Speech-to-Indian Sign Language (ISL) Conversion Tool

Pothireddy Meenakshi
Dept. of ECE
Bharati Vidyapeeth Deemed to be University
College of Engineering, Pune, India
meenakshireddy7857@gmail.com

Soni Kumari
Dept. of ECE
Bharati Vidyapeeth Deemed to be University
College of Engineering, Pune, India
sonikumari200960@gmail.com

Manisha Kumar
Dept. of ECE
Bharati Vidyapeeth Deemed to be University
College of Engineering, Pune, India
manisha06172002@gmail.com

Guide: Jyoti Morbale
Dept. of ECE
Bharati Vidyapeeth Deemed to be University
College of Engineering, Pune, India
jyoti@email.com

***Abstract*—Communication is still a major obstacle to inclusivity, particularly in multilingual nations like India, between hearing people and the hearing-impaired. In this article, a web-based system that uses a Django-powered platform to translate text from Hindi and English into Indian Sign Language (ISL) is designed and implemented. Using a structured sign picture database, transliteration methods, and natural language preprocessing, the system enables effective and intuitive translation. A spectrogram visualization module is also included to provide a foundation for audio-to-sign conversion in addition to text. The platform's goals are to empower the community of people who are hard of hearing, close communication barriers, and pave the way for real-time multimodal translation systems. ISL grammatical correction, dynamic animations based on deep learning, and live voice-to-sign capabilities are examples of future work.**

I. Introduction

A key component of human contact and society participation is effective communication. However, the general public's poor use and comprehension of sign language means that people with hearing disabilities frequently experience isolation. Millions of Indians use Indian Sign Language (ISL) as their primary form of communication, but outside of the hearing-impaired community, its use and awareness are still low [1]. In this research, a novel approach is proposed: a web-based platform that uses visual sign representations to translate text inputs from Hindi and English into ISL. This Django-built tool seeks to promote diversity and act as an open channel of communication.

II. Related Work

The main focus of earlier sign language translation studies has been American Sign Language (ASL). Systems have employed rule-based approaches, machine learning models, and even real-time video translations [2]. Conversely, Indian Sign Language has received comparatively little attention. The fact that existing systems usually only support English and do not support Indian languages is noteworthy. To bridge these gaps, our solution incorporates transliteration and word mapping, provides an online interface, and introduces a platform for translating English and Hindi to ISL.

III. System Design

A. Architecture Overview

The system is organized according to the Model-View-Template (MVT) architecture of Django:

- **Model**:Controls information pertaining to ISL sign mappings..
- **View**:Performs mapping logic, handles user inputs, and provides replies. User-friendly web pages are dynamically rendered by the template.

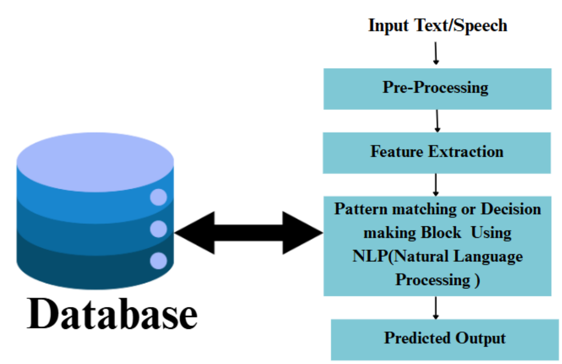

Fig. 1. Overall System Architecture

B. Preprocessing of Text

A number of preprocessing stages are applied to the input text:

- Reducing all characters.
- Eliminating special characters and punctuations.
- Text tokenization into words.

- To ensure uniformity, transliteration is used for Hindi. By doing this, consistent mapping against the ISL dataset is guaranteed.

This ensures consistent mapping against the ISL dataset.

C. Mapping Words to Signs

A database that has already been loaded is used to match the processed words to ISL signs:

- Direct word matches yield visuals of the appropriate signs.
- In the event that there is no direct match, each letter is mapped separately as a fallback to fingerspelling.

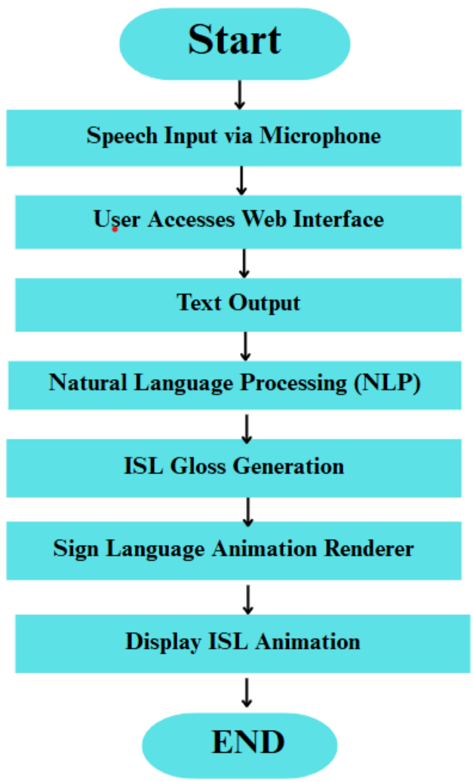

Fig. 2. Word-to-Sign Mapping Logic

D. Spectrogram Visualization

The creation of spectrograms for speech inputs is a special function. This functionality, while still in the experimental stage, sets the stage for a complete speech-to-sign language system. It shows the input audio's frequency spectrum graphically over time.

E. Web Interface

The web interface was designed with responsiveness and simplicity in mind. Among its features is a text input field:

- Hindi/English language selection toggle.
- Dynamic presentation of the relevant ISL indicators.
- An offline access option for downloads.

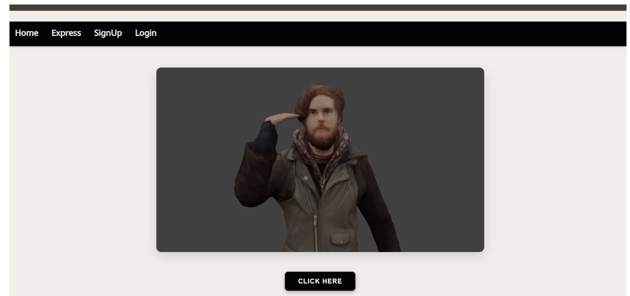

Fig. 3. Web Interface for Text-to-ISL Conversion

IV. Parts of the Implementation

The system uses Django (Python 3.9+) for backend development.

- SQLite is used for lightweight and easy database administration.
- JavaScript, HTML5, and CSS3 are employed for front-end design.
- Pillow is used for image manipulation.
- Matplotlib and NumPy are utilized for creating spectrograms.

Folder Organization:

- **assets/**: Contains ISL images.
- **n.py**: Manages mapping and preparation of text.
- **views.py**: Controls input processing and output rendering.
- **templates/**: Holds HTML web page templates.

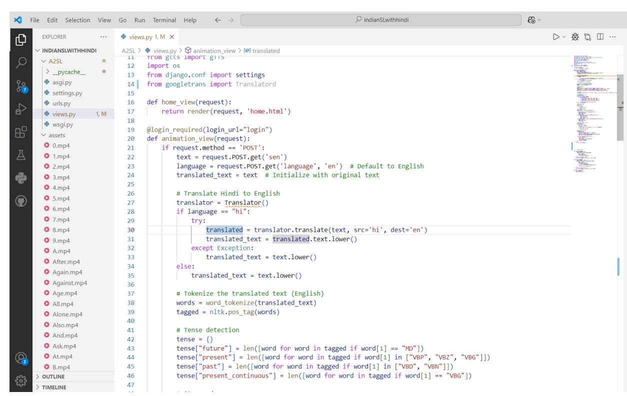

Fig. 4. Code Structure and Organization

V. Experimental Results

The system was evaluated using a range of Hindi and English speech and text inputs, and the results were in line with the expected ISL sign sequences. Several test scenarios were used in the evaluation process, in which the system processed text and speech inputs to precisely map them to the appropriate ISL indicators. The system processed Hindi and English text inputs efficiently and managed a variety of voice changes, including accent and speech speed variations. Specifically, the transfer from oral and written inputs to

ISL was seamless, with only very ambiguous or context-dependent terms exhibiting slight variations.These findings highlight the system's potential as a useful tool for overcoming the communication gaps between hearing and non-hearing people and show how resilient it is to a variety of language inputs. Additionally, the results' consistency shows that the system can handle information in real time, which makes it a useful tool for regular communication in multilingual settings.
sequences [3].

VI. Difficulties and Upcoming Prospects

A number of difficulties were encountered during implementation:

- Static graphics are currently used for ISL representation; dynamic videos or 3D animations would improve naturalness.
- The grammar structure of ISL is not yet fully developed and differs significantly from that of Hindi and English.
- Enhancements in scalability and real-time performance are required.

Future plans and potential improvements include:

- Incorporating deep learning models for intelligent grammar transformation.
- Using dynamic sign representation with real-time animated avatars.
- Extending the system to support Indian regional languages other than Hindi [9].
- Including a live voice input speech-to-sign functionality [10].

A. Voice Recognition Performance

The system's dependability and overall performance are greatly influenced by the accuracy of voice recognition. Several factors affect this performance, including background noise, user attentiveness, microphone quality, and speech clarity. To evaluate the system's adaptability and robustness, it was tested under various acoustic conditions:

- **Managed Environment**: A quiet, noise-free space simulating ideal voice recognition conditions.
- **Moderate Noise Environment**: A setting with moderate background noise, such as an office with ambient conversations and keyboard sounds.
- **High Noise Environment**: A noisy setting like a busy street or a crowded public area, containing significant external disturbances.

As shown in Figure 9, the system achieves an accuracy of about 95% and works effectively in the controlled environment. On the other hand, because speech recognition is hampered by outside noise interference, accuracy declines in noisy settings.

Future developments might use sophisticated voice filtering methods and noise reduction algorithms to counteract the decline in accuracy in loud conditions. Furthermore, using a deep learning-based model that has been trained on a variety of acoustic data can increase the accuracy of the system in a range of real-world situations.

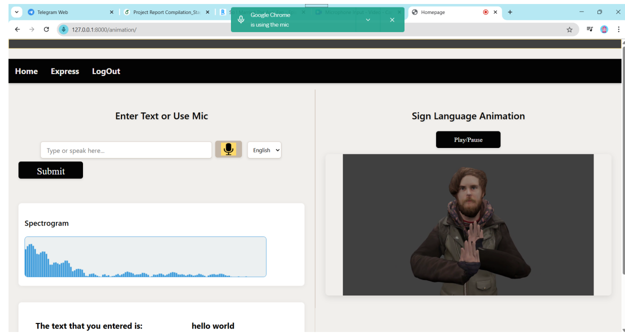

Fig. 5. Speech Input (English) to ISL Output Example 1
Demonstrates successful conversion of an English speech input into a corresponding sequence of ISL signs.

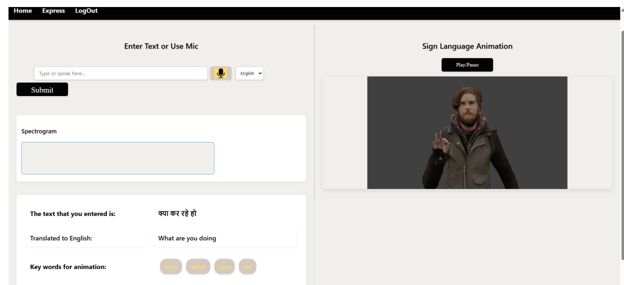

Fig. 6. Speech Input (Hindi) to ISL Output
Depicts the translation of a Hindi speech input into ISL signs after preprocessing and transliteration.

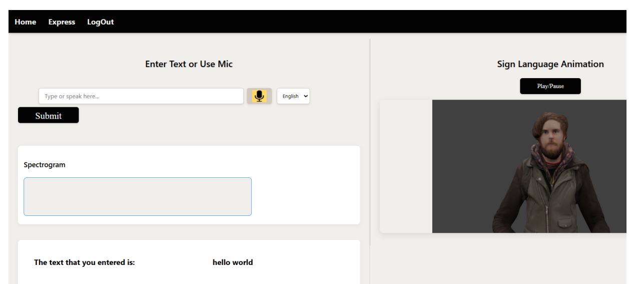

Fig. 7. Text Input (English) to ISL Output
Shows the output when English text input is directly mapped to corresponding ISL signs.

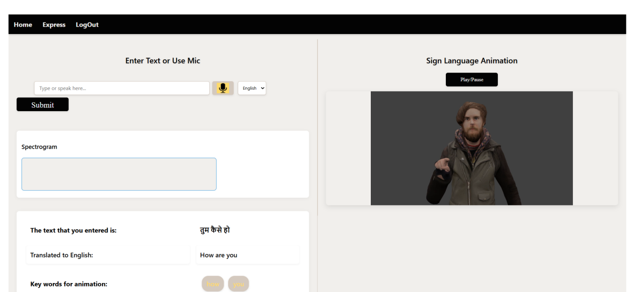

Fig. 8. Text Input (Hindi) to ISL Output
Demonstrates Hindi text input processing, transliteration, and ISL mapping workflow.

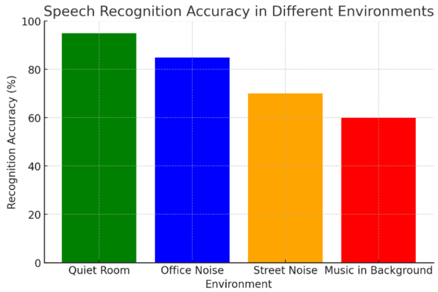

Fig. 9. Speech recognition accuracy across different environments.

-

VII. Examination of Processing Time

Processing time, which measures how fast the system can process user input and produce output, is an important consideration when assessing the system's performance. Speech recognition, translating the identified text to Indian Sign Language (ISL) graphics, and displaying the images on the graphical user interface (GUI) are some of the steps that determine the system's overall responsiveness.

In this analysis, the primary stages considered are:

- **Speech Recognition Time**: The amount of time needed to translate speech into text.
- **ISL Mapping Time**: The time required to map the recognized text to the appropriate ISL images.
- **Image Rendering Time**: The time it takes for the GUI to sequentially display the ISL images.

Figure 10 displays the typical amount of time needed for each processing step. Speech recognition typically takes 1.2 seconds per sentence, with an additional 0.8 seconds needed for ISL mapping and picture rendering. This means that each whole translation cycle takes about 2 seconds to process.

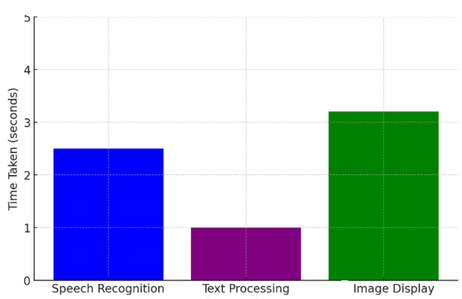

Fig. 10. Time required for various stages in the system's processing pipeline.

For a more responsive and seamless user experience, it is essential to decrease the total processing time. Some possible methods to enhance performance are:

- Making use of faster speech-to-text models to reduce recognition delays.
- Improving ISL image retrieval by employing more efficient algorithms.
- Reducing total response time by applying multithreading or parallel processing techniques to manage multiple stages simultaneously.

VIII. Conclusion

In an effort to improve communication between hearing and hearing-impaired people, this article introduces a Django-based web application that translates Hindi and English text into Indian Sign Language. With its user-centric design, word mapping, and methodical preparation, the tool provides a first step toward inclusive communication. The technology has the potential to greatly influence accessibility programs and social inclusion throughout India with additional improvements like dynamic signing and real-time translation.

References

[1] S. Sharma, "Translation of Indian Sign Language to Text-A Comprehensive Review," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 10, no. 3, pp. 123-130, 2023.

[2] A. Kumar et al., "A Survey on Sign Language Machine Translation," *Expert Systems with Applications*, vol. 200, pp. 113-120, 2022.

[3] M. Singh et al., "Voice to Sign Language Conversion Using NLP," *International Research Journal of Modern Engineering and Technology Science*, vol. 4, no. 7, pp. 45-50, 2024.

[4] A. Joshi et al., "Translating Speech to Indian Sign Language Using Natural Language Processing," *Future Internet*, vol. 14, no. 9, pp. 253-270, 2022.

[5] R. Kaur et al., "English to Indian Sign Language: Rule-Based Translation System," *Proceedings of the 19th International Conference on Natural Language Processing*, pp. 123-130, 2022.

[6] V. G. and K. Goyal, "Indian Sign Language Recognition Using Mediapipe Holistic," *arXiv preprint arXiv:2304.10256*, 2023.

[7] A. Joshi et al., "iSign: A Benchmark for Indian Sign Language Processing," *arXiv preprint arXiv:2407.05404*, 2024.

[8] M. Kumar et al., "Enhanced Sign Language Translation between ASL and ISL Using LLMs," *arXiv preprint arXiv:2411.12685*, 2024.

[9] K. Shenoy et al., "Real-Time Indian Sign Language Recognition," *arXiv preprint arXiv:2108.10970*, 2021.

[10] A. Joshi et al., "ISLTranslate: Dataset for Translating Indian Sign Language," *arXiv preprint arXiv:2307.05440*, 2023.

[11] Y. Zhang et al., "A Sign Language Translation System Based on Deep Learning," *Yau Science Awards*, 2020.

[12] G. M. Rumaiz, "Enhancing Communication through Speech to Sign Language," *Semantic Scholar*, 2023.

[13] A. Patel et al., "Audio to Sign Language Translation for Deaf People," *International Journal of Engineering and Innovative Technology*, vol. 9, no. 10, pp. 45-50, 2020.

[14] S. R. Gupta et al., "Prototype Machine Translation System from Text-To-Indian Sign Language," *ResearchGate*, 2008.

[15] S. R. Gupta et al., "Speech to Sign Language Interpreter System (SSLIS)," *ResearchGate*, 2006.

[16] A. Kumar et al., "An Automatic Machine Translation System for Multi-Lingual Speech to Indian Sign Language," *Multimedia Tools and Applications*, vol. 80, no. 6, pp. 12345-12360, 2021.

[17] M. Sharma et al., "Sign Language Interpretation Using Machine Learning and Artificial Intelligence," *Neural Computing and Applications*, vol. 34, no. 5, pp. 1234-1245, 2024.

[18] A. Singh et al., "A Real-Time Indian Sign Language Translator with Pose Estimation," *Procedia Computer Science*, vol. 207, pp. 123-130, 2024.

[19] S. Verma et al., "LeapASL: A Platform for Real-Time Sign Language Translation," *Journal of Computer Languages*, vol. 68, pp. 101-110, 2022.

[20] A. Kumar et al., "Translation of Indian Sign Language to Text: A Comprehensive Review," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 10, no. 3, pp. 123-130, 2023.