Contextual Translation System to Sign Language

Mrs Ch.Srivatsa[1] Mogusala Anjali [2] Jeedimadla Pushpan[3]
Nadagana Pranay[4] Varkala Akshitha[5]

[1]Professor, ACE Engineering College Hyderabad, India [2]Student, ACE Engineering College Hyderabad, India
[3]Student, ACE Engineering College Hyderabad, India
[4]Student, ACE Engineering College Hyderabad, India
[5]Student, ACE Engineering College Hyderabad, India
Email : [1] Srivatsajava@gmail.com [2]anjali@gmail.com
[3] 21ag1a05f9@gmail.com [4] nadaganapranay14@gmail.com
[5] varkalaakshitha03@gmail.com

ABSTRACT:

People with hearing and speech disabilities face significant challenges in communicating with others, as not everyone understands sign language. This project aims to create a system that helps bridge this communication gap by converting spoken English into Indian Sign Language (ISL). The system works by recognizing voice input, the recognized speech is converted into text, which is then simplified using natural language processing techniques.

Finally, the text is translated into ISL and displayed as a series of images or motion videos using Python libraries. This system provides an easy and accessible way for people with hearing or speech disabilities to communicate effectively, promoting inclusivity and understanding in everyday interactions.

I. INTRODUCTION:

Communication is a key part of daily life, but it can be very challenging for people with hearing and speech disability.

Since most people don't understand sign language, it becomes hard for them to express their thoughts and needs. This creates a need for tools that can bridge the gap between spoken languages and sign language. This tool will help people who use sign language to communicate more easily, especially in schools and other learning environments. Indian Sign Language (ISL) is widely used by the deaf and hard-of-hearing community in India. However, not everyone knows ISL, and learning it can be difficult for those who mainly use spoken English. To solve this problem, this project aims to create a real-time system that converts spoken English into ISL.The system works by recognizing spoken

words, converting them into text, simplifying the text to match ISL grammar, and finally showing the ISL translation as images or videos on a screen.

II. OBJECTIVES:

- Enhance Communication: Help individuals with hearing and speech disabilities communicate effectively by providing accurate and context-aware sign language translation.
- Simplify Sentences: Break down English sentences by removing unnecessary words and using simpler forms that are easy to translate into ISL.
- Create Visual Outputs: Display ISL translations as a sequence of images or videos to represent the corresponding signs.
- User-Friendly Design: Develop an easy-to-use interface tailored to the needs of the deaf and hard-of-hearing community, with a focus on continuous improvement based on user feedback.

III. PROBLEM STATEMENT:

The communication problem faced by people with hearing and speech disabilities is significant, as most people are not familiar with sign language. Learning sign language can be challenging, making it hard for these individuals to express themselves. This paper proposes a solution that converts voice input into Indian Sign Language (ISL) to help overcome this barrier. The system uses speech recognition tools (like Pyaudio, NLTK, and Google Speech Recognition API) to convert spoken words into text, and then displays the corresponding sign language gestures as images or videos on the screen. This solution aims to make communication easier for people with hearing and speech disabilities.

IV. PROPOSED SYSTEM

The proposed system aims to bridge communication gaps between individuals with hearing and speech disabilities and non-sign language users by translating English to Indian Sign Language (ISL) and vice versa. It employs advanced Natural Language Processing (NLP) techniques to simplify English sentences, creating accurate ISL gloss, which is then animated using a library of ISL signs for relevant and precise translations. To ensure the translated text fits the structure of sign language, the system eliminates unnecessary words for clarity and conciseness, uses basic forms of words to align with ISL grammar, and replaces missing words with synonyms from a dictionary to maintain translation accuracy. This system enhances communication, making it more effective for people with hearing and speech disabilities.

V. SOFTWARE REQUIREMENTS

- Platform: Visual Studio Code
- Frontend Technologies: Visualization Techniques.
- Backend Technologies: Python

VI. TECHNOLOGY DESCRIPTION

Python:

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++ or Java. It enables rapid development, making it ideal for a wide range of applications, including web development,

data analysis, artificial intelligence (AI), machine learning (ML), and natural language processing (NLP). Python's straightforward syntax and cross-platform compatibility allow for easy development and execution across different operating systems without modification. Python features a dynamic type system and automatic memory management. Python's support for both object-oriented and functional programming paradigms allows developers flexibility in structuring their code. Furthermore, the large and active Python community ensures continuous support, making it an excellent choice for both beginners and experienced developers. Python boasts a rich ecosystem of libraries and frameworks that significantly speed up the development process.

PACKAGES USED:

A few packages have been used in order to build this project. The project uses PyAudio for capturing audio input and Speech Recognition to convert voice into text. For natural language processing, NLTK simplifies sentences, while SpaCy generates Indian Sign Language (ISL) glosses. Pillow (PIL) handles image processing, and Matplotlib visualizes ISL signs as images and GIFs. Finally, Tkinter is used to create an interactive graphical user interface, ensuring seamless integration and user interaction.

VII. ALGORITHM

Divide the code into smaller functions, each handling a specific task. This makes the code modular and easier to understand.

1.Data Acquisition:

The first step is collecting the dataset for analysis. This is critical, as some datasets may lack required features or contain numerous null values, impacting the quality of the analysis.

2.Data Cleaning:

Data cleaning ensures the dataset is ready for analysis by addressing inconsistencies. Null values are identified and filled with appropriate measures such as the mean or default values, unwanted columns are removed, and outliers are handled to maintain data reliability .

3.Data Integration:

Data integration combines data from multiple sources to create a comprehensive, accurate, and up-to-date dataset. This step is crucial for business intelligence, data analysis, and applications requiring a unified dataset for effective decision-making.

4.Data Preprocessing:

This step includes exploratory data analysis and other preparatory techniques to understand patterns, relationships, and anomalies in the dataset. Data preprocessing helps transform raw data into a format suitable for model building.

5. Model building and Evaluation:

Model building is a vital phase where machine learning or statistical models are created to extract insights from the dataset. The goal is to develop an accurate and efficient model for the given dataset. Once built, the model is evaluated using performance metrics to ensure it meets the desired accuracy and reliability.

The entire process follows a step-by-step approach, using PyAudio for audio capture, Speech Recognition for voice-to-text conversion, and NLTK and SpaCy for text processing. ISL glosses are mapped to images or GIFs stored in a database, processed using Pillow (PIL), and displayed via Matplotlib or Tkinter for visualization. The system offers a user-friendly interface through Tkinter for interaction, handling errors and allowing users to reset or quit the application as needed.

VII. METHODS:

Frontend: The user interface of the application is designed using Tkinter, which provides a simple yet effective framework for building the GUI, ensuring seamless interaction for users. EasyGUI is used for simple dialog-based interactions, facilitating voice or text input and user communication.

Audio Input Handling: PyAudio is used for capturing audio input through the microphone, allowing the system to process voice commands. SpeechRecognition is then used to convert the audio into text for further processing.

Natural Language Processing (NLP): Text simplification and translation are handled by NLTK, which processes and analyzes English text by breaking it into manageable tokens and simplifying sentences. SpaCy is used for advanced NLP tasks, such as generating Indian Sign Language (ISL) glosses by removing unnecessary words and restructuring sentences

Image Processing: The Pillow (PIL) library is used to handle and display images of ISL signs, while

Matplotlib is employed for visualizing ISL signs as images or GIFs in the graphical user interface (GUI), providing clear visualization of the translated signs.

Model Training and Evaluation: In this system, machine learning models may not be explicitly used for complex tasks like prediction but are based on predefined rules for translation between text and ISL signs.

Backend: Python is used as the core programming language for system development and to integrate all components. Flask or FastAPI could be used for creating RESTful APIs, handling backend communication between the frontend and other system components, ensuring data processing, model inference, and result generation.

User Interaction and Flexibility: The system allows for flexible user interaction, handling errors and prompting users for re-input if issues arise, like unclear voice input. The user interface is designed to be intuitive, with easy navigation between input and output phases.

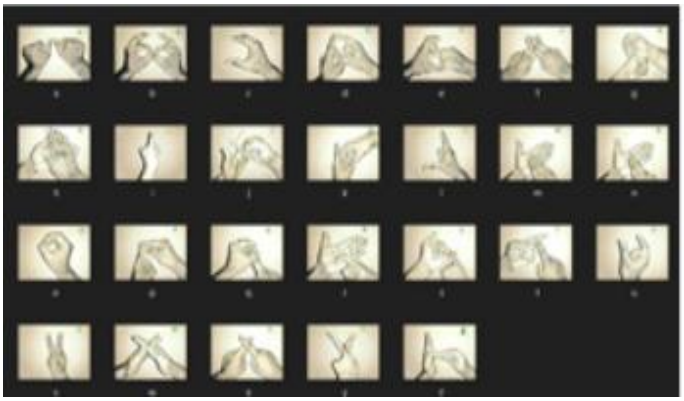
IX. OUTPUT SCREENS:

```
def func(input_type="voice"):
    arr = list(string.ascii_lowercase)

    if input_type == "voice":
        with sr.Microphone() as source:
            r.adjust_for_ambient_noise(source)
            while True:
                print("I am Listening")
                audio = r.listen(source)
                try:
                    a = r.recognize_google(audio).lower()
                    print(f'You Said: {a}')

                    if a in ["goodbye", "good bye", "bye"]:
                        print("Oops! Time to say goodbye!")
                        break

                    handle_input(a, isl_gif, arr)

                except:
                    print("Sorry, could not recognize your voice.")
```

```
elif input_type == "text":
    while True:
        a = input("Type your message (or type 'bye' to exit): ").lower()
        if a in ["goodbye", "good bye", "bye"]:
            print("Oops! Time to say goodbye!")
            break

        handle_input(a, isl_gif, arr)
```

```
C:\Users\hp\Downloads\VoiceVista-voice-to-sign-converter-main\VoiceVista-voice-to
Type your message (or type 'bye' to exit): nice to meet you
```

```
Listening...
You said: nice to meet you
Exiting after 10 seconds.
Program has stopped executing.
```



X. CONCLUSION:

This project represents a crucial step in improving communication for individuals who are deaf or hard of hearing. By addressing the specific needs of ISL users, it significantly enhances the learning experience in educational environments, allowing for more inclusive interactions between students and faculty. The use of NLP and sign language generation techniques improves the real- world applicability of the system, making it a valuable tool for bridging communication

XI. REFERENCES:

[1]. Alisha Kulkarni 1*, Archith Vinod Kariyal2, Dhanush V3, Paras Nath Singh4,"Speech to Indian Sign Language Translator " , ICIIC December 2021 DOI:10.2991/ahis.k.210913.035

[2] K. Saija, S. Sangeetha and V. Shah, "WordNet Based Sign Language Machine Translation: from English Voice to ISL Gloss," 2019 IEEE 16th India Council International Conference (INDICON), 2019, pp. 1-4, DOI: 10.1109/INDICON47234.2019.9029074.

[3] Aditi Bailur, Yesha Limbachia, Moksha Shah, Harshil Shah, Prof. Atul Kachare, " American Sign Language Recognition and its Conversion from Text to Speech".

[4] Mahmoud Elmezain, Ayoub Al-Hamadi, Jorg Appenrodt, Bernd Michaelis"A Hidden Markov Model-Based Continuous Gesture Recognition System for Hand Motion Trajectory"

[5] Pankaj Sonawane,Karan Shah,Parth Patel,Shikhar Shah,Jay Shah,"Speech to Indian Sign language", IEEE 2021 ,DOI: 10.1109/ICCCIS51004.2021.9397097