INDIAN SIGN LANGUAGE TO TEXT/SPEECH TRANSLATION

Dissertation submitted in partial fulfilment of the requirements for the award of the Degree of

BACHELOR OF COMPUTER APPLICATIONS

Submitted by

PATURU SANDEEP	**22102C010247**
PINJARI DASTAGIRI	**22102C010255**
SYAMTAN SAI YADAV BONTHA	**22102C010319**
THARUGU SOMA SHEKAR REDDY	**22102C010329**

Under the Guidance of

Mrs. P. Yasminbhanu,
Assistant Professor,
Dept. of CA.

DEPARTMENT OF COMPUTER APPLICATIONS

SCHOOL OF COMPUTING

MOHANBABUUNIVERSITY

Sree Sainath Nagar, Tirupati–517102

(2025)

MOHAN BABU UNIVERSITY

DEPARTMENT OF COMPUTER APPLICATIONS

Vision

To become a center of excellence in the field of computer science and applications.

Mission

- ➢ Imparting knowledge and skills through contemporary curriculum to the diverse group of students.
- ➢ Creating a talent pool of faculty in diverse domains of computer applications through continuous training.
- ➢ Domain and transferable skill development for the holistic personality of students to inculcate values and ethics for effective professional practice and as an entrepreneur.

BACHELOR OF COMPUTER APPLICATIONS (BCA)

PROGRAM EDUCATIONAL OBJECTIVES

After few years of completion of the Program, the graduates of BCA will be able to:

PEO1. Pursue higher education in the core and allied areas of computer science by applying computing knowledge and domain-specific knowledge, demonstrating their innovative skills, and considering social and environmental concerns.

PEO2. Become professionals in industry and academia with ability to investigate, and solve complex computing problems using modern tools in evolving technologies in the core and allied areas of computer science.

PEO3. Become successful entrepreneurs to excel in diverse application skills in the core or allied area of computer science of societal importance.

PEO4. Exhibit professionalism, and uplifting health, safety, legal, environmental, ethical, and cultural diversity issues for serving the society and communicating with local, and national peers, bound within regulations and leading to lifelong learning.

PROGRAM OUTCOMES

On successful completion of the Program, the graduates of B.C.A Program will be able to:

PO1. Computational Knowledge: Apply knowledge of computing fundamentals, computing specialization, mathematics, and domain knowledge appropriate for the computing specialization to the abstraction and conceptualization of computing models from defined problems and requirements.

PO2. Problem Analysis: Identify, formulate, research literature, and solve complex computing problems reaching substantiated conclusions using fundamental principles of mathematics, computing sciences, and relevant domain disciplines.

PO3. Design / Development of Solutions: Design and evaluate solutions for complex computing problems, and design and evaluate systems, components, or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.

PO4. Conduct Investigations of Complex Computing Problems: Use research-based knowledge and research methods including design of experiments, analysis, and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5. Modern Tool Usage: Create, select, adapt and apply appropriate techniques, resources, and modern computing tools to complex computing activities, with an understanding of the limitations.

PO6. Professional Ethics: Understand and commit to professional ethics and cyber regulations, responsibilities, and norms of professional computing practices.

PO7. Life-long Learning: Recognize the need, and have the ability, to engage in independent learning for continual development as a computing professional.

PO8. Project management and finance: Demonstrate knowledge and understanding of the computing and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO9. Communication Efficacy: Communicate effectively with the computing community, and with society at large, about complex computing activities by being able to comprehend and write effective reports, design documentation, make effective presentations, and give and understand clear instructions.

PO10. Societal and Environmental Concern: Understand and assess societal, environmental, health, safety, legal, and cultural issues within local and global contexts, and the consequential responsibilities relevant to professional computing practices.

PO11. Individual and Team Work: Function effectively as an individual and as a member or leader in diverse teams and in multidisciplinary environments.

PO12. Innovation and Entrepreneurship: Identify a timely opportunity and using innovation to pursue that opportunity to create value and wealth for the betterment of the individual and society at large.

PROGRAM SPECIFIC OUTCOMES

On successful completion of the Program, the graduates of B.C.A Program will be able to:

PSO1: Design, implement and test applications for complex computing problems for desired specifications using programming skills.

PSO2: Analyze and adapt managerial and domain skills of Information Management to model an application's data requirements to extract information for interpreting the datasets for Decision Making.

PSO3: Apply suitable techniques and algorithms to integrate Operating System Services, Network devices, Security mechanisms and Infrastructure to meet the requirements for the deployment of an application and to communicate on networks.

MOHAN BABU UNIVERSITY

Sree Sainath Nagar, Tirupati 517 102

SCHOOL OF COMPUTING

DEPARTMENT OF COMPUTER APPLICATIONS

Certificate

This is to certify that the project report entitled **"Indian Sign Language to Text/Speech Translation"** is the bona fide work carried out and submitted by

PATURU SANDEEP (22102C010247),

PINJARI DASTAGIRI (22102C010255),

SYAMTAN SAI YADAV BONTHA (22102C010319),

THARUGU SOMA SHEKAR REDDY (22102C010329),

in the Department of **Computer Applications, School of Computing** of **Mohan Babu University**, Tirupati, in partial fulfilment of the requirements for the award of the degree of **Bachelor of Computer Applications** during 2024 - 25.

Supervisor	Head of the Department
Mrs. P. Yasminbhanu, **Assistant Professor,** **Dept. of CA,** **MBU, Tirupati.**	**Dr. M. Sowmya Vani,** **Associate Professor,** **Dept. of CA,** **MBU, Tirupati.**

Date:

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

This acknowledgement transcends the reality of formality when we would like to express our deep gratitude and respect to all those people behind the screen who guided, inspired and helped us with the completion of our project work.

We express our deep sense of gratitude to our beloved Chancellor, **Dr. M. Mohan Babu,** Padma Shri awardee, for his encouragement throughout the program.

We express our deep sense of gratitude to our beloved Vice-Chancellor, **Dr. K. Karunakaran,** for his encouragement throughout the program.

We are extremely thankful to **Dr. M. Sowmya Vani**, HOD, Department of **Computer Applications,** for all the provisions made and for her constant encouragement throughout our work.

We wish to express our deep sense of gratitude to our Project Coordinator, **Mr. P M D Ali Khan**, Assistant Professor, Computer Applications and Project Supervisor **Mrs. P. Yasminbhanu**, Assistant Professor, Computer Applications for extending her valuable co-operation, moral support, kind attention, guidance, suggestions and encouragement to complete our Project Work successfully.

We thank all our beloved **Faculty**, Department of CA, for giving their valuable suggestions and maximum cooperation.

We owe a deep sense of gratitude to our beloved **Parents** for extending their moral support in this Endeavour. We would like to thank all my **friends** who extended their help, encouragement and moral support either directly or indirectly in completing our project work.

PATURU SANDEEP	**22102C010247**
PINJARI DASTAGIRI	**22102C010255**
SYAMTAN SAI YADAV BONTHA	**22102C010319**
THARUGU SOMA SHEKAR REDDY	**22102C010329**

DECLARATION

We, PATURU SANDEEP 22102C010247,
PINJARI DASTAGIRI 22102C010255,
SYAMTAN SAI YADAV BONTHA 22102C010319,
THARUGU SOMA SHEKAR REDDY 22102C010329,

hereby declare that, the project entitled "**Indian Sign Language to Text/Speech Translation**" developed by us at **MOHAN BABU UNIVERSITY**, Tirupati during the Academic year 2024-2025 and submitted to The Department of Computer Applications, School of Computing, MOHAN BABU UNIVERSITY for partial fulfilment for the award of Bachelor of Computer Applications (BCA).

We also declare that, the project is resulted by my own effort and that it has not been copied from anyone and not been submitted by anybody in any of the University or Institution or Research Centre.

Place: Tirupati
Date:

PATURU SANDEEP (22102C010247)
PINJARI DASTAGIRI (22102C010255)
SYAMTAN SAI YADAV BONTHA (22102C010319)
THARUGU SOMA SHEKAR REDDY (22102C010329)

ABSTRACT

The Indian Sign Language to Text or Speech Translation project represents a groundbreaking integration of technology in the field of communication and accessibility. Through this system, ISL communication connects to verbal or written expressions for continuous dialogue. The system functions as an assistive tool for deaf and hard-of-hearing people because it generates real-time text or speech output from sign language inputs. The system facilitates diverse interaction between users of sign language and individuals who do not speak that language. Renewed Indian Sign Language (ISL) translation requires advanced technological functions such as natural language processing (NLP) and computer vision along with machine learning operations for precise translation. Deep learning models, including CNNs and transformers, enhance gesture recognition and speech synthesis. By incorporating real-time processing, the project ensures a smooth and natural user experience. It also contributes to language accessibility, making education and daily interactions more inclusive. The use of AI-driven models enables continuous learning and improvement for higher accuracy. This creative approach shows a notable development that helps better communication and supports a more linked society.

Keywords—Accessibility Technology, Sign Language Interpretation, Audio Processing, Speech-to-Text Conversion, Natural Language Processing (NLP), Computer Vision Deep Learning, Artificial Intelligence.

Table of contents

List of Figures

List of Tables

1.INTRODUCTION

1.1.Introduction

The primary motivation behind "INDIAN SIGN LANGUAGE TO TEXT/SPEECH TRANSLATION" is to address the communication barriers faced by the deaf and hard-of-hearing community. Despite the prevalence of sign language, there exists a significant gap in everyday interactions between those who rely on sign language and those who use spoken language. This project is driven by the desire to create a more inclusive society where this communication gap is minimized. The motivation also stems from the technological challenge of integrating different domains like NLP, AI, and computer vision to create a cohesive system capable of understanding and translating complex human languages and gestures. This technological innovation aims to foster better understanding, empathy, and interaction among people with diverse communication needs.

The project addresses the challenge of facilitating communication between the deaf and hard-of-hearing community and those who communicate through spoken language. There is a significant gap in accessible communication tools that effectively translate between audio and sign language in real-time. This gap leads to a lack of inclusivity and barriers in education, employment, and social interactions for many individuals. The project seeks to solve this by creating a reliable and efficient system that can translate spoken words into sign language and interpret sign language back into text, thereby enabling seamless communication across these different modalities.

The primary objective of this project is to develop a functional and accurate system for translating audio to sign language and sign language to text. This involves achieving a high level of accuracy in voice recognition, sign language interpretation, and text generation. The system aims to cater to a wide range of languages and dialects in both speech and sign language, making it universally accessible. Another objective is to ensure the system is user-friendly, with a simple interface that can be easily used by people of all ages and technological proficiencies. Moreover, the project seeks to contribute to academic and practical knowledge in the fields of language processing, accessibility technology, and human-computer interaction.

The scope of the project encompasses developing a comprehensive system capable of converting spoken language into sign language and vice versa. This involves the creation of a robust framework for audio processing, sign language recognition, and text generation. The project aims to cover a wide range of vocabularies and expressions in both audio and sign language, ensuring a versatile and effective communication tool. Additionally, the project has the potential to expand into various applications, including educational tools, real-time communication platforms, and assistive technologies for differently-abled individuals. Its scope extends beyond mere translation, aiming to provide a seamless and intuitive user experience for both speech and sign language users.

This is an innovative project that aims to revolutionize the way we approach communication between different language modalities. By leveraging state-of-the-art technology in voice recognition, sign language processing, and text generation, this project seeks to create a bridge between the deaf and hearing communities. The introduction of such a system has profound implications for social inclusion, education, and professional opportunities for deaf and hard-of-hearing individuals. This project is not just about language translation; it's about creating new avenues for understanding and interaction in a diverse society. It represents a significant step forward in making our world more accessible and inclusive for all.

1.2.Aim of the project

The aim of this project, "Indian Sign Language to Text/Speech Translation," is to bridge the longstanding communication gap between the deaf and hard-of-hearing communities and the hearing population. In India, where Indian Sign Language (ISL) is widely used among the hearing-impaired, there exists a significant lack of accessible tools that enable seamless interaction between signers and non-signers. This project seeks to break down those barriers by developing an innovative system capable of translating sign language gestures into text and spoken words, and vice versa—spoken language into visual sign representations. The overarching goal is to foster inclusive communication by leveraging the power of modern technology, including Natural Language Processing (NLP), computer vision, machine learning, and real-time video processing.

One of the primary objectives of the project is to empower individuals with hearing disabilities by providing them with tools that allow for natural interaction in mainstream society. Whether in classrooms, workplaces, healthcare settings, or public services, deaf individuals often face challenges due to a lack of accessible communication. Through this system, sign language users will be able to convey their messages through recognised hand gestures, which will be converted into written or spoken language. On the other hand, individuals who rely on speech can communicate back by speaking or typing, and their input will be transformed into corresponding sign language animations. This bidirectional communication model is at the core of the project's mission—making communication fluid and natural for all, regardless of hearing ability.

The project also aims to push the boundaries of current technological limitations by using real-time gesture recognition frameworks like MediaPipe and deep learning models such as CNNs, RNNs, GRUs, and ViT (Vision Transformers). These models are crucial for accurately detecting and interpreting complex hand movements, body language, and facial expressions that form the essence of ISL. Furthermore, the integration of NLP with tools like NLTK and T5 models ensures that the system doesn't just recognise words in isolation but understands grammatical structure, sentence context, and meaning, allowing it to handle full sentences, questions, tenses, and idiomatic expressions effectively.

In addition to real-time functionality, the project strives to be scalable and adaptable. It is designed in a modular fashion so that new gestures, expressions, and dialect variations can be continuously added. This is particularly important for a country like India, where regional differences in sign usage can vary widely. The system is also intended to be accessible on web platforms, making it widely usable across devices and locations. Its user-friendly interface supports people of all technological backgrounds, ensuring that the system is not only powerful but also practical and inclusive.

Beyond the technical implementation, the aim of the project is rooted in social transformation. By eliminating communication barriers, this project contributes to greater equality, accessibility, and awareness in society. It promotes empathy and understanding between communities, encouraging inclusive education, better job opportunities, and increased independence for people with hearing impairments. Through innovation and thoughtful design, the system aspires to be a meaningful step toward creating a world where everyone's voice—whether signed or spoken—can be heard and understood.

1.3.Project Domain

Accessibility and Inclusivity in Communication

The primary domain of this project lies in enhancing accessibility and inclusivity for the deaf and hard-of-hearing community. Traditional communication methods fail to bridge the gap between spoken and sign languages, leaving many individuals marginalized in social, educational, and professional contexts. This project aims to create a real-time, user-friendly system that provides equal communication opportunities, ensuring that individuals using sign language can express themselves effectively and understand others without barriers.

Sign Language Recognition and Processing

At the heart of the project is the domain of sign language recognition, which involves interpreting hand gestures, movements, and expressions to derive meaningful communication. Indian Sign Language (ISL), which includes a combination of static and dynamic gestures, is detected using computer vision technologies like MediaPipe. By processing gestures through real-time video input, the system translates them into text, enabling signers to communicate naturally with non-signers.

Natural Language Processing (NLP)

Another critical domain is Natural Language Processing (NLP), which handles the conversion of spoken or written language into structured data that can be translated into sign language. Tasks such as tokenization, part-of-speech tagging, lemmatization, tense detection, and sentence reconstruction are performed using libraries like NLTK. NLP enables the system to understand grammar, sentence context, and user intent, ensuring accurate sign language translations.

Speech Recognition and Audio Processing

This domain deals with capturing spoken input through microphones and converting it into text using speech-to-text APIs. The processed text is then passed on to the sign language animation module. Audio processing is especially important for non-signers who prefer speaking over typing, enabling the system to act as a real-time translator between speech and sign language, thus making interactions smoother and more natural.

Computer Vision and Gesture Tracking

The use of computer vision for tracking hand movements, finger positions, and body posture is essential for understanding sign language gestures. MediaPipe, an advanced framework, is used for real-time gesture recognition and tracking. This domain also includes handling environmental variations such as lighting and background, ensuring accurate detection of gestures regardless of external conditions.

Web Development and User Interface Design

The project includes a significant focus on front-end and back-end web development. Using Django for the server side and HTML, CSS, JavaScript, and Bootstrap for the interface, the system is designed to be accessible and easy to navigate. The user interface allows individuals to input audio, text, or sign language and receive real-time translations. Accessibility features are embedded to ensure usability across different devices and user demographics.

Real-Time System Integration

Integrating various domains—NLP, speech recognition, computer vision, and database management—into a single seamless system requires strong backend development and synchronization. Real-time processing is crucial so users don’t experience lag or inaccuracies in translation. This domain ensures that the entire process from input to output—whether text, speech, or sign—is fast, reliable, and synchronized.

Machine Learning and Deep Learning

Machine learning plays a vital role in the gesture recognition and translation process. Models such as CNN, GRU, and ViT are utilized to detect gestures, classify them accurately, and model their temporal sequence. These models are trained on custom ISL datasets to ensure high accuracy, robustness to variation, and the ability to learn new gestures over time, making the system adaptive and intelligent.

Assistive Technology and Human-Computer Interaction (HCI)

This project falls under the domain of assistive technology, where systems are built specifically to support individuals with disabilities. Human-computer interaction principles are applied to ensure that the system is intuitive, engaging, and beneficial for users with diverse needs. Features such as gesture animation, voice feedback, and session history are designed to make the interaction comfortable and efficient.

Database Management and System Scalability

Lastly, the system requires an efficient database to manage gesture animations, user accounts, session histories, and model outputs. Using MySQL with Django ORM, the system ensures secure and structured data storage. Scalability is a key focus, enabling the addition of new signs, phrases, and dialects without major architectural changes. This makes the system future-ready and expandable across languages and regions.

How These Domains Interact in the Project

In the Indian Sign Language to Text/Speech Translation project, the process begins when a user provides input through either audio, text, or live video. If the input is audio, the system uses a speech recognition module to capture spoken language through a microphone. This audio is processed using a speech-to-text API, which converts the speech into written text for further processing. In the case of direct text input, the system bypasses this stage and sends the text straight to the Natural Language Processing (NLP) module.

Within the NLP domain, the input text undergoes several stages of processing, including tokenisation, part-of-speech tagging, lemmatisation, and stop word removal. These steps help the system understand the grammatical structure and semantics of the sentence, allowing it to determine tense and contextual meaning. This understanding is vital for generating accurate sign language representations, as sign language grammar and syntax often differ from spoken language.

Once the text is fully processed, it is matched to corresponding sign language gestures. These gestures can be either retrieved from a pre-existing database of animations or generated dynamically. The animations are then presented on the user interface, allowing hearing individuals to communicate their message to the deaf community using sign language visuals. Meanwhile, if the user is deaf and communicates through sign language, the system captures their gestures through a webcam.

The captured video input is processed using computer vision, particularly leveraging the MediaPipe framework to track hand movements and finger positioning in real time. These visual data points are passed to deep learning models, such as CNNs, GRUs, or Vision Transformers (ViT), which have been trained on Indian Sign Language datasets. These models classify the gestures and interpret their meaning, which is then converted into text. NLP may further process this text to ensure sentence clarity and meaning.

If the output needs to be spoken, the system uses a text-to-speech (TTS) module to convert the interpreted text into natural-sounding speech. This completes the cycle of translating sign language into both text and audio, enabling seamless two-way communication. The web interface, built using HTML, CSS, and JavaScript, displays all translations, while Django manages backend functions such as routing, server-side logic, and user authentication.

All translation records, gesture mappings, and user interactions are stored in a MySQL database, accessed through Django's ORM. This allows for efficient data retrieval, user session management, and system scalability. The entire platform is designed for real-time interaction, meaning each component—from input capture to output rendering—must work in harmony with minimal latency. Cross-platform compatibility ensures users can access the system on various devices, including desktops and smartphones.

Finally, the assistive technology domain ensures that the interface and interactions are optimized for users with hearing impairments. By combining NLP, computer vision, machine learning, and responsive design, the system offers a powerful, inclusive tool for real-time language translation. Each domain contributes a critical layer of functionality, and their smooth integration ensures that users experience fast, accurate, and meaningful communication across different language modalities.

Key components

Natural Language Processing (NLP)

NLP is essential for analyzing and understanding human language in both its written and spoken forms. It helps break down sentences into components like words and tenses using tools like tokenization and POS tagging. This allows the system to generate grammatically and semantically accurate translations. It forms the linguistic backbone of the project, enabling intelligent and context-aware sign language conversion.

Speech Recognition

This component allows users to speak naturally into the system, where their voice is converted into text. APIs like Google Speech-to-Text are employed to capture spoken input efficiently. The resulting text can then be processed and translated into sign language animations. It simplifies communication for non-sign language speakers and enhances user-friendliness.

Text-to-Speech (TTS)

TTS enables the system to vocalize text that has been translated from sign language. This is particularly helpful when deaf or mute individuals communicate with people who rely on spoken language. It ensures that the system supports bi-directional communication effectively. The speech output is generated in a natural and intelligible manner using modern TTS engines.

Sign Language Gesture Recognition

The system uses MediaPipe to detect hand landmarks and finger positions from webcam video input. These gestures are interpreted using machine learning models to identify the intended signs. It works in real-time, enabling fluid communication from signers to non-signers. This module is crucial for converting physical sign language into digital data.

Deep Learning Models (CNN, GRU, ViT)

Deep learning models such as Convolutional Neural Networks (CNNs) and Gated Recurrent Units (GRUs) play a vital role in recognizing sign gestures. CNNs capture spatial features while GRUs model gesture sequences over time. Vision Transformers (ViTs) help in capturing global visual patterns for higher accuracy. These models work together to ensure robust gesture classification and translation.

Real-Time Video Processing

This component enables the system to capture and process live video input from the user's camera. It ensures that sign language gestures are recognized and responded to instantly. Low latency is critical for maintaining natural communication flow. It also helps in tracking and analyzing dynamic gestures without delay.

Sign Language Animation Module

The animation module is responsible for converting text into sign language animations. These animations use either pre-recorded gesture clips or avatar-based visual outputs. It ensures that users who rely on visual cues can understand spoken or written communication. The module helps bridge the gap between text and visual languages

Text Preprocessing

Before NLP operations begin, the system preprocesses input text by converting it to lowercase, removing stop words, and normalizing content. Lemmatization is used to reduce words to their base forms. This step ensures that the text fed into the NLP pipeline is clean and consistent. It greatly enhances translation quality and model performance.

Contextual Understanding

Understanding the context of words and sentences is critical for generating accurate sign translations. The system uses NLP techniques to analyze sentence structure and semantics. Context helps in avoiding misinterpretation of ambiguous words or idioms. It ensures that the intended meaning is preserved during translation.

User Interface (UI)

The UI provides a clean, intuitive interface for users to interact with the system. It includes sections for entering text, recording audio, and displaying sign language animations. Designed using responsive web technologies, it adapts well across devices. Its simplicity ensures that even non-technical users can operate the system easily.

Web Development (Frontend)

Frontend technologies like HTML, CSS, JavaScript, and Bootstrap are used to create a responsive layout. The frontend handles user interactions such as form inputs and button clicks.

It also displays translation outputs clearly and attractively. Accessibility and mobile responsiveness are built-in to support diverse user needs.

Backend Development (Django)

Django is used to manage server-side logic, routing, and user authentication. It coordinates communication between the frontend, machine learning models, and databases. Django’s built-in ORM makes database operations efficient and secure. It ensures the backend is scalable, maintainable, and fast.

Gesture Animation Database

This database stores sign language animations that correspond to words or phrases. It allows for quick retrieval during translation from text to sign. The database is regularly updated to include new signs and idioms. This ensures the system remains relevant and expansive over time.

Machine Learning Training Module

The training module allows deep learning models to learn from gesture datasets. It helps in refining model accuracy by retraining with new data. The more it learns, the better it performs in recognizing a variety of gestures. It supports continuous improvement and adaptation of the system.

Named Entity Recognition (NER)

NER is used to identify special terms like names, locations, or brands in the input text. These entities often require custom gestures or emphasis. Recognizing them ensures that translations are more natural and meaningful. This component adds depth and intelligence to the NLP layer.

User Authentication System

Security is maintained through user registration, login, and session management. Django’s built-in authentication handles password protection and access control. It ensures that each user’s data and history remain private. It also enables personalized user experiences by storing preferences.

Session Management and History

The system keeps a record of user interactions and translation history. This allows users to review past inputs and outputs if needed. It improves usability and also helps in evaluating system performance. Session logs can also aid in retraining models or debugging.

Database Management (MySQL)

A structured MySQL database stores user data, translations, and gesture mappings. Django ORM facilitates easy querying and modification of records. The database supports multiple users simultaneously and maintains data integrity. It plays a vital role in backend functionality.

Cross-Platform Compatibility

The system is designed to run smoothly across various operating systems and devices. Whether accessed from a smartphone, tablet, or desktop, the platform remains fully functional. This maximizes accessibility for users from different backgrounds. It ensures wide usability across real-world scenarios.

Assistive Technology Integration

Special features like screen reader support and voice command compatibility enhance accessibility. These features are especially useful for users with additional physical challenges. The system adheres to inclusive design principles to serve diverse needs. It helps ensure the technology can truly empower all users.

1.4.Scope of the project

The scope of the project "Indian Sign Language to Text/Speech Translation" extends across multiple technological and social domains, aiming to create a comprehensive communication bridge between the hearing-impaired community and the hearing population. It involves the development of a real-time, bidirectional translation system that can convert spoken or typed text into sign language and, conversely, interpret sign language gestures into written or spoken output. This ensures that both signers and non-signers can communicate seamlessly, without the need for a human interpreter.

The project leverages cutting-edge technologies such as Natural Language Processing (NLP), computer vision, deep learning, and gesture recognition frameworks to achieve accurate translations. NLP is employed for analyzing the structure and meaning of input text, enabling grammatically and contextually appropriate sign language conversion. Computer vision, on the other hand, is used to recognize hand gestures through a webcam or camera in real-time, making the system interactive and intuitive.

A major aspect of the scope is its adaptability to Indian Sign Language (ISL), which includes a wide range of regional variations and unique grammatical structures. The system is designed to be modular and scalable, allowing new signs, gestures, and dialects to be added over time without changing the overall architecture. This ensures that the system remains relevant as ISL evolves and more users engage with it.

Furthermore, the scope includes integration with speech-to-text and text-to-speech technologies. This makes the platform inclusive not only for hearing-impaired users but also for individuals who prefer auditory communication. The inclusion of sign language animations allows users to visualize text or speech in an engaging and understandable format.

The project also aims to serve multiple sectors, such as education, healthcare, customer service, public administration, and social networking, where communication gaps often exist due to language barriers. By making this tool web-based and mobile-friendly, the project expands its accessibility to users across various platforms and locations.

Additionally, the project focuses on user-friendliness, ensuring that individuals with minimal technical skills can interact with the system effectively. With secure login features, real-time feedback, and session histories, the system offers a personalized experience that users can rely on daily.

Another important dimension is its role in assistive technology. The system not only facilitates communication but also empowers the hearing-impaired community by promoting independence and confidence. It supports the vision of a more inclusive and equitable digital world.

In terms of research and development, the project also opens avenues for future innovation in AI-driven language translation, gesture modelling, and cultural adaptability in communication tools. It sets the groundwork for more complex systems capable of understanding facial expressions, body posture, and even emotional cues in sign language.

Ultimately, the scope of this project is both technical and humanitarian. It addresses a pressing social challenge while demonstrating the power of technology in creating meaningful change. By reducing communication barriers, it contributes to social inclusion, equal opportunities, and digital accessibility for all.

Bidirectional Language Translation

The project is designed to facilitate two-way communication by translating spoken or typed language into sign language and vice versa. This allows both hearing and non-hearing individuals to interact naturally without external interpreters. It’s a vital step in closing communication gaps in daily conversations and formal settings.

Real-Time Gesture Recognition

Using computer vision tools like MediaPipe, the system processes sign language gestures in real-time. This allows for immediate feedback and instant translations, enabling fluid communication. It enhances usability in scenarios like live classes, meetings, or customer service.

Text and Audio Integration

The system supports both text and audio inputs, allowing users to type or speak their messages. This dual input mechanism makes the application versatile and user-friendly for people with varied communication preferences. It also supports converting sign language into both text and spoken audio.

Application of NLP for Contextual Understanding

Natural Language Processing (NLP) is used to understand sentence structure, tense, and context before translation. This helps in maintaining the meaning and accuracy when converting between text and sign language. It also handles complex sentences and regional variations effectively.

Support for Indian Sign Language (ISL)

The project is specifically tailored for Indian Sign Language, acknowledging its grammatical uniqueness and regional dialects. By focusing on ISL, the system becomes highly relevant and practical for the Indian context, which is often underserved in global accessibility tech.

Scalability for Additional Gestures and Dialects

The system architecture allows for future expansion by adding new gestures, signs, and regional dialects. This ensures that the platform can grow over time, adapting to language evolution and user feedback. It keeps the system dynamic and sustainable.

Educational and Learning Applications

This project has a broad application in education, especially for inclusive classrooms where deaf students can follow along through live translations. It also helps non-signers learn ISL through visual feedback, making it a potential self-learning platform.

Accessibility in Public Services

The system can be deployed in public service areas like hospitals, police stations, and government offices. This allows deaf individuals to interact with service providers without needing an interpreter. It enhances independence and equal access to essential services.

Integration with Assistive Technologies

The system is designed to be compatible with screen readers, voice input tools, and other assistive technologies. This ensures accessibility for users with varying disabilities, not just hearing impairment. It aligns with inclusive design principles.

Mobile and Web Compatibility

The application is built to function across multiple platforms, including web browsers and mobile devices. This cross-platform capability allows users to access the system anywhere, anytime, which is essential for day-to-day use in varied settings.

Secure User Authentication

The scope includes user authentication features such as registration and login to keep session data secure. It helps users track their translation history and customize settings, ensuring a personalized and protected user experience.

User Feedback and Adaptability

A feedback system allows users to report inaccuracies or suggest new signs. This makes the platform community-driven and adaptable. Continuous user interaction improves the system's effectiveness over time through machine learning updates.

Research and Innovation Platform

Beyond practical use, the project provides a foundation for academic research in AI, NLP, and computer vision. It encourages further study and innovation in sign language recognition and translation technologies.

Social Empowerment and Inclusion

One of the broadest scopes of this project is its role in social impact. By enabling independent communication for the deaf, it promotes dignity, self-reliance, and equal opportunity in society. It supports the vision of a truly inclusive digital environment.

Future-Ready Technology Infrastructure

The project uses modular coding and scalable frameworks to prepare for future upgrades. This includes deeper AI integration, 3D avatar animations, facial expression tracking, and support for multiple languages. It’s designed to evolve with technological advancements.

2.LITERATURE REVIEW

"Integrating NLP and Computer Vision for Effective Sign Language Translation": This paper explores the integration of NLP for contextual understanding and computer vision for gesture recognition in sign language translation tools. It highlights the challenges in real-time translation systems and offers insights into future research directions for the deaf and hard-of-hearing community.

Drawback: The integration of NLP and computer vision faces challenges in handling complex gesture variations and contextual ambiguity in real-time translations. Additionally, real-time processing demands significant computational resources.

"Advancements in Real-Time Speech-to-Sign Language Systems": This survey examines recent advancements in converting real-time speech to sign language using various machine learning models, analyzing case studies and experimental results. It sheds light on the progress in assistive communication technologies and their implications for accessibility.

Drawback: The accuracy of real-time speech-to-sign translation is often compromised in noisy environments, and handling diverse dialects or regional variations of sign language remains a challenge.

"Challenges in Designing User Interfaces for Speech-to-Sign Translation Tools": This literature review discusses the challenges in designing user interfaces for speech-to-sign translation applications, emphasizing inclusive design principles. It analyzes interface models and user feedback to identify key factors for creating user-friendly systems for individuals with hearing impairments.

Drawback: The paper lacks practical implementation examples of user interfaces for diverse user groups, and it may not fully address the scalability of designs for real-world deployment.

"A Comparative Analysis of Gesture Recognition Technologies in Sign Language Translation": This paper presents a detailed analysis of gesture recognition technologies, evaluating techniques from traditional image processing to advanced deep learning approaches. It critically assesses the accuracy, speed, and reliability of these technologies in real-world scenarios.

Drawback: The paper may not provide in-depth comparisons for all possible gesture recognition models, and it lacks focus on newer, cutting-edge techniques like deep learning-based approaches.

"Pansare et al. (2020) developed a hybrid CNN-RNN model for Indian Sign Language recognition": Pansare et al. proposed a hybrid CNN-RNN model for recognizing Indian Sign Language gestures, combining spatial feature extraction with temporal sequence modeling. The system achieved high accuracy with custom datasets.
Drawback: The model's reliance on custom datasets limits its generalizability to real-world applications, and it may struggle with recognizing gestures from diverse populations.

"Kumar and Rani (2019) used contour-based hand segmentation and SVM for static ISL gesture recognition": Kumar and Rani's system employed contour-based hand segmentation and SVM for static Indian Sign Language gesture recognition. It worked well for isolated gestures but struggled with dynamic signs. Noise filtering and preprocessing improved results.
Drawback: The system is limited to isolated gestures and may perform poorly with dynamic or continuous sign language, reducing its real-world applicability.

"Yadav et al. (2021) created a dataset of dynamic ISL gestures and used LSTM models for recognition": Yadav et al. developed a dataset of dynamic Indian Sign Language gestures and used LSTM models for gesture recognition. Their system achieved high accuracy for sequential gestures in real-time webcam use.
Drawback: The LSTM model's reliance on hand landmark detection may face difficulties with complex or overlapping gestures, affecting accuracy in some cases.

"Sharma and Gupta (2022) combined CNNs and GRUs to build an end-to-end ISL recognition system": Sharma and Gupta combined CNNs for frame-level feature extraction and GRUs for modeling gesture sequences in their end-to-end ISL recognition system. The system showed high accuracy, even under various lighting and user conditions.
Drawback: The model may be sensitive to environmental changes like lighting, and it could struggle with noisy or obstructed hand gestures in real-world scenarios.

"Nayak and Patel (2020) focused on full translation from ISL to text and speech": Nayak and Patel developed a system that translated Indian Sign Language gestures to text and speech using T5 NLP and Google's TTS API. Their system showed strong performance in translating gestures to natural English text and speech.
Drawback: The system may face difficulties with real-time performance and handling a wide variety of sign language gestures, limiting its scalability for large vocabularies.

"Deshmukh et al. (2021) used Vision Transformers (ViT) and GRUs for advanced ISL recognition": Deshmukh et al. proposed a system using Vision Transformers (ViT) and GRUs to capture both global patterns and temporal flow in Indian Sign Language gestures. Their model achieved high accuracy, even in noisy conditions.
Drawback: The model's high accuracy may not be easily reproducible on varied datasets, and it could be computationally intensive for real-time application.

3. PROBLEM DEFINITION

Real-Time Sign Language Translation for Communication Accessibility

Sign language remains the primary mode of communication for millions of deaf and hard-of-hearing individuals. However, there is a gap in effective communication between deaf and non-deaf communities, as traditional sign language requires face-to-face interaction. Real-time sign language translation systems are essential to bridge this gap and enable seamless communication in everyday environments. Despite advancements, the development of systems capable of real-time, accurate sign language interpretation for diverse users remains an ongoing challenge.

Improvement of Gesture Recognition Models for Indian Sign Language (ISL)

Indian Sign Language (ISL) has its own unique syntax and vocabulary that differ significantly from other sign languages. Developing gesture recognition models that can accurately capture these features in both static and dynamic gestures is a significant challenge. Current recognition systems often struggle with complex gestures or require large datasets to train models effectively. Improving the recognition accuracy and scalability of ISL gesture models is critical for building robust sign language translation systems.

Integration of NLP and Computer Vision for Contextual Understanding in Sign Language

Sign language translation systems often face difficulties in understanding contextual nuances, as sign language is not just about hand gestures but also includes facial expressions, body postures, and even the surrounding environment. Integrating NLP (Natural Language Processing) with computer vision models can enhance the contextual understanding of signs and improve translation accuracy. However, designing a system that can accurately process these non-manual signals while performing gesture recognition is still a major research challenge.

Handling Diverse Regional Variations in Sign Language Translation

Like spoken languages, sign languages exhibit regional variations, which means that the same sign can differ in shape, movement, and meaning depending on the region. For example, the Indian Sign Language (ISL) may have different regional dialects.

Developing a translation system that can handle these variations without compromising on translation accuracy remains a complex task. Dataset creation, recognition models, and contextual understanding need to account for these variations to improve system adaptability.

Real-Time Gesture Recognition in Noisy and Dynamic Environments

Gesture recognition systems often struggle in environments with significant background noise or dynamic elements. In real-world settings, lighting conditions, cluttered backgrounds, and movement of the user can distort gesture recognition accuracy. Real-time systems need to be resilient to these changes, providing accurate gesture translations even in less-than-ideal conditions. Building a system capable of recognizing dynamic gestures across diverse environments is a major problem in current sign language translation systems.

Scalability and Real-Time Performance in Large Vocabularies

A major challenge in building sign language translation systems is their scalability, especially as the vocabulary size increases. Systems must be able to recognize and translate a large number of signs in real time, which often requires significantly more computational resources. Training models to handle larger vocabularies while maintaining accuracy and real-time performance is a problem that continues to challenge developers in the field of sign language translation.

Creating Comprehensive Datasets for Dynamic Sign Language Gestures

Datasets for dynamic sign language gestures are critical for training accurate recognition models. However, most existing datasets are limited in terms of the variety of signs, diverse user conditions, or dynamic gestures. A comprehensive dataset that includes various hand movements, facial expressions, and diverse lighting conditions is necessary for training machine learning models that can handle real-world scenarios. The lack of such diverse and high-quality datasets limits the development of more accurate and adaptable sign language translation systems.

Designing User-Centered Interfaces for Sign Language Translation Systems

User interfaces for sign language translation systems must cater to the needs of individuals with hearing impairments, ensuring they are easy to use, intuitive, and accessible. The challenge lies in balancing the functionality of the system with simplicity, making sure that users of all ages and cognitive abilities can navigate the system effectively.

The lack of universally accepted design guidelines for such specialized interfaces presents a major barrier to widespread adoption of these tools.

Real-Time Translation from Sign Language to Text and Speech

While gesture recognition is a key component of sign language translation, the real-time conversion of recognized gestures into text or speech remains a challenge. Effective real-time conversion requires not only accurate gesture recognition but also the integration of natural language processing (NLP) systems that can generate meaningful text or speech. The latency and errors in these real-time translation systems can hinder their effectiveness, particularly in communication-heavy environments such as classrooms, meetings, or public spaces.

Adapting to Occluded and Overlapping Hand Gestures in ISL Recognition

In real-world settings, users may perform sign language gestures with partially obstructed hands or overlapping gestures. This leads to difficulties in accurate gesture recognition and translation. Most models trained for Indian Sign Language (ISL) rely on clean hand movements, but in practice, gestures can be occluded by objects or other body parts, affecting the accuracy of recognition. Developing systems that can handle these real-world challenges remains a significant problem in sign language translation technology.

3.1. Existing System

Traditional sign language recognition systems often rely on pre-recorded videos for each word or phrase. These videos are collected into a database and then retrieved when the corresponding word needs to be translated. The approach is simple and effective for a limited vocabulary, with each video representing a specific sign. Basic Natural Language Processing (NLP) techniques like tokenization and tagging are applied to process the text and generate the corresponding sign language gesture. However, the NLP component often lacks advanced contextual understanding, which means it may not accurately interpret sentences in a nuanced manner. The system does not allow for real-time interactions, making it less dynamic and adaptable to spontaneous conversations. Additionally, the pre-recorded videos do not accommodate variations in signing styles or regional dialects, limiting their applicability. While this method is straightforward to implement, it suffers from the drawback of needing constant updates as new words and phrases emerge. Maintenance becomes intensive as every new word or phrase requires the creation and storage of new video clips, which can quickly become unmanageable.

Drawbacks:

Limited Vocabulary: Pre-recorded video databases are limited to the number of words and phrases they contain, making the system inefficient for larger, more dynamic conversations.

Lack of Contextual Understanding: Basic NLP techniques cannot capture the full context of sentences, which often results in inaccurate translations of phrases, particularly when there are ambiguous meanings.

No Real-Time Interaction: The reliance on pre-recorded videos means the system cannot interact in real-time, making it unsuitable for live conversations or spontaneous communication.

Maintenance Intensive: Continuously updating the system to include new signs or words requires the creation and addition of new videos, which can be time-consuming and costly.

No Adaptation to Regional Variation: Pre-recorded systems often fail to account for regional variations in sign language, which can lead to misinterpretations, especially in diverse linguistic communities.

Dependence on Fixed Data: The system’s reliance on static video data makes it unable to adapt to new gestures or signs without manual updates, limiting its flexibility and scalability.

Scalability Issues: As the vocabulary and database grow, the system may struggle with performance, especially when handling large datasets of pre-recorded videos, making it difficult to scale.

3. 2. Hardware Requirements

Processor - I3/Intel Processor

The processor is the central component of a computer that performs most of the processing inside a system. An Intel i3 processor is an entry-level option that provides sufficient processing power for running lightweight applications such as sign language recognition systems that don't require heavy computational resources. The I3 processor supports basic tasks and ensures smooth performance for general operations, making it suitable for systems with moderate requirements.

Hard Disk - 160GB

The hard disk serves as the primary storage device where the operating system, applications, and data are stored. A 160GB hard disk provides enough space to store the operating system, essential software, and a moderate amount of data, including video files, training datasets, and system files. This size is appropriate for a system that doesn't need large-scale storage but still requires adequate space for everyday usage and backups.

Keyboard - Standard Windows Keyboard

A standard Windows keyboard is designed to be compatible with most operating systems, including Windows, which is commonly used in sign language recognition systems. It provides all the necessary keys for typing, navigating the system, and performing basic operations, ensuring smooth user input. This type of keyboard is reliable and cost-effective, making it a practical choice for basic usage in office or educational environments.

Mouse - Two or Three Button Mouse

A two or three-button mouse is a standard input device used for pointing, clicking, and scrolling through graphical interfaces. It allows for precise control of the system, enabling users to interact with software applications such as sign language recognition tools. The mouse's basic functionality is sufficient for tasks like navigating menus, selecting files, or adjusting settings, and its simplicity makes it user-friendly for both novice and experienced users.

Monitor - SVGA

SVGA (Super Video Graphics Array) monitors are a common type of display used in many computing systems. With a resolution of 800x600 pixels or higher, SVGA monitors provide clear and readable visuals for general tasks such as software interaction, viewing sign language gestures, and analyzing video data. While SVGA resolution is lower than modern HD displays, it remains suitable for basic applications and is often chosen for budget-friendly setups or environments where high-resolution visuals are not critical.

RAM - 8GB

RAM (Random Access Memory) is essential for temporarily storing data that is actively used or processed by the computer. An 8GB RAM is a standard configuration that offers adequate performance for most mid-range applications, including sign language recognition systems.It allows the system to handle multiple tasks simultaneously without significant slowdowns, such as running software, processing video data, or handling user input. This amount of RAM provides a good balance between cost and performance for general-purpose computing.

3.3 Software Requirements

Operating System - Windows 7/8/10

The operating system (OS) is the software that manages hardware resources and provides a platform for running applications. Windows 7, 8, or 10 are commonly used OS versions that support a wide range of applications and are user-friendly. These versions of Windows provide stability, compatibility, and security updates, making them ideal for running sign language recognition systems and general-purpose applications with reliable performance.

Server Side Script - HTML, CSS, Bootstrap & JS

HTML (Hypertext Markup Language) is used for structuring web content, while CSS (Cascading Style Sheets) is responsible for the design and layout of the web pages. Bootstrap is a front-end framework that uses CSS and JavaScript to create responsive, mobile-first websites with ease. JavaScript (JS) enables dynamic behavior and interaction on web pages. These technologies work together to create a user-friendly, visually appealing interface for web applications, providing the structure and functionality for user input and display.

Programming Language – Python

Python is a high-level, interpreted programming language known for its simplicity and readability. It is widely used for developing web applications, data analysis, machine learning, and automation tasks. In sign language recognition systems, Python is ideal because of its extensive libraries, ease of integration with machine learning models, and strong community support, making it a go-to choice for both beginners and advanced developers.

Libraries - Django, Pandas, Mysql.connector, Os, Smtplib, Numpy

Django: A high-level Python web framework that promotes rapid development of secure and maintainable websites. Django is used for creating backend services, handling requests, and managing databases.

Pandas: A powerful library used for data manipulation and analysis. It simplifies working with data structures like dataframes, making it ideal for managing large datasets used in machine learning tasks or data processing.

Mysql.connector: A Python library that allows interaction with MySQL databases, enabling database queries, updates, and retrievals directly from Python code.

OS: A Python library that provides a way to interact with the operating system, including file management and execution of system commands.

Smtplib: A built-in Python library used for sending emails using the Simple Mail Transfer Protocol (SMTP), useful for notification or alert features in web applications.

Numpy: A core library for numerical computing in Python, Numpy is used for handling arrays and performing advanced mathematical operations, essential for data manipulation and machine learning tasks.

IDE/Workbench - PyCharm

PyCharm is a popular integrated development environment (IDE) specifically designed for Python development. It offers powerful features like code completion, debugging tools, version control integration, and project management. PyCharm enhances developer productivity by providing a comprehensive environment for writing, testing, and debugging Python code, making it ideal for sign language recognition and web development projects.

Technology - Python 3.6+

Python 3.6 and above refers to versions of Python that support modern features like f-strings, improved performance, and enhanced security. Python 3.6+ ensures better compatibility with newer libraries and frameworks. It’s important for running modern Python code and leveraging the latest features, making it a great choice for developing advanced applications like sign language translation.

Server Deployment - Xampp Server

XAMPP is a free and open-source cross-platform web server solution stack package that includes Apache, MySQL, and PHP. It's a popular choice for local web development as it simplifies the deployment and testing of web applications on personal computers. With XAMPP, developers can easily create and test web applications before deploying them to a live server, making it ideal for initial development and testing phases of sign language recognition systems.

Database - MySQL

MySQL is an open-source relational database management system (RDBMS) that stores and manages data in structured formats using SQL queries. It is highly efficient for handling large datasets, ensuring fast retrieval and update of information. MySQL is commonly used in web applications to store user data, records, and other important information, and it integrates seamlessly with Python through the Mysql.connector library. This makes it a perfect choice for storing and managing the data generated by sign language translation systems.

4.PROPOSED SYSTEM

The proposed system introduces an advanced and efficient approach to sign language translation by integrating sophisticated Natural Language Processing (NLP) techniques and real-time hand gesture recognition. Unlike traditional systems that rely on static databases of pre-recorded videos for each word or phrase, this method uses Python's Natural Language Toolkit (NLTK) to enhance language understanding. NLTK enables lemmatization, part-of-speech tagging, and tense detection, allowing the system to grasp the full context and grammatical structure of sentences. This ensures that the generated sign language is not only accurate but also meaningful and relevant to the conversation.

For gesture recognition, the system utilizes MediaPipe, a robust framework developed by Google for real-time hand tracking. MediaPipe enables the system to detect and interpret hand movements instantly, eliminating the latency found in pre-recorded video methods. This allows users to interact in real-time, which is especially valuable in live communication scenarios. The dynamic nature of this approach makes the system highly scalable, as it can adapt to a wide range of phrases and sentence structures without needing to expand a library of gesture videos.

One of the primary advantages of this proposed method is its ability to deliver real-time interaction. Users receive instant feedback through gesture recognition, making the system responsive and seamless to use. Additionally, the integration of NLP allows for contextual understanding, meaning the system can process not just individual words but entire phrases and sentences in context. This results in more accurate and human-like sign language generation.

Moreover, the system is user-friendly and designed to provide an interactive experience, making it especially useful for educational purposes or for those learning sign language. Its dynamic and scalable architecture ensures that it can grow and adapt over time without the need for excessive manual updates, such as recording new gesture videos. All these features combined result in a modern, intelligent solution that is far more efficient and accessible than existing static systems, ultimately making communication easier for the deaf and hard-of-hearing community.

4.1.Objectives

To Enable Bi-directional Translation Between Speech and Sign Language

The core aim of the project is to facilitate seamless translation from spoken language to sign language and vice versa. This dual functionality ensures that communication between hearing and non-hearing individuals is smooth, enabling inclusivity in real-world interactions such as education, customer service, and everyday communication.

To Achieve High Accuracy in Voice and Gesture Recognition

A major focus of the system is to ensure high precision in both voice recognition and gesture interpretation. By leveraging modern tools such as speech-to-text engines and hand gesture detection models like MediaPipe, the system is trained to minimize errors and produce reliable outputs that accurately reflect the intended message.

To Handle Multiple Languages and Dialects

The system is designed with multilingual support in mind, aiming to recognize and translate a variety of languages and regional dialects in both spoken and sign formats. This ensures the tool can be used in diverse communities, making it a global solution for accessibility and communication.

To Implement Advanced NLP for Contextual Understanding

Natural Language Processing (NLP) is integrated to understand sentence structures, grammar, and meaning. This enables the system to go beyond word-to-word translation and instead provide context-aware interpretations that capture the true intent behind a sentence, which is crucial for effective communication.

To Utilize Real-Time Processing Capabilities

The project emphasizes real-time processing, allowing users to receive immediate translation results. This is particularly important in dynamic environments like classrooms, meetings, or public spaces where live interaction is essential and delay can hinder communication.

To Provide a User-Friendly Interface for All Age Groups

Ease of use is central to the project’s design. The system will feature a clean and intuitive interface that can be used by people regardless of their technological proficiency. Accessibility features like large buttons, voice commands, and simple navigation are prioritized to accommodate users from children to the elderly.

To Minimize System Latency and Enhance Speed

The project aims to deliver fast processing speeds without sacrificing accuracy. By optimizing the back-end logic and utilizing efficient algorithms, the system reduces lag and ensures a smooth user experience, which is vital for real-time applications like live conversations or interactive learning.

To Support Integration with Assistive Devices and Platforms

Another objective is to make the system adaptable for integration with existing assistive technologies such as screen readers, smart gloves, or hearing aids. It also aims to be deployable on different platforms like mobile apps, web apps, and desktop environments, broadening its usability.

To Contribute to Research in Language and Accessibility Technologies

This project is not only a practical tool but also a contribution to academic research in accessibility, machine learning, and human-computer interaction. Insights gained during development can aid future innovations in sign language processing and inclusive technology design.

To Enhance Social Inclusion for the Deaf and Hard-of-Hearing Community

Ultimately, the project seeks to improve the lives of individuals with hearing impairments by empowering them with tools that promote independence and social interaction. By breaking communication barriers, the system fosters inclusivity and allows users to actively participate in educational, professional, and social environments.

4.2.Problem Formulation

The core problem addressed by this project is the communication gap between the hearing and the deaf or hard-of-hearing community. The lack of a dynamic, real-time translation tool that can interpret both audio into sign language and sign language into readable text creates challenges in daily communication, education, and social interaction. This project proposes a solution that combines advanced Natural Language Processing (NLP) and Computer Vision to build a real-time, interactive translation system. To implement this, three major development areas are identified: database schema design, backend development, and frontend development.

Database Schema Design

The foundation of any application is its data structure. In this project, the database will store user information, recognized gestures, corresponding text outputs, and logs of translation history. The schema is carefully designed to support both scalability and performance. It also ensures data integrity, proper indexing for faster access, and secure storage of user inputs. A well-structured schema helps in effective data retrieval and future system enhancements.

Backend Development

The backend serves as the brain of the system, managing the logic, communication with the database, gesture recognition processes, audio-to-text conversion, and NLP integration. Python will be the core language, using frameworks like Django to handle server-side functionality. The backend also incorporates MediaPipe for gesture recognition and speech recognition libraries for audio processing. Efficient API design will allow seamless communication between frontend and backend, ensuring fast and reliable translation.

Frontend Development

The frontend is responsible for presenting the system to users in a clear and accessible way. Developed using HTML, CSS, Bootstrap, and JavaScript, the interface is designed to be responsive and user-friendly. It will offer features like speech input, webcam gesture capture, real-time sign language display, and text output. A smooth and intuitive user experience is a top priority, ensuring that users of all technical skill levels can comfortably use the system.

4.3 Methodology of the Proposed System

NLP:

Natural Language Processing (NLP) is a crucial component of your project, which aims to bridge spoken language and sign language through advanced computational techniques. In this context, NLP serves multiple purposes, from understanding and processing user input to enhancing the interaction between users and the sign language translation system. Let's break down the various aspects of NLP utilized in this project:

Text Tokenization:

Definition and Purpose: Tokenization is the process of breaking down text into smaller units, typically words or phrases. In your system, tokenization is the first step in analyzing the user's input text.

Implementation: The NLTK library is used to tokenize sentences into individual words. This step is fundamental because it prepares the text for further processing and analysis.

Part-of-Speech (POS) Tagging:

Definition and Purpose: POS tagging involves identifying the grammatical category of each word in a sentence, such as whether a word is a noun, verb, adjective, etc. This is critical in understanding the structure of the sentence and the role of each word.

Implementation: The system employs NLTK's POS tagger to assign tags to each word. This information is used to determine the tense of the sentence and the grammatical structure, which is vital for accurate sign language translation.

Tense Detection:

Definition and Purpose: Understanding the tense of a sentence is essential for accurate sign language translation. Tense detection helps in conveying the correct time frame of the action or event described.

Implementation: The system analyzes the POS tags to identify verbs and their respective tenses. The tense is determined based on the frequency and type of verbs present (e.g., past, present, future).

Lemmatization:

Definition and Purpose: Lemmatization involves reducing words to their base or dictionary form. Unlike stemming, it considers the context and converts the word to its meaningful base form.

Implementation: NLTK’s WordNetLemmatizer is used. It's particularly useful in handling different forms of a word, ensuring that the system recognizes 'running,' 'ran,' and 'runs' as the same base word 'run.'

Stop Word Removal:

Definition and Purpose: Stop words are common words like 'is,' 'and,' 'the,' etc., that are often filtered out because they contribute little to the overall meaning of the sentence in the context of sign language translation.

Implementation: The system utilizes a predefined list of stop words provided by NLTK. Removing these words can help in focusing on the more meaningful words that need to be translated into sign languages.

Contextual Understanding:

Definition and Purpose: This involves understanding the context in which words are used, which is crucial for accurate translation. This can mean the difference between correctly interpreting phrases and idioms versus taking words at face value.

Implementation: Advanced NLP techniques, possibly involving machine learning models, are used to understand the context. This could be further enhanced with Named Entity Recognition (NER) to identify and properly translate proper nouns.

Sentence Reconstruction:

Definition and Purpose: After processing the text, it may be necessary to reconstruct the sentence, especially in sign language translation, where the sentence structure can be different from spoken language.

Implementation: The system rearranges the words based on the rules of sign language grammar, which often differs from the syntax of spoken language.

Handling Special Cases:

Definition and Purpose: Some words or phrases might not have a direct translation in sign language or may be specific to a region or culture.

Implementation: The system can have a special handling mechanism for such cases, possibly redirecting to a human translator or using a default sign.

Integration with Sign Language Translation:

Definition and Purpose: The final step involves integrating the processed text with the sign language translation module.

Implementation: Once the text is processed, it is fed into the sign language translation system, which then converts it into corresponding sign language gestures, possibly through animated avatars or videos.

Feedback and Learning:

Definition and Purpose: Continuously improving the accuracy and effectiveness of the translation system.

Implementation: Implementing a feedback mechanism where users can provide input on the accuracy of translations, which can be used to train and refine the system further.

In summary, NLP plays a multifaceted role in your project, handling everything from initial text processing to ensuring that the translation into sign language is contextually and grammatically accurate. The use of NLTK for tasks like tokenization, POS tagging, lemmatization, and stop word removal forms the backbone of the text processing module. This is complemented by advanced techniques for contextual understanding and sentence reconstruction, ensuring that the system not only translates words but also conveys the intended meaning accurately in sign language.

5.DATASET COLLECTION

5.1.Dataset Description

In order to train and evaluate the performance of the Indian Sign Language to Text/Speech Translation system, a diverse set of datasets covering multiple modalities — including text, audio, and video — were required. Each dataset was carefully selected or created to serve a specific purpose in the system: from natural language understanding to gesture recognition. Below are the key datasets used and curated for the project, organized under different functional needs.

Indian Sign Language Gesture Dataset

This dataset includes thousands of video clips of individuals performing Indian Sign Language (ISL) gestures. Each video is labeled with the corresponding English or Hindi word, making it suitable for gesture recognition training. The dataset includes common words, verbs, pronouns, and some short phrases, recorded under different lighting conditions and backgrounds for robustness.

Self-Curated Sign Language Alphabet Dataset

To train the system on recognizing individual signs representing the alphabet (A–Z), a custom dataset was curated using webcam recordings. Multiple participants performed each letter of the ISL alphabet in front of a neutral background. The dataset was annotated manually and is used primarily in the initial stages of gesture recognition training.

RWTH-PHOENIX-Weather 2014T Dataset (Benchmark)

Though originally based on German Sign Language, this dataset served as a benchmark for understanding how temporal models (e.g., GRU or LSTM) can map sequences of video frames to text. It includes aligned video, gloss, and text translations, which are helpful for validating sequence models.

Indian Regional Audio Dataset

To handle diverse accents and speech patterns across India, a regional audio dataset was compiled from open-source repositories and crowd-sourced audio recordings. The dataset contains audio clips in English and Hindi with labels, allowing the system's speech-to-text module to be trained across variations in pronunciation, tone, and speed.

Google Speech Commands Dataset

This dataset was used for pretraining the audio recognition model. It includes short audio clips (1-second) of simple voice commands, which helped the system understand phoneme transitions and develop a baseline for English speech recognition.

Indian Names and Places Dataset

A curated list of commonly used Indian names (people and places) was compiled to improve the performance of NLP modules, particularly Named Entity Recognition (NER). These names are often missed or misinterpreted by standard language models and required specialized handling for accurate sign translation.

Indian Language WordNet (Hindi-English)

The IndoWordNet dataset was used for semantic mapping of words and improving contextual understanding. It allows for better lemmatization and translation accuracy by providing synonyms and part-of-speech classifications, which helps align spoken or typed inputs to their correct sign language equivalents.

NLTK Corpora (Brown, Gutenberg, Stopwords)

Several pre-built datasets from the NLTK library were used for natural language processing, including the Brown and Gutenberg corpora for general English usage patterns, and stopword lists for noise reduction. These datasets helped in tokenization, POS tagging, and sentence structuring.

Custom Text Input Dataset

To fine-tune the NLP models for the kind of short, informal phrases typically entered by users, a custom dataset of 5,000+ user-generated sentences was collected via online forms. These included common expressions, daily usage phrases, and short conversational entries with labeled grammatical structures.

Sign Language Animation Dataset

This dataset includes animated 3D avatar sequences representing ISL gestures. These animations are linked to corresponding words and phrases in the gesture database.

It helped render smoother real-time translations for text-to-sign output using avatar animations.

Facial Expression Sign Dataset

This dataset contains video clips of facial expressions associated with non-manual signs in ISL, such as questioning, affirmation, or emotional tone. It is crucial for enhancing the realism and grammatical correctness of sign language output.

User Feedback Dataset

A live feedback dataset was created during beta testing of the system, where users reported accuracy scores and gave qualitative responses to gesture interpretations. This dataset was used to tune the model's performance and prioritize improvement areas, especially in ambiguous cases.

6.SYSTEM DESIGN

6.1. High-Level Design Documentation

Input Design The input design for our audio-sign and sign-audio translation project is pivotal for achieving accurate and efficient processing by the system. It involves several components, including text input, audio input, and sign language gesture input, all of which are processed using different technologies such as NLP and Media Pipe. Below are the key aspects of the input design:

Text Input for Sign Language Conversion

Format: The system should accept text in standard formats, including typed input or pasted text.

Preprocessing Steps:

Tokenization: Breaking down sentences into individual words or phrases.

Normalization: Converting all characters to a standard case (usually lower case) for consistency.

POS Tagging and Lemmatization: Assigning parts of speech to words and reducing them to their base or dictionary forms for more accurate NLP processing.

Audio Input for Text Conversion

Audio Capture: The system must efficiently capture audio through a microphone or an uploaded audio file.

Voice Recognition: Converting spoken words into text using speech-to-text technology. This text is then further processed for sign language conversion.

Sign Language Gesture Input for Text Conversion

Video Capture: Real-time capturing of hand gestures using a webcam or a camera.

Gesture Recognition: Utilizing MediaPipe to interpret the sign language gestures and convert them into text.

Output Design The output design focuses on presenting the translation results in an accessible and useful manner to the user. This includes:

Translation Results

Textual Display: Showing the converted text from sign language or audio input.

Sign Language Animation: Displaying sign language animations or videos corresponding to the text input.

Visualization and Feedback

Real-Time Feedback: Providing immediate visual feedback on the screen for the sign language gestures being recognized.

User Interface Elements:

Interactive Dashboard: A user-friendly dashboard displaying the input (text, audio, or video) and the corresponding output (text or sign language animation).

Feedback Mechanism: Allowing users to provide feedback on the translation accuracy, which can be used to improve the system.

Reporting and Export Options

Session History: Option to view the history of translations in a session.

Export Functionality: Providing the ability to export the translated text for documentation or further use.

Integrations and Compatibility

Compatibility with Assistive Technologies: Ensuring that the system is accessible and compatible with various assistive technologies used by individuals with hearing or speech impairments.

6.1.1 Use Case Diagram

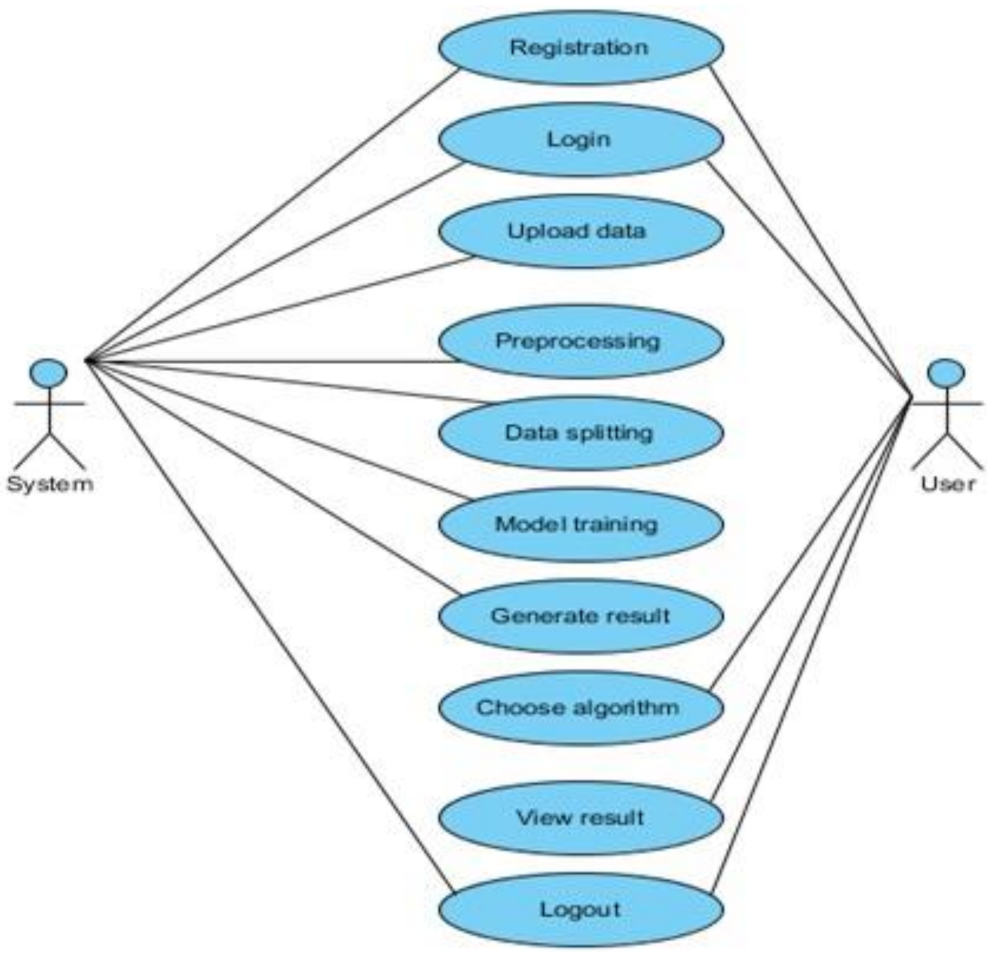


Figure 6.1.1: Use case Diagram

According to the Unified Modeling Language (UML), a use case diagram is a particular kind of behavioral diagram that is produced from and defined by a use-case study. In terms of actors, their objectives (shown as use cases), and any dependencies among those use cases, it serves to graphically summarize the functionality offered by a system. A use case diagram's primary objective is to demonstrate which actors use the system's functionalities. It is possible to illustrate the roles of the system's actors.

6.1.2 Class Diagram

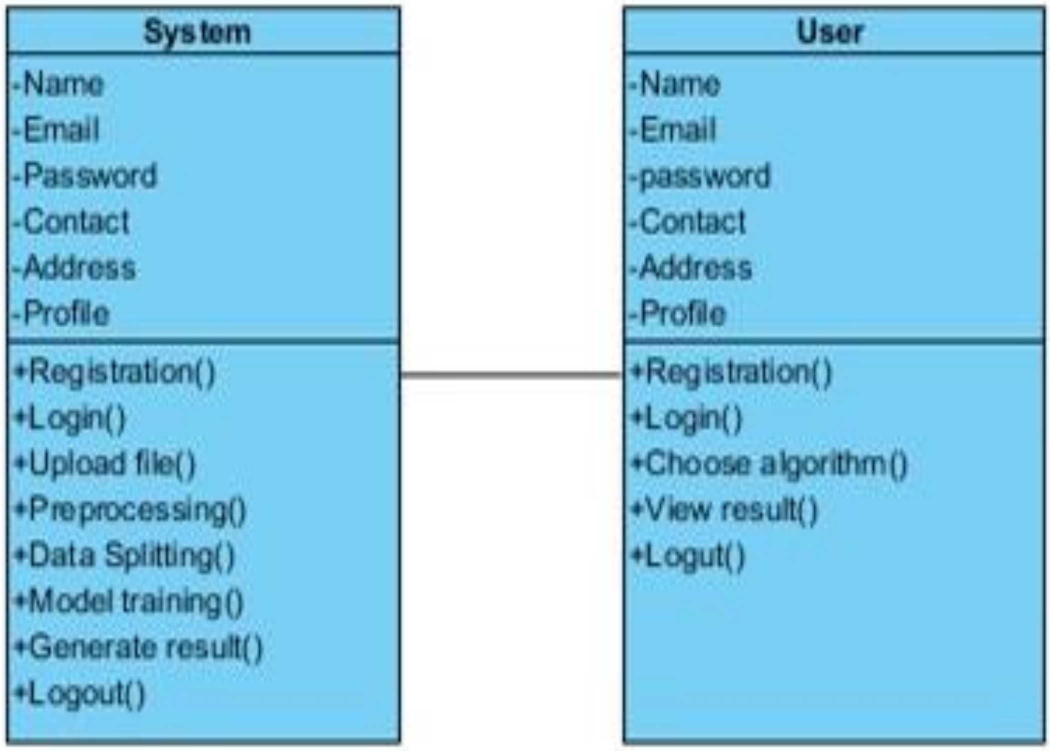


Figure 6.1.2: Class Diagram

A class diagram, as defined by the Unified Modelling Language (UML), is a kind of static structural diagram used in software engineering that illustrates a system's classes, attributes, actions (or methods), and relationships between the classes. It indicates which class includes.

6.1.3 Sequence Diagram

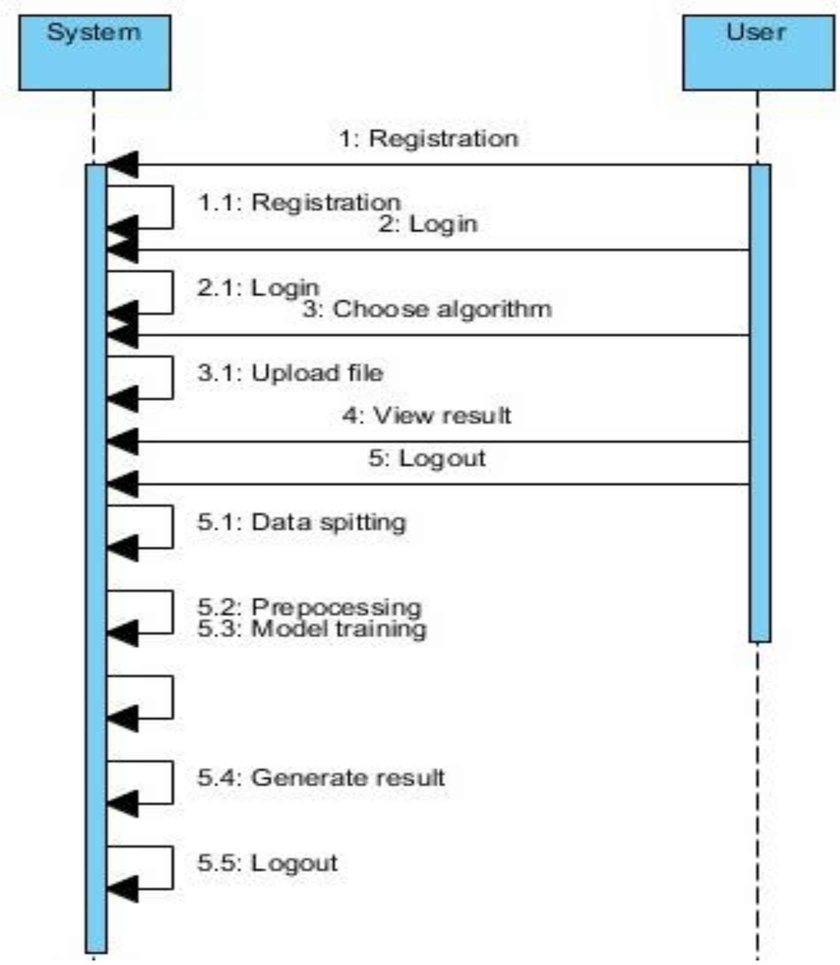


Figure 6.1.3: Sequence Diagram

In the Unified Modeling Language (UML), a sequence diagram is a type of interaction diagram that illustrates the order and manner in which processes interact with one another. It is a MessageSequence Chart construct. Event diagrams, event situations, and timing are other names for sequence diagrams.

6.1.4 Deployment Diagram

Figure 6.1.4: Deployment Diagram

The deployment view of a system is represented by a deployment diagram. The component diagram has anything to do with it. because the deployment diagrams are used to deploy the components. Nodes make up a deployment diagram. The physical hardware used to install the program is known as a node.

6.1.5 Activity Diagram

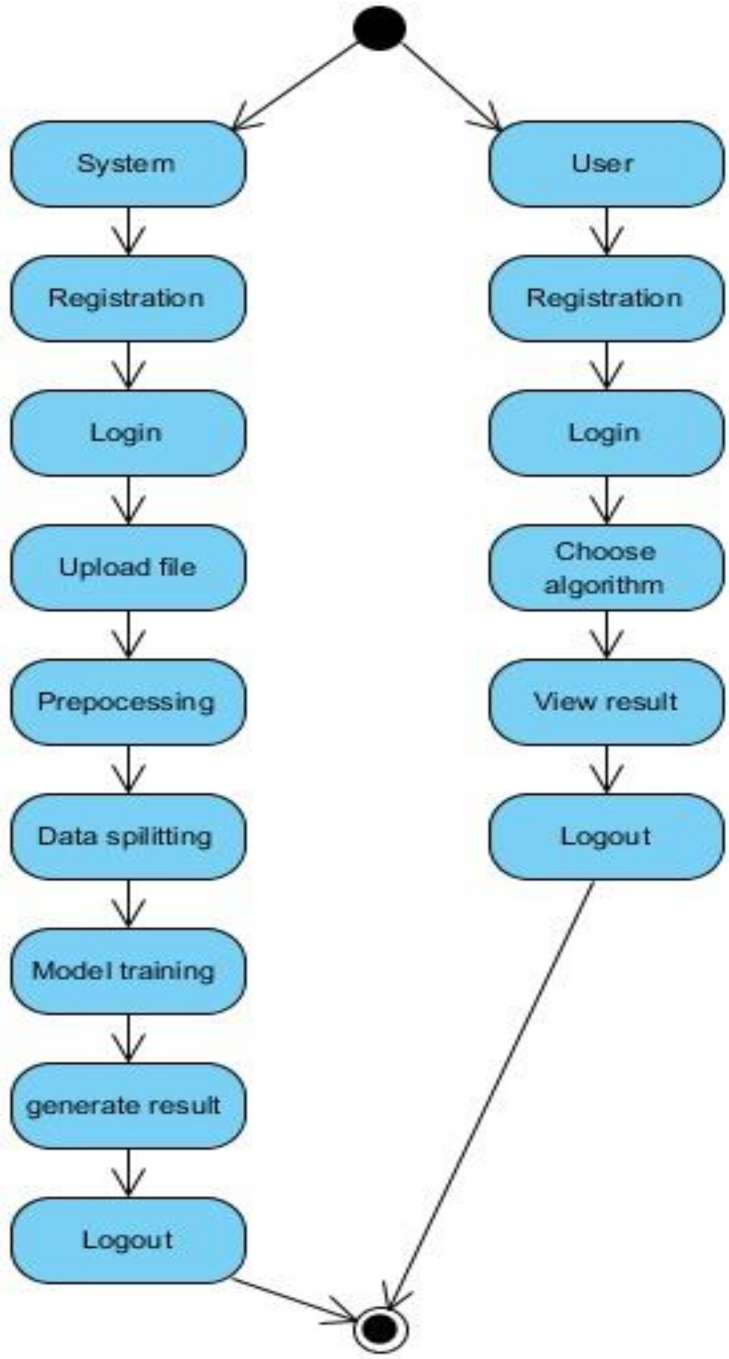
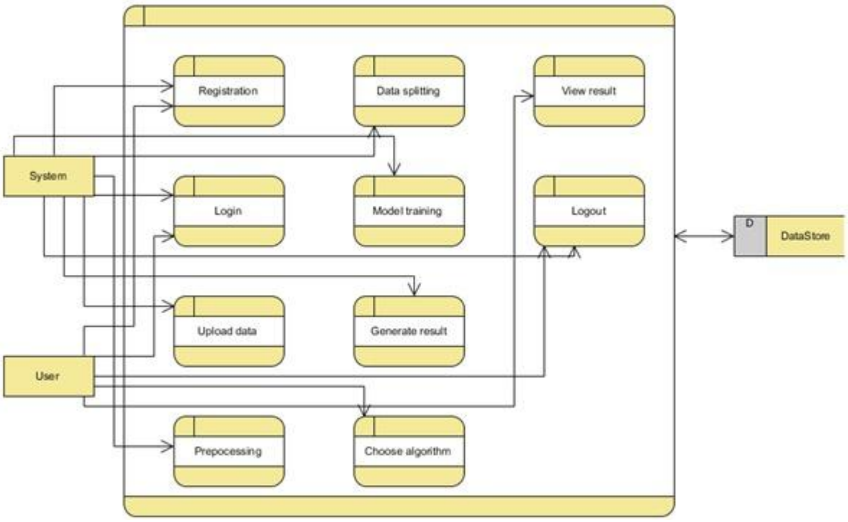


Figure 6.1.5 : Activity Diagram

Activity diagrams are graphical depictions of step-by-step actions and activities that support concurrency, choice, and iteration. The business and operational step-by-step processes of system components can be described using activity diagrams in the Unified Modeling Language. An activity diagram illustrates the general control flow.

6.1.6 Dataflow Diagram

Figure 6.1.6: Dataflow Diagram



The conventional method of visualizing the information flows inside a system is to use a Data Flow Diagram (DFD). A substantial portion of the system requirements can be graphically represented by a clean and unambiguous DFD. It may be automatic, manual, or a mix of the two. It displays where information is stored, how it enters and exits the system, and what modifies it. A DFD is used to illustrate the limits and extent of a system overall. As the first step in redesigning a system, it can serve as a communication tool between a systems analyst and any individual involved in the system.

6.1.7 ER Diagram

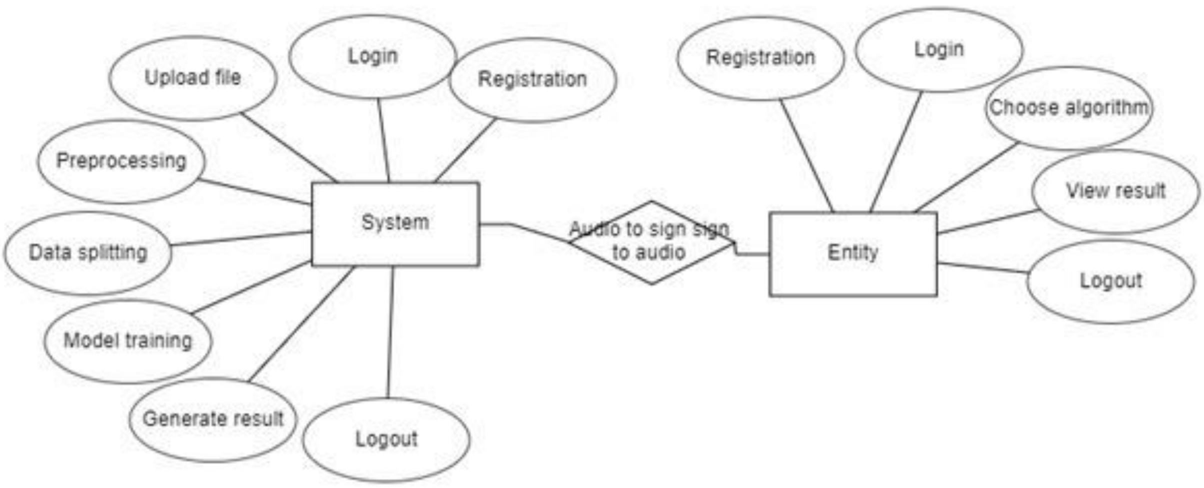


Figure 6.1.7 : ER Diagram

With the use of an entity relationship diagram, an entity–relationship model explains the structure of a database. An ER model is a database design or blueprint that can be used to create a database in the future. The entity set and relationship set are the two primary parts of the E-R model. The relationships between entity sets are depicted in an ER diagram. A collection of related entities, some of which may possess attributes, is called an entity set. Since an entity in DBMS refers to a table or an attribute of a table in a database, an ER diagram illustrates the full logical structure of a database by displaying the relationships between tables and their attributes. Let's examine a basic ER diagram to better grasp this idea.

6.1.8. Component Diagram

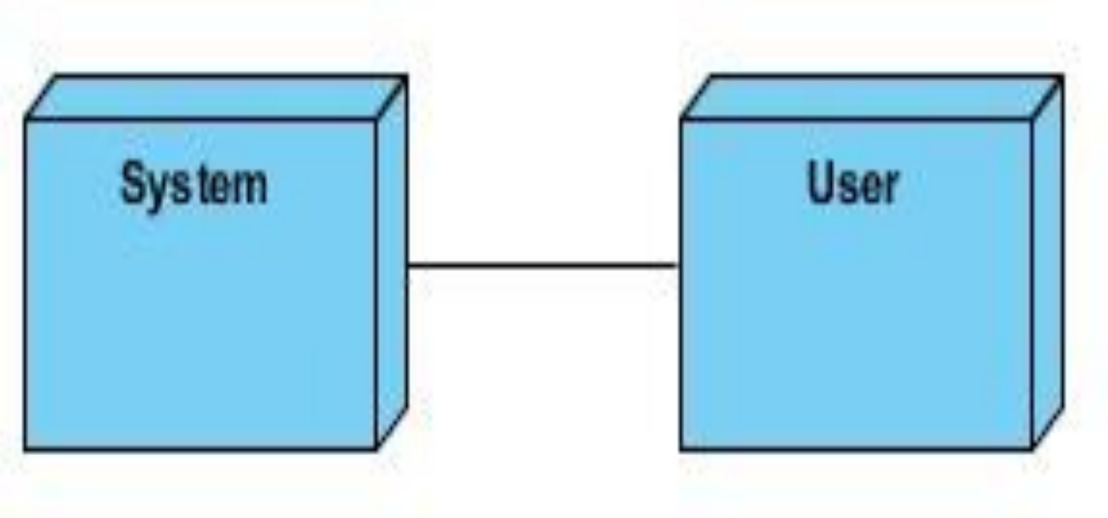


Figure 6.1.8 : Component Diagram

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development.

6.1.9 Component Diagram

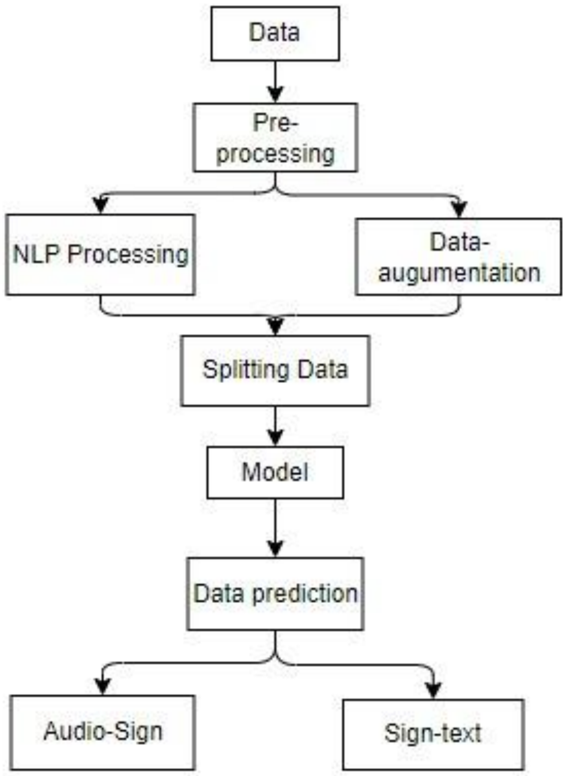


Figure 6.1.9: System Flow Diagram

This flowchart represents the process of translating sign language using machine learning and natural language processing (NLP). Here's a step-by-step explanation:

Data Collection:

The system first collects data, which includes images, videos, or text related to sign language.

Pre-processing:

The collected data is cleaned and prepared for further analysis. This may involve removing noise, normalizing images, or converting text to a suitable format.

Two Parallel Processes:

NLP Processing: If the input is text or speech, it undergoes natural language processing to understand the context.

Data Augmentation: If the input is images or videos, techniques like flipping, rotating, or adding variations help create more training data.

Splitting Data:

The data is divided into training and testing sets to train the model effectively.

Model Training:

A deep learning model (such as CNN, ViT, or RNN) learns patterns from the training data to recognize sign language gestures or translate text.

Data Prediction:

The trained model makes predictions based on new inputs.

Final Output:

The system generates two types of outputs:

Audio-Sign: Converts speech or text into sign language.

Sign-Text: Converts sign language into text for easier understanding.

7. IMPLEMENTATIONS

7.1 Platform/Technologies Used

Platform Setup and Development Environment:

Web Framework: Utilize Django, a Python-based web framework, for building the web application. Django will handle user authentication, routing, server-side logic, and database interactions.

Development Tools: Use tools like Git for version control and a suitable IDE (Integrated Development Environment) like PyCharm or Visual Studio Code for writing and testing code.

User Interface and Experience:

Front-End Development: Employ HTML, CSS, and JavaScript to design a user-friendly and accessible interface. This includes forms for text input, buttons for audio recording, and a display area for sign language animations.

Responsive Design: Ensure the web application is responsive and accessible across various devices and screen sizes.

Text Processing and NLP:

NLTK Library: Implement NLP functionalities using the Natural Language Toolkit (NLTK) for tasks like tokenization, POS tagging, and lemmatization.

Text Preprocessing: Design functions to process user input text, including lowercasing, removing stop words, and normalizing text for consistent processing.

Audio Processing:

Speech-to-Text API: Integrate a speech-to-text API (like Google Cloud Speech API) to convert user speech into text.

Audio File Handling: Implement functionality to handle and process audio input, either live through a microphone or via uploaded audio files.

Sign Language Gesture Recognition:

MediaPipe for Gesture Recognition: Utilize MediaPipe, a machine learning framework, to recognize sign language gestures from video input.

Video Processing: Develop a system to capture live video from the user's webcam, process the video frames in real-time, and recognize sign language gestures.

Sign Language Animation:

Animation Library/Software: Use an animation library or software to create and display sign language animations based on the processed text input.

Database of Gestures: Maintain a database of sign language gestures/animations that can be mapped to corresponding text.

Database and User Management:

Django's ORM: Use Django's ORM (Object-Relational Mapping) for database interactions to store user data, session histories, and possibly a cache of common translations.

User Authentication: Implement Django's authentication system to manage user accounts, including registration, login, and password management.

Integration and Testing:

API Integration: Seamlessly integrate different APIs (like speech-to-text and possibly text-to-speech) into the application.

Testing: Conduct thorough testing, including unit tests, integration tests, and user acceptance testing to ensure all components work as intended.**SMTP/Email (via smtplib - Built-in):** Python's built-in `smtplib` library is used to send alert emails when AQI values exceed certain thresholds. This feature helps notify users about potential health risks in real time.

6.2.System Testing

6.2.1.Testing Strategies

6.2.1.1.Unit Testing

Creating test cases that verify that the internal program logic is operating correctly and that program inputs result in legitimate outputs is known as unit testing. It is necessary to validate the internal code flow and all decision branches. It involves testing each of the application's separate software components. Prior to integration, it is carried out following the completion of a separate unit. This is an intrusive structural test that depends on understanding how it was built. Unit tests test a particular application, system configuration, or business process at the component level. Unit tests guarantee that every distinct business process route operates precisely in accordance with the stated specifications and has inputs and expected outcomes that are well-defined.

6.2.1.2.Integration Testing

The purpose of integration tests is to verify if integrated software components function as a single program. Event-driven testing focuses more on the fundamental results of fields or screens. Although the components were satisfactorily tested independently, as demonstrated by successful unit testing, integration tests show that the components are correctly and consistently combined. Identifying issues that result from the merging of components is the special goal of integration testing. In order to generate failures resulting from interface flaws, software integration testing involves incrementally integrating two or more integrated software components on a single platform. Checking that software applications or components, such as those in a software system or, at a higher level, software applications at the corporate level, interact error-free is the goal of the integration test. Test Results: Every one of the aforementioned test cases was successfully completed. No flaws were found. -> Acceptance Testing User Acceptance Testing is a crucial stage of any project in which the end user must be heavily involved. It also guarantees that the system satisfies the functional specifications.

6.2.1.3.Functional Testing

Functional tests offer methodical proof that the functions being tested are available in accordance with the technical and business requirements, system documentation, and user manuals. The following things are the focus of functional testing: Valid Input: The recognized categories of valid input need to be approved. Invalid Input: Classes of input that have been determined as invalid must be refused. Functions: The functions that have been identified must be used. Output: It is necessary to exercise the designated types of application outputs. Systems/processes: It is necessary to call interacting systems or processes. Functional tests are organized and prepared with requirements, important functionalities, or unique test cases in mind. Additionally, testing needs to take into account data fields, specified procedures, and subsequent processes in order to discover business process flows. Additional tests are found and the usefulness of the existing tests is assessed prior to functional testing being finished.

7.2.1.4. White Box Testing

White box testing is a type of software testing where the tester is aware of the inner workings, language, and structure of the program, or at least its intended use. It has a purpose. It is employed to test regions inaccessible from a black box level.

6.2.1.4.Black Box Testing

Testing software without being aware of the inner workings, structure, or language of the module being tested is known as "black box" testing. Like the majority of other test types, black box tests need to be created from a definite source document, like a requirements or specification document. This type of testing treats the software being tested as a "black box." You are unable to "see" into it. Without taking into account how the software functions, the test generates inputs and reacts to outputs.

7.2.2Test Cases

A The primary objective of testing is to ensure that all system components are functioning as intended, with a special focus on input validation, user interface responsiveness, and proper navigation throughout the application. It is critical that all field entries work correctly and accept only valid data in the required format. Each page and section of the application must be easily accessible through the provided links, ensuring seamless navigation for users. Furthermore, the entry screens should respond instantly, without noticeable delays in processing user inputs or generating system messages.

The testing process will verify whether the application correctly accepts input data in the appropriate formats and rejects invalid or incomplete entries. An important aspect of the validation process is the prevention of duplicate entries, ensuring data integrity and uniqueness across the database. All navigational links embedded within the application should redirect users to their corresponding pages accurately and without error, contributing to an intuitive and error-free user experience.

- **Test Cases**

Input	**Output**	**Result**
Input Text Input gesture	Output should be Sign Output should be Text	Success Success

- **Test Case Module Building**

S.NO	Test cases	I/O	Expected O/T	Actual O/T	Paas/Fail
1	Read the dataset.	Dataset path.	Dataset need to read successfully.	Dataset fetched successfully.	Pass
2	Performing pre-processing on the dataset	Pre-processing part takes place	The dataset should undergo pre-processing.	Preprocessing was finished successfully	Pass
3	Building Models	Building a model for the clean data.	Model creation must be done with the necessary algorithms.	The model was successfully created.	Pass
4	Text-sign Sign-Text	An input image is supplied.	The sign and text should be the output.	Successful model.	Pass

Table 7.1: *Test Cases*

7.3 RESULTS

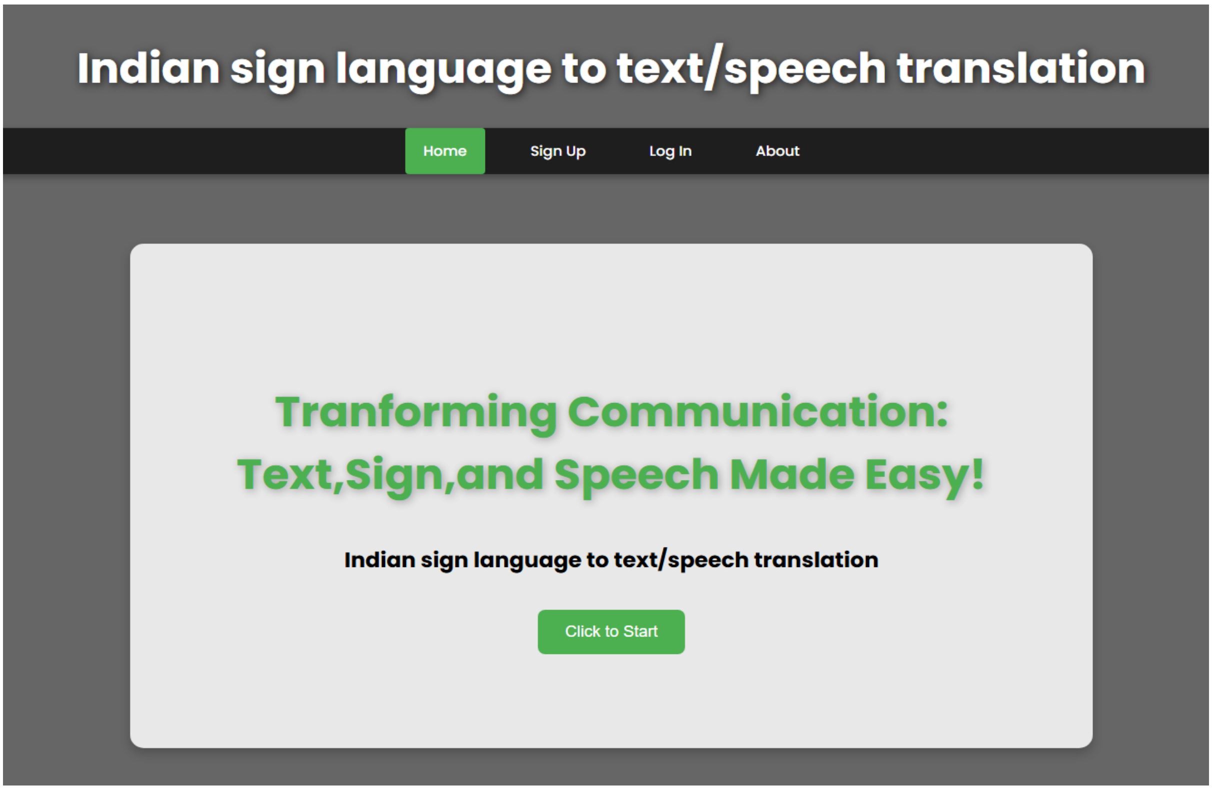


Figure 7.3.1:Home Page

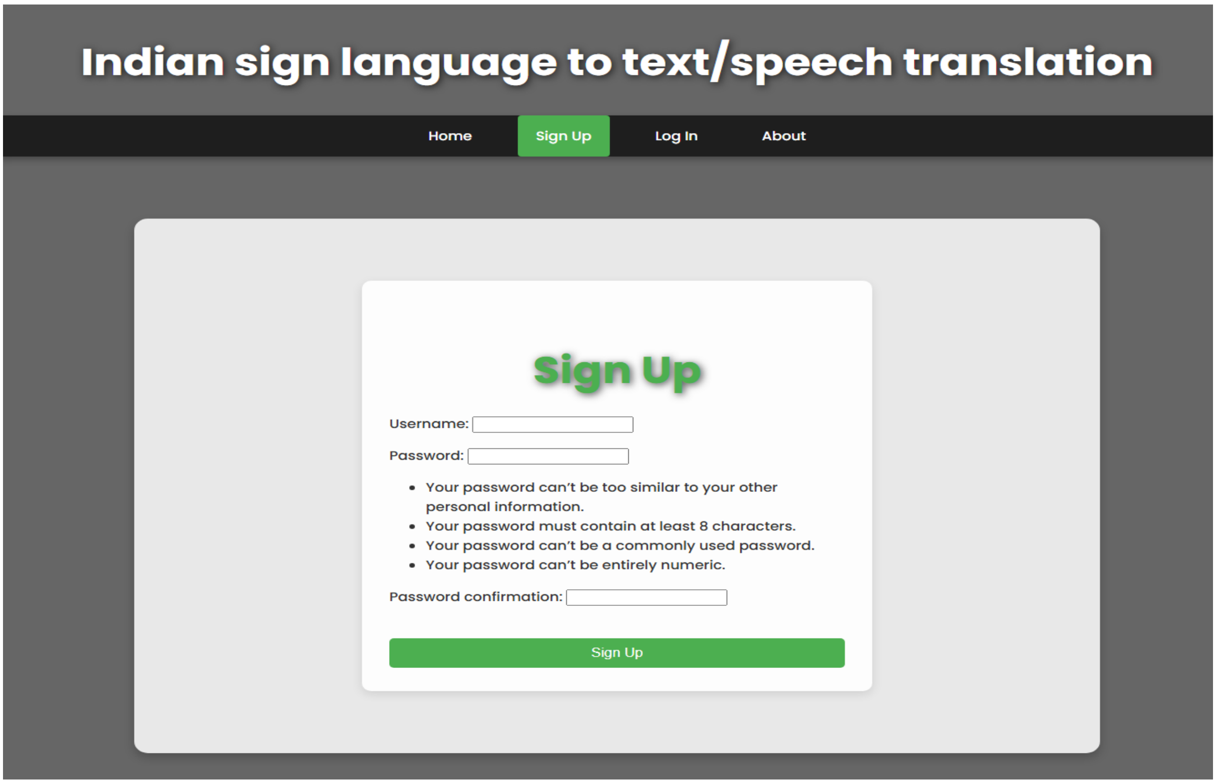


Figure 7.3.2:SignUp Page

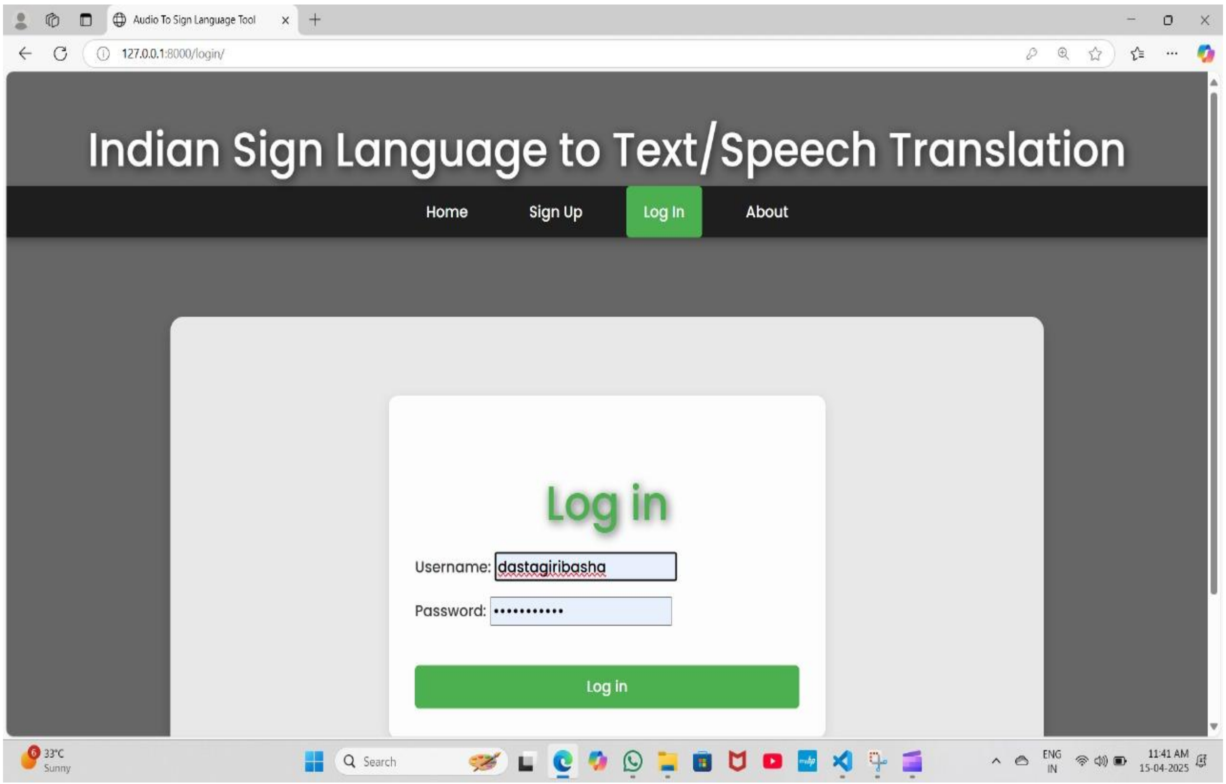


Figure 7.3.3:Login Page

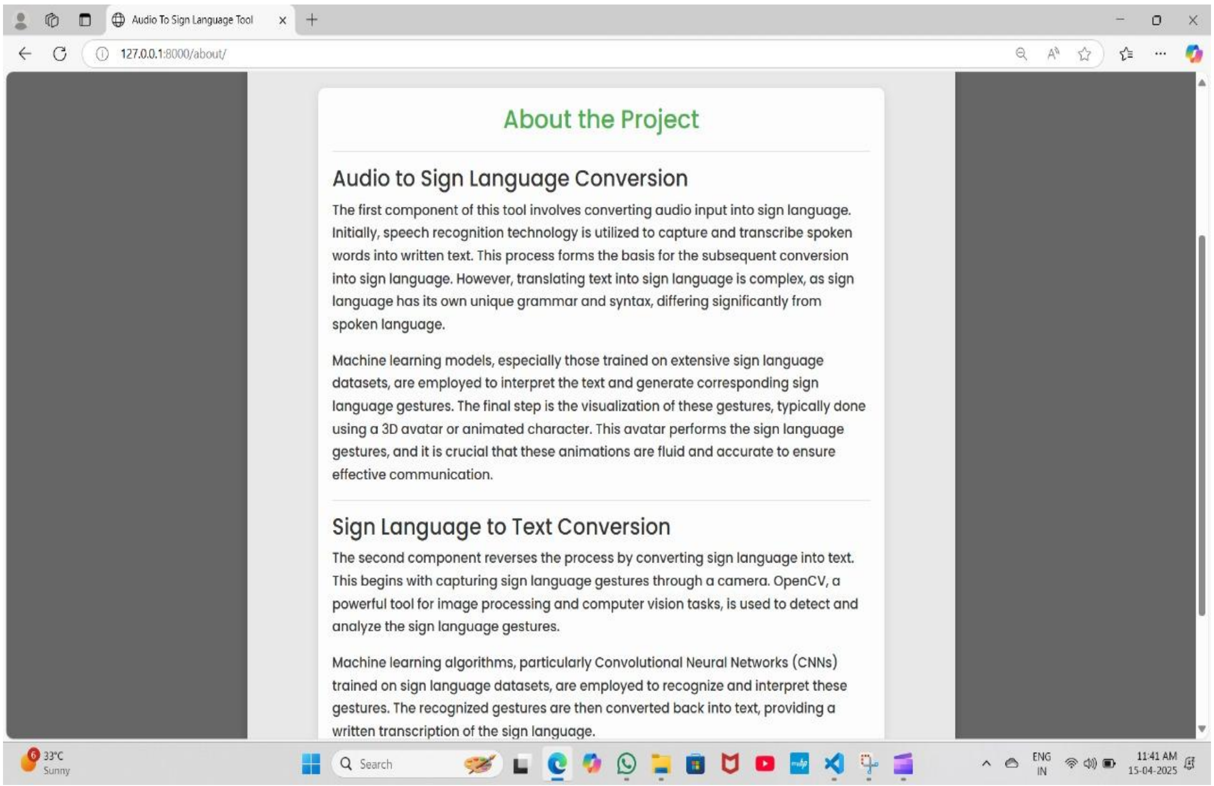
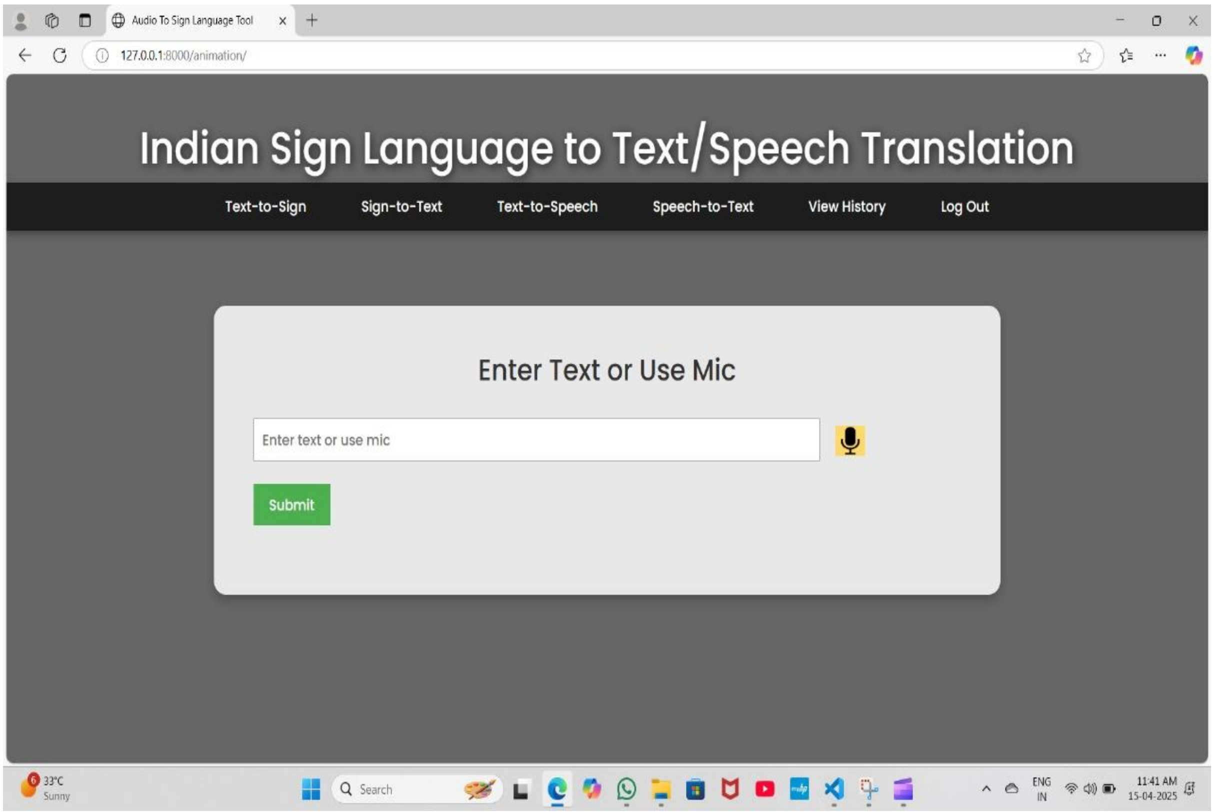


Figure 7.3.4: About Page



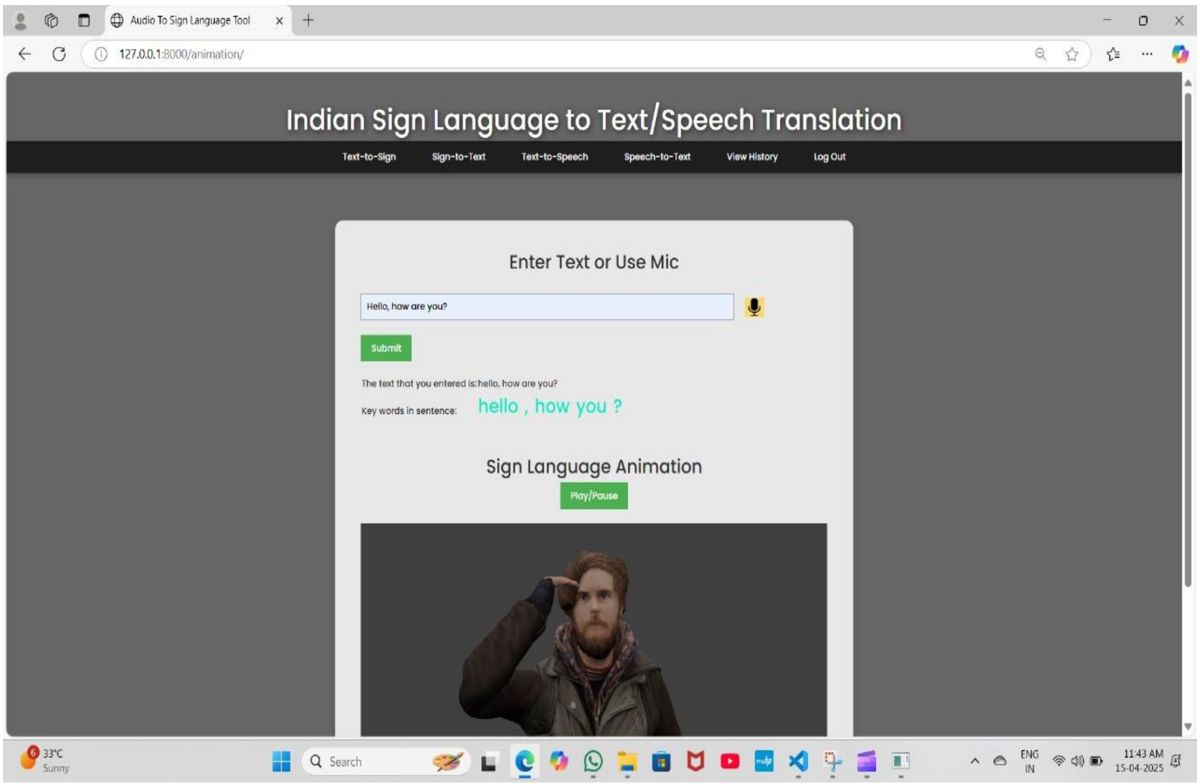


Figure 7.3.5: Text-to-sign Page

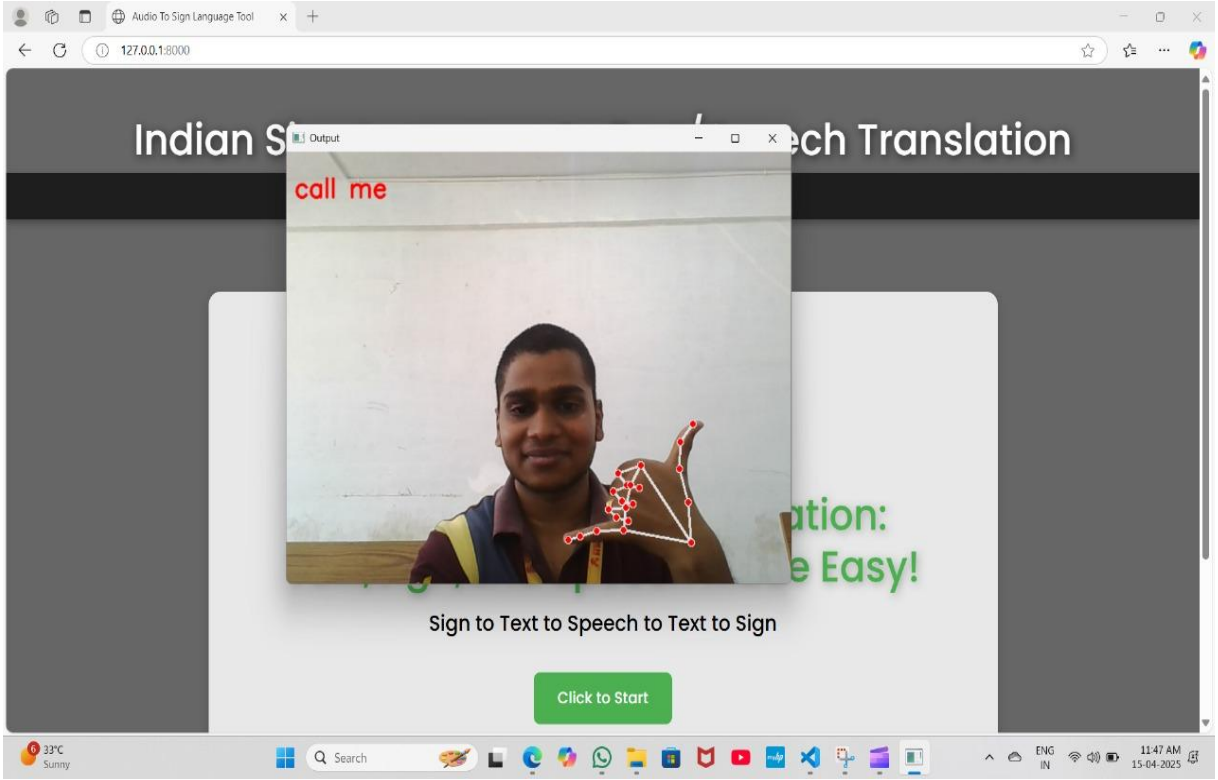


Figure 7.3.6: Sign-to-text Page

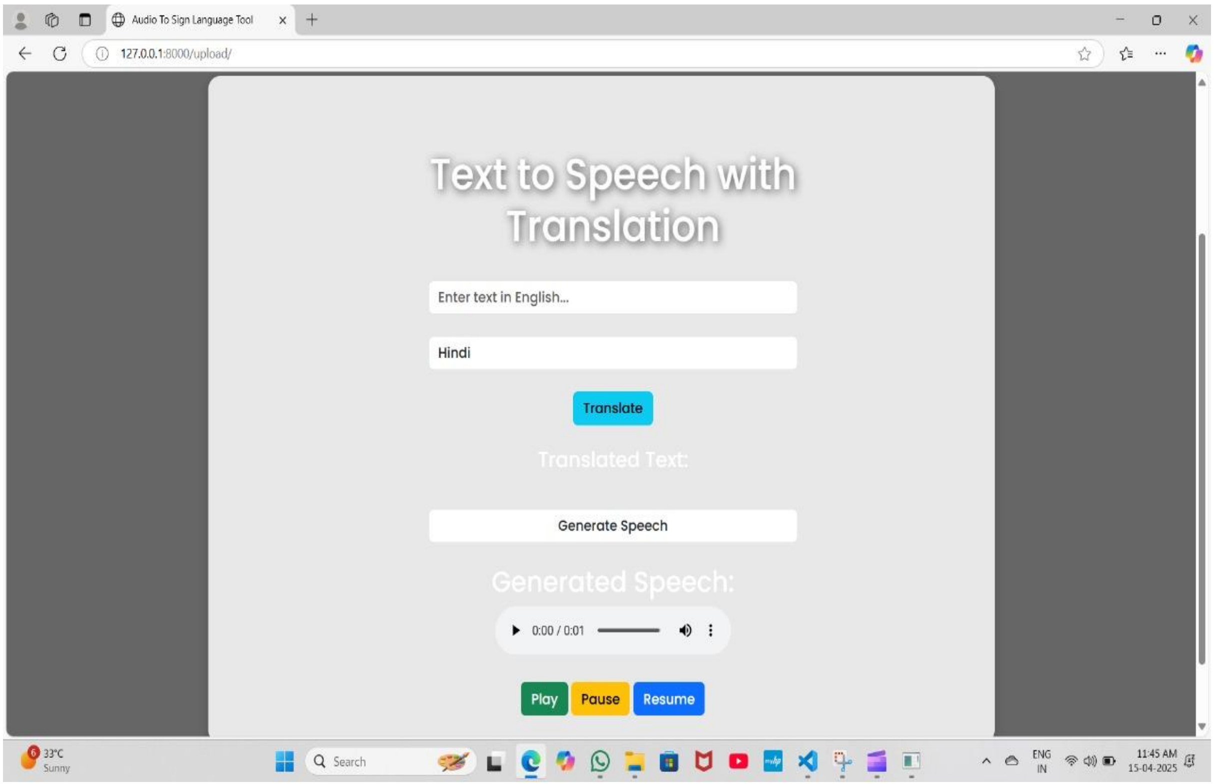


Figure 7.3.7: Text-to-Speech Page

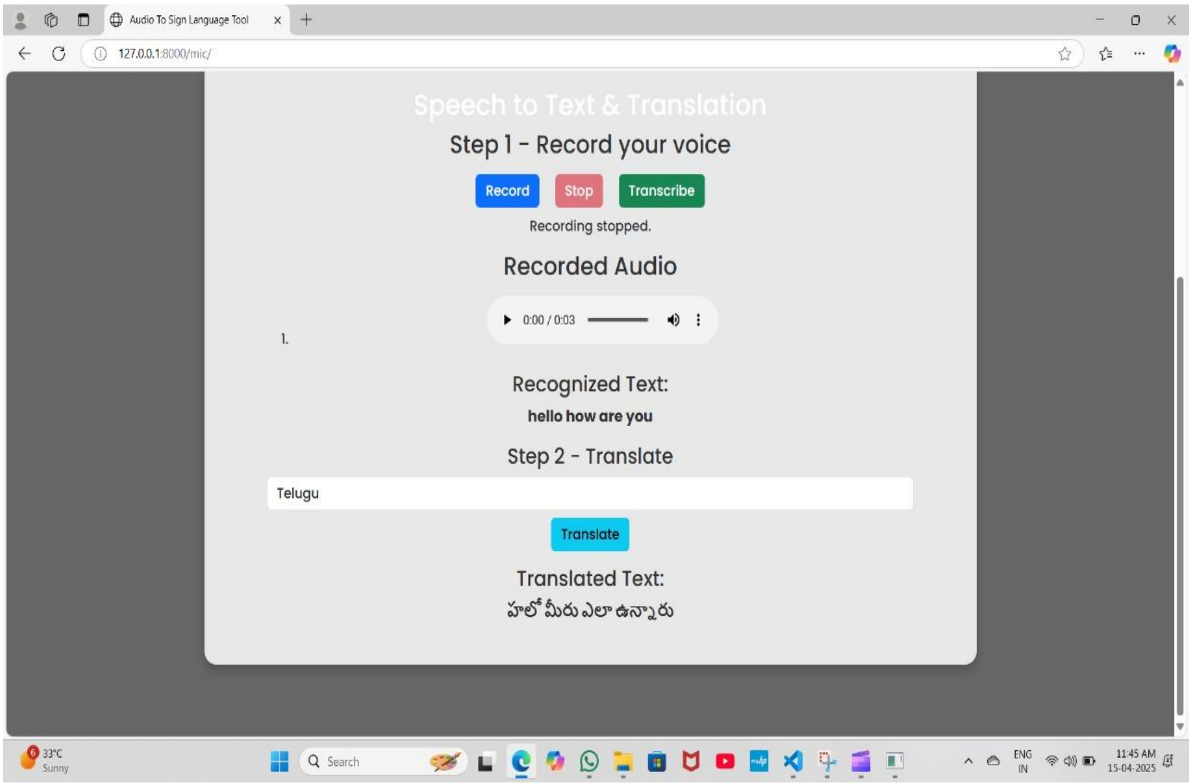


Figure 7.3.8: Speech-to-text Page

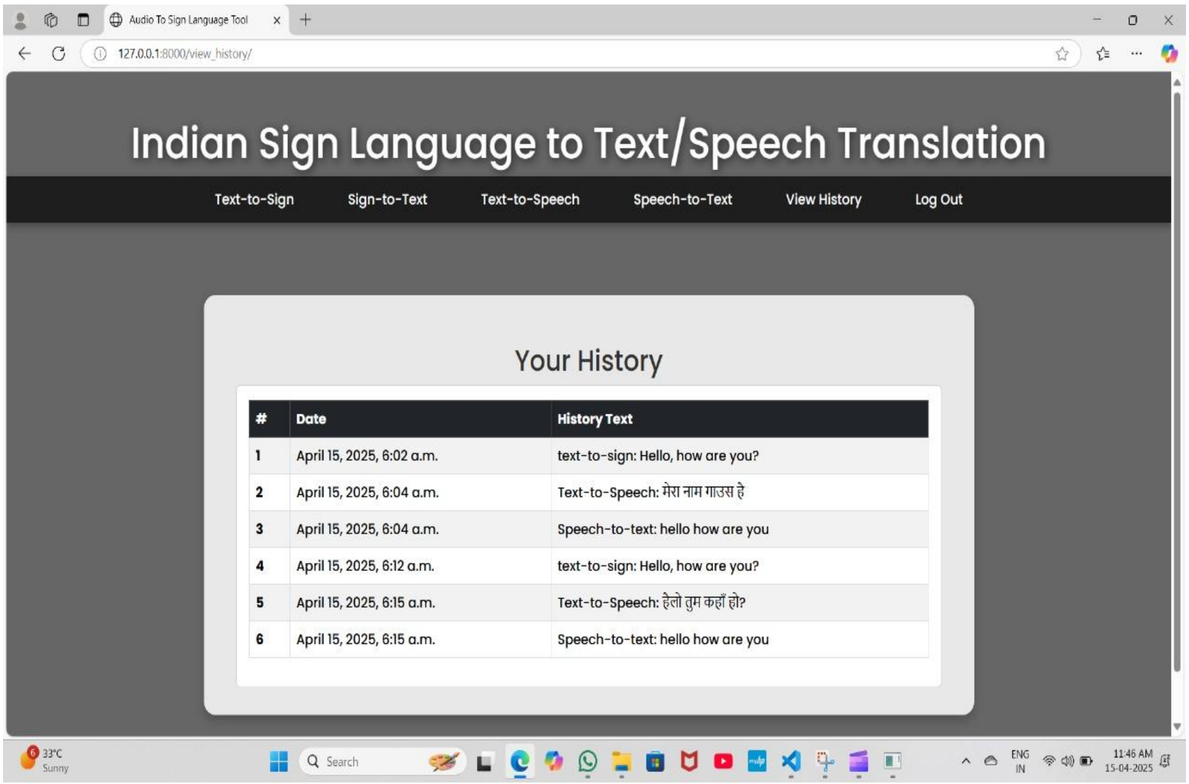


Figure 7.3.8: History Page

8. CONCLUSION

The development and implementation of the Indian sign language to text/speech translation project mark a major step forward in assistive technology and inclusive communication. This innovative system brings together multiple advanced technologies to create a seamless experience for users. Natural Language Processing (NLP) is utilized to understand and process textual input, while speech-to-text conversion allows spoken language to be accurately transcribed. Gesture recognition, powered by MediaPipe, interprets sign language gestures with impressive accuracy. The system also features dynamic sign language animation, enabling clear and intuitive visual representation of translated content. Together, these technologies form a comprehensive platform that supports real-time interaction between deaf or hard of hearing individuals and those who use spoken language. The user interface is designed to be simple and accessible, ensuring that people of varying technical abilities can operate it with ease. By addressing both audio-to-sign and sign-to-audio translation, the system offers a complete communication bridge. This solution is especially valuable in educational, medical, and public service environments where inclusive interaction is crucial. Ultimately, the project plays a vital role in reducing communication barriers and promoting equal access to information and conversation.

8.1. Limitations

Despite its promising capabilities, the Indian sign language to text/speech translation project does have certain limitations. One of the primary challenges lies in the system's handling of regional sign language variations and idiomatic expressions, which may not be accurately interpreted due to limited training data. The accuracy of speech-to-text conversion can also be affected by background noise, varied accents, and unclear pronunciation. Real-time gesture recognition may struggle with complex or fast-paced signs, occasionally resulting in misinterpretation or lag. The database of sign animations, while extensive, still lacks coverage of many less common words and expressions. Non-manual signals such as facial expressions, which are critical in sign language, are not yet fully integrated. The system's reliance on stable internet connectivity can limit its usability in remote or underdeveloped areas. User customization and personalization options are minimal, which may affect the experience for individuals with specific needs.

Furthermore, the system has limited support for languages other than English, restricting its accessibility to a broader audience. Continuous training and updates are necessary to maintain accuracy and relevance. Lastly, the technology requires further optimization to ensure smoother performance across different devices and platform.

8.2. Future Work

Building upon the foundation laid by the current audio-sign and sign-audio translation project, there are several promising avenues for future development and enhancement. These improvements aim not only to refine the existing system but also to broaden its impact and accessibility. A major focus will be the integration of advanced artificial intelligence and machine learning models to enhance the accuracy of speech recognition, particularly in noisy environments or when interpreting varied accents. In addition, the development of more sophisticated algorithms for gesture recognition will enable the system to handle a wider range of sign language dialects and capture subtleties such as facial expressions and non-manual signals. Another important area is the optimization of real-time processing to reduce latency and improve the synchronization between audio, text, and animated sign outputs. This will result in a smoother, more seamless user experience. Moreover, expanding the linguistic reach of the system by including support for multiple spoken languages will enhance its global accessibility. The database of sign language animations will also be continuously expanded to include more regional and idiomatic variations, ensuring that the platform remains inclusive and relevant to the diverse needs of the deaf and hard of hearing community. These future enhancements will significantly strengthen the system's effectiveness, reliability, and scalability.

9. REFERENCES

1. McNeil, M. R. (Ed.). (2016). "Clinical Management of Sensorimotor Speech Disorders." Thieme.

2. Sutton-Spence, R., & Woll, B. (1999). "The Linguistics of British Sign Language: An Introduction." Cambridge University Press.

3. Goldin-Meadow, S., & Mylander, C. (1998). "Spontaneous sign systems created by deaf children in two cultures." Nature.

4. Stokoe, W. C. (2005). "Sign Language Structure: An Outline of the Visual Communication Systems of the American Deaf." Journal of Deaf Studies and Deaf Education.

5. Valli, C., & Lucas, C. (2000). "Linguistics of American Sign Language: An Introduction." Gallaudet University Press.

6. Schick, B., de Villiers, P., de Villiers, J., & Hoffmeister, R. (2007). "Language and Deafness." Jones & Bartlett Learning.

7. Vogler, C., & Metaxas, D. (2001). "A Framework for Recognizing the Simultaneous Aspects of American Sign Language." Computer Vision and Image Understanding.

8. Ladefoged, P., & Maddieson, I. (1996). "The Sounds of the World's Languages." Blackwell.

9. Corina, D. P., & Knapp, H. (2006). "Sign Language Recognition and Translation: A Multidisciplined Approach From the Field of Artificial Intelligence." Journal of Deaf Studies and Deaf Education.

10. Bragg, D., et al. (2019). "Sign Language Recognition, Generation, and Translation: An Interdisciplinary Perspective." The 21st International ACM SIGACCESS Conference on Computers and Accessibility.

11. Ong, S. C. W., & Ranganath, S. (2005). "Automatic Sign Language Analysis: A Survey and the Future beyond Lexical Meaning." IEEE Transactions on Pattern Analysis and Machine Intelligence.

12. Koller, O., Ney, H., & Bowden, R. (2016). "Deep Learning of Mouth Shapes for Sign Language." In Proceedings of the IEEE International Conference on Computer Vision Workshops.

13. Parton, B. S. (2005). "Sign Language Recognition and Translation: A Multidisciplined Approach from the Field of Artificial Intelligence." Journal of Deaf Studies and Deaf Education.

14. Sandler, W., & Lillo-Martin, D. (2006). "Sign Language and Linguistic Universals." Cambridge University Press.

15. Mitchell, R. E., & Karchmer, M. A. (2004). "Chasing the Mythical Ten Percent: Parental Hearing Status of Deaf and Hard of Hearing Students in the United States." Sign Language Studies.

16. Hochberg, I., et al. (1997). "Generation of Virtual Reality with 3-D Animation of Sign Language." Journal of Deaf Studies and Deaf Education.

17. Zhao, L., Kipper, K., Schuler, W., Vogler, C., & Palmer, M. (2000). "A Machine Translation System from English to American Sign Language." Proceedings of the Association for Machine Translation in the Americas.

18. Brennan, M. (1990). "Word Formation in British Sign Language." Stockholm University.

19. Lane, H., Hoffmeister, R., & Bahan, B. (1996). "A Journey into the Deaf-World." DawnSignPress.

20. Liddell, S. K., & Johnson, R. E. (1989). "American Sign Language: The phonological base." Sign Language Studies.

.

Appendix - A: Plagiarism Report

28% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text

Match Groups

- **341** Not Cited or Quoted 28%
  Matches with neither in-text citation nor quotation marks
- **5** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

- 10% Internet sources
- 9% Publications
- 26% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Appendix – B: Time and cost analysis of your project work

Time Analysis

Month 1: Requirement Gathering & Planning

Activities Involved:

- Understanding user requirements, accessibility needs, and defining the system's scope.
- Researching assistive technology tools, NLP, speech APIs, gesture recognition models.
- Identifying modules: speech-to-text, text-to-sign, sign-to-text, TTS.
- Outlining feasibility, project goals, and high-level architecture.

Estimated Time: 2 weeks

- 1 week: Requirement gathering and research
- 1 week: Feasibility study and architecture design

Month 1: Dataset Preparation & Preprocessing

Activities Involved:

- Collecting or curating a dataset of Indian Sign Language gestures (via open datasets or videos).
- Preprocessing: frame extraction, resizing, normalization.
- Augmenting gesture videos/images (flip, rotate, crop) to improve model robustness.
- Preparing text datasets for NLP (grammar rules, sentence correction).

Estimated Time: 2 weeks

- 1 week: Dataset collection & cleaning
- 1 week: Data preprocessing & augmentation

Month 2: Model Development

Activities Involved:

- Implementing gesture recognition using MediaPipe + CNN (or ViT).
- Creating NLP pipeline using NLTK for text parsing, grammar correction, lemmatization.
- Integrating Speech-to-Text (Google API) and Text-to-Speech (gTTS/pyttsx3).
- Testing and fine-tuning model for accuracy and real-time inference.

Estimated Time: 4 weeks

- 2 weeks: Gesture & speech model development
- 1 week: NLP module development
- 1 week: Testing & hyperparameter tuning

Month 3: Backend Development

Activities Involved:

- Building Flask/Django backend for API routing and model serving.
- Handling real-time webcam/audio input and output streaming.
- Implementing secure login/authentication (optional).
- Integrating model outputs into JSON responses.

Estimated Time: 2 weeks

Month 3: Frontend Development

Activities Involved:

- Designing UI for user inputs: webcam, audio, and text.
- Implementing sign animation/visualization component.
- Real-time display of translation results (signs, text, speech).
- Cross-platform support for desktop/mobile.

Estimated Time: 3 weeks

- 1 week: UI/UX design
- 2 weeks: Integration with backend APIs

Month 4: Testing & Debugging

Activities Involved:

- Unit testing, integration testing, and gesture recognition validation.
- Cross-browser and device testing.
- Ensuring gesture prediction accuracy and model reliability.
- Fixing bugs in interface and model predictions.

Estimated Time: 3 weeks

- 2 weeks: Testing
- 1 week: Debugging

Month 4: Deployment & Documentation

Activities Involved:

- Deploying app using Render, Heroku, or local host.
- Writing final project report, user manual, and presentation slides.
- Creating demo videos and tutorials.

Estimated Time: 2 weeks

- 1 week: Deployment
- 1 week: Documentation

Cost Analysis

Development Tools & IDE

Details:

Using open-source tools like VS Code, PyCharm (Community Edition), Flask, TensorFlow, PyTorch, MediaPipe, OpenCV, and NLTK.

Estimated Cost: ₹0

Hosting Server

Details:

Deploying on free-tier cloud platforms (Render, Heroku, AWS Free Tier, or GitHub Pages for frontend).

Estimated Cost: ₹0

APIs & Libraries

Details:

Google Speech API (free tier), NLTK, gTTS, MediaPipe, OpenCV — all have free access tiers suitable for academic use.

Estimated Cost: ₹0

Database

Details:

Using SQLite (for local) or MongoDB Atlas (free tier) for storing gesture mappings, user profiles, and logs.

Estimated Cost: ₹0

Miscellaneous Expenses

Details:

Electricity, internet, and hardware use assumed to be covered under academic environment.

Estimated Cost: ₹0

Appendix – C: Journal Certificate


International Journal of Scientific Development and Research
IJSDR.ORG | ISSN : 2455-2631
An International Open Access, Peer-reviewed, Refereed Journal

Certificate of Publication

The Board of
International Journal of Scientific Development and Research
Is hereby awarding certificate to
Paturu Sandeep
In recognition of the publication of the paper entitled
Indian Sign Language to text/speech translation
Published in Volume 10 Issue 4, April-2025, | Impact Factor: 9.15 by Google Scholar
Co-Authors - Syamtan Sai Yadav Bontha, Pinjari Dastagiri, Tharugu Soma Shekar Reddy, P Yasmin bhanu

Paper ID - IJSDR2504070
Registration ID - 301565

ISSN 2455-2631 IJSDR

Editor-In Chief

IJSDR | ISSN : 2455-2631

An International Scholarly, Open Access, Multi-disciplinary, Monthly, Indexing in all Major Database & Metadata, Citation Generator
IJSDR - International Journal of Scientific Development and Research | ESTD: 2016
www.ijsdr.org | editor@ijsdr.org

International Journal of Scientific Development and Research
IJSDR.ORG | ISSN : 2455-2631
An International Open Access, Peer-reviewed, Refereed Journal

Certificate of Publication

The Board of

International Journal of Scientific Development and Research

Is hereby awarding certificate to

Pinjari Dastagiri

In recognition of the publication of the paper entitled

Indian Sign Language to text/speech translation

Published in Volume 10 Issue 4, April-2025, | Impact Factor: 9.15 by Google Scholar

Co-Authors - Syamtan Sai Yadav Bontha, Paturu Sandeep, Tharugu Soma Shekar Reddy, P Yasmin bhanu

Paper ID - IJSDR2504070
Registration ID - 301565

ISSN 2455-2631 IJSDR

Editor-In Chief

IJSDR | ISSN : 2455-2631

An International Scholarly, Open Access, Multi-disciplinary, Monthly, Indexing in all Major Database & Metadata, Citation Generator

IJSDR - International Journal of Scientific Development and Research | ESTD: 2016

www.ijsdr.org | editor@ijsdr.org

International Journal of Scientific Development and Research

IJSDR.ORG | ISSN : 2455-2631

An International Open Access, Peer-reviewed, Refereed Journal

Certificate of Publication

The Board of

International Journal of Scientific Development and Research

Is hereby awarding certificate to

Syamtan Sai Yadav Bontha

In recognition of the publication of the paper entitled

Indian Sign Language to text/speech translation

Published in Volume 10 Issue 4, April-2025, I Impact Factor: 9.15 by Google Scholar

Co-Authors - Pinjari Dastagiri, Paturu Sandeep, Tharugu Soma Shekar Reddy, P Yasmin bhanu

Paper ID - IJSDR2504070
Registration ID - 301565

ISSN 2455-2631 IJSDR

Editor-In Chief

IJSDR | ISSN : 2455-2631

An International Scholarly, Open Access, Multi-disciplinary, Monthly, Indexing in all Major Database & Metadata, Citation Generator

IJSDR - International Journal of Scientific Development and Research | **ESTD: 2016**

www.ijsdr.org | editor@ijsdr.org

International Journal of Scientific Development and Research
IJSDR.ORG | ISSN : 2455-2631
An International Open Access, Peer-reviewed, Refereed Journal

Certificate of Publication

The Board of

International Journal of Scientific Development and Research

Is hereby awarding certificate to

Tharugu Soma Shekar Reddy

In recognition of the publication of the paper entitled

Indian Sign Language to text/speech translation

Published in Volume 10 Issue 4, April-2025, | Impact Factor: 9.15 by Google Scholar

Co-Authors - Syamtan Sai Yadav Bontha, Pinjari Dastagiri, Paturu Sandeep, P Yasmin bhanu

IJSDR | ISSN : 2455-2631

Paper ID - IJSDR2504070
Registration ID - 301565

ISSN 2455-2631 IJSDR
International Journal of Scientific Development and Research

Editor-In Chief

An International Scholarly, Open Access, Multi-disciplinary, Monthly, Indexing in all Major Database & Metadata, Citation Generator
IJSDR - International Journal of Scientific Development and Research | ESTD: 2016
www.ijsdr.org | editor@ijsdr.org

Appendix – D: Journal Paper

Indian Sign Language to Text/Speech Translation

Mrs. P. YASMINBHANU, Assistant Professor, Department of CA, School of Computing,
Mohan Babu University (Erstwhile
Sree Vidyanikethan Engineering College
(Autonomous)), Tirupati,
Andhra Pradesh, India,
yasminbhanu33@gmail.com

Mr. SYAMTAN SAI YADAV BONTHA
UG Scholar, Department of BCA
Mohan Babu University (Erstwhile
Sree Vidyanikethan Engineering College
(Autonomous)) Tirupati,
Andhra Pradesh, India,
syamtansaiyadavedu@gmail.com

Mr. PINJARI DASTAGIRI
UG Scholar, Department of BCA
Mohan Babu University (Erstwhile
Sree Vidyanikethan Engineering College
(Autonomous)) Tirupati,
Andhra Pradesh, India,
pinjaridastagiribasha@gmail.com

Mr. PATURU SANDEEP,

UG Scholar, Department of BCA,
Mohan Babu University (Erstwhile
Sree Vidyanikethan Engineering College
(Autonomous)), Tirupati,
Andhra Pradesh, India,
sandeepitis206@gmail.com

Mr. THARUGU SOMA SHEKAR REDDY
UG Scholar, Department of BCA
Mohan Babu University (Erstwhile
Sree Vidyanikethan Engineering College
(Autonomous)), Tirupati,
Andhra Pradesh, India,
somushekar3005@gmail.com

Abstract—The Indian Sign Language to Text or Speech Translation project represents a groundbreaking integration of technology in the field of communication and accessibility. Through this system, ISL communication connects to verbal or written expressions for continuous dialogue. The system functions as an assistive tool for deaf and hard-of-hearing people because it generates real-time text or speech output from sign language inputs. The system facilitates diverse interaction between users of sign language and individuals who do not speak that language. Renewed Indian Sign Language (ISL) translation requires advanced technological functions such as natural language processing (NLP) and computer vision along with machine learning operations for precise translation. Deep learning models, including CNNs and transformers, enhance gesture recognition and speech synthesis. By incorporating real-time processing, the project ensures a smooth and natural user experience. It also contributes to language accessibility, making education and daily interactions more inclusive. The use of AI-driven models enables continuous learning and improvement for higher accuracy. This creative approach shows a notable development that helps better communication and supports a more linked society.

Keywords—Accessibility Technology, Sign Language Interpretation, Audio Processing, Speech-to-Text Conversion, Natural Language Processing (NLP), Computer Vision Deep Learning, Artificial Intelligence.

INTRODUCTION

Among people or groups, communication is a vital tool for expressing ideas and feelings. Good communication promotes good thinking and helps to move things forward. Communication depends on language; it includes not just words but also deeds. Deaf or mute people use sign language to express themselves using physical movement and hand gestures. Many people lack sign language knowledge, which makes it difficult for those who are deaf, hearing impaired, or speech unable to communicate and convey ideas. This issue creates a barrier between those who are deaf or mute and others.

The challenge requires a strong tool called a sign language recognition system which has prompted numerous ongoing research projects about this field. Good for society. Technological developments are always advancing in this competitive climate, hence this interpreter is vital in offering equal chances for everyone, regardless of their handicap.

This world is home to many languages, each with unique regional variations including sign language, which also varies depending on the local language. While the discussion takes place in English, The research adopts American Sign Language (ASL) as its main communication method. The recognition of sign language exists in two different types: static and dynamic. The research adopts static sign language for its methodologies. This means the data is shown as pictures, and it monitors hand motions using a good hand-tracking system. [1]. This system detects hand movements in real time from camera capture.

With data collected and supplied to a deep learning model, this system uses computer vision and natural language processing to provide real-time predictions.

LITERATURE SURVEY

[1] Proposed hand gesture detection utilizing the Karhunen-Loeve (K-L) transform, in conjunction with convolutional neural networks (CNN). The system used skin filtering then separated the palm area through palm cropping and finalised the process with edge detection for hand detection purposes. Subsequently, feature extraction of the hand was performed utilizing the K-L transform approach, followed by picture classification employing Euclidean distance. They evaluated 10 distinct hand motions with an accuracy of 96%.

[2] Proposed Contour tracing characteristics help to identify single-handed sign language actions. This paper shows that The system extracts hand outlines by separating them from background elements through skin colour identification in RGB along with YCbCr colour domains, together with threshold intensities of grey levels. Gesture outlines detected by segmentation were traced using the contour tracing descriptor. To evaluate the accuracy, they used picture categorization using KNN and SVM-supervised machine learning techniques.

[3] Identification of hand gestures based on Principal Component Analysis (PCA). The system applies hand vein recognition through colour models with thresholding methods and fast template matching that enables efficient hand verification identification. Portable skin colour modelling in the YCbCr colour space defines recognition. The foreground and background are separated using Otsu thresholding. Gesture recognition uses template matching based on Principal Component Analysis (PCA). For low-brightness pictures, the algorithm achieved 91.43% accuracy.

[4] Suggested deep convolutional neural networks for letters and numbers proposed American Sign Language gesture identification. This paper addresses the preparation of images, whereby a background removal technique was used to remove the backdrop. Using CNN for image categorization, The dataset was split into two parts: one for training and the other for testing. Concerning the motions of the alphabet, the system was 82.5% accurate.

[5] Proposed Suggested examination of plans for language and hand gesture identification. Executing data collecting and pre-processing, The system used median and Gaussian filters to reduce noise whereas it applied histogram equalization to clear unnecessary data in morphological approaches. It then segmented, tracking for hand detection and skin colour segmentation included. Culminating in picture categorization, the following stage was feature extraction using several approaches. This talk provides a comprehensive overview of the subject of automatic gesture and language recognition.

[6] Proposed dynamic sign language recognition tool. For image categorization, forecasting, and identification, they used a supervised learning technique called SVM. This system triggers sign gesturing detection from live video stream capture. The author utilizes darkened images and isolated white hand borders to achieve their goal. Pulls hand outlines from video frames. This boundary defines the hand's contours.

[7] Proposed Deep learning methods help to identify static signs in sign language. Hand identification using this approach is based on skin colour modelling; it uses a defined skin colour range to isolate hand pixels (foreground) from non-pixels (background). Using images with a consistent background, the approach classified images using a CNN. In American Sign Language alphabet recognition, the system achieved 90.04% accuracy; in testing accuracy, 93.67%.

[8] Suggesting hand gesture identification for static images using convolutional neural networks (CNN). The system contains photo pre-processing methods which include Morphological operations, contour extraction, polygon approximation and segmentation. The research team employed CNN architectures for extracting photos' characteristics before performing classification to obtain results from multiple CNN designs.

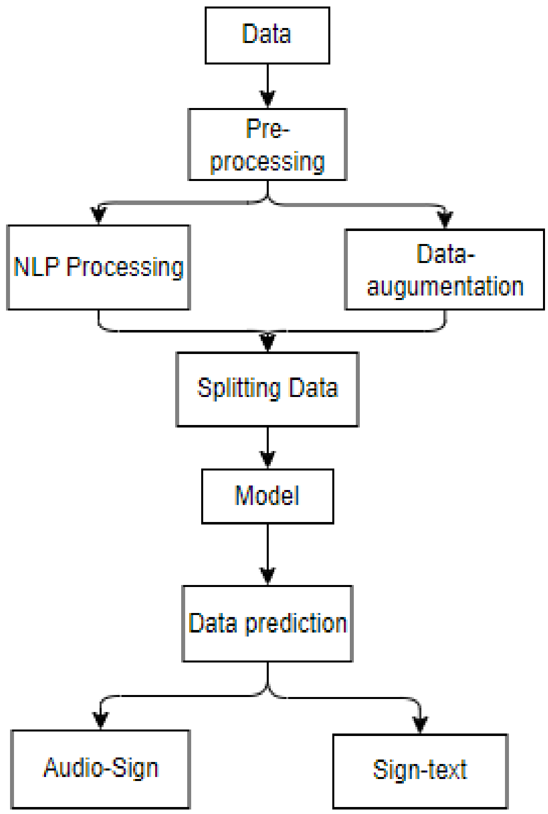
[9] Proposed The authors designed a sign language recognition system combining CNN with computer vision technology. [9] HSV colour technology functions as the detection algorithm for hand gestures while the researchers designated the background to show black colour. The image pre-processing involves conversion to grayscale then dilation and mask operation followed by gesture segmentation. The CNN architecture was used for picture categorization, and then feature extraction in the first layer. The system correctly identified 10 letters with a 90% accuracy.

[10] Stokoe's 2005 study highlights the linguistic structure of American Sign Language (ASL). He identifies key phonological components—handshape, location, and movement—comparable to phonemes in spoken languages. His work proves ASL is a fully structured language with its grammar. The study emphasizes ASL's visual-spatial nature, differing in modality but not complexity. It was instrumental in changing perceptions of sign languages. Stokoe's research remains a cornerstone in sign linguistics and deaf studies.

PROPOSED SYSTEM

The proposed system aims to develop a real-time translation tool that converts spoken language to sign language and vice versa using advanced NLP, AI, and computer vision technologies. Unlike traditional approaches that rely on pre-recorded videos, this system dynamically interprets and generates sign language using MediaPipe for gesture recognition and NLP techniques for accurate text translation. This enhances promoting communication for the deaf and hard-of-hearing community.

PROJECT FLOW



SYSTEM DESIGN

The system design is structured to handle a comprehensive pipeline from data ingestion to actionable insights. It starts with data collection, where raw data from various sources, such as text, audio, and visual inputs, is aggregated. The preprocessing phase involves cleaning, normalizing, and transforming this data to ensure consistency and quality. For textual data, NLP processing is employed to tokenize, lemmatize, and extract meaningful features, enabling the system to understand and process human language effectively.

Then, methods of data augmentation are used to strengthen the quality of the dataset., particularly useful in scenarios with limited data. This step involves generating synthetic data or modifying existing data to improve model generalization. The data is subsequently split into training, validation, and testing sets to facilitate a structured approach to model development and evaluation.

The model design phase focuses on selecting and fine-tuning appropriate algorithms, whether they are traditional machine learning models or advanced deep learning architectures, depending on the task's complexity. Once the model is trained and validated, it is deployed for data prediction, where it processes new inputs to generate accurate outputs.

For multimodal data, such as audio and sign language, the system incorporates specialized modules. Audio signals are converted into text using speech recognition techniques, while sign language gestures are interpreted into textual representations through computer vision and gesture recognition models. This integration ensures the system can handle diverse data types seamlessly.

The final design emphasizes scalability, accuracy, and efficiency, ensuring that the system can adapt to varying data volumes and types while maintaining high performance. Continuous monitoring and updates are incorporated to keep the system relevant and effective in dynamic environments.

IMPLEMENTATION

Index Page

Provides an overview of the Indian Sign Language to Text/Speech Translation System and navigation options to other pages such as Registration, Login, User Dashboard, and Information Sections.

Registration Page

Allows users to sign up for an account by providing the required details.

Login Page

Enables registered users to log in using their credentials, hence granting access to their tailored dashboard and features.

User Home Page

The main dashboard for registered users offers access to key features such as text translation, sign language recognition, and accuracy metrics.

Text Processing & NLP Page

Handles text input processing using NLP techniques like tokenization, lemmatization, and grammar correction before conversion into sign language or speech.

Speech-to-Text Page

Allows users to record or upload audio, which is converted into text using speech recognition APIs, helping to link sign language and spoken language.

Sign Language Recognition Page

Captures live video input and uses computer vision models (MediaPipe, CNN, ViT, GRU) to detect hand gestures and facial expressions for sign language translation.

Sign Language Animation Page

Displays sign language animations corresponding to the processed text, using a gesture library and real-time rendering techniques.

Accuracy Page

Displays performance metrics and visualizations of the gesture recognition and NLP models, including translation accuracy, error rates, and processing speed.

Prediction Page

Uses machine learning models to predict text, speech, and sign gestures, ensuring real-time and accurate translation results.

Logout Page

Allows users to securely log out, terminating active sessions and protecting their account data and privacy.

SYSTEM STUDY AND TESTING

The feasibility study ensures that the proposed system is practical and beneficial. It includes economic feasibility, which evaluates whether the project stays within budget, utilizing free technologies except for necessary customizations. Technical feasibility ensures that the system does not overburden existing resources, requiring minimal changes for implementation. Social feasibility assesses user acceptance, ensuring proper training and a positive experience to encourage adoption. System testing is conducted to identify and fix errors, verifying that the software meets requirements and functions as expected.

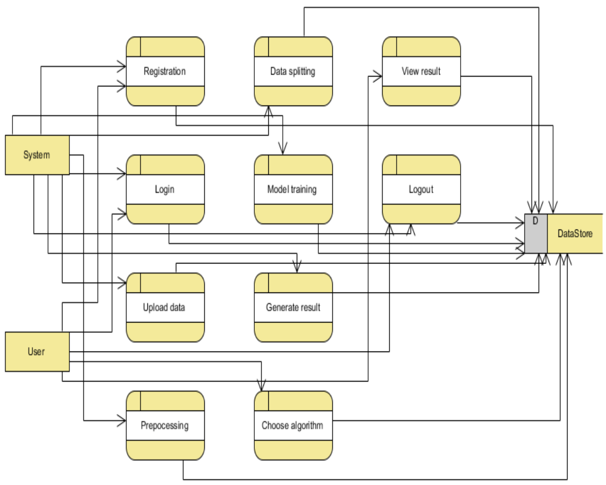
Test cases :

S. NO	Test cases	I/O	Expected O/T	Actual O/T	P/F
1	Read the dataset.	Dataset path.	Dataset need to read successfully.	Dataset fetched successfully.	P
2	Performing pre-processing on the dataset	Pre-processing part takes place	Pre-processing should be performed on dataset	Pre-processing successfully completed.	P
3	Model Building	Model Building for the clean data	Need to create model using required algorithms	Model Created Successfully.	P
4	Text-sign Sign-Text	Input image provided.	Output should be the Sign and text	Model successful	P

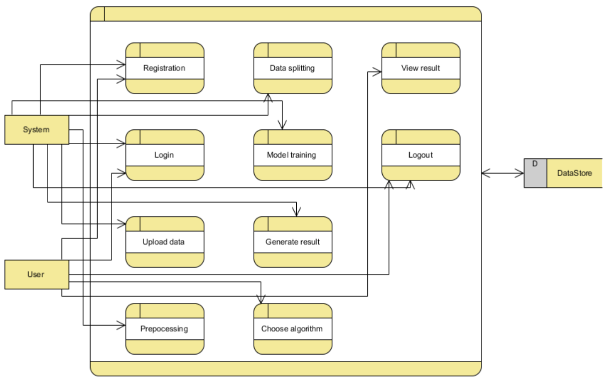
DATAFLOW DIAGRAM

Information flow within a system is typically depicted through the use of a data flow diagram. DFD a well-structured and Visually organised DFD can effectively illustrate a substantial portion of the system requirements. The process can be either manually automated or a combination of both methods. This illustrates how data flows in and out of the system, what modifies the data and where data is kept the primary objective of ADFD is to visually represent the overall scope Unlimited of a system. The tool facilitates communication between system analysts and any individual involved in the system, serving as a launchpad for system reconfiguration.

Level 1 diagram :



Level 2 diagram :



CONCLUSION

The development and implementation of the audio-sign and sign-audio translation project represent a significant development in communication access and assistive technologies a significant advancement in the field of assistive technology and communication accessibility. This project successfully integrates several complex technologies, including natural language processing (NLP), speech-to-text conversion, gesture recognition using Media Pipe, and dynamic sign language animation. The result is a versatile and user-friendly platform that facilitates real-time interaction between non-users and sign language users bridging a critical gap in interpersonal interactions.

FUTURE ENHANCEMENT

Future advancements in the Indian Sign Language to Text/Speech Translation System will focus on improving its intelligence, speed, and inclusivity. AI-driven enhancements will refine speech recognition to better handle background noise, diverse accents, and variations in tone. Gesture recognition models will be trained on larger datasets, allowing the system to identify complex sign language dialects and non-manual signals like facial expressions, body posture, and movement speed, which are crucial for accurate interpretation. Real-time processing improvements will minimize latency, ensuring instant translation between sign language, text, and speech for a natural and fluid user experience. The sign animation module will be upgraded with 3D avatars that replicate human signing with lifelike movements and expressions, making digital communication more engaging. Expanding language and dialect support, the system will accommodate regional variations of sign language and provide translations in multiple spoken languages, bridging communication gaps worldwide. To increase accessibility, mobile and wearable device integration will be explored, allowing users to interact through smartphones, AR glasses, or IoT devices. Further, cloud-based processing will be introduced to enable scalability and real-time global access. Lastly, user customization options will be developed, allowing individuals to train the system with personalized gestures and speech patterns for more adaptive and user-friendly interactions. These enhancements will push the boundaries of assistive technology, fostering greater inclusivity, independence and social inclusion for those who are hard of hearing or deaf.

REFERENCES

1. Liddell, S. K., & Johnson, R. E. (1989). "American Sign Language: The phonological base." Sign Language Studies.
2. McNeil, M. R. (Ed.). (2016). "Clinical Management of Sensorimotor Speech Disorders." Thieme.
3. Sutton-Spence, R., & Woll, B. (1999). "The Linguistics of British Sign Language: An Introduction." Cambridge University Press.
4. Goldin-Meadow, S., & Mylander, C. (1998). "Spontaneous sign systems created by deaf children in two cultures." Nature.
5. Stokoe, W. C. (2005). "Sign Language Structure: An Outline of the Visual Communication Systems of the American Deaf." Journal of Deaf Studies and Deaf Education.
6. Valli, C., & Lucas, C. (2000). "Linguistics of American Sign Language: An Introduction." Gallaudet University Press.
7. Schick, B., de Villiers, P., de Villiers, J., & Hoffmeister, R. (2007). "Language and Deafness." Jones & Bartlett Learning.
8. Vogler, C., & Metaxas, D. (2001). "A Framework for Recognizing the Simultaneous Aspects of American Sign Language." Computer Vision and Image Understanding.
9. Ladefoged, P., & Maddieson, I. (1996). "The Sounds of the World's Languages." Blackwell.
10. Corina, D. P., & Knapp, H. (2006). "Sign Language Recognition and Translation: A Multidisciplined Approach From the Field of Artificial Intelligence." Journal of Deaf Studies and Deaf Education.

Appendix – E: Sample Source Code:

1.about.html

```
 {% extends 'base.html' %}

{% block navbar %}

  <li class="li"><a href="{% url 'home' %}">Home</a></li>

  <li class="li"><a href="{% url 'signup' %}">Sign Up</a></li>

  <li class="li"><a href="{% url 'login' %}">Log In</a></li>

  <li class="li"><a class="active" href="{% url 'about' %}">About</a></li>

{% endblock %}

{% block content %}

  <div class="about-section" style="max-width: 800px; margin: 50px auto; padding: 20px; background-color: rgba(255, 255, 255, 0.9); border-radius: 10px; box-shadow: 0 4px 12px rgba(0, 0, 0, 0.1);">

        <h2 style="text-align: center; color: #4CAF50; margin-bottom: 20px;">About the Project</h2>

        <hr>

        <h3 style="color: #333;">Audio to Sign Language Conversion</h3>

        <p style="font-size: 1.1em; line-height: 1.6; color: #333;">

                The first component of this tool involves converting audio input into sign language. Initially, speech recognition technology is utilized to capture and transcribe spoken words into written text. This process forms the basis for the subsequent conversion into sign language. However, translating text into sign language is complex, as sign language has its own unique grammar and syntax, differing significantly from spoken language.

        </p>
```

<p style="font-size: 1.1em; line-height: 1.6; color: #333;">

Machine learning models, especially those trained on extensive sign language datasets, are employed to interpret the text and generate corresponding sign language gestures. The final step is the visualization of these gestures, typically done using a 3D avatar or animated character.

This avatar performs the sign language gestures, and it is crucial that these animations are fluid and accurate to ensure effective communication.

</p>

<hr>

<h3 style="color: #333;">Sign Language to Text Conversion</h3>

<p style="font-size: 1.1em; line-height: 1.6; color: #333;">

The second component reverses the process by converting sign language into text. This begins with capturing sign language gestures through a camera. OpenCV, a powerful tool for image processing and computer vision tasks, is used to detect and analyze the sign language gestures.

</p>

<p style="font-size: 1.1em; line-height: 1.6; color: #333;">

Machine learning algorithms, particularly Convolutional Neural Networks (CNNs) trained on sign language datasets, are employed to recognize and interpret these gestures. The recognized gestures are then converted back into text, providing a written transcription of the sign language.

</p>

<hr>

</div>

```
{% endblock %}
```

2. animation.html

```
        {% extends 'base.html' %}

{% load static %}


{% block navbar %}

        <!-- <li class="li"><a class="active" href="#">View Profile</a></li> -->

        <li class="li"><a href="{% url 'animation' %}">Text-to-Sign</a></li>

    <li class="li"><a href="{% url 'sign' %}">Sign-to-Text</a></li>

    <li class="li"><a href="{% url 'upload' %}">Text-to-Speech</a></li>


 <li class="li"><a href="{% url 'mic' %}">Speech-to-Text</a></li>

    <li class="li"><a href="{% url 'view_history' %}">View History</a></li>

    <li class="li"><a href="{% url 'logout' %}">Log Out</a></li>

{% endblock %}


{% block content %}

<div class="split left" style="padding: 20px;">

    <h2 style="text-align: center;">Enter Text or Use Mic</h2>

    <br>
```

```
    <form action="" method="post" style="text-align: left;">

        {% csrf_token %}



        <input type="text" name="sen" class="mytext" id="speechToText" placeholder="Enter
text or use mic" style="width: 80%; padding: 10px; margin-right: 10px;">

        <button type="button" name="button" class="mic" onclick="record()"
style="background: none; border: none;">

            <img src="{% static 'mic3.png' %}" height="32px" width="38px" />

        </button>

        <br><br>



        <input type="submit" name="submit" class="submit" style="padding: 10px 20px;
background-color: #39c9bd; color: white; border: none; cursor: pointer;">

    </form>

    <br>



  {% if text %}

      <table cellspacing="20px">



          <tr>

              <td class="td">The text that you entered is:</td>

              <td class="td">{{ text }}</td>
```

```
  </tr>

        <tr>

          <td class="td">Key words in sentence:</td>

          <td class="td">

            <ul class="td" id="list" style="list-style-type: none; padding: 0; text-align:
center;">


  {% for word in words %}

              <li id="{{ forloop.counter }}" style="display: inline-block; margin-right:
8px;">{{ word }}</li>

              {% endfor %}

            </ul>

          </td>

        </tr>

      </table>

    {% endif %}

</div>

{% if text %}

      <div class="split right" style="padding: 20px;">

            <h2 style="text-align: center;">Sign Language Animation</h2>

            <div style="text-align: center;">
```

```
                <button class="submit" onclick="playPause()" style="padding: 10px 20px; background-color: #39c9bd; color: white; border: none; cursor: pointer;">Play/Pause</button>

                <br><br>

                <video id="videoPlayer" style="max-width: 100%; height: auto;" preload="auto" autoplay>

                    <source src="" type="video/mp4">

                    Your browser does not support HTML5 video.

                </video>

            </div>

        </div>

{% endif %}


<script>

    // WebkitSpeechRecognition API for speech-to-text conversion

    function record(){

        const recognition = new webkitSpeechRecognition();

        recognition.lang = 'en-IN';



        recognition.onresult = function(event){

            console.log(event)

            document.getElementById('speechToText').value = event.results[0][0].transcript;

        }
```

```
        recognition.start();

    }

    function play() {

        const videoSource = [];

        const videos = document.getElementById("list").getElementsByTagName("li")



        for (let j = 0; j < videos.length; j++) {

            videoSource[j] = "/static/" + videos[j].innerHTML + ".mp4";

        }

        let i = 0;

        const videoCount = videoSource.length;



        function videoPlay(videoNum) {

            document.getElementById("list").getElementsByTagName("li")[videoNum].style.color =
"#09edc7";


document.getElementById("list").getElementsByTagName("li")[videoNum].style.fontSize =
"xx-large";

            document.getElementById("videoPlayer").setAttribute("src", videoSource[videoNum]);

            document.getElementById("videoPlayer").load();

            document.getElementById("videoPlayer").play();
```

```
}

document.getElementById('videoPlayer').addEventListener('ended', myHandler, false);


        videoPlay(0);

        function myHandler() {


            document.getElementById("list").getElementsByTagName("li")[i].style.color =
    "#feda6a";


    document.getElementById("list").getElementsByTagName("li")[i].style.fontSize = "20px";

            i++;

            if (i < videoCount) {

                videoPlay(i);

            } else {

                document.getElementById("videoPlayer").pause();

            }

        }

    }

    function playPause(){

        const videoPlayer = document.getElementById("videoPlayer");
```

```
  if (videoPlayer.paused) {
            play();
        } else {
            videoPlayer.pause();
        }
    }

</script>

{% endblock %}
```

3. home.html

```
        {% extends 'base.html' %}
{% load static %}
{% block navbar %}
        <li class="li"><a class="active" href="{% url 'home' %}">Home</a></li>
        <li class="li"><a href="{% url 'signup' %}">Sign Up</a></li>
        <li class="li"><a href="{% url 'login' %}">Log In</a></li>
        <li class="li"><a href="{% url 'about' %}">About</a></li>
```

```
{% endblock %}




{% block content %}

    <div class="form-style" style="display: flex; flex-direction: column; justify-content: center; align-items: center; height: 80vh; text-align: center;">



        <h1 style="font-size: 3em; margin-bottom: 20px; color: #4CAF50; text-shadow: 2px 2px 10px rgba(0, 0, 0, 0.3);">Tranforming Communication: Text,Sign,and Speech Made Easy!</h1>

        <h3 style="font-size: 1.5em; margin-bottom: 40px; color: black;">Sign to Text to Speech to Text to Sign</h3>

        <a href="{% url 'animation' %}">



            <button class="button" style="padding: 15px 30px; font-size: 18px; background-color: #4CAF50; border: none; border-radius: 8px; color: white; cursor: pointer; transition: 0.3s;">

                Click to Start

            </button>

        </a>

    </div>

{% endblock %}
```

CO-PO-PSO mapping Table

COURSE DESCRIPTION: Identification of topic for the project work; Literature survey; Collection of preliminary data; Identification of implementation tools and methodologies; Performing critical study and analysis of the topic identified; Time and cost analysis; Implementation of the project work; Preparation of thesis and presentation.

COURSE OUTCOMES: After successful completion of this course, the students will be able to:

CO1. Create/Design computer science engineering systems or processes to solve complex computer science engineering and allied problems using appropriate tools and techniques following relevant standards, codes, policies, regulations and latest developments.

CO2. Consider society, health, safety, environment, sustainability, economics and project management in solving complex computer science engineering and allied problems.

CO3. Perform individually or in a team besides communicating effectively in written, oral and graphical forms on computer science engineering systems or processes.

CO-PO-PSO Mapping Table:

Course Outcome	**Program Outcomes**											**Program Specific Outcomes**				
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	3	3	3	3	3			3				3				
CO2						3	3				3					
CO3									3	3						

Correlation Level: 3-High; 2-Medium; 1-Low

Short Bio-data of the Students

Name of the Student:	Paturu Sandeep.
Roll Number:	22102C010247.
Date of Birth:	20-06-2004.
Qualification:	BCA.
Phone number:	+91 6301087844.
Mail ID:	sandeepitis206@gmail.com
Address:	Tirupati.

Name of the Student:	Pinjari Dastagiri.
Roll Number:	22102C010255.
Date of Birth:	21-11-2003.
Qualification:	BCA.
Phone number:	+91 6300352686.
Mail ID:	pinjaridastagiribasha@gmail.com
Address:	Kurnool.

Name of the Student:	Syamtan Sai Yadav Bontha.
Roll Number:	22102C010319.
Date of Birth:	05-05-2005.
Qualification:	BCA.
Phone number:	+91 7981833192.
Mail ID:	syamtansaiyadavedu@gmail.com
Address:	Tirupati.

Name of the Student:	Tharugu Soma Shekar Reddy.
Roll Number:	22102C010329.
Date of Birth:	14-10-2006.
Qualification:	BCA.
Phone number:	+91 9390754621.
Mail ID:	somushekar3005@gmail.com
Address:	Tirupati.