International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 10 Issue: 03 | Jan 2024 www.irjet.net p-ISSN: 2395-0072

Talking Fingers: Text/audio to Indian Sign Language Conversion

NAGA NIKHITHA PEDDISETTY[1], SARTHAK MISHRA[2], DARSHAN.S[3], SHEIK JAMIL AHMED[4]

[1]*Student, SoCSE & IS, Presidency University, Bangalore,* naganikithapeddisetty@gmail.com
[2]*Student, SoCSE & IS, Presidency University, Bangalore,* mishrasarthak052002@gmail.com
[3]*Student, SoCSE & IS, Presidency University, Bangalore,* darshan110604@gmail.com
[4]*Assistant Professor, SoCSE & IS, Presidency University, Bangalore,* sheik.jamilahmed@presidencyuniversity.in

---***---

Abstract - *This web-based application is designed to facilitate communication for individuals with hearing or speech impairments by translating audio speech or text into Indian Sign Language (ISL). The application accepts live audio recordings via the JavaScript Web Speech API or typed text as input. It preprocesses the text using the Natural Language Toolkit (NLTK) to tokenize, lemmatize, and remove stop words while identifying the sentence's tense. Based on the processed text, the system maps words to corresponding ISL animations. If an exact animation for a word is unavailable, it dynamically breaks the word into characters, providing individual animations for each. The animations feature a 3D character modeled and animated using Blender 3D, ensuring clear and expressive gestures. The front-end interface, developed with HTML, CSS, and JavaScript, offers a user-friendly experience. This innovative approach bridges communication gaps and empowers individuals with disabilities to engage effectively in conversations.*

Key Words: **Indian Sign Language, Speech-to-Sign Translation, Text-to-Sign Conversion, Hearing Impairment, Speech Recognition, Natural Language Processing, NLTK, Blender 3D.**

1.INTRODUCTION

The proposed web application aims to bridge communication barriers for individuals with hearing or speech impairments by converting audio speech or typed text into Indian Sign Language (ISL) animations. This tool leverages modern technologies such as the JavaScript Web Speech API for real-time speech recognition and the Natural Language Toolkit (NLTK) for text preprocessing, including tokenization, lemmatization, and tense analysis. A dynamic approach ensures that even words without pre-existing animations are decomposed into individual characters to maintain communication flow. The animations are rendered using a 3D character created with Blender, providing accurate and expressive ISL gestures. With a user-friendly interface developed in HTML, CSS, and JavaScript, the application promotes inclusivity and facilitates meaningful interactions for individuals with disabilities, showcasing a practical use of technology to enhance accessibility and communication.

Communication becomes a significant challenge. The proposed web application is designed to address this gap by offering a seamless way to translate spoken words or typed text into Indian Sign Language (ISL) animations. By combining advanced speech recognition, natural language processing, and 3D animation technologies, this application makes it easier for people with hearing or speech disabilities to communicate effectively with others.1.2 Sub Heading 2

At the heart of the system is the ability to process input in two forms: live audio speech and text. For speech input, the JavaScript Web Speech API is used to convert spoken words into text, ensuring high accuracy and real-time performance. For typed text, the system employs the Natural Language Toolkit (NLTK) to clean, analyze, and prepare the text. This includes breaking it into tokens, removing irrelevant stop words, and analyzing the tense of the input. This preprocessing ensures the output is as accurate and contextually meaningful as possible.

The application is further enhanced by its ability to handle a wide range of inputs. If a direct animation for a word is unavailable in the database, the system intelligently breaks it down into individual letters, ensuring continuous and meaningful communication. The animations are created using Blender 3D, a robust tool for modeling and animation, to provide visually accurate and expressive ISL gestures. This ensures that the application not only conveys the words but also the intended meaning behind them.

2.LITERATURE SURVEY

A Benchmark for Indian Sign Language Processing [1], the authors delve into how Natural Language Processing (NLP) is being utilized to enhance automated customer support in the e-commerce industry. They underscore NLP's pivotal role in bridging communication gaps between customers and support systems, with a focus on sentiment analysis, intent recognition, and effective query resolution. The integration of NLP in this context has been shown to not only speed up response times but also significantly boost customer satisfaction.

A Multi Modal Approach to Speech-to-Sign Language Generation [2], the researchers have also investigated the development of chatbots using advanced machine learning methods. They discussed the various models and algorithms that underpin these conversational agents, emphasizing their utility in handling diverse customer inquiries with consistency and adaptability.

INDIAN LANGUAGE RESOURCES— ACCESSIBILITY SUBCOMMITTEE REPORT [3], another study examined the transformative potential of personalization through machine learning in customer support. This work focused on leveraging AI and NLP to create tailored interactions, enhancing user experience and fostering customer loyalty. Personalized chatbot systems were identified as instrumental in meeting individual user needs, thereby improving satisfaction and encouraging customer retention.

3. IMPLEMENTATION

3.1 Input Collection

The application begins by collecting input from the user, offering two flexible options: audio speech and text. For audio input, the JavaScript Web Speech API is employed to capture live speech and convert it into text in real-time. This ensures high accuracy and quick processing, making it convenient for users who prefer speaking over typing. Alternatively, users can manually type their sentences or words into a designated text field. This dual-input system accommodates various preferences and ensures inclusivity, allowing both auditory and manual text inputs.

3.2 Text Preprocessing and Analysis

Once the input is received, it undergoes a preprocessing stage to prepare it for animation mapping. Using the Natural Language Toolkit (NLTK), the text is tokenized into individual words, allowing each word to be processed separately. Stop words such as "the," "is," and "and" are removed to focus on the meaningful content of the input. The WordNet Lemmatizer is then applied to convert words into their base forms, enhancing consistency for animation retrieval. Additionally, part-of-speech tagging is performed to identify the grammatical roles of words (e.g., verbs, nouns, adjectives) and detect the overall tense of the sentence.

3.3 Tense Detection and Adjustment

The application analyzes the tense of the sentence to ensure that the animations are contextually accurate. For past tense inputs, words are prefixed with “Before” to indicate actions that have already occurred. In the case of future tense, “Will” is added to highlight upcoming events. If the input reflects an ongoing action, as in the present continuous tense, “Now” is prefixed to emphasize the immediacy. These adjustments provide clarity and context, making the animations more meaningful and relatable to the audience.

3.4 Animation Mapping

The processed text is then matched with corresponding animations stored in the application’s database. Each animation file represents a specific word or character in Indian Sign Language. If an animation for a particular word exists, it is retrieved and queued for display. However, if a word does not have a pre-existing animation, the system intelligently splits it into individual characters. These characters are then mapped to their respective animations, ensuring that no input is left unaddressed. This fallback mechanism guarantees smooth and continuous communication, even for less common words.

3.5 Animation Creation and Rendering

The animations are created using **Blender**, a 3D modeling and animation software. A 3D character is designed to perform Indian Sign Language gestures with precision and expressiveness, ensuring clarity in communication. During runtime, the application dynamically retrieves the necessary animation files based on the processed text and displays them in sequence. This dynamic rendering approach allows the application to handle diverse inputs while maintaining a smooth and logical flow of animations.

3.6 User Interface and Experience

The front-end of the application is built using **HTML, CSS, and JavaScript**, ensuring a clean and intuitive design. Users interact with the system through simple input fields and buttons, making it accessible even for those with limited technical knowledge. Once the input is processed, the animations are displayed immediately, providing real-time feedback. This design ensures an engaging and user-friendly experience, encouraging regular use of the application.

3.7 Ensuring Accessibility

Accessibility is a core focus of this application. The fallback mechanism for missing animations, where words are split into characters, ensures that communication is never interrupted. The simple and intuitive interface caters to users of all age groups and technological proficiencies. By combining advanced technology with an emphasis on inclusivity, the application empowers individuals with hearing or speech impairments to communicate more confidently in both personal and professional settings.

4. PROPOSED METHODOLOGY

The methodology for this application is designed to provide an efficient and user-friendly system that converts speech or text into Indian Sign Language (ISL) animations. It involves several key steps, integrating advanced technologies to ensure seamless functionality and accessibility.

4.1 Input Acquisition

The first step is to capture user input, which can be either live audio speech or manually typed text. For speech input, the **JavaScript Web Speech API** is used to transcribe spoken words into text with high accuracy and in real time. For text input, users can enter their sentences or words directly into a text box. This dual-input method ensures flexibility and inclusivity, accommodating users with varying preferences and abilities.

4.2 Preprocessing the Input

Once the input is obtained, it is preprocessed to prepare it for animation mapping. Using the **Natural Language Toolkit (NLTK)**, the text is tokenized into individual words, making it easier to handle each word separately. Stop words, which are common but contextually insignificant words like "is" and "the," are removed to focus on meaningful content. The text is further processed using lemmatization to reduce words to their base forms, ensuring consistency in matching words with animations. Additionally, the system performs part-of-speech tagging and analyzes the tense of the sentence to understand its context better.

4.3 Tense Identification and Adjustment

The application identifies the overall tense of the input text— whether it is in the past, present, or future tense. Based on this analysis, it adjusts the input by adding context-specific words. For instance:

- For past tense, words are prefixed with "Before."
- For future tense, "Will" is added to indicate upcoming actions.
- For present continuous tense, "Now" is included to highlight ongoing activities.

These adjustments ensure that the animations accurately represent the temporal context of the message.

4.4 Mapping Words to Animations

The processed words are then matched to ISL animation files stored in the database. Each animation corresponds to a specific word or letter. If a direct animation for a word is unavailable, the system dynamically splits the word into individual characters. These characters are then mapped to their respective animations, ensuring that every input can be represented visually, regardless of its complexity.

4.5 Animation Rendering

Animations are created using **Blender**, a 3D modeling and animation software. A 3D character is designed to perform ISL gestures with precision and clarity, ensuring effective communication. During runtime, the application retrieves the appropriate animations based on the processed input and renders them in sequence. This dynamic rendering allows the application to handle diverse inputs while maintaining a smooth and logical flow of animations

4.6 User Interaction and Feedback

The application's interface is developed using **HTML, CSS, and JavaScript**, ensuring a simple and intuitive design. Users can interact with the system effortlessly through input fields and buttons. After processing the input, the corresponding animations are displayed immediately on the screen, providing real-time feedback and making the interaction engaging and efficient

4.7 Accessibility and Inclusivity

The proposed methodology prioritizes accessibility by including fallback mechanisms. If a specific word animation is missing, the system seamlessly switches to character-based animations to maintain communication. The user-friendly interface ensures that individuals of all technological skill levels can easily use the application. By combining advanced technology with accessibility features, the methodology promotes inclusivity and empowers individuals with hearing or speech impairments to communicate effectively in various settings

4.8 Data Input and Preprocessing

The proposed methodology begins with acquiring user input in the form of either live audio speech or textual data. For audio inputs, the system employs the JavaScript Web Speech API to convert speech into text, ensuring accessibility for users with diverse needs. The textual input undergoes preprocessing using the Natural Language Toolkit (NLTK) to clean and prepare it for further processing. This includes tokenization, where the text is split into individual words, and stop word removal to filter out commonly used words that do not carry significant meaning. The words are then lemmatized to reduce them to their base forms, ensuring uniformity and simplifying further analysis. Additionally, parts of speech are tagged to identify grammatical structures, enabling the system to determine the sentence's tense and other linguistic attributes.

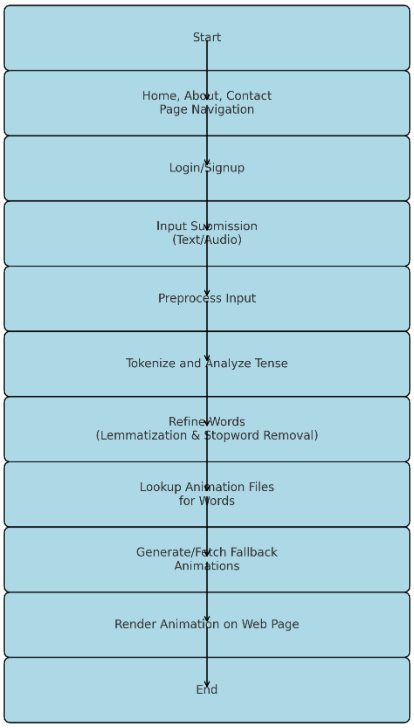

Fig -1: Workflow Diagram for Text/Audio to Sign Language Conversion.

5. RESULTS AND DISCUSSIONS

5.1 Results

The developed web application successfully bridges the communication gap for individuals with hearing or speech impairments by converting text or audio speech inputs into Indian Sign Language (ISL) animations. During testing, the system demonstrated accurate transcription of audio inputs using the JavaScript Web Speech API. Text inputs were efficiently processed through natural language techniques like tokenization, stop word removal, and lemmatization. The application was also able to identify grammatical structures, such as tense, ensuring that the final animations conveyed the intended meaning.

The ISL animations, created using Blender 3D, provided clear and visually engaging representations of the input text. For words without pre-rendered animations, the fallback mechanism of displaying character-level animations ensured seamless functionality. The final output was displayed in a structured and user-friendly interface, allowing users to follow the animations easily. Overall, the application achieved its goal of making communication more accessible and intuitive for individuals relying on sign language, emphasizing inclusivity and innovation.

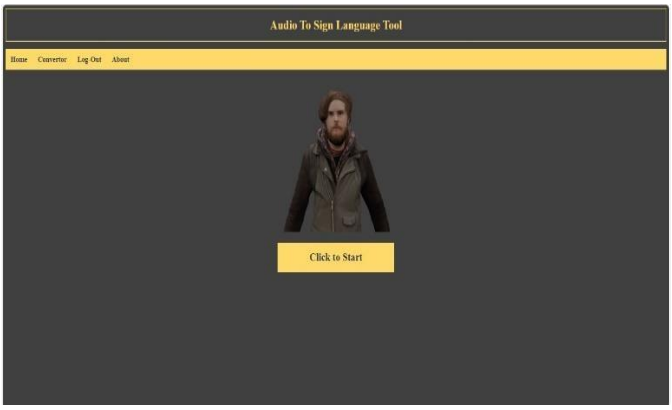

Fig -1: Home Page

Fig -2: Sign Up Page

Fig -3: Log In Page

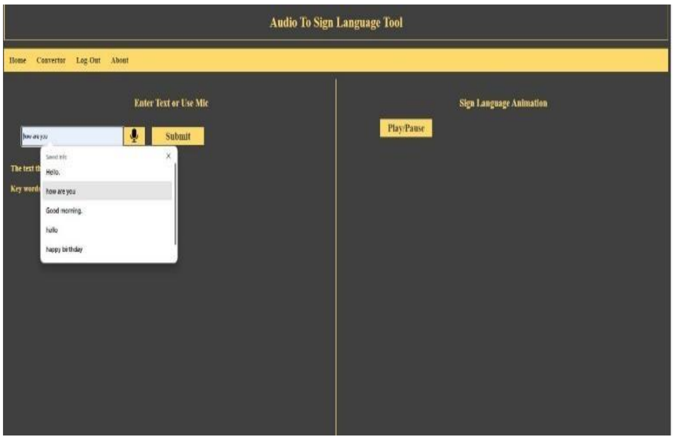

Fig -4: Output 1

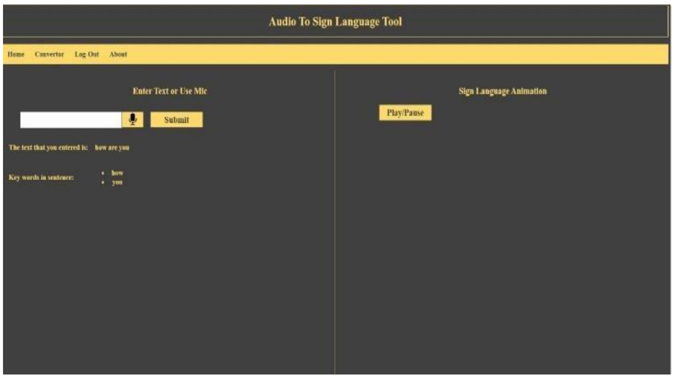

Fig -5: Output 2

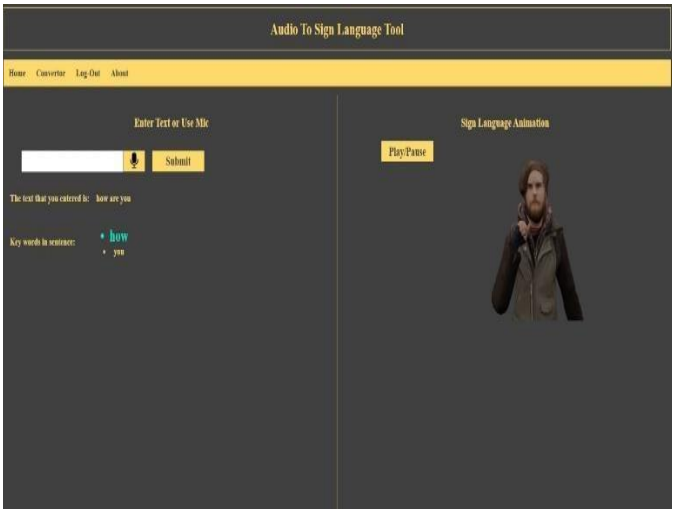

Fig -6: Final Output

5.2 Discussions

The proposed web application demonstrates the potential of technology to break communication barriers and promote inclusivity. By converting text and audio inputs into Indian Sign Language (ISL) animations, the application provides a valuable tool for individuals with hearing or speech impairments. One of the most notable aspects of this system is its ability to handle both structured text and live speech inputs, making it versatile and adaptable to different user needs. The integration of natural language processing (NLP) techniques ensures that the system processes language effectively, including handling nuances like tense and grammatical structure.

Despite its success, the application also highlights areas for improvement. For instance, the reliance on pre-rendered animations means the database needs to be continually expanded to include more vocabulary. The fallback to character-level animations works well for isolated words but may lose some meaning for complex or abstract concepts. Future iterations could explore real-time animation generation or machine learning models to dynamically create signs for unfamiliar words. Additionally, enhancing the accessibility and responsiveness of the user interface will further improve the experience for diverse users. Overall, the project serves as a promising step toward leveraging technology for inclusive communication solutions.

6. CONCLUSIONS

This project highlights the powerful role technology can play in making communication more inclusive. By transforming text and audio inputs into Indian Sign Language (ISL) animations, the application provides a bridge for individuals with hearing or speech impairments, allowing them to engage more effectively with others. The seamless combination of natural language processing, speech recognition, and 3D animation ensures that the system is both accurate and visually appealing, delivering a meaningful user experience.

While the application has proven successful in its current form, it also opens the door to exciting future possibilities. Expanding the animation database, improving support for complex words or phrases, and exploring real-time generation of signs can make the tool even more robust and flexible. This project is a testament to how thoughtful innovation can break down barriers, fostering a more inclusive world for everyone.

This application serves as a step forward in using technology to create an inclusive communication platform for individuals with hearing or speech impairments. By

converting spoken or written language into Indian Sign Language (ISL) animations, it bridges gaps and fosters understanding, offering users a tool that is both functional and intuitive. The integration of speech-to-text conversion, NLP techniques, and 3D animations ensures that the application not only translates words but also conveys context and meaning effectively.
The success of the system lies in its ability to simplify complex tasks, such as identifying grammatical structures and adapting outputs for different tenses, while maintaining a user-friendly interface. However, its potential for growth is vast. Enhancing the animation library and introducing real-time sign generation could elevate the system further. Ultimately, this project showcases how innovative solutions can bring about significant changes, making communication accessible and empowering individuals in their daily lives.

7. FUTURE WORK

The proposed application has laid a solid foundation for facilitating communication through Indian Sign Language (ISL) animations. However, there are numerous enhancements and expansions that can be undertaken to improve its functionality and reach. A critical area for future work is the expansion of the animation database. While the current system covers a range of words, adding more vocabulary, including idiomatic expressions, cultural references, and domain-specific terms, will make the application more versatile. Additionally, incorporating context-aware animations could help better convey nuances, such as tone or intent, which are essential for effective communication.

Another promising direction is the integration of advanced machine learning techniques to enable real-time animation generation. Using natural language understanding models, the system could dynamically create ISL signs for words or sentences not currently available in the database. This would significantly reduce the dependency on pre-rendered animations and improve scalability. Furthermore, incorporating gesture recognition and sign language translation capabilities could transform the system into a two-way communication tool, allowing users to input signs and receive spoken or written translations.

On the usability front, enhancing the interface to be more intuitive and responsive will be key. Features like adaptive font sizes, voice- assisted navigation, and customizable display settings can make the application accessible to users with varying needs. Multilingual support for both input and output would further expand its utility, especially in a linguistically diverse country like India. Lastly, developing a mobile-friendly version or offline functionality would enable users in areas with limited internet access to benefit from the application. By addressing these areas, the system can evolve into a comprehensive, real-time communication tool that bridges the gap between spoken and signed languages.

8. ACKNOWLEDGMENT

We extend our sincere thanks to Sheik Jamil Ahmed, whose guidance and mentorship were instrumental in shaping this project. We are also thankful to Presidency University for creating an environment that fostered growth and learning, and for providing us with the necessary tools and resources to bring this project to fruition.

9. REFERENCES

1. C. Ahuja, D. W. Lee, R. Ishii, and L.-P. Morency. "No gestures left behind: Learning relationships between spoken language and freeform gestures". In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 1884–1895, Online, Nov. 2020. Association for Computational Linguistics.

2. S. Alexanderson, G. E. Henter, T. Kucherenko, and J. Beskow". Style-controllable speech-driven gesture synthesis using normalizing flows." Computer Graphics Forum, 39(2):487–496, 2020.

3. M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. "2d human pose estimation: New benchmark and state of the art analysis". In 2014 IEEE Conference on Computer Vision and Pattern Recognition, pages 3686–3693, 2014.

4. N. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden. "Neural sign language translation. 03 2018."

5. M. Crawshaw. "Multi-task learning with deep neural networks: A survey", 2020.

6. E. P. da Silva, P. D. P. Costa, K. M. O. Kumada, and J. M. de Martino. "Facial action unit detection methodology with application in Brazilian sign language recognition". Pattern Analysis and Applications, 25:549 – 565, 2021

7. E. P. da Silva, K. M. O. Kumada, and P. D. P. Costa. "Analysis of facial expressions in Brazilian sign language (libras)". European Scientific Journal, ESJ, 2021.

8. S. Dachkovsky and W. Sandler. "Visual intonation in the prosody of a sign language". Language and Speech, 52(2-3):287–314, 2009. PMID: 19624033.

9. J. Devlin, M. Chang, K. Lee, and K. Toutanova. "BERT: pre-training of deep bidirectional transformers for language understanding". CoRR, abs/1810.04805, 2018.

10. I. Grishchenko and V. Bazarevsky. "Media pipe holistic — simultaneous face, hand and pose prediction, on device", Dec. 2020.