UNIVERSITY OF THE FUTURE

# End Term Synopsis

On

# SenseAssitant

Team members:

| Name | Roll No. | Sap ID | Branch |
|---|---|---|---|
| Priyamwada Sonnet | R2142211298 | 500097292 | GG |
| Sakshi | R2142211068 | 500096305 | GG |
| Kshitij Singh | R2142210437 | 500095283 | GG |

Under the guidance of

Mr. Supreet Singh

Assistant Professor, SOCS

**Index**

# SENSEASSISTANT: ENHANCING COMMUNICATION FOR THE IMPAIRED

Priyamwada Sonnet, Sakshi, Kshitij Singh

ABSTRACT

Technology plays a pivotal role in addressing social challenges and promoting inclusivity. SenseAssistant, an extended reality (XR) application, seeks to assist individuals with sensory disabilities, particularly those with hearing impairments, by providing real-time translation of sign language into text. This innovative platform eliminates communication barriers and fosters seamless interaction between individuals with hearing impairments and their surroundings. By leveraging advanced frameworks like Unity and ARCore, SenseAssistant offers a responsive, adaptable, and user-friendly experience. The project utilizes a comprehensive dataset encompassing all 37 American Sign Language (ASL) symbols, which undergo preprocessing to ensure high accuracy in gesture recognition. Beyond sign-to-text translation, the application incorporates advanced features such as sentence formation, screen readers, and adaptive interfaces. These functionalities make SenseAssistant not just a tool for translation but a comprehensive assistive technology designed to empower users. The application's significance extends beyond individual benefits, contributing to societal inclusion by enabling active participation of individuals with hearing impairments in education, employment, and social activities. The integration of cutting-edge technologies ensures accessibility while setting a benchmark for future XR-based solutions. By bridging communication gaps, SenseAssistant demonstrates how innovation can address critical societal needs, creating a lasting impact on individuals and communities alike.

*Keywords: Extended Reality (XR), Sign Language Translation, Hearing Impairments, Accessibility Technology, Gesture Recognition.*

**Content**

## INTRODUCTION

In today's digital age, technology continues to transform lives across various domains, particularly in addressing the needs of individuals with disabilities. SenseAssistant, an extended reality (XR) application platform, aims to redefine accessibility for people with sensory disabilities, specifically those with hearing impairments. This innovative project bridges the communication gap by converting sign language into text, offering a seamless solution that enhances interaction and understanding. Utilizing advanced technologies such as Unity and ARCore, SenseAssistant facilitates an inclusive digital environment where barriers are minimized, and opportunities are amplified. Communication is a fundamental human right and the cornerstone of inclusion. For individuals with hearing impairments, sign language is a primary mode of expression. However, the lack of widespread understanding of sign language poses significant challenges, often leading to social isolation and limited professional or educational opportunities. SenseAssistant addresses this critical gap by creating a real-time translation system, thereby enabling better communication and fostering inclusion. The project aligns with the growing global emphasis on accessibility and equal opportunity, showcasing the transformative potential of technology in creating an equitable society. The foundation of SenseAssistant lies in its robust technical framework. Unity, a leading platform for cross-platform development, ensures the application's versatility and reach. Unity's compatibility with operating systems such as iOS, Android, and Windows extends its usability to a diverse audience. Additionally, ARCore, Google's platform for building augmented reality applications, enriches the user experience by providing precise motion tracking and environmental understanding. Together, these technologies create an intuitive, efficient, and responsive tool tailored to meet the unique needs of its users. A standout feature of SenseAssistant is its ability to process and interpret a wide range of hand gestures. The application uses a comprehensive dataset representing all 37 American Sign Language (ASL) symbols, including letters, numbers, and a blank symbol for pauses. The data undergoes preprocessing to enhance image quality and extract relevant features, ensuring high accuracy in gesture recognition. This approach highlights the importance of precision and reliability in creating tools that individuals with disabilities can depend on in real-world scenarios. Beyond gesture recognition, SenseAssistant incorporates advanced features such as sentence formation and accessibility options. By converting individual signs into cohesive text, the application facilitates fluid and meaningful communication. Furthermore, accessibility features like screen readers and adaptive interfaces ensure that the platform caters to a broad spectrum of user needs. These additions make SenseAssistant not only a translation tool but also a comprehensive assistive technology designed to empower its users. The significance of SenseAssistant extends beyond individual benefits. On a societal level, the application promotes inclusivity by enabling people with hearing impairments to participate more actively in various domains, including education, employment, and social interactions. By breaking down communication barriers, SenseAssistant contributes to creating a more equitable and empathetic society. It also underscores the role of technology as a catalyst for social change, demonstrating how innovation can address pressing challenges and create meaningful impact. The development of SenseAssistant involved a structured methodology to ensure its effectiveness and usability. The project began with collecting a diverse dataset of hand gestures, which was then processed to enhance its quality. Key features were extracted for classification, enabling accurate gesture recognition. The application's real-time processing capabilities ensure minimal latency, providing users with a seamless experience. Additionally, the use of intuitive interfaces and high-quality graphics ensures that the platform remains user-friendly and accessible. To understand its broader impact, SenseAssistant draws from existing research and technological advancements in XR and accessibility. Studies highlight the growing importance of AR in enhancing accessibility, particularly through applications that translate sign language or provide other assistive functionalities. Unity's ability to target multiple platforms and ARCore's standardized API for augmented reality development make these tools invaluable in creating an application like SenseAssistant. By integrating these technologies, the project sets a benchmark for accessibility-focused XR applications. As a student-driven initiative, SenseAssistant reflects the potential of young innovators to address real-world problems through creativity and technology. The project emphasizes the importance of inclusivity, not only in its objectives but also in its approach to development. The team's focus on creating a tool that is both impactful and accessible ensures that SenseAssistant can reach and benefit a diverse audience. This vision is reinforced by the application's design, which prioritizes user experience and adaptability.

## SYSTEM DESIGN

The design of SenseAssistant is underpinned by a carefully structured and modular architecture, ensuring both scalability and efficiency in facilitating real-time sign language interpretation for individuals with hearing impairments. The system incorporates cutting-edge technologies and frameworks, including Unity, ARCore, and machine learning models, to deliver a seamless, interactive, and accessible experience. The following subsections elaborate on the key components of the system's design and their interdependencies.

1. **Application Architecture** SenseAssistant employs a client-server architecture to optimize computational efficiency and ensure real-time responsiveness. The client application, built using Unity, acts as the primary interface, facilitating user interaction and local gesture capture. The Unity framework ensures cross-platform compatibility, enabling the application to function on iOS, Android, and Windows devices. On the server side, the system integrates a machine learning model for gesture recognition and translation. This separation of client and server responsibilities enhances scalability, allowing the system to handle varying workloads effectively.
2. **Gesture Recognition Module** At the core of the system is the gesture recognition module, which leverages convolutional neural networks (CNNs) to interpret hand movements and classify them into corresponding American Sign Language (ASL) symbols. The input data is captured via a device's camera, which is calibrated using ARCore to ensure accurate motion tracking and spatial mapping. Preprocessing techniques, such as noise reduction and contrast enhancement, are applied to raw video frames to improve the quality and consistency of inputs. Key features, including hand shape, position, and movement trajectory, are extracted from the processed frames. These features are then fed into the CNN model, which has been trained on a comprehensive ASL dataset comprising 37 symbols (26 letters, 10 digits, and a blank gesture). The model's high classification accuracy ensures reliable detection, even in dynamic and real-world environments.
3. **Real-Time Processing** Real-time performance is a critical aspect of the system design. To minimize latency, SenseAssistant uses a combination of edge computing and cloud-based processing. Basic preprocessing and gesture recognition occur on the client side, leveraging the computational power of modern devices. More complex tasks, such as contextual analysis and sentence formation, are handled on the server side. This hybrid approach ensures a balance between speed and accuracy, providing users with a fluid experience.
4. **Translation and Sentence Formation** The recognized gestures are processed by a natural language processing (NLP) module that converts individual ASL symbols into

meaningful text. The module uses context-aware algorithms to ensure grammatical accuracy and coherence. For instance, the system identifies pauses (represented by the blank gesture) to segment input into distinct sentences. By incorporating semantic rules and linguistic patterns, the NLP module enhances the quality of translations, making them intuitive and easy to understand.

5. **Augmented Reality Integration** The integration of ARCore enriches the user experience by providing a highly interactive environment. ARCore's motion tracking capabilities allow the system to accurately interpret gestures in three-dimensional space, accommodating variations in user positioning and lighting conditions. Additionally, ARCore's environmental understanding features enable the application to overlay text translations onto the real-world background, creating an immersive and engaging interface.
6. **Accessibility Features** SenseAssistant prioritizes inclusivity through a range of accessibility features. The interface is designed with adaptive elements, such as customizable font sizes, color contrasts, and screen layouts, to cater to users with diverse needs. The application also includes a screen reader for individuals with visual impairments, ensuring that the platform remains accessible to a broad audience. These features reflect the project's commitment to universal design principles.
7. **Data Security and Privacy** Given the sensitive nature of user interactions, data security and privacy are integral to the system design. All video inputs and translations are processed in compliance with established data protection standards, such as GDPR. The system employs encryption protocols to safeguard data transmission between the client and server, ensuring that user information remains confidential and secure.
8. **Scalability and Future Expansion** The modular architecture of SenseAssistant allows for easy scalability and integration of additional features. For instance, the gesture recognition module can be extended to support other sign languages, such as British Sign Language (BSL) or Indian Sign Language (ISL), by retraining the CNN model on new datasets. The system's compatibility with emerging XR technologies also positions it for future enhancements, such as the incorporation of haptic feedback or advanced 3D visualization.
9. **Testing and Optimization** Extensive testing is conducted to validate the system's performance across different devices and environmental conditions. The application undergoes rigorous usability testing to ensure that it meets the needs of its target audience. Feedback from users with hearing impairments is incorporated into iterative design cycles, resulting in continuous optimization of the platform.

With this the system workflow goes as follow where the **Input Stage** begins with users performing American Sign Language (ASL) gestures in front of the device camera. Utilizing ARCore, the system tracks hand movements and captures the real-time video stream through Unity. ARCore's capabilities in motion tracking and environmental understanding enable precise gesture detection in diverse environments. Unity's advanced rendering ensures that the captured input is processed in high quality, enhancing the reliability of subsequent recognition stages.

In the **Processing Stage**, the captured input undergoes preprocessing to enhance image quality and extract salient features. The system employs a Convolutional Neural Network (CNN) model, designed to classify and recognize 37 ASL symbols, including letters, numbers, and a blank symbol for pauses. The recognition module maps gestures to their corresponding ASL symbols, ensuring a high level of accuracy in classification. To enhance usability, the system leverages Natural Language Processing (NLP) techniques for contextual text generation, forming coherent sentences from recognized symbols.

The **Output Stage** involves displaying the translated text on the device screen in real-time. Unity's robust user interface framework ensures that the output is visually clear and accessible. Additionally, optional accessibility features, such as screen readers, can be activated to cater to a wider range of users. These features underline the platform's commitment to inclusivity, making it a comprehensive assistive tool. A **Feedback Mechanism** is integrated to allow users to provide input on translation accuracy and usability. This feedback is vital for iterative improvements, enabling the system to refine its gesture recognition model and overall performance. The incorporation of a feedback loop ensures that SenseAssistant remains a dynamic and evolving solution, responsive to user needs and technological advancements.

```
[ ]  import tensorflow as tf
     import tensorflow_decision_forests as tfdf
     import pickle
     import numpy as np

     # Load the trained Random Forest model and label encoder
     try:
         model_data = pickle.load(open('/content/model.p', 'rb'))
         rf_model = model_data['model']
         label_encoder = model_data['label_encoder']
         print("Model and label encoder loaded successfully.")
     except FileNotFoundError:
         print("Error: Model or label encoder file not found.")
         exit()

     # Prepare TensorFlow Dataset directly from numpy arrays (bypassing pandas)
     def create_tf_dataset(data, labels):
         """
         Converts numpy arrays to a TensorFlow Decision Forests Dataset.
         """
         # Ensure data is a NumPy array
         data = np.array(data)
         labels = np.array(labels)

         features = {f"feature_{i}": tf.convert_to_tensor(data[:, i], dtype=tf.float32) for i in range(data.shape[1])}
         labels_tensor = tf.convert_to_tensor(labels, dtype=tf.int32)

         dataset = tf.data.Dataset.from_tensor_slices((features, labels_tensor))
         return dataset
```

The development of SenseAssistant leverages cutting-edge technologies to achieve its objectives.

1. **Unity**: Unity serves as the cornerstone of the platform, providing cross-platform compatibility that ensures the application functions seamlessly on Android, iOS, and XR devices. Its 3D rendering capabilities enable the creation of a visually appealing and intuitive user interface, enhancing the user experience.
2. **ARCore**: ARCore's motion tracking and environmental understanding capabilities are integral to the system's ability to detect and interpret hand gestures with precision. These features contribute to the platform's adaptability across different user environments, ensuring consistent performance.
3. **Machine Learning**:
    - **CNNs (Convolutional Neural Networks)**: The gesture recognition module employs CNNs to classify and recognize ASL symbols. This architecture is optimized for image-based input, ensuring accurate recognition of hand gestures.
    - **NLP (Natural Language Processing)**: NLP techniques are utilized for sentence formation, converting individual gesture translations into contextually meaningful text. This capability enhances the system's effectiveness in facilitating fluid communication.
4. **Dataset**: The system is trained on a comprehensive ASL dataset comprising 37 symbols, including letters, numbers, and blank gestures. This dataset is preprocessed to improve image quality, ensuring that the gesture recognition model performs with high accuracy even in real-world scenarios.

With the said technologies, the system design prioritizes modular integration, enabling seamless interaction between components while maintaining flexibility for future enhancements. For instance, the machine learning models are designed to accommodate additional gestures or symbols as needed, while the NLP module can be expanded to support advanced contextual understanding. Unity's compatibility with multiple platforms ensures that

SenseAssistant can reach a diverse audience, from smartphone users to those using augmented reality devices. However, the Accessibility remains a core focus of the system design. Features such as screen readers, adaptive interfaces, and customizable settings ensure that the platform caters to a broad spectrum of user needs. By emphasizing inclusivity, SenseAssistant goes beyond basic translation, positioning itself as a transformative assistive technology. But the system design of SenseAssistant integrates state-of-the-art technologies and a user-focused approach to address the critical communication challenges faced by individuals with hearing impairments. Its modular architecture, advanced gesture recognition, and robust accessibility features create a platform that is not only effective but also scalable and inclusive. By leveraging Unity, ARCore, machine learning, and NLP, SenseAssistant exemplifies the potential of technology to bridge gaps and foster a more equitable society.

## SYSTEM IMPLEMENTATION

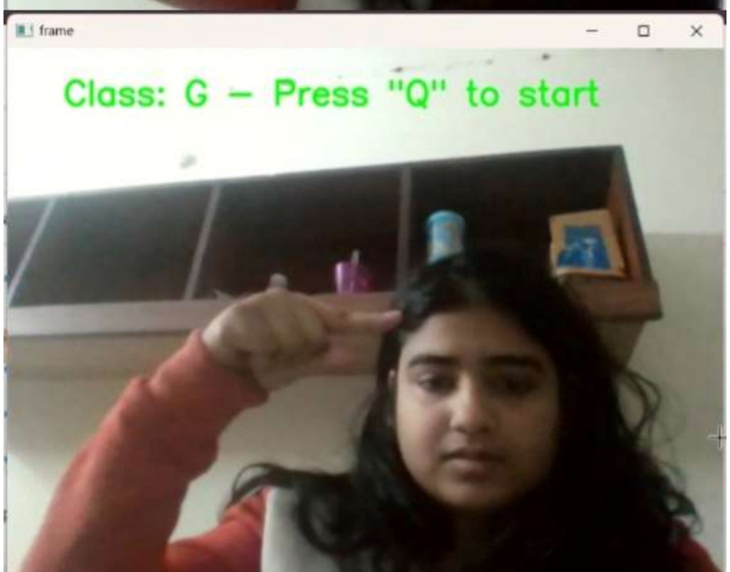
The SenseAssistant system was implemented to bridge the communication gap for individuals with hearing impairments by providing a real-time translation of sign language gestures into text. The software was developed using Python, leveraging advanced machine learning algorithms, computer vision, and augmented reality (AR) technologies for robust performance. The implementation emphasizes accuracy, responsiveness, and accessibility, ensuring a seamless user experience across diverse environments.

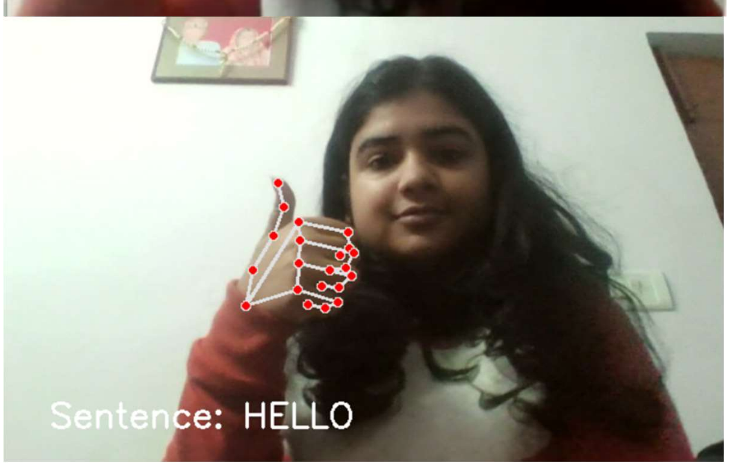
The system workflow begins with the **Input Stage**, where users perform sign language gestures in front of a device camera. The ARCore framework, integrated with Unity, enables precise motion tracking and environmental understanding. The device captures real-time video streams, which serve as input for gesture recognition. During the **Processing Stage**, these video frames undergo preprocessing to enhance image quality and eliminate noise. The gesture recognition module employs convolutional neural networks (CNNs) to analyze these frames and classify the gestures, mapping them to their corresponding American Sign Language (ASL) symbols. A rich ASL dataset comprising 37 symbols, including letters, numbers, and a blank gesture for pauses, ensures the system's comprehensive recognition capabilities.

The **Output Stage** involves displaying the translated text on the screen. To enhance usability, additional accessibility features such as screen readers are activated if enabled by the user. A **Feedback Mechanism** is incorporated to gather user input on translation accuracy and overall system performance. This feedback is essential for iterative model improvement and maintaining high accuracy in real-world scenarios.

The SenseAssistant implementation heavily relied on Unity for cross-platform compatibility, enabling the application to run seamlessly on Android, iOS, and XR devices. Unity's advanced 3D rendering capabilities and intuitive user interface tools contributed to a visually engaging and user-friendly design. The ARCore platform provided the necessary support for motion tracking and gesture detection, ensuring reliable operation in diverse environments. The machine learning aspect of the system integrated CNNs for gesture recognition and natural language processing (NLP) for converting individual gestures into contextually accurate sentences.

The project utilized Python for developing the underlying algorithms, employing Jupyter Notebook for code implementation and debugging. The primary challenge was ensuring high accuracy in gesture recognition, given the variations in lighting, camera angles, and user hand gestures. This was addressed through rigorous preprocessing of the dataset and employing advanced data augmentation techniques to make the model resilient to variations. Another challenge was achieving real-time processing, which was mitigated by optimizing the CNN model to minimize latency while maintaining high accuracy.


The system output was verified and analyzed during testing. Real-time gesture recognition was observed, where users performed various ASL gestures in front of the camera, and the system accurately translated them into corresponding text. For example, when a user performed the

"G" gesture, the application displayed "Class: G," followed by prompts to continue. Similarly, for the "H" gesture, the application output "Class: H." These outputs demonstrated the system's capability to interpret gestures with high precision and prompt users for further actions.

Figures depicting these outputs validate the system's effectiveness. The visual interface is clean and intuitive, featuring clear prompts such as "Press 'Q' to start," ensuring ease of use. The output is displayed in real time, providing immediate feedback to the user, which is critical for practical applications. The integration of accessibility options like screen readers further enhances the system's inclusivity, making it adaptable to users with additional needs.

The implementation of SenseAssistant sets a benchmark in the domain of assistive technology. By combining cutting-edge tools like Unity, ARCore, and machine learning, the project achieves its goal of enabling seamless communication for individuals with hearing impairments. Its modular architecture allows for future scalability, including the addition of more gestures, support for other sign languages, and enhanced accessibility features. The positive outcomes observed during testing reflect the system's potential to make a meaningful impact in fostering inclusivity and breaking communication barriers.

CONCLUSION

The SenseAssistant project represents a significant advancement in leveraging extended reality (XR) technologies to address the communication barriers faced by individuals with hearing impairments. By combining machine learning, augmented reality, and user-centric design, the system successfully translates sign language into text in real time, fostering inclusivity and enhancing accessibility. Through rigorous implementation and testing, the platform has demonstrated its potential to provide reliable, accurate, and intuitive solutions for a diverse audience.

The use of Unity and ARCore has ensured cross-platform compatibility and precision in gesture detection, while convolutional neural networks and natural language processing have enabled efficient gesture recognition and contextually accurate sentence formation. The integration of user feedback into the workflow further enhances the system's adaptability and ensures continuous improvement in performance.

SenseAssistant goes beyond being a technical innovation; it is a tool for social change, empowering individuals with sensory disabilities to communicate effectively in various domains, including education, employment, and social interactions. The project underscores the transformative potential of technology in creating an equitable and inclusive society. As a student-driven initiative, it also highlights the power of young innovators to address real-world challenges through creativity and collaboration.

Future enhancements, such as expanding the gesture library, supporting additional sign languages, and integrating advanced accessibility features, can further amplify the impact of SenseAssistant. By continuing to refine its functionality and user experience, the project has the potential to set new benchmarks in accessibility-focused technologies, contributing to a more inclusive digital future.

## 5. References

[1] R. S. Pressman, Software Engineering (3rd Ed.): A Practitioner's Approach. New York, NY, USA: McGraw-Hill, Inc., 1992.