This ensures accessibility and scalability for real-time sign communication**.** Figure 4 represents the training process of the Text/Speech to Sign and Figure 5 represents the output of Text/Speech to Sign.

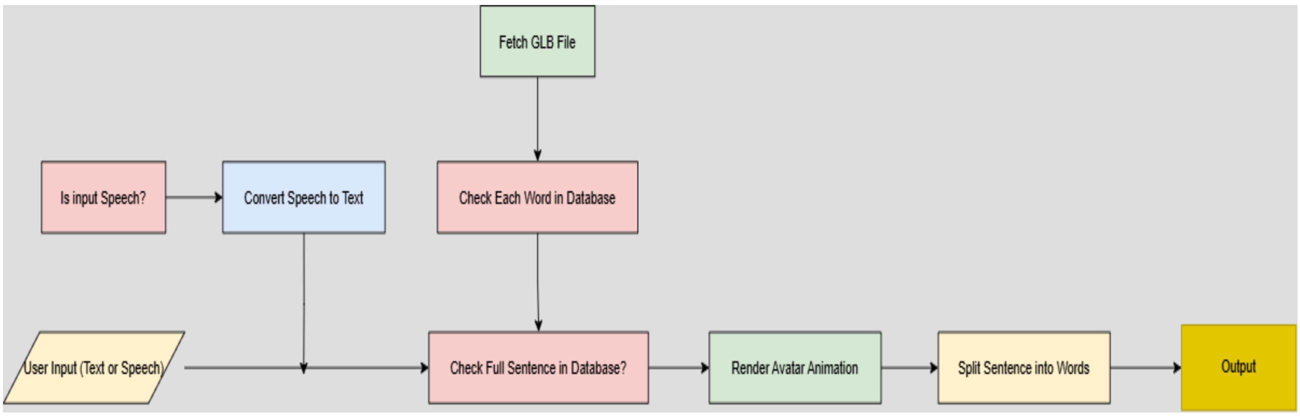

Figure.4 Training

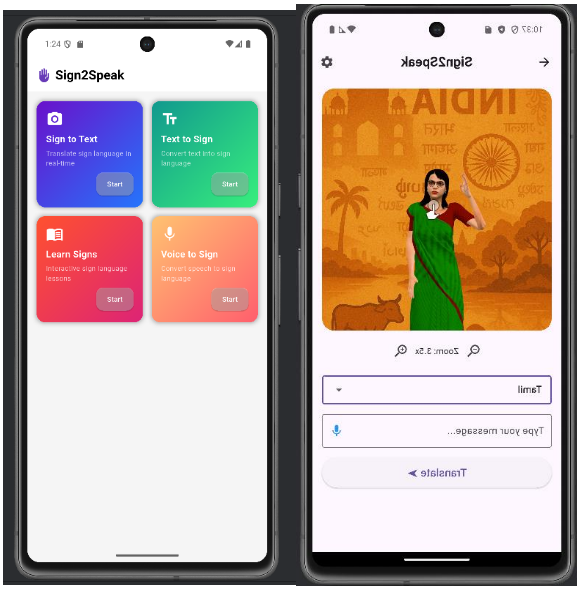

Figure.5 Output of Text/Speech to Sign

3.2.1 User Input (Text or Speech)

The user begins by entering a sentence either by:

- Typing the text directly, or
- Speaking into the system's microphone.

This dual input support ensures the application is accessible to both literate and non-literate users.

3.2.2 Speech Input Check

A conditional check is performed:

- If the input is speech, it proceeds to the Speech-to-Text (STT) module.
- If it's already in text form, it skips to the next step.

3.2.3 Convert Speech to Text

When speech is detected:

- A Speech Recognition engine (e.g., Google Speech API or Mozilla DeepSpeech) converts the spoken sentence into plain text.
- The output is a coherent sentence string that can be processed further.

3.2.4 Sentence/Word Matching in Database

Two levels of linguistic mapping occur:

Full Sentence Check:

- ○ The system first checks if the entire sentence exists in the ISL database (text-to-animation mapping).
- ○ If found, the corresponding animation is retrieved and played.

Word-by-Word Check:

- ○ If no full sentence match is found, the system splits the sentence into individual words.
- ○ Each word is then searched in the ISL word database.
- ○ If found, its corresponding avatar gesture is queued.

3.2.5 Fetch GLB File for Avatar

Once all relevant sign words or sentence animations are identified:

- The system loads corresponding GLB files (3D sign animations).
- These files contain rigged avatar animations for signs.

3.2.6 Render Avatar Animation

The selected GLB file(s) are rendered on-screen using a 3D animation engine such as Three.js or Babylon.js.

- The avatar performs sign gestures in proper sequence.
- Transition timing and visual clarity are managed for readability.

3.2.7 Output

The final output is:

- A 3D animated avatar performing Indian Sign Language signs for each word or sentence.
- This animation helps bridge communication for hearing-impaired users by converting spoken/text communication into visual ISL gestures.

3.3. Learning Mode

The figure 6 represents the training process of the Structure of Learnig mode and Figure 7 represents the output of Learning Mode.

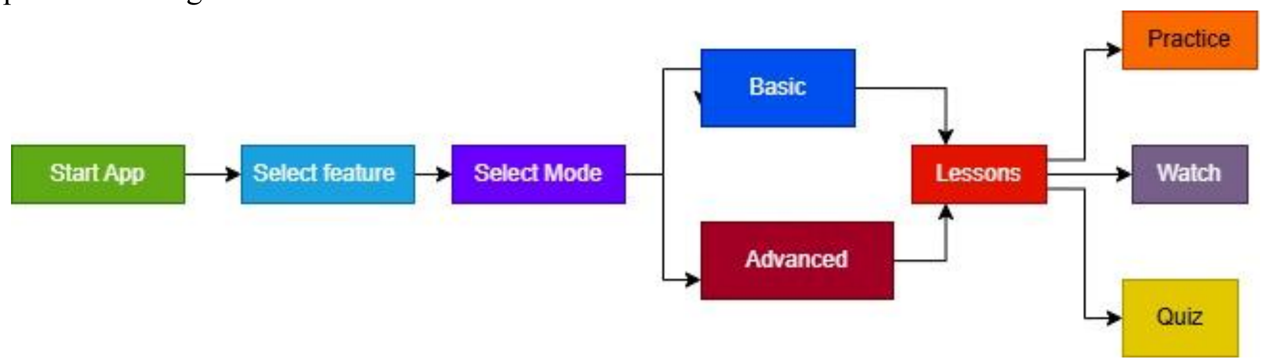

Figure.6 Structure of Learnig mode

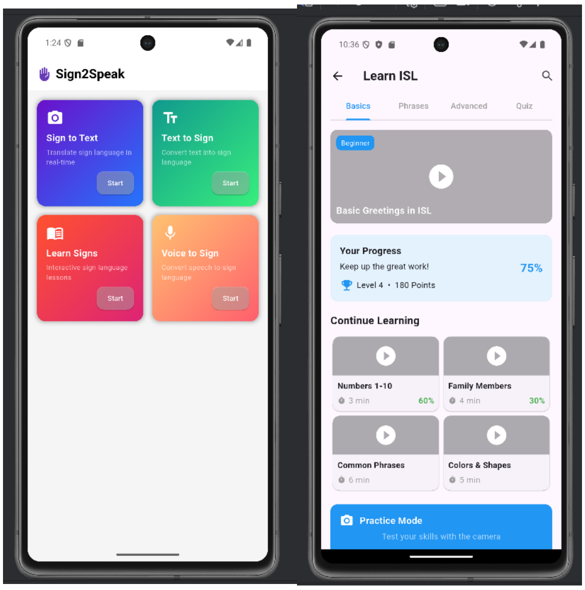

Figure.7 Output of Learning Mode

3.3.1 Content Structuring and Video Integration

The learning modules are categorized into levels:

- Basic Level: Introduction to simple signs, everyday phrases, and foundational vocabulary.
- Advanced Level: In-depth signs, complex phrases, and advanced conversational gestures. Each module is supported by:
- Sign Language Videos: Real human performers demonstrate the correct gestures for each sign, allowing learners to observe and replicate the gestures.

3.3.2 Learning Interface

Users can:

- Select a level or category (Basic/Advanced).
- Watch sign language videos embedded within the lesson, demonstrating each sign in real-time.
- Bookmark and replay content as needed for better retention and understanding.

3.3.3 Interactive Learning Tools

- Flashcards with videos and sign gestures for efficient review and practice.
- Practice Mode where users can mimic the signs demonstrated in videos (future upgrade: camera-based AI tracking to provide real-time feedback on user gestures).
- Quiz Mode to strengthen memory retention through:
  - Multiple-choice questions about signs and their meanings.
  - Video-based recognition challenges to identify the correct sign.
  - "Guess the sign" games to test and reinforce knowledge.

3.3.4 Progress Tracking and Motivation

- User progress is monitored across different learning modules.
- Visual dashboards display completion rates, streaks, and accuracy to track progress.

- Badges and gamification elements are incorporated to motivate continued learning and improvement

4. CONCLUSION

Our work focuses on developing an ISL translation system using machine learning techniques to bridge communication gaps between the deaf and hard-of-hearing community and the hearing world. The system uses a real-time camera-based recognition method to identify ISL gestures and translate them into text and speech in multiple Indian regional languages. We utilized a CNN for gesture recognition and feature extraction, while employing a natural language processing (NLP) model for generating accurate translations. The app also integrates a learning mode to help users learn ISL effectively and provides real-time feedback on gestures. The system's performance was evaluated using accuracy metrics, demonstrating the effectiveness of the approach. Overall, our project shows significant potential in enhancing communication for the deaf and hard-of-hearing individuals, while promoting the widespread adoption of ISL as a language. The application is designed to be accessible, user-friendly, and adaptable to various devices, making it a valuable tool for both communication and learning

5. FUTURE WORK

The project aims to expand its dataset by incorporating a larger and more diverse collection of signs, including regional variations from different parts of India, to enhance the accuracy of Sign to Text and Audio recognition. Additionally, the avatar will be further trained to recognize and express a broader range of words and sentences for effective Text to Sign Language conversion. Improvements will also be made to the avatar's facial expressions and gesture clarity to ensure more natural and understandable communication. A potential new feature includes the development of a YouTube to Sign Language translation tool, enabling the avatar to interpret video audio or subtitles into Sign Language in real time, significantly improving content accessibility for deaf users. Furthermore, the system will be enhanced with multilingual support by integrating more Indian regional languages such as Tamil, Telugu, Marathi, and others for both input and output, promoting wider inclusivity and usability**.**

REFERENCES

1. Priti Jadhav et al., "Real-Time Sign Language Detection Using CNN," International Journal of Computer Applications (IJCA), Vol. 184, No. 25, pp. 10–15, 2022.
2. Adithya R. et al., "Sign Language Translator Using CNN and LSTM," International Journal of Engineering and Technology (IJET), Vol. 11, No. 4, pp. 205–212, 2023.
3. Chandrasekar et al., "ISL Recognition Using Image Processing," International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), Vol. 9, Issue 12, pp. 45–50, 2022.
4. Swapnil S. et al., "Mobile App for Gesture-to-Text Translation," Journal of Mobile Computing and Application Development (JMCAD), Vol. 7, No. 1, pp. 33–40, 2023.
5. Ravi Kumar et al., "Text-to-Sign Virtual Avatar System," International Journal of Human-Computer Interaction (IJHCI), Vol. 12, No. 2, pp. 88–94, 2021.