

# A Real-Time Avatar Animation System for Deaf People

Srushti D. Shinde<sup>1</sup>, Shreya M. Gade<sup>2</sup>, Snehal U. Gutte<sup>3</sup>, Samruddhi D. Pawar<sup>4</sup>,  
Prof. Gauri K. Kharat<sup>5</sup>

Students, Department of Computer Technology<sup>1,2,3,4</sup>

Lecturer, Department of Computer Technology<sup>5</sup>

Sou. Venutai Chavan Polytechnic, Pune, Maharashtra, India

**Abstract:** *Communication barriers between the general public and hearing-impaired people continue to be a significant problem, particularly in situations where sign language proficiency is low. Current options, such as speech-to-sign translators and 3D avatar systems, frequently encounter problems including high computing costs, little datasets, and poor real-time response.*

*This study describes a real-time, lightweight web-based Django system that uses fingerspelling to translate text into sign language. The approach ensures speed and clarity while lowering complexity by displaying each character as a sign image. It enhances accessibility for a variety of users by supporting both ASL and ISL.*

*With features like OTP authentication and input validation, the platform prioritizes security, usability, and real-time performance. It provides a scalable and useful solution for communication and education, while being reliant on static graphics. Future improvements could incorporate facial expressions, 3D animations, and NLP and machine learning-based speech-to-sign integration...*

**Keywords:** Text-to-sign translation , Sign image representation , Indian Sign Language (ISL) ,American Sign Language (ASL)

## I. INTRODUCTION

Despite the fact that communication is a basic human need, people with hearing impairments nevertheless encounter many difficulties, especially in areas where the general public is not very proficient in sign language. There are significant obstacles in social interaction, education, healthcare, and work when non-sign language users are unable to interact with deaf and hard-of-hearing people. Isolation, a lack of possibilities, and a lower quality of life are frequently the outcomes of this communication gap. While sign languages such as American Sign Language (ASL) and Indian Sign Language (ISL) are rich, structured, and expressive modes of communication, their usage remains largely confined to the deaf community, thereby restricting seamless interaction with the broader society.

Rapid developments in assistive technology in recent years have sparked the creation of creative solutions meant to close this gap. To improve communication, systems including 3D avatar-driven animation platforms, vision-based recognition models, and speech-to-sign translators have been proposed. These systems use a variety of computational methods, such as Natural words Processing (NLP), computer vision, and motion capture technologies, to try and translate spoken or written words into sign language representations. Despite their potential, these methods frequently have a number of intrinsic drawbacks. High computational complexity, reliance on extensive annotated datasets, lack of real-time responsiveness, and inadequate modeling of subtle aspects like hand transitions, contextual movements, and facial expressions are some of these. As a result, their application in practical situations—particularly in settings with low resources—remains restricted.

In order to overcome these difficulties, this study suggests a fingerspelling-based, lightweight, effective, and real-time system SSS for converting textual input into sign language representations. The Django framework, which



guarantees scalability, platform freedom, and ease of deployment, is used to develop the system as a web-based application. By mapping each letter of the input text to its appropriate sign image, the suggested method streamlines the translation process in contrast to conventional systems that mostly rely on intricate 3D motion modeling and animation. This makes the system appropriate for real-time applications by drastically lowering computational overhead while preserving responsiveness, interpretability, and clarity.

Additionally, dual-language support for both American Sign Language (ASL) and Indian Sign Language (ISL) is incorporated into the proposed system, improving its adaptability and inclusivity across various linguistic and cultural groups.

A user-friendly and responsive user interface, real-time input validation, and strong security features like OTP-based authentication and session management are all examples of how the system is built with a strong focus on user-centric principles. Together, these characteristics guarantee a smooth and safe user experience.

According to experimental evaluation, the system can produce precise and quick visual representations of text in sign language, making it a useful tool for teaching and communication support.

## **II. LITERATURE REVIEW**

To promote inclusive communication, research on sign language translation systems has expanded dramatically. Early systems lacked scalability, context comprehension, and real-time performance; instead, they employed static image-based representations and basic gesture mapping.

The first methods concentrated on simple text-to-sign conversion, which involved mapping text to sign images directly. Although these systems were effective, they were only able to recognize fingerspelling; they were unable to recognize natural sign transitions or grammar.

Advances led to the introduction of 3D avatar-based systems that translate text into sign gestures using natural language processing (NLP). Some models had an accuracy of about 89.50%, but they were constrained by tiny datasets and lacked features like smooth transitions and facial emotions.

Although speech-to-ISL systems increased accessibility even more, they lacked scalability and real-time efficiency and relied on predetermined data. Additionally, research reveals that the visual aspect of sign language and the lack of a uniform written form make translation difficult, requiring a trade-off between computing cost and accuracy.

Despite advancements, a lot of systems are still computationally demanding, not deployed in the real world, and do not facilitate bilingual communication (ASL and ISL). Most significantly, very few offer a comprehensive end-to-end solution that incorporates user interaction, animation, and translation.

## **III. METHODOLOGY**

### **SYSTEM OVERVIEW**

The Real-Time Avatar Animation System for Deaf People is intended to facilitate efficient communication between hearing and hearing-impaired people by translating textual input into visual sign language representations. The Django framework was used to create the system, which functions as a web-based application that guarantees dependability, scalability, and safe user interaction.

The system supports several sign languages, including ASL and ISL, and takes text input from users. The input is mapped to matching sign representations kept in the database after being analyzed and structured using fundamental Natural Language Processing (NLP) techniques.

A clear and intelligible visual output is then produced by sequentially displaying these signs using pictures or animations based on avatars. To improve usability, the system also has features including secure authentication, a fluid interface, and real-time validation.



**WORKING OF THE SYSSTEM**

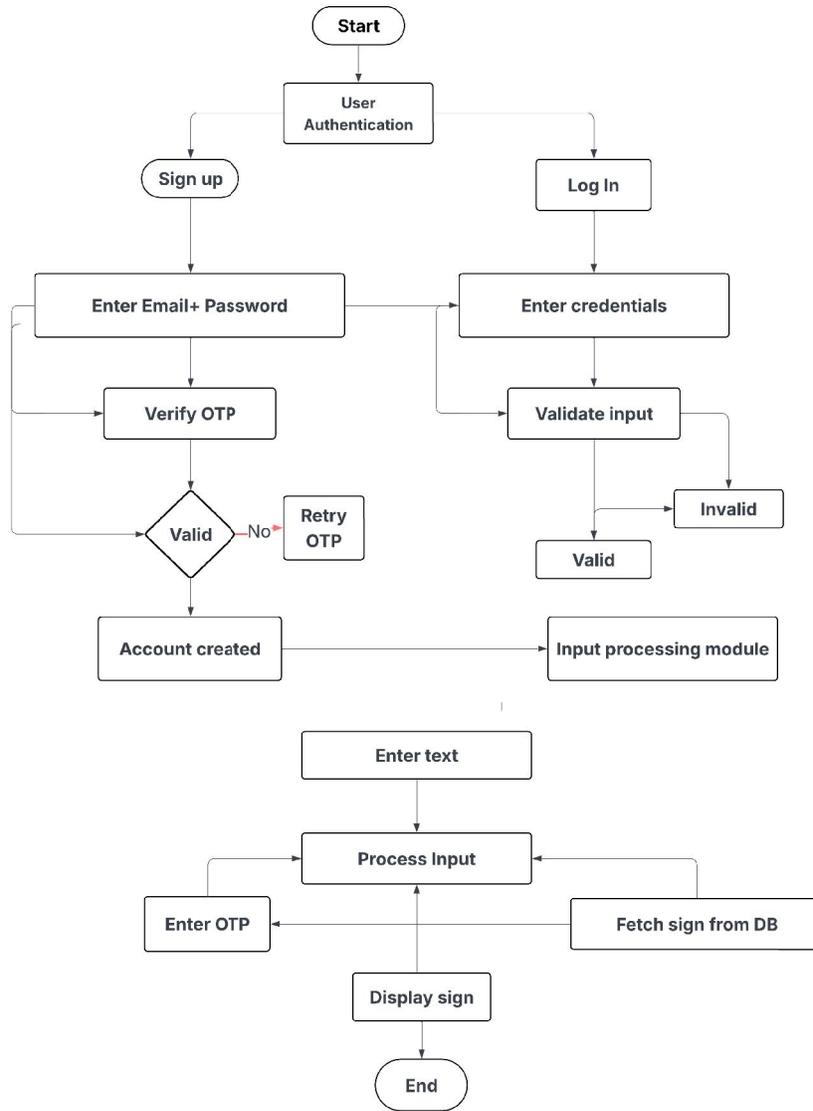


Fig. 1 Workflow of Project

The workflow of the proposed system is intended to provide a secure and efficient platform for transforming textual input into sign language representations while also assuring correct user authentication via OTP verification. The entire process is divided into two primary components: user authentication through OTP verification and text-to-sign language conversion via the Sign model. These components work together to produce a smooth and dependable application workflow.

The process starts when a user attempts to engage with the system, usually by creating a new account or completing a sensitive activity like password reset. At this point, the system is using the OTP Verification methodology to enable safe identity validation.



When a user registers, they supply basic information such as their email address, username, and password. Instead of creating the user account immediately, the system temporarily stores this information before generating a One-Time Password (OTP) with the `generate_otp()` method.

This approach generates a random 6-digit numeric number to ensure unpredictability and security.

Once the OTP is generated, the `create_otp()` method is called. This procedure completes a number of critical tasks in order to keep the system running properly. First, it invalidates any previously generated OTPs for the same email and purpose by designating them as verified, preventing their reuse or duplication. Then, it assigns the new OTP a validity duration of 10 minutes by establishing an expiration time stamp.

The OTP, along with user information and metadata like purpose and timestamps, is saved in the database. This organized storage enables the system to track verification attempts and manage the OTP lifecycle effectively.

After obtaining the OTP (usually by email), the user enters it into the system for verification. The system then uses the `verify_otp()` method to validate the entered code. This method uses numerous layers of validation to increase security. It increases the attempt counter each time a verification is attempted, preventing brute-force assaults. If the number of attempts surpasses five, the system prevents further attempts and prompts the user to obtain a new OTP.

The method additionally checks if the OTP has expired by comparing the current time to the stored expiration time. If the OTP has expired, the system will reject it and urge the user to produce a new one. It also checks to see if the OTP has previously been used, which prevents replay attacks.

If all validation tests are successful and the entered OTP matches the stored value, the system identifies the OTP as confirmed and finishes the authentication procedure. At this stage, the user account can be safely created using the temporary credentials. This ensures that only confirmed users have access to the system, hence protecting data and preventing unwanted access. To ensure database sanitation, the system periodically executes the `cleanup_expired()` procedure, which deletes OTP records

older than 24 hours. This reduces excessive data accumulation while improving performance.

Once authenticated, the user can use the application's basic functionality, which is to transform text into sign language representations. This functionality is managed by the Sign model. The user provides a text string, which is subsequently supplied to the class method `get_signs_for_text()`. This method is in charge of breaking down the input text into individual characters and mapping each character to the appropriate sign representation.

To ensure consistency in database queries, the procedure first changes all of the input text to uppercase. It then cycles over each character in the text. The system establishes the nature of each character, which is either an alphabet, a number, or a space.

Based on this classification, the system searches the database for a matching sign by word, sign language (ASL or ISL), and sign type (letter, number, or space). These attributes are indexed in the database, allowing for rapid and efficient retrieval even with enormous datasets.

If a matching sign is discovered in the database, it is added to a list of sign objects that will eventually become the output. However, the system is also intended to handle missing data graciously. If a sign for a specific character is not found, the system does not fail or disrupt the workflow. Instead, it generates a placeholder sign object that contains little information and no related image. This guarantees that the converting procedure runs smoothly.

Even with insufficient datasets.

Each sign object has an `image` field that contains a visual representation of the symbol. To access this picture safely, the system use the `get_image_url()` method. This method offers advanced error handling to prevent the application from crashing due to missing or inaccessible picture files. It verifies if the picture exists and if the file path is correct. If the picture is missing or inaccessible, the system issues a warning and returns the default placeholder image URL. This ensures that the user interface always contains a visual element to display, resulting in a consistent user experience.



Furthermore, the `has_valid_image()` method is used to ensure that the image associated with a sign is accessible. This strategy is especially beneficial in situations where photographs can be stored remotely, such as cloud storage systems. By validating picture availability, the system ensures reliability and eliminates broken links in the user interface.

After analyzing all of the characters in the input text, the system organizes the list of sign objects into a hierarchical order. This sequence expresses the input text in sign language, usually using fingerspelling (character-by-character representation). The processed Signs are then routed to the front-end interface and displayed to the user in order.

This visual output, which may include images or animations depending on the implementation, allows users to understand the text using sign language.

Overall, the workflow shows a well-structured integration of authentication with essential functionality. The OTP Verification model assures that only authorized users may access the system, whilst the Sign model effectively converts text to sign language. The usage of indexing, error handling, placeholder methods, and cleanup routines demonstrate the system's stability and scalability. This approach not only improves security but also provides a consistent and seamless user experience, making the program ideal for real-time sign language creation.

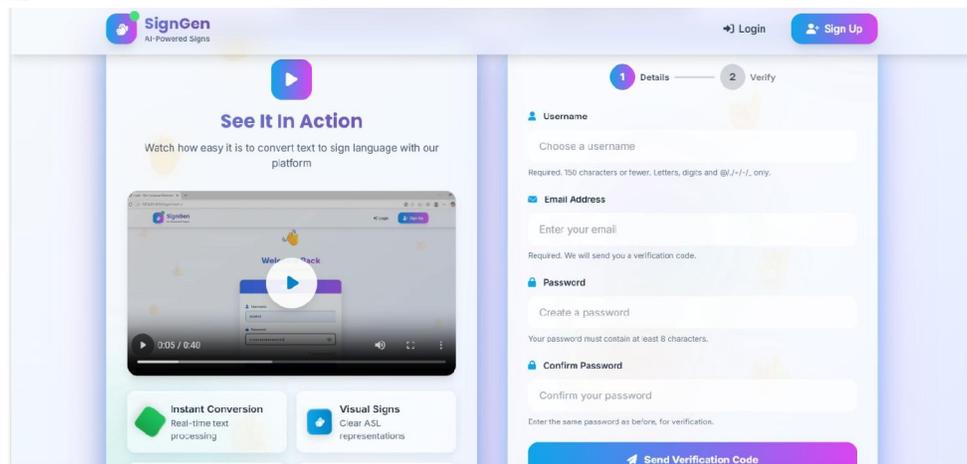
#### IV. RESULTS AND DISCUSSION

The suggested system, A Real-Time Avatar Animation System for Deaf People, was successfully created and tested to determine how well it translated text into visual representations of sign language. Important features of the system include secure user authentication, real-time validation, dual language compatibility (ASL and ISL), and text-to-sign conversion.

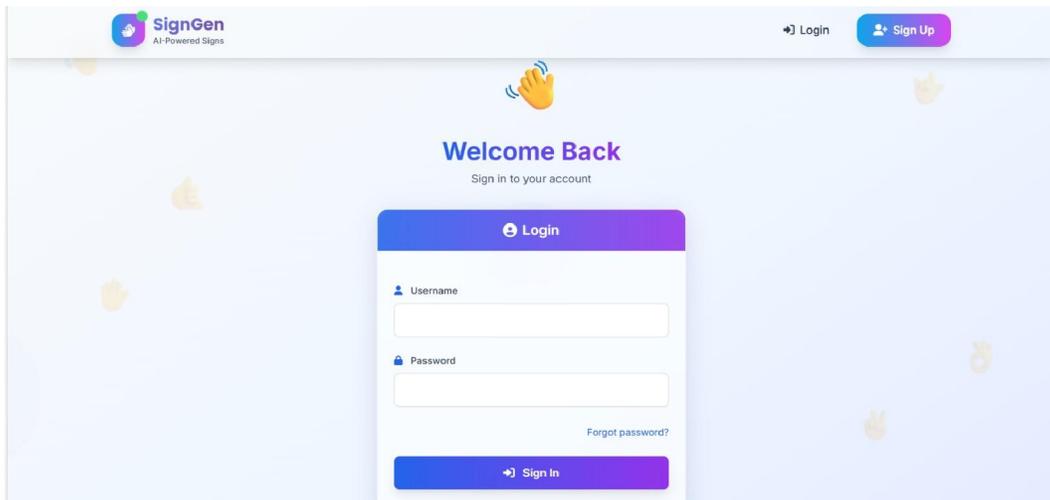
The system effectively processed user input during testing and produced matching sign outputs in real time. Clarity and consistency were ensured by the conversion mechanism's precise mapping of text into fingerspelling-based signals. Usability was enhanced by the bilingual capability, which made it easy to switch between ASL and ISL.

The user interface was determined to be responsive and user-friendly across devices, and the system also showed consistent performance with little latency. Nevertheless, dynamic animations and facial expressions are not included in the existing system, which is restricted to static photos.

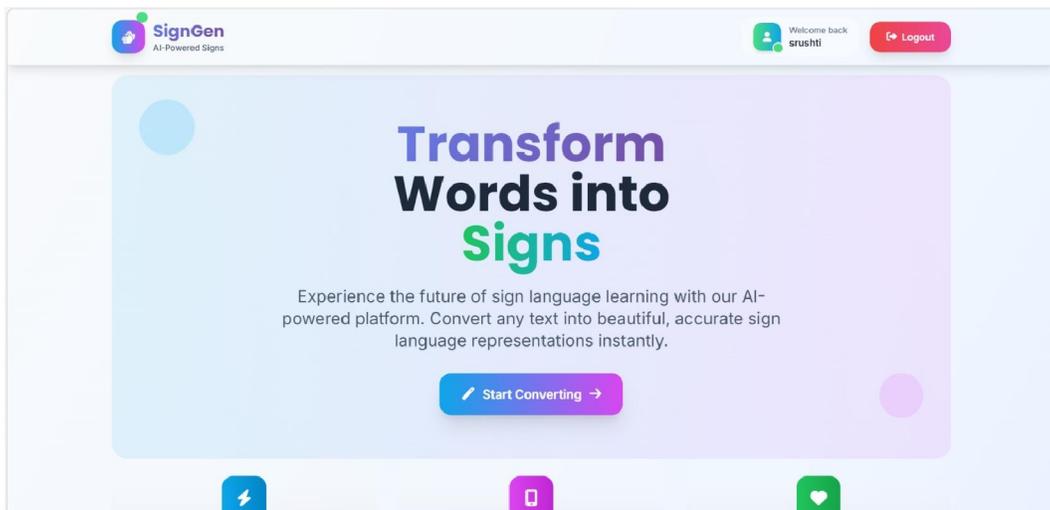
#### SIGN PAGE



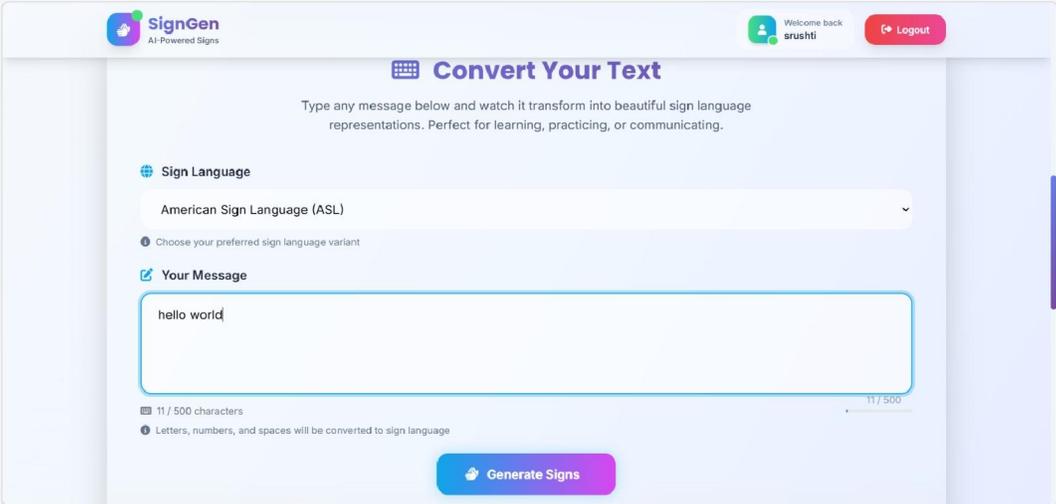
**LOGIN PAGE**



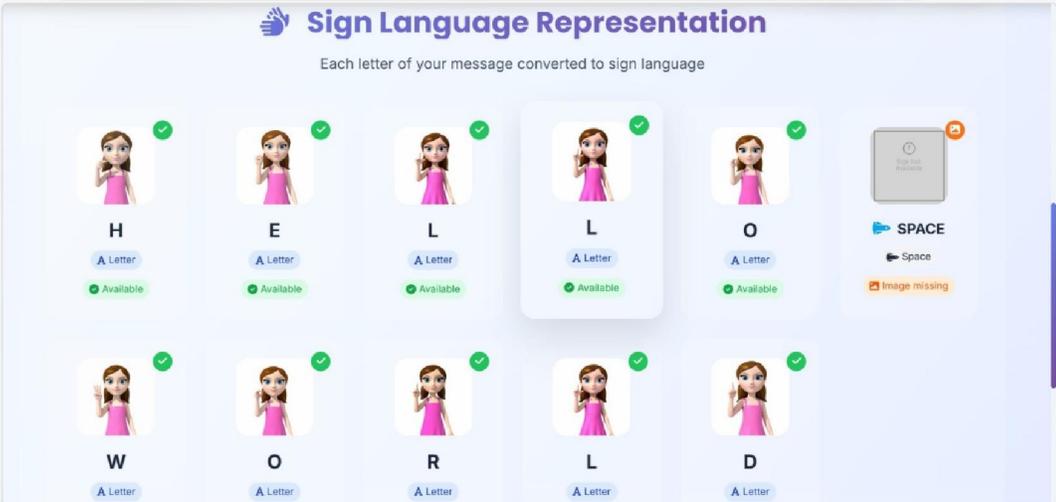
**HOME PAGE**



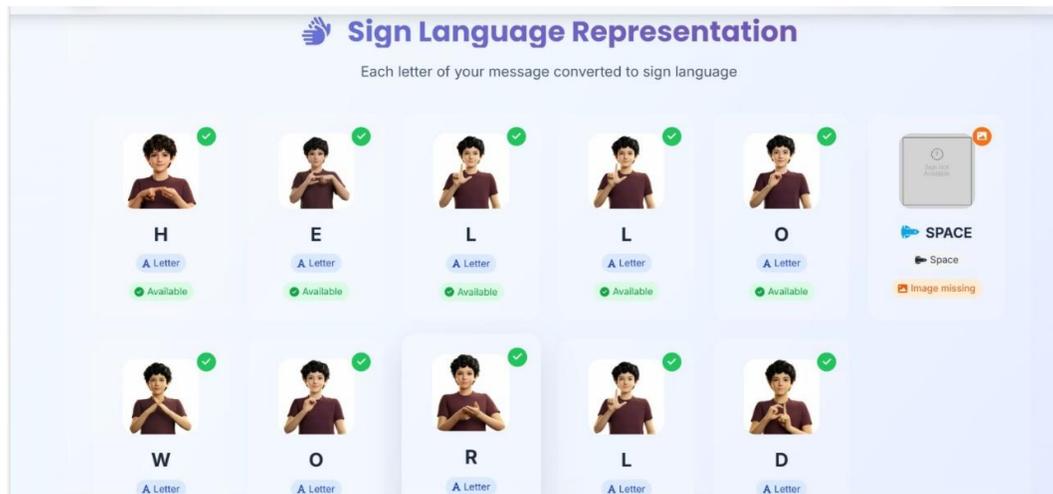
**INPUT**



**ASL OUTPUT**




## ISL OUTPUT



## V. ADVANTAGES OF THE PROPOSED SYSTEM

### Advantages and Limitations

#### Advantages

The suggested system successfully combines practical implementation, efficiency, and accessibility to provide a reliable and scalable solution. By transforming textual input into sign language representations with little latency, it enables real-time communication and greatly lowers the communication barrier between hearing and hearing-impaired people. The solution guarantees platform independence, ease of deployment, and seamless accessibility across a variety of devices without requiring specific software installations because it is a web-based application created using the Django framework.

Additionally, the system is compatible with both Indian Sign Language (ISL) and American Sign Language (ASL), which increases its usefulness to a variety of language and geographical user groups. Using a lightweight fingerspelling technique improves computing efficiency and allows for quicker processing without sacrificing system responsiveness. Additionally, the user interface is made to be dynamic and intuitive, making it simple for people with different degrees of technical expertise to utilise.

From a security perspective, dependable and safe user interactions are guaranteed by the combination of secure session management systems and OTP-based authentication. Additionally, the system's architecture is modular, which promotes future scalability and increases flexibility. This enables the smooth integration of sophisticated features like improved 3D avatar animations, artificial intelligence-driven gesture detection, and speech-to-text translation.

## VI. LIMITATIONS

### Absence of Organic Expressions

The facial expressions, lip movements, and body posture that are essential for expressing emotions and meaning in sign language are sometimes inaccurately represented by avatar systems.

### Reliance on Predetermined Datasets

These systems use small, pre-trained sign datasets. Because of this, individuals are unable to quickly identify or produce signals for unfamiliar words, phrases, or complicated sentences outside of their training set.



### **Transitions of Unnatural Gestures**

The animation appears robotic and the natural flow of communication is diminished since the transitions between successive signs are sometimes not seamless.

### **High Requirements for Computation**

Generating realistic 3D animations in real time requires significant processing power, which can cause delays or poor performance, especially on low-end devices.

### **Excessive Computational Needs**

It takes a lot of processing power to produce realistic 3D animations in real time, which can lead to delays or subpar performance, particularly on low-end devices.

## **VII. CONCLUSION**

In order to help hearing-impaired people communicate with others, a real-time avatar animation system for deaf people has been developed. Through an organised pipeline, the system transforms textual input into sign language representations, guaranteeing effective and instantaneous processing. It offers a scalable, safe, and user-friendly web-based solution and was developed with the Django framework. With support for both ASL and ISL, the system exhibits efficient text-to-sign conversion, improving its usefulness for a wide range of users. Overall dependability and user experience are enhanced by features including a responsive UI, real-time validation, and secure login. With support for both ASL and ISL, the system exhibits efficient text-to-sign conversion, improving its usefulness for a variety of users. Overall dependability and user experience are enhanced by features including a dynamic UI, secure authentication, and real-time validation. Nevertheless, the system lacks sophisticated features like face expressions and seamless gesture transitions and is restricted to predetermined signs and fingerspelling. Larger datasets, AI-based models, and 3D avatar motions to increase realism and accuracy are possible future improvements.

## **VIII. ACKNOWLEDGMENT**

The authors would like to sincerely thank everyone who helped them finish this study on the Real-Time Avatar Animation System for Deaf People. We would especially want to express our gratitude to our mentors and project guide for their unwavering support, insightful advice, and professional direction during the course of this investigation. We also thank our university for providing the technical tools, infrastructure, and encouraging learning atmosphere that allowed us to successfully complete this project. We also want to express our gratitude to our teammates, colleagues, and reviewers for their assistance, helpful criticism, and support, all of which greatly improved the overall calibre of our research project.

## **REFERENCES**

- [1] M. Hasan et al., "Advancing real-time sign language detection using YOLOv8," *BMC Artificial Intelligence*, 2025. Available: [Read Paper]([https://www.ijert.org/sign-language-recognition-2?utm\\_source=chatgpt.com](https://www.ijert.org/sign-language-recognition-2?utm_source=chatgpt.com))
- [2] "Real-time Arabic avatar for deaf-mute communication enabled by deep learning sign language translation," *Computer & Electrical Engineering*, 2024. Available: [Read Paper]([https://www.sciencedirect.com/science/article/abs/pii/S0045790624004026?utm\\_source=chatgpt.com](https://www.sciencedirect.com/science/article/abs/pii/S0045790624004026?utm_source=chatgpt.com))
- [3] P. Anbumani et al., "Bidirectional Communication: ISL and Text/Speech Converter using 3D Animated Avatar," *IJAIDR Journal*, 2026. Available: [Read Paper]([https://www.ijaidr.com/research-paper.php?id=1719&utm\\_source=chatgpt.com](https://www.ijaidr.com/research-paper.php?id=1719&utm_source=chatgpt.com))
- [4] M. Aziz and A. Othman, "Evolution and Trends in Sign Language Avatar Systems: A Systematic Review," *MDPI*, 2023. Available: [Read Paper]([https://www.mdpi.com/2414-4088/7/10/97?utm\\_source=chatgpt.com](https://www.mdpi.com/2414-4088/7/10/97?utm_source=chatgpt.com))



- [5] D. Das Chakladar et al., "3D Avatar Approach for Continuous Sign Movement Using Speech/Text," Applied Sciences, 2021. Available: [Read Paper]([https://www.mdpi.com/2076-3417/11/8/3439?type=check\\_update&version=1&utm\\_source=chatgpt.com](https://www.mdpi.com/2076-3417/11/8/3439?type=check_update&version=1&utm_source=chatgpt.com))
- [6] R. Rastgoi, K. Kiani, and S. Escalera, "Sign Language Recognition: A Deep Survey," Expert Systems with Applications, 2021. Available: <https://www.sciencedirect.com/science/article/pii/S0957417420306465>
- [7] "ES2ISL: An Advancement in Speech to Sign Language Translation using 3D Avatar Animator," IEEE Conference. Available: [Read Paper]([https://ieeexplore.ieee.org/document/9255783/?utm\\_source=chatgpt.com](https://ieeexplore.ieee.org/document/9255783/?utm_source=chatgpt.com))
- [8] V. Punjabi et al., "Signify: An Enhancing ASL Communication Model with Deep Learning," IJRASET, 2024. Available: [Read Paper]([https://www.ijraset.com/research-paper/signify-an-enhancing-asl-communication-model-with-deep-learning?utm\\_source=chatgpt.com](https://www.ijraset.com/research-paper/signify-an-enhancing-asl-communication-model-with-deep-learning?utm_source=chatgpt.com))
- [9] J. Wolfe et al., "3D Sign Language Avatar for Learning and Communication," IEEE Global Engineering Education Conference, 2019. Available: <https://ieeexplore.ieee.org/document/8717435>
- [10] S. Saipatwar et al., "Dataset for Everyday Phrases in ISL and ASL," Mendeley Data, 2025. Available: <https://data.mendeley.com/datasets/w7fgy7jvs8>

