Grenze International Journal of Engineering and Technology, January Issue

An Approach of Translating Speech to Sign Language using Machine Learning techniques

Gouthami Velakanti[1], Dr. N. Gayatri[2], Gogikar Naveen Kumar[3] and K. Srinivas[4]
[1-2]Kakatiya Institute of Technology and Science Warangal, India
Email: vg.cse@kitsw.ac.in, ng.cse@kitsw.ac.in
[3-4]Vaagdevi Engineering College, Warangal
Email: naveengogikar@vecw.edu.in, kalyansr555@gmail.com

***Abstract*— This paper presents a real-time speech-to-sign language translation system aiming to bridge communication gaps between the hearing and deaf communities. Leveraging advanced machine learning and computer vision techniques, the system translates spoken language into sign language gestures in real-time. The abstract highlights the system's innovation in converting verbal communication to visual sign language, enhancing inclusivity and accessibility for the deaf and hard-of-hearing individuals. The paper discusses the underlying architecture, including the integration of speech recognition, natural language processing, and sign language animation modules. The system's evaluation highlights its accuracy, speed, and effectiveness in enhancing cross-modal communication, breaking barriers between hearing and deaf individuals for greater integration and understanding.**

***Keywords*— Speech, Sign Language, Machine Learning, Computer vision techniques, Natural Language processing (NLP), Animation modules.**

I. INTRODUCTION

In order to communicate the context of topic and make it understandable, sign language is the best option. Because this language incorporates expressions with gestures. In the world, there large number of languages, sign language is the one among them used by physically impaired people like hearing and speaking, in order to communicate. Around 20%-21% of world population in the world are suffering from hearing loss. Nearly, 5-6 crores people in India are using sign language for their communication.

It is observed that there is a minor improvement in the access to content for this population. Still, it is a great struggle and constant issue for deaf-mute people in order to gain access to information on education, employment and communications. This is due to the available small number of sign language implementing schools located in big cities than rural regions. Therefore, it has become a major issue to concentrate on the path providing solution for these deaf-mute people making platform independent and ensuring self-learning. This article proposes a novel approach, easy-to-use and time efficient online platform for deaf-mute people that would serve as an effective mode of communication and learning for them.

Technology is rapidly changing and the way the world operates is improving. This necessitates the breaking of communication barriers for the hearing and speech impaired community. To this day, there are only about 250 certified sign language interpreters in India for a deaf-mute population between 1.8 million and 7 million.

Grenze ID: 01.GIJET.12.1.86

The Audio to Sign Language Converter is a unique application designed to assist individuals with hearing difficulties in communicating effectively with others. Using advanced technologies such as Google Speech API and Sphinx, the app translates spoken words into written text, offering a simple and user-friendly interface through EasyGui. This accessible tool captures live speech or audio recordings via PyAudio and then transforms them into Indian Sign Language (ISL) images or GIFs on the screen.

The application goes beyond mere translation, striving to enhance overall communication between those with hearing impairments and those unfamiliar with sign language. It has a straightforward design that makes it easy for anyone to use on computers or mobile phones. The primary aim is to create a seamless and inclusive communication experience.

As the converter evolves, it seeks to encompass the entire ISL vocabulary, including both manual and non-manual signs. This scalability positions the application as a versatile and dynamic tool, adaptable to the changing needs of the hearing-impaired community. The overarching goal is to promote inclusivity, ensuring that everyone, regardless of their communication abilities, can engage meaningfully in various aspects of life.

Beyond being a technological advancement, the Audio to Sign Language Converter is a significant step towards breaking down communication barriers. It aspires to foster a world where people can easily express themselves and understand one another, contributing to a more connected and compassionate society. Through this application, the vision is to make communication universally accessible, promoting understanding and empathy among diverse groups of people.

Before the invention of sign language, techniques for speech translation included manual interpretation by individuals familiar with both spoken and visual languages, such as priests or family members. Gestures and facial expressions were also used to convey meaning. Some cultures relied on written communication or simplified visual representations. Non-verbal cues and context played a significant role in understanding and conveying messages.

After the invention of sign language, techniques for speech translation include:

- Speech recognition to convert spoken language into text.
- Natural language processing for understanding the meaning of the text.
- Machine translation to convert the text into sign language.
- Animation or avatar creation for displaying sign language.
- User interface design for seamless interaction with the translation system.

II. LITERATURE REVIEW

Ref [1] & Ref [7] states that various students in their research used many ways to identify various sign languages and movement of hands. Few used static movement gestures Ref[8] and some used videos. This paper deals with Indian sign languages. In this paper authors takes written statements in English as input and generates output as ISL structure.

Ref [2] suggested a speech-to-British sign language technology which is used in communication for a deaf person with post office clerk. The authors used a formulatic grammar that the system uses in communication. The technology makes use of phrase lookup database and a set of phrases which are predefined. This is domain specific system.

Ref[3] developed a project named SignSynth that uses ASCII-Stokoe to represent signs that displays an animated output which is generated through the conversion of Virtual Reality Modelling Language. During translation process any language should be translated to be valid both syntactically and semantically.

Ref [4] & Ref [9] proposed a translator called ViSiCAST which is an example of a transfer architecture system which translates English to BSL by using HPSG (Head-driven Phrase Structure Grammar) and a Prolog based freeware system ALE (Attribute Logic Engine). HamNoSys is used to represent the phonology of the system.

Ref [5] developed an application for railway reservation domain. HamNoSys developed INGIT, which is a Ref[10] machine translation system which takes input from a reservation clerk, converts it to ISL, and presents the result as animated ISL-gloss strings. TESSA which uses a purely formulaic approach. Hindi and Ref[11] have same word order so that the differentiation of structure is not required.

Ref [6] suggested an Automated Speech to Sign Language Conversion Ref [12] making use of Google API and NLP in which speech is taken as input. Using Google API, it is then converted to speech. The text that is obtained will contain some infected parts which are removed using Natural Language Processing in the further step. For each word or character, a matching operation is carried out with the video database of the sign language. The matched videos are then retrieved from the database and are made a single video that depicts the complete string in sign language. The output of the system is a video finally.

III. METHODOLOGY

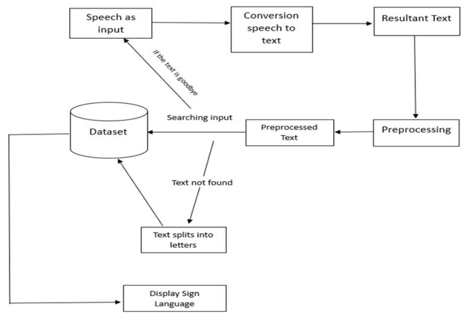

Fig 1: Architecture Diagram showing the process of Speech translation to Sign language

A. Speech as input

Users speak into the microphone, providing verbal input to the application. For example, a user might say, "Are you hungry?"

B. Conversion from Speech to Text with Recurrent Neural Networks (RNNs)

Sequential data are handled by specific class of artificial intelligence neural networks called Recurrent Neural Networks (RNNs). One of the important features of RNN’s,

Unlike feedforward neural networks, which process inputs independently of each other, RNNs maintain an internal state or memory that allows them to retain information about past inputs. This memory mechanism enables RNNs to capture temporal dependencies in sequential data, making them well-suited for tasks such as speech recognition.

In the context of speech-to-text applications, RNNs analyze sequential audio frames extracted from an input speech signal. At each time step, the RNN processes an audio frame along with its internal state from the previous time step. By updating its internal state based on both the current input and the context learned from past inputs, the RNN can effectively model the temporal dynamics of speech.

The recurrent nature of RNNs allows them to understand the sequential nature of spoken language, recognizing patterns and structures[13] inherent in speech. This understanding enables RNNs to decipher the context of the spoken words, such as distinguishing between words with similar sounds but different meanings, understanding grammatical structures, and recognizing speech patterns specific to different languages or dialects.

Additionally, RNNs can adapt to variations in speech tempo, accent, and pronunciation, as they learn from the sequential patterns present in the training data. Through the training process, RNNs adjust their internal parameters to minimize the difference between the predicted text output and the ground truth transcription, improving their accuracy and performance over time. Overall, the ability of RNNs to capture temporal dependencies in sequential data makes them a powerful tool for speech recognition tasks, enabling the conversion of spoken language into text with high accuracy and efficiency.

- Each word in the spoken phrase is converted into a sequence of features, such as spectrogram frames representing the audio signal.
- These features are fed into the RNN one at a time, starting from "Are", then "you", and finally "hungry".
- At each time step, the RNN processes the current input along with the hidden state from the previous time step.
- The RNN updates its internal state based on the current input and the context learned from previous inputs.
- As the RNN processes each word, it maintains an internal representation of the sequence so far, incorporating information from all previously seen words.
- This allows the RNN to capture the context of the entire phrase, understanding that "hungry" follows "you" and "Are" initiates a question.
- Once the entire sequence is processed, the RNN generates the corresponding text output based on its learned representation of the input sequence.
- In this case, the output would ideally be "Are you hungry?", accurately transcribing the spoken phrase.

C. Resultant Text

The recognized text obtained from the speech recognition step is: "Are you hungry?"

D. Preprocessing

The obtained text may undergo preprocessing steps to make it suitable for further processing. Preprocessing can involve tasks such as converting the text to lowercase, removing punctuation, and handling special cases or commands. For example, the text may be preprocessed to remove punctuation, resulting in: "are you hungry"

E. Pre-processed Text

Once the preprocessing is complete, the text is considered preprocessed. This preprocessed text is then ready for translation into sign language. In this case, the preprocessed text is: "are you hungry"

F. Dataset

The application utilizes a dataset containing mappings between spoken words or phrases and their corresponding Indian Sign Language (ISL) representations. However, if the phrase spoken by the user is not found in the dataset, the application has an additional feature.

G. Displaying sign language or symbolic representation

If the phrase is not found in the dataset, the application switches to displaying a symbolic representation for each character in the phrase. This symbolic representation could be a simplified visual representation or icon for each character.

H. Display Sign Language

Frames represent individual images comprising an animation, crucial for handling GIFs. Upon loading an image, frames are extracted and stored in the `self.frames` list. The `next_frame` method sequentially displays these frames, creating an illusion of animation. The `self.loc` variable tracks the current frame being displayed. Frames are converted into `PhotoImage` objects for compatibility with Tkinter. The `ImageLabel` widget effectively manages the display of GIF animations. This approach enables smooth playback of GIFs within the Tkinter interface. The code ensures efficient handling and rendering of animated content.
Based on the preprocessed text "Hello how are you today", the application retrieves the relevant ISL images or GIFs for each word from the dataset. It then displays the corresponding ISL signs for "are," "you," and "hungry" sequentially, enabling communication in sign language.

IV. Implementation

The provided code is a Python implementation of a "Hearing Impairment Assistant" program. Let's break down the implementation for a paper presentation:

- Sign language GIFs are displayed in a tkinter window using the `ImageLabel` class, which continuously cycles through the frames of the GIF.
- Alphabet images are displayed using matplotlib's `imshow()` function.

The provided implementation offers an interactive platform catering to the needs of individuals with hearing impairments, enabling them to communicate effectively through speech input. By leveraging state-of-the-art speech recognition technology, the program accurately transcribes spoken words into text, ensuring real-time interpretation and interaction. This integration of speech recognition capabilities not only facilitates seamless communication but also empowers users to express themselves naturally and spontaneously.
Moreover, the program's incorporation of sign language GIFs and alphabet images provides users with valuable visual feedback corresponding to their speech input. This visual feedback mechanism serves as a bridge between spoken language and visual representations, enhancing comprehension and aiding in conveying messages effectively. The inclusion of sign language GIFs caters to users familiar with sign language, offering an intuitive means of communication that transcends linguistic barriers. Through the display of relevant sign language gestures, the program fosters better understanding and engagement, promoting inclusivity and accessibility.
Furthermore, for users preferring written language, the program displays alphabet images corresponding to recognized speech. This feature enables users to visualize the textual content, facilitating communication through written text. By accommodating diverse communication preferences, the program ensures inclusivity and empowers users to choose the communication mode that best suits their needs.
The utilization of graphical user interface (GUI) components, facilitated by libraries such as `easygui` and `tkinter`, further enhances user interaction and navigation. Through intuitive dialogs and windows, users can initiate actions and receive feedback seamlessly, enhancing the overall user experience. The program's multimodal approach, combining speech recognition with visual feedback, underscores its versatility and adaptability in meeting the diverse needs of users with hearing impairments.

In essence, this implementation serves as a practical assistive technology solution, exemplifying the transformative impact of technology in promoting inclusivity and empowerment. By bridging communication gaps and providing tailored support, it exemplifies the pivotal role of assistive technology in enhancing quality of life and fostering inclusive environments for individuals with disabilities.

V. Results

The provided code is a Python script designed to function as a hearing impairment assistant. When run, it creates a graphical user interface (GUI) using the `easygui` library, presenting the user with two options: "Live Voice" and "All Done!". Here's what happens when each option is selected:

A. Live Voice

- When the "Live Voice" button is clicked, the script activates the `func()` function.
- The `func()` function starts listening to the microphone input for speech using the `speech_recognition` library.
- Upon recognizing speech, the script checks if the recognized speech matches any predefined phrases stored in the `isl_gif` list.
- If a match is found, it displays the corresponding American Sign Language (ASL) gif associated with that phrase using a tkinter window.
- If the recognized speech contains alphabets, it displays images of those alphabets.
- The script continues to listen for speech input until the user says "goodbye", "good bye", or "bye", at which point it terminates the `func()` function and returns to the GUI.

B. All Done!

- When the "All Done!" button is clicked, the script exits the loop and terminates.

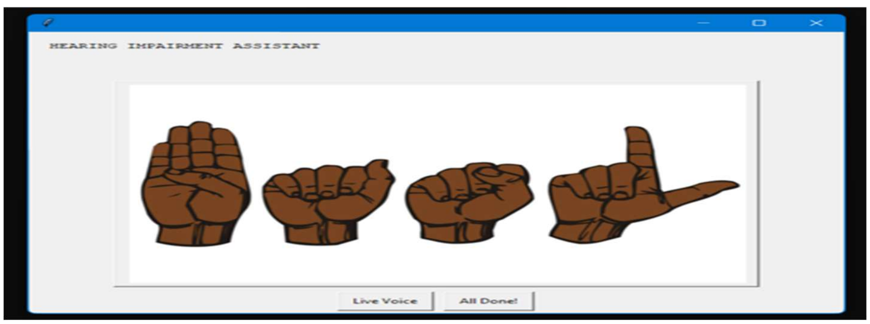

Fig 2: Hearing Impairment Assistance window

If after clicking the "Live Voice" button, the script displays "I am Listening", it indicates that the script has successfully entered the `func()` function and is currently listening for speech input.

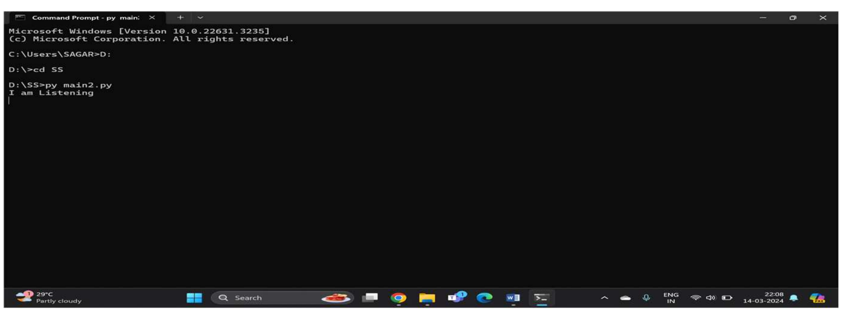

Fig 3: Hearing window accepting speech

If the recognized phrase is "are hungry", the script will match it against the list of predefined phrases (`isl_gif`). Since "are hungry" is included in this list, the script will display the corresponding ASL gif associated with this phrase.

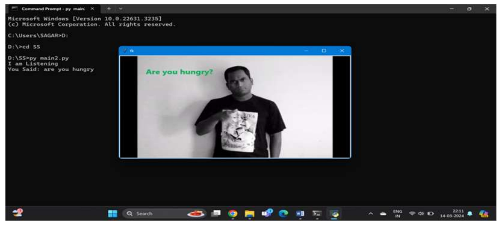

Fig 4: Output of Hearing window - displaying sign for given input speech

*If the recognized phrase is not present in the predefined list (`isl_gif`), the script will execute the portion of code responsible for displaying alphabet images corresponding to the recognized phrase. Here's how it works:

- The script iterates through each character in the recognized phrase.
- For each character, it checks if it is present in the list `arr` which contains alphabets.
- If the character is an alphabet (present in `arr`), the script displays the corresponding image of that alphabet.

Let's say a phrase that is not in the list and is relatively short. For example:
"I like dog."
This phrase is not present in the `isl_gif` list, and it's short enough to provide a quick demonstration of how the script handles unrecognized phrases. When the script encounters this phrase, it will display images of the alphabets corresponding to each letter in the phrase ("I", "l", "i", "k", "e", "d", "o", "g") as visual feedback.
Here's how the script would handle this:

- It will recognize that "I like dog" is not present in the `isl_gif` list.
- It will then proceed to display images of the alphabets "I", "l", "i", "k", "e", "d", "o", "g" using the alphabet images stored in the directory.

This behavior ensures that even if the recognized phrase is not explicitly listed in the `isl_gif` list, the script still provides visual feedback by displaying the corresponding alphabet images.

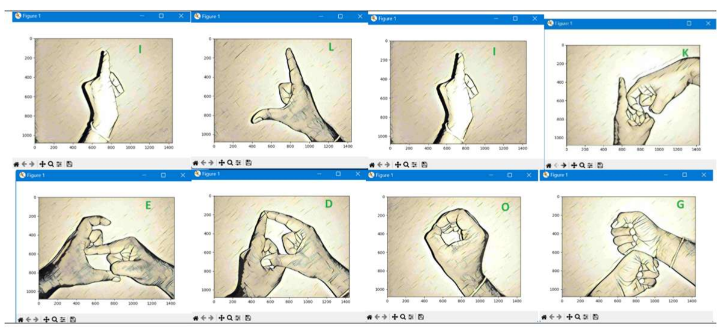

Fig 5: Sign images of corresponding alphabets

If the recognized phrase matches one of the termination phrases ("goodbye", "good bye", or "bye"), the script prints "Time to say goodbye!" and breaks out of the `while True` loop, effectively terminating the `func()` function and returning to the GUI loop. This ensures that the user can exit the speech recognition loop and return to the GUI whenever they want to end the interaction.

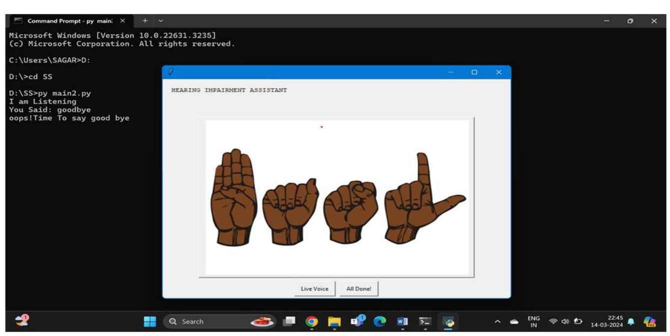

Fig 6: Hearing Impairment Assistance window – exit operation

VI. CONCLUSION

In conclusion, real-time translation of speech to sign language holds promise for enhancing communication between the deaf and hearing communities. Future trends will focus on improving translation accuracy and efficiency through advances in machine learning and natural language processing. Integrating multimodal approaches like speech, gesture, and facial expression recognition will enhance translation comprehensiveness. User-centric design and collaboration with the deaf community will be vital for developing user friendly translation solutions. Overall, ongoing research and innovation will continue to improve communication accessibility for the deaf community.

REFERENCES

[1] Dasgupta, Tirthankar & Anupam, Basu. (2008). Prototype machine translation system from Text-To-Indian Sign Language. 313-316. 10.1145/1378773.1378818.

[2] Alisha Kulkarni, Archith Vinod Kariyal, Dhanush V, Paras Nath Singh, Atlantis Highlights in Computer Sciences, volume 4 Proceedings of the 3rd International Conference on Integrated Intelligent Computing Communication & Security (ICIIC 2021)

[3] Grieve-Smith, A.B. (2002). SignSynth: A Sign Language Synthesis Application Using Web3D and Perl. In: Wachsmuth, I., Sowa, T. (eds) Gesture and Sign Language in Human-Computer Interaction. GW 2001. Lecture Notes in Computer Science(), vol 2298. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-47873-6_14

[4] JA Bangham et al. “Virtual Signing: Capture, Animation, Storage and Transmission – an Overview of the ViSiCAST Project”, IEEE, IEE Seminar on Speech and Language Processing for Disabled and Elderly People, April 2000.

[5] Purushottam Kar et al. “INGIT: Limited Domain Formulaic Translation from Hindi Strings to Indian Sign Language”, published by COLING/ACL 2006 proceeding.

[6] Ritika Bharati et al. “Automated Speech to Sign language Conversion using Google API and NLP”, Proceedings of the International Conference on Advances in Electronics, Electrical & Computational Intelligence (ICAEEC) 2019, Elsevier

[7] Rao, R R, Nagesh, A, Prasad, K. and Babu, K E (2007) Text-Dependent Speaker Recognition System for Indian Languages. International Journal of Computer Science and Network Security, Vol. 7, No.11

[8] T.Starner, “Visual Recognition of American Sign Language Using Hidden Markov Models,” Master’s thesis, MIT, Media Laboratory, Feb. 1995.

[9] Neha Poddar, Shrushti Rao, Shruti Sawant, Vrushali Somavanshi, Prof.Sumita Chandak "Study of Sign Language Translation using Gesture Recognition" International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 2, February 2015.

[10] P. Kar, M. Reddy, A. Mukherjee, A. M. Raina. 2017. INGIT: Limited Domain Formulaic Translation from Hindi Strings to Indian Sign Language. ICON.

[11] M. Vasishta, J. Woodward and S. DeSantis. 2011. An Introduction to Indian Sign Language. All India Federation of the Deaf (Third Edition)

[12] Dr. Kalyanapu Srinivas, K Ranjithkumar, V Rakesh Datta and K. Rama Devi “An Approach for Sign Language Recognition With Deep Learning Algorithm” 6th International Joint Conference on Advances in Computational Intelligence IJCACI 2022 (IJCACI22), South Asian University, OCTOBER 15-16, 2022. SPRINGER.

[13] Kavitha, P., Prabakaran, S., Designing a feature vector for statistical texture analysis of brain, International Journal of Engineering and Advanced Technology, V-8, I-5, PP:1228-1230, 2019.