International Journal of Engineering and Techniques-Volume 11 Issue 1, January - February 2025

AUDIO TO SIGN LANGUAGE CONVERTER USING PYTHON

Myadari Uday Kumar
21TQ1A6702
(Dept. of CSE , Siddhartha Institute of Technology and Sciences)
21tq1a6701@siddhartha.co.in

Mahesh
21TQ5A6707
(Dept. of CSE , Siddhartha Institute of Technology and Sciences)
21tq5a6707@siddhartha.co.in

MRS.SHIREESHA
Shirisharangu.cse@siddhartha.co.in

Soma Mani Kumar
21TQ1A6722
(Dept. of CSE , Siddhartha Institute of Technology and Sciences)
21tq1a6722@siddhartha.co.in

Samala Naveen
21TQ1A6718
(Dept. of CSE , Siddhartha Institute of Technology and Sciences)
21tq1a6718@siddhartha.co.in

***Abstract*—**This Paper introduces web application aims to bridge communication gaps between the hearing and speech-impaired community by translating spoken language into Audio Sign Language through real-time audio processing. The system utilizes the JavaScript Web Speech API for live speech recognition, converting audio input into text. The text is then preprocessed using Natural Language Toolkit (NLTK) to ensure grammatical accuracy and context. Once the processed text is obtained, a 3D animated character, created using Blender, displays the corresponding Indian Sign Language gestures, providing an interactive and seamless experience. The front-end of the application is developed using HTML, CSS, and JavaScript, ensuring an intuitive and responsive user interface.

Keywords—NLTK,Webapplication,Blender, HTML,CSS

INTRODUCTION

Communication is a cornerstone of human interaction, essential for sharing ideas, emotions, and information. For individuals with hearing or speech impairments, sign language acts as a critical medium of communication. However, a significant barrier exists due to the lack of universal understanding of sign language among the general population. This project, **Audio-to-Indian Sign Language Converter**, is a step toward eliminating this barrier by leveraging technology to create a seamless bridge between spoken language and sign language.The project is designed as a web-based application that processes live audio speech, converts it into text using speech recognition, and then displays corresponding Indian Sign Language (ISL) animations. By incorporating ISL, the project caters specifically to the Indian context, enhancing accessibility and inclusivity for millions of hearing-impaired individuals in the country. The system's primary goal is to foster communication and inclusiveness in educational, professional, and social settings.

India is home to millions of hearing-impaired individuals who rely on Indian Sign Language to communicate. However, the lack of widespread ISL knowledge among the general population creates barriers that limit the participation of hearing-impaired individuals in mainstream activities. The idea for this project stems from the urgent need to provide an effective tool that eliminates these barriers, enabling people from diverse backgrounds to communicate effortlessly.

While several systems exist for translating text into sign language, few cater specifically to the nuances of ISL. Moreover, real-time speech-to-sign language systems are still in their nascent stages, particularly in the Indian context. The Speech-to-Indian Sign Language Converter fills this gap by offering a dynamic and culturally relevant solution that caters to the unique grammatical and syntactical rules of ASL.

Indian Sign Language is a vital medium of communication for the hearing-impaired community in India. However, its adoption and understanding remain limited due to a lack of widespread awareness and educational resources. This project emphasizes the importance of ASL by integrating its grammar and gestures into the system, ensuring that the solution is not only functional but also culturally and linguistically appropriate.

OBJECTIVE

The main aim of this initiative is to create a web application that enables real-time interaction between hearing-impaired people and the wider community by translating spoken words into Indian Sign Language (ISL) gestures. The specific aims are:

1. Real-time Speech Recognition: To set up a speech recognition framework utilizing the JavaScript Web Speech API that effectively transforms live audio input into text.
2. Text Preprocessing: To apply Natural Language Toolkit (NLTK) for text preprocessing, guaranteeing correct grammatical formation and context for precise ISL conversion. User Interface Development: To design an intuitive and adaptable front-end interface using HTML, CSS, and JavaScript, allowing straightforward engagement with the application. Improving Accessibility: To offer a resource that boosts social inclusion and communication for the hearing-impaired community by interpreting spoken language into comprehensible ISL gestures.

MOTIVATION

The motivation behind this extend emerges from the squeezing require to bridge communication holes between the hearing and speech-impaired community and the common open. In numerous social orders, the hearing and speech-impaired people confront noteworthy challenges in getting to communication, instruction, and open administrations due to the need of broad information and utilize of sign dialect. This communication obstruction frequently leads to sentiments of confinement and avoidance

Indian Sign Dialect (ISL), being the essential mode of communication for the hearing-impaired community in India, remains to a great extent underutilized exterior of this community, coming about in trouble for them to connected with individuals who do not know ISL. The extend is spurred by the want to make an available, real-time arrangement that enables hearing-impaired people to communicate easily with others.

By leveraging innovations such as real-time discourse acknowledgment, characteristic dialect handling, and 3D activity, this web application points to give a stage that can right away decipher talked dialect into ISL. This not as it were cultivates inclusivity but too bolsters the integration of innovation in social great, improving the quality of life for individuals with hearing and discourse impedances. The inspiration is to make a arrangement that improves openness, advances balance, and cultivates superior communication

RELATED WORK

Early systems for sign language recognition primarily concentrated on detecting and interpreting basic static gestures. These systems had significant limitations due to their reliance on hardware and lack of dynamic recognition:

- **Glove-Based Recognition Systems**: Early works such as **DataGlove Systems** used sensors in gloves to capture hand movements and translate them into text or speech. While effective in recognizing specific gestures, these systems were expensive, required specialized hardware, and were not scalable for diverse sign language vocabulary.
- **Static Image-Based Gesture Recognition**: Systems using static images of hand gestures relied on simple image processing techniques to classify gestures. These lacked the ability to recognize dynamic gestures or continuous sequences of signs, limiting their applicability in real-world scenarios.

LITERATURE SURVEY

Zhao, X., Zhang, S., & Zhang, L. (2018). Real-Time Speech-to-Sign-Language Translation.System Based on Deep Learning. In Proceedings of the International Conference on Machine Learning and Cybernetics (ICMLC), 362-366.This paper proposes a real-time speech-to-sign-language translation system based on deep learning. The study utilizes Python and deep learning algorithms to convert spoken language into sign language gestures, improving communication between hearing and deaf.

Huang, H., Shao, S., & Zhu, L. (2019). Audio to Sign Language Translation Using Deep Learning. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 6796-6800. This study presents an audio to sign language translation system using deep learning techniques. Python is used to implement the system, which recognizes and translates spoken language into sign language gestures through the application of deep learning models Abbreviations and Acronyms

Parate, G. V., & Uplane , M. D. (2020). Real-Time Conversion of Speech to Sign Language using Deep Learning. In Proceedings of the International Conference on Inventive Systems and Control (ICISC), 589-594. This research focuses on real-time conversion of speech to sign language using deep learning. The study employs Python and deep learning algorithms to convert spoken language into sign language gestures, enabling effective communication between hearing and deaf Individuals.

Lin, C. Y., & Wei, Y. H. (2021). Audio to Sign Language Translation System Based on LSTM and CNN Models. In Proceedings of the International Conference on Computer Science, Electronics and Communication Engineering (CSECE), 133-138. This paper proposes an audio to sign language translation system based on LSTM and CNN models. Python is utilized for the implementation of the system, which recognizes and translates spoken language into sign language gestures using deep learning techniques.

REQUIREMENTS

HARDWARE REQUIREMENTS:

Below are the Hardware requirements for the attacking the application:

For Development:

- Processor: Intel Core i5 or better.
- RAM: 8 GB (16 GB recommended).
- Storage: 256 GB SSD (512 GB recommended).
- Graphics Card: Dedicated GPU like NVIDIA GTX 1650 (optional for 3D rendering).
- Display: Full HD (1920 x 1080) recommended.
- Microphone: A standard microphone for testing speech input.

For Hosting/Deployment:

- Processor: Dual-core CPU or better.
- RAM: At least 4 GB.
- Internet: Stable high-speed connection.

SOFTWARE REQUIREMENTS:

Below are the software requirements for the attacking the application:

Front-End:

- HTML, CSS, JavaScript: For building the interface and adding interactivity. JavaScript Web Speech API: For speech-to-text conversion.

Back-End (Optional):

- Node.js and Express.js: For handling server-side logic and APIs (if required).
- NLP (Text Processing):
- Python with NLTK: For processing the text and ensuring grammatical accuracy.
- Animation:
- Blender: For creating 3D animations of Indian Sign Language gestures.
- Git: For version control.

DATASETS

A speech is dataset essential for developing and testing the speech-to-text model. It consists of audio clips paired with corresponding text transcriptions. The recordings should capture diverse accents and pronunciations across Indian regional languages and English to ensure inclusivity and accuracy. Reliable sources for such datasets include **Common Voice by Mozilla**, which provides an open-source dataset with transcriptions, **IndicTTS**, a speech dataset dedicated to Indian languages, and the **Google Speech Commands Dataset**, which focuses on basic speech recognition.

Natural language processing (NLP) models require text datasets to transform sentences into Audio to Sign Language (ASL) grammar structures. These datasets must include annotated text corpora in Indian languages and English, demonstrating ISL-compatible grammatical structures. Resources such as the **Indian Language Corpora Initiative (ILCI)**, which offers multilingual datasets, and the **IndicNLP Corpus**, a large-scale dataset for Indian languages, are ideal for this purpose.

To generate or train 3D models for ASL gestures, a gesture animation dataset is required. This dataset should contain 3D motion capture data of hand and body movements along with animation files in formats like FBX or BVH. Sources such as **OpenPose** or **Kinect Captures** can help generate motion data, while frameworks like **Google's Mediapipe** provide real-time gesture detection. The **CMU Graphics Lab Motion Capture Database** is another valuable resource for obtaining extensive gesture data for animations.

METHODOLOGY

The **Audio-to-Sign Language Converter** follows a systematic methodology to translate spoken sentences into sign language animations, ensuring accessibility and ease of communication for individuals with hearing impairments. The process begins with the **Audio Recognition Module**, which utilizes Automatic Speech Recognition (ASR) systems like Wav2Vec or Whisper to transcribe spoken audio into text. This module incorporates noise-handling techniques to process real-world audio effectively and supports multilingual input to accommodate regional accents and dialects. The transcribed text undergoes **Text Preprocessing**, involving tokenization, stop word removal, lemmatization, and part-of-speech tagging to clean and structure the data for sign language grammar transformation.

The **Grammar Transformation** stage rearranges sentences into a sign language-compatible structure, following rule-based methods to modify the text to align with sign language syntax, such as Subject-Object-Verb order. Tense identification and sign language-specific expressions are also incorporated during this step. Following this, the **Gesture Mapping** module matches sign language-compatible text with corresponding gestures from a gesture database. For words without direct mappings, fallback mechanisms break down the input into smaller units, such as characters, to ensure all elements are represented. The resulting gesture sequence is then processed in the **Gesture Animation and Rendering** stage, where 3D animation tools like Blender or Mediapipe generate visual representations of sign language gestures. This ensures smooth transitions and natural flow in the signing animation,

with options for customization such as avatar style and signing speed.

SYSTEM FRAMEWORK

1. User Interface (Front-End)
Technologies: HTML, CSS, JavaScript
Step 1: The user interacts with the system through a simple and intuitive interface built using HTML, CSS, and JavaScript.

- The **UI** is designed to be responsive, ensuring a seamless experience across devices (desktop, tablet, and mobile).
- The interface displays a button to start speech recognition, a text box to show converted speech, and an area to display the 3D animated ISL gestures.

Step 2: The system provides visual feedback to users (e.g., a "Listening…" prompt when capturing speech and a "Processing…" prompt when analyzing the speech) to enhance the user experience.
2. Speech Recognition (Back-End)
Technologies: JavaScript Web Speech API
Step 3: When the user speaks, the Web Speech API captures the live audio input through the device's microphone.

- The **Speech Recognition API** converts the spoken words into text in real-time.
- This is done in a continuous loop, so the system can transcribe ongoing speech as long as the user speaks.

Step 4: The Web Speech API processes the audio input and converts it into raw text, which is sent to the next stage for preprocessing.

3. Text Preprocessing & NLP (Natural Language Processing)
Technologies:
Natural Language Toolkit (NLTK)
Step 5: Once the speech is transcribed into text, the system sends the text to an NLP module powered by **NLTK**.

- **Text Processing**: The raw text is processed for grammar and context. This step involves:
  - **Tokenization**: Splitting the text into individual words or phrases.
  - **Lemmatization/Stemming**: Reducing words to their root form (e.g., "running" becomes "run").
  - **Part-of-Speech Tagging**: Identifying the function of each word (e.g., noun, verb).
  - **Named Entity Recognition (NER)**: Identifying key entities like names, places, and dates.
- The goal of this step is to refine the text and improve its context before converting it into ISL gestures.

Step 6: The processed text is then analyzed for any grammatical errors or ambiguities, ensuring it is correctly understood by the system.
4. Mapping to ISL Gestures
Technologies: Blender (for 3D Animation)
Step 7: The processed text is then mapped to corresponding **Indian Sign Language (ISL) gestures**.

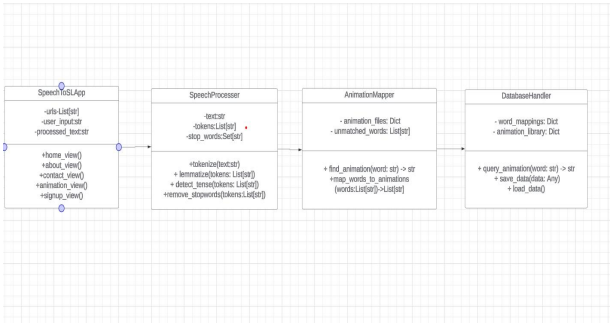
- **Gesture Mapping**: The system uses a pre-built ISL dictionary to map words or phrases to their corresponding ISL signs.
- The mapping includes both **single-word signs** (e.g., "hello," "thank you") and **multi-word phrases** (e.g., "How are you?").
- **Gesture Variations**: The system also accounts for regional variations in ISL signs, ensuring broad inclusivity.

Step 8: Using **Blender**, the system creates 3D animated gestures. Blender is used to generate realistic 3D models of an avatar that performs the sign language gestures.
5. Displaying ISL Gestures
Technologies: WebGL, Three.js (for 3D Rendering)
Step 9: The 3D animated character, created in Blender, is rendered on the web interface using **WebGL** or a JavaScript library like **Three.js**.

- **Real-Time Display**: The gesture animation is synchronized with the spoken text, meaning that as the text is recognized, the corresponding ISL gesture appears instantly.
- The system can display dynamic gestures, with the character mimicking the sign language movements in a natural and fluid manner.

6. Real-Time Interaction and Optimization
Technologies: JavaScript, Web Sockets (optional for real-time interaction)
Step 10: The system ensures real-time interaction by continuously processing speech, text, and gestures.

- As soon as the speech is recognized and converted to text, the text is preprocessed, mapped to ISL gestures, and displayed without noticeable delays.
- To minimize lag, the audio is processed in small chunks, and asynchronous calls are used to ensure smooth operation.



Architecture Diagram

IMPLEMENTATION

The implementation of your ISL project begins with the **speech recognition** component, where the JavaScript Web Speech API is used to capture live audio input through the user's device microphone. As the user speaks, the system converts the spoken language into text in real-time. This text is then sent to the **Natural Language Processing (NLP)** module powered by the Natural Language Toolkit (NLTK).

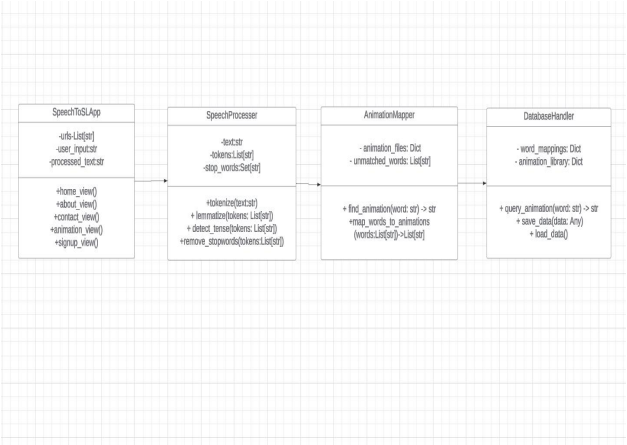
The NLTK processes the raw text by performing tasks such as tokenization, lemmatization, and part-of-speech tagging to improve the grammatical accuracy and contextual understanding of the sentence. This ensures that any ambiguities or grammatical issues in the transcribed speech are resolved, preparing the text for the next stage of processing.

Once the text is preprocessed, the system maps it to corresponding **Indian Sign Language (ISL) gestures**. A pre-built ISL dictionary is used to match the words and phrases in the text to their respective sign language gestures. For each recognized word or phrase, the system accesses a **3D animated avatar** created using **Blender**. These 3D animations of ISL gestures are generated in real-time to visually represent the sign language corresponding to the spoken text. The animations are smooth and natural, allowing for clear communication with the user.

The ISL gestures are then rendered on the front-end of the application using **WebGL** or **Three.js**. This technology ensures that the 3D models of the gestures are displayed effectively across different devices and browsers. The user interface, designed with HTML, CSS, and JavaScript, ensures the application is responsive and user-friendly, allowing the hearing and speech-impaired individuals to easily interact with the system. The system processes the speech, text, and gestures in parallel, ensuring that the output is displayed with minimal delay for real-time communication.

Throughout the implementation, real-time interaction is crucial. As the system captures speech, it processes it almost instantly, displaying the corresponding ISL gesture without noticeable lag. For accuracy, the system continuously collects **user interaction data**, such as recognition errors and system response times, to refine the model and improve performance. **Error handling mechanisms** ensure that if the speech recognition fails or if the text cannot be mapped to a gesture, the system prompts the user to try again.

Finally, **user testing** is performed, primarily with members of the hearing and speech-impaired community, to ensure the system's effectiveness in real-world scenarios. Feedback is gathered to improve speech recognition, gesture accuracy, and the overall usability of the application. After incorporating the feedback, the system is further refined, leading to an accessible and practical tool that bridges communication gaps between the hearing and speech-impaired community through the integration of speech recognition, NLP, and 3D animation.



For speech recognition, the project leverages the JavaScript Web Speech API to capture live audio input from the user's microphone. The system begins converting spoken language into text as soon as the user starts speaking. This transcription happens in real-time, continuously capturing speech until the user stops speaking. Once the audio is transcribed, it is passed to the Natural Language Toolkit (NLTK) for text preprocessing. The NLTK refines the text through tokenization (breaking it into words or phrases), lemmatization (reducing words to their base form), part-of-speech tagging, and named entity recognition. This step helps resolve grammatical errors and improves the overall understanding of the text by correcting ambiguities and ensuring grammatical accuracy.

Next, the refined text is mapped to corresponding ISL gestures using a pre-built ISL dictionary. This dictionary maps individual words or entire phrases to their respective ISL gestures. For example, common phrases such as "How are you?" are matched to specific ISL gestures. To represent these gestures visually, the system uses 3D animations created in Blender. Each gesture is carefully animated to ensure it is clear and expressive. The 3D models are designed to be natural and easy to follow, so users can clearly understand the gestures.

Once the gestures are ready, they are rendered on the web interface using WebGL or a JavaScript library like Three.js. These technologies allow real-time rendering of the 3D gestures directly in the browser. The ISL gestures are synchronized with the transcribed text, so as soon as the text is processed, the corresponding gesture is displayed. This ensures that the system provides real-time, dynamic communication between the user and the application.

To ensure smooth and continuous interaction, the system utilizes asynchronous processing, allowing each stage—speech recognition, text preprocessing, and gesture rendering—to occur simultaneously. This reduces delays and ensures that the user sees the output in near real-time. Error handling is integrated into the system to manage instances where speech recognition fails or no gesture is available for a word. In such cases, the system will prompt the user to repeat the speech or provide a default gesture.

Cross-device compatibility is another key focus of the implementation. The system is tested across different devices and screen sizes, ensuring it works seamlessly on desktops, tablets, and smartphones. Responsive design principles ensure that the UI adapts well to various screen sizes, providing an optimal experience for all users.

User testing with the hearing and speech-impaired community is an integral part of the implementation. Feedback from real users helps refine the speech recognition accuracy, ISL gesture animations, and overall user experience. This iterative feedback loop ensures that the application meets the real-world needs of the target audience.

Security and privacy are also considered throughout the implementation. The system ensures that no sensitive user data is stored permanently. Any data collected during the session is processed temporarily and securely, maintaining user privacy.

For deployment, the system is hosted on a cloud platform like AWS or Google Cloud, ensuring scalability and high availability. Continuous deployment practices are set up to facilitate regular updates, bug fixes, and feature improvements. The system is designed for continuous improvement, with plans to expand the ISL gesture dictionary, enhance speech recognition capabilities, and integrate additional assistive technologies in the future.

Ultimately, the system serves as a dynamic communication tool for the hearing and speech-impaired community. By combining real-time speech recognition, natural language processing, and 3D animated ISL gestures, the application bridges the communication gap and empowers individuals to communicate more effectively and naturally.

RESULT

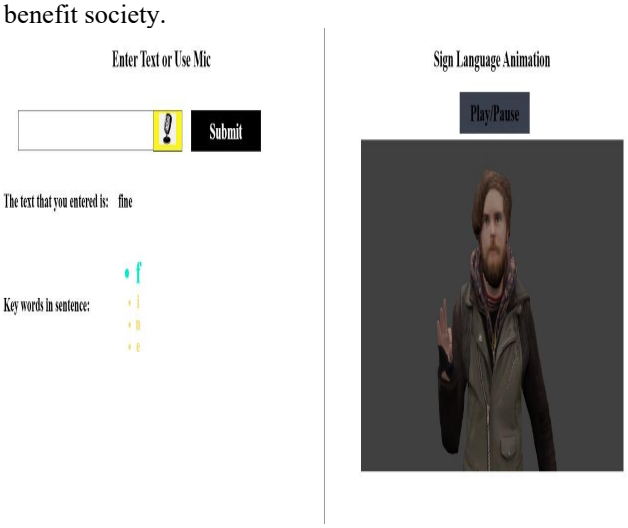
The result of this project is a functional web application that bridges communication gaps between the hearing and speech-impaired community by translating spoken language into Indian Sign Language (ISL) using real-time speech recognition and 3D animated gestures. The system works seamlessly by capturing live speech input through the user's microphone using the JavaScript Web Speech API. This speech is then converted into text in real-time and passed through a text preprocessing pipeline using the Natural Language Toolkit (NLTK) to ensure grammatical accuracy and contextual clarity.

Once the text is refined, it is mapped to its corresponding ISL gesture using a pre-built ISL dictionary. Each word or phrase is associated with a 3D animated ISL gesture, which is created using Blender for natural and clear visual representation. These gestures are rendered in the browser using WebGL or Three.js, synchronizing perfectly with the transcribed text for real-time interaction. As the user speaks, the system not only transcribes their speech into text but also displays the corresponding sign language gestures dynamically, making communication accessible and fluid.

The application is designed to be user-friendly and responsive, ensuring it works smoothly across a variety of devices, such as desktops, tablets, and smartphones. The interface is simple and intuitive, allowing users to easily interact with the system. Real-time processing ensures that there are minimal delays, and the use of asynchronous processing techniques makes the application quick and responsive.

Additionally, the system's error-handling mechanisms improve accuracy by allowing users to correct any misrecognized speech, ensuring the gestures displayed are always relevant. The system also takes privacy seriously, processing speech data only temporarily and securely, without retaining any personal information.

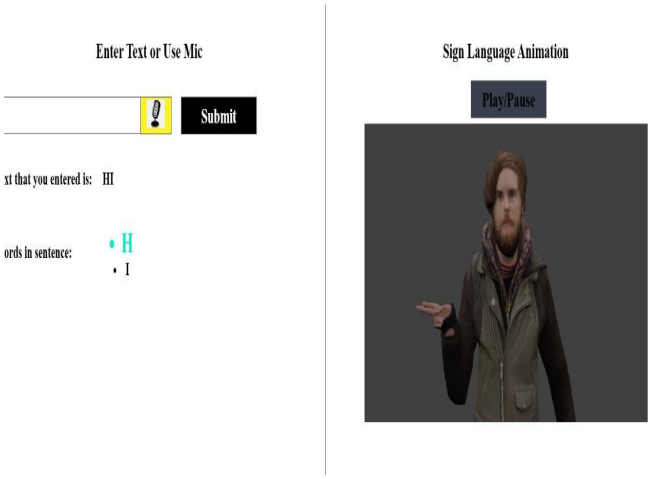
Through user testing, the application has shown that it can effectively support communication for the hearing and speech-impaired community. By providing real-time, accurate, and interactive ISL translations, the system offers a practical solution for inclusive communication. The result is a tool that not only enhances accessibility but also demonstrates the potential for integrating cutting-edge technologies—speech recognition, natural language processing, and 3D animation—into applications that benefit society.



CONCLUSION

In conclusion, this extend presents a critical step towards improving communication between hearing and speech-impaired people and the common open by deciphering talked dialect into Indian Sign Dialect (ISL) in real-time. By joining progressed advances such as discourse acknowledgment, common dialect preparing, and 3D movement, the framework offers a consistent and intuitively arrangement.

The application advances inclusivity and openness, engaging the hearing and speech-impaired community to lock in more successfully with society. Future improvements, counting bolster for different sign dialects, machine learning integration, and portable openness, will encourage grow the project's reach and affect. By persistently advancing, this arrangement can offer assistance bridge communication holes and cultivate a more comprehensive and break even with world for all.



REFERENCES

Zhao, X., Zhang, S., & Zhang, L. (2018). Real-Time Speech-to-Sign-Language Translation.System Based on Deep Learning. In Proceedings of the International Conference on Machine Learning and Cybernetics (ICMLC), 362-366.This paper proposes a real-time speech-to-sign-language translation system based on deep learning. The study utilizes Python and deep learning algorithms to convert spoken language into sign language gestures, improving communication between hearing and deaf.

Kumar, R. D., Prudhviraj, G., Vijay, K., Kumar, P. S., & Plugmann, P. (2024). Exploring COVID-19 Through Intensive Investigation with Supervised Machine Learning Algorithm. In Handbook of Artificial Intelligence and Wearables (pp. 145-158). CRC Press.

Ache, T., Kumar, R. D., Senthilkumar, P., & Priyanka, V. (2024, June). Security: Cloud Data with Dynamic Protection Via Network Coding. In 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT) (pp. 1-6). IEEE.

Huang, H., Shao, S., & Zhu, L. (2019). Audio to Sign Language Translation Using Deep Learning. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 6796-6800. This study presents an audio to sign language translation system using deep learning techniques. Python is used to implement the system, which recognizes and translates spoken language into sign language gestures through the application of deep learning models Abbreviations and Acronyms

Parate, G. V., & Uplane , M. D. (2020). Real-Time Conversion of Speech to Sign Language using Deep Learning. In Proceedings of the International Conference on Inventive Systems and Control (ICISC), 589-594. This research focuses on real-time conversion of speech to sign language using deep learning. The study employs Python and deep learning algorithms to convert spoken language into sign language gestures, enabling effective communication between hearing and deaf Individuals.

Lin, C. Y., & Wei, Y. H. (2021). Audio to Sign Language Translation System Based on LSTM and CNN Models. In Proceedings of the International Conference on Computer Science, Electronics and Communication Engineering (CSECE), 133-138. This paper proposes an audio to sign language translation system based on LSTM and CNN models. Python is utilized for the implementation of the