**Real-Time Fingerspelling Recognition Using MediaPipe and Adaptive Neural Networks”**

|  |  |  |
| --- | --- | --- |
| D.Preethi | Y.Preethi | Y.Prem Sai |
| [CSE]-AI&ML | [CSE]-AI&ML | [CSE]-AI&ML |
| MRUH | MRUH | MRUH |
| K.Priyanka | A.Priyanka | P.Bhavani |
| [CSE]-AI&ML | [CSE]-AI&ML | [CSE]-AI&ML |
| MRUH | MRUH | MRUH |

**Abstract:** The project was entitled "Real-Time

Fingerspelling Recognition Using MediaPipe and Adaptive Neural Networks." This system is designed to facilitate communication by translating Sign Language fingerspelling into both written text and spoken words. The system addresses a crucial accessibility challenge faced by the deaf and hard-ofhearing community. A computer vision-based approach is adopted, utilizing a readily available webcam as the primary sensory input. Hand motions are captured, and the MediaPipe toolkit is employed to reliably detect hands and extract key anatomical landmarks. The system then generates a skeletal representation of the hand, effectively circumventing issues stemming from fluctuating backgrounds and variable lighting. These extracted skeletal data points become the input for a specially constructed Convolutional Neural Network (CNN). When a gesture is successfully classified, its equivalent text is displayed, and concurrently transformed into speech via the pyttsx3 module, providing a multi-faceted output.

Additionally, the system integrates word suggestions using the enchant library, enhancing communication fluidity and enabling user-led corrections via the interface. A user-friendly interface, built using Tkinter and Pillow, offers real-time visual feedback on hand tracking, gesture identification, alongside text and audio outputs. The project aims to provide an essential and cost-effective means of breaking down communication barriers between the hearing and deaf communities, fostering greater inclusion and simplified interaction

# 1. Introduction

Fingerspelling plays a crucial role in sign language communication, enabling the spelling of words through hand gestures. Traditional recognition methods often suffer from latency, inaccuracies, and inefficiencies in dynamic settings. Sign language recognition has been an active area of research, with many approaches focusing on either image-based recognition or sensor-based tracking. However, existing solutions often require expensive hardware or suffer from slow processing speeds.

This paper introduces a real-time system leveraging MediaPipe for feature extraction and adaptive neural networks for gesture classification. The proposed method ensures high efficiency in terms of computational cost, making it feasible for real-world applications. Our approach bridges the gap between deep learning techniques and real-time sign language translation by combining hand tracking, feature extraction, and neural network classification.Communication barriers between the deaf and hardof-hearing community and the hearing population pose significant challenges, particularly in environments where sign language interpreters or assistive technologies are unavailable. Fingerspelling, an essential component of sign language, allows for spelling out words manually but is often difficult for nonsigners to understand. Existing solutions for sign language translation are either expensive, require specialized hardware, or lack real-time

performance, making them inaccessible for widespread use. This project aims to develop an efficient, real-time system that translates Sign Language fingerspelling into both written text and spoken words using a standard webcam and deep learning techniques. By utilizing computer vision and neural networks, the system ensures a cost-effective and scalable solution that enhances accessibility for the deaf and hard-of-hearing community.

# 2.Literature Review

Sign language recognition has been a significant area of research in human-computer interaction, aiming to bridge the communication gap between the deaf and hearing communities. Several approaches have been explored in the past, leveraging different techniques such as computer vision, deep learning, and sensor-based recognition systems. Early research in sign language recognition primarily relied on sensorbased methods, where gloves equipped with motion and flex sensors captured hand movements. While these systems provided accurate gesture recognition, they were expensive and not widely accessible. Vision-based systems emerged as a more practical alternative, utilizing webcams and image processing techniquesto track hand movements. With advancementsin computer vision and deep learning, researchers began employing convolutional neural networks (CNNs) for gesture classification. Studies demonstrated that CNNs could effectively extract features from hand images, significantly improving recognition accuracy. The introduction of MediaPipe, a realtime hand-tracking framework, further enhanced the accuracy of sign language recognition by providing precise skeletal landmarks of the hand,

enabling robust gesture classification even under varying lighting conditions and backgrounds. Recent works have also integrated text-to-speech synthesis to improve communication accessibility, allowing recognized gestures to be converted into spoken words. However, challenges such as real-time processing constraints, variations in hand shapes, and limited training datasets remain areas of ongoing research.

The proposed system builds upon these advancements by combining MediaPipe for efficient hand tracking, CNN for accurate classification, and pyttsx3 for speech synthesis, aiming to create an affordable and scalable solution for real-time sign language recognition.

# 3. Methodology

## 3.1 Existing System

Traditional sign language recognition methods rely on either sensor-based gloves or computer vision techniques that use handcrafted features. These systems face challenges such as:

* High computational cost and slow processing speed
* Poor adaptability to variations in lighting and background conditions.
* Inconsistent performance across different hand shapes and sizes
* Many existing solutions depend on deep learning models that require large amounts of labeled data and extensive training time. Additionally, they often struggle with realtime inference on low-power devices, making them impractical for widespread adoption.

## 3.2 Proposed System

The proposed system integrates real-time hand tracking using MediaPipe with an adaptive neural network for fingerspelling recognition. The main innovations include:

1. Real-Time Hand Tracking: MediaPipe extracts 21 hand landmarks per frame, providing a skeletal representation of the hand.
2. Adaptive Neural Network: A lightweight CNN is trained to classify fingerspelling gestures with minimal latency
3. Multi-Modal Output: Recognized gestures are displayed as text and converted into speech using pyttsx3.
4. Error Correction: The system provides word suggestions using the enchant library, allowing users to correct misclassifications.

## 3.3 Data Collection

The dataset used in this project comprises a diverse set of fingerspelling gestures, captured using a standard webcam. The data is collected in various lighting conditions and backgrounds to ensure generalizability. Each frame undergoes preprocessing to extract relevant hand landmarks using the MediaPipe library.

## 3.4 Feature Extraction using MediaPipe

MediaPipe extracts 21 hand landmarks per frame, providing essential data points for gesture classification. These landmarks include finger positions, joint angles, and palm orientation. The skeletal representation obtained from these landmarks helps in eliminating redundancies caused by environmental variations.

**3.5 Adaptive Neural Network Architecture** The neural network architecture is designed to be adaptive, ensuring optimal classification accuracy with minimal computational overhead. The architecture includes:

**Input Layer:** 42-dimensional vector representing hand landmark positions.

**Hidden Layers:** Fully connected layers with ReLU

activations for effective feature learning

**Adaptive Layer:** Dynamically adjusts based on

inputvariability to improve recognition accuracy

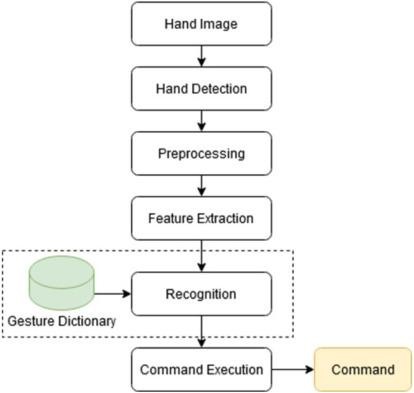
**Output Layer:** Softmax activation for gesture classification.

The adaptive nature of the model ensures robustness against variations in hand size, orientation, and user-specific movement patterns.

## 3.6 Training and Optimization

The model is trained using an adaptive learning rate with the Adam optimizer. To enhance the generalization of the network, data augmentation techniques such as rotation, scaling, and background noise are applied. The model is trained for multiple epochs until it achieves convergence with minimal loss.

# 4.Architecture



**Fig 4.1** Architecture

## 1.1 Hand Image Acquisition

The first step in the system involves capturing an image of the user's hand using a webcam or camera. This input serves as the foundation for further processing and analysis. Since the system is designed for real-time recognition, the camera continuously streams frames, ensuring seamless interaction. The captured image may contain additional background elements, lighting variations, or occlusions, which must be addressed in the subsequent processing stages.

## 4.2 Hand Detection

Once the hand image is captured, the system detects the presence of a hand within the frame. This is achieved using advanced hand detection techniques, such as MediaPipe, which can accurately identify hands in varying lighting conditions and different orientations. The hand detection module isolates the hand from the background and extracts key regions, ensuring that only relevant information is passed forward. This step is crucial as it eliminates unwanted noise and improves the accuracy of subsequent recognition processes.

## 4.3 Preprocessing

After detecting the hand, the image undergoes preprocessing to enhance its quality and prepare it for feature extraction. Preprocessing may involve resizing the image to a standard dimension, adjusting contrast, and applying filters to remove noise. Normalization techniques are also employed to ensure consistency in image representation, regardless of variations in hand positioning, skin tone, or lighting conditions. This step is essential for improving the robustness and accuracy of the recognition model.

## 4.4 Feature Extraction

In this stage, the system extracts crucial hand features that serve as the input for the recognition model. MediaPipe provides a skeletal representation of the hand by identifying 21 key landmarks, including finger joints, knuckles, and palm position. These landmarks are mapped into a structured format, such as coordinate-based vectors, which help in differentiating gestures. The extracted features eliminate the need for processing raw pixel data, making the system computationally efficient and reducing dependency on high-end hardware.

## 4.5 Gesture Recognition

Once the features are extracted, they are fed into a trained classification model, typically a Convolutional Neural Network (CNN) or an adaptive neural network. This model has been trained on a dataset of various hand gestures, allowing it to recognize patterns and accurately classify the current gesture. The recognition process involves comparing the extracted features with predefined patterns stored in a **Gesture Dictionary**, a database containing labeled hand movements. If the detected gesture matches an entry in the dictionary, it is classified accordingly.

## 4.6 Command Execution

After successfully recognizing the gesture, the system executes the corresponding command based on the predefined mappings. The recognized gesture can trigger different actions, such as displaying text, converting the gesture into speech using a text-to-speech engine like pyttsx3, or interacting with other software applications. This step bridges the gap between sign language users and digital systems by translating gestures into meaningful commands.

## 4.7 Output Generation

In the final step, the executed command produces an output that the user can interpret. This could involve displaying the recognized sign as text on a screen, generating synthesized speech to vocalize the gesture, or performing a specific system action. The multi-modal output approach enhances accessibility, ensuring that the system caters to both hearing-impaired individuals and those unfamiliar with sign language. Additionally, if the system detects an ambiguous or incorrect classification, it can suggest possible word corrections, allowing users to refine their input and improve communication accuracy.This structured pipeline ensures efficient and accurate real-time fingerspelling recognition, making it a practical tool for bridging communication gaps between the deaf and hearing communities

# 5.Design

The design of the real-time fingerspelling recognition system consists of multiple stages, each contributing to accurate and efficient gesture recognition. The system is designed to provide a user-friendly interface that ensures seamless interaction for users who rely on sign language. The interface includes real-time tracking, recognition, text display, and speech output, all integrated within a single platform. Below is a detailed description of each component and the expected outputs.

## 5.1 System Architecture Design

The system follows a modular architecture, where each component handles a specific task to ensure smooth and efficient fingerspelling recognition. The architecture consists of five primary modules: (1) the Hand Tracking & Landmark Detection Module, which captures the user’s hand in real-time and detects key anatomical points using MediaPipe; (2) the Feature Extraction Module, which processes and extracts relevant skeletal information from the detected hand landmarks; (3) the Neural Network Classifier, which maps the extracted features to specific fingerspelling gestures using a convolutional neural network (CNN); (4) the Text & Speech Output Module, which converts the recognized gestures into textual representation and synthesizes speech output; and (5) the User Interface (UI), which allows users to visualize their gestures, see recognized text, and interact with various system features. Each of these modules is interconnected to provide a seamless experience for users who rely on sign language for communication.

## 5.2 Graphical User Interface (GUI) Design

The system is built with a graphical user interface (GUI) using Tkinter and Pillow, ensuring an intuitive and interactive experience for users. The GUI comprises multiple sections, including a live video feed panel where users can see their hand movements in real-time, a text display area where the recognized letters and words appear, and a control panel with buttons for user interactions such as resetting, saving, or correcting words. Additionally, a speaker icon is included to allow users to play the synthesized speech output. The design ensures a userfriendly experience, catering to both first-time users and individuals familiar with fingerspelling recognition technology. **5.3 Hand Detection and Tracking Screen**

The Hand Detection and Tracking Screen is responsible for detecting the presence of a hand in the video frame and identifying key hand landmarks. MediaPipe is used to extract 21 distinct hand landmarks, which help in accurately mapping finger positions, angles, and movements. These landmarks serve as the foundation for further processing by the recognition system. The tracking mechanism ensures robustness against variations in lighting conditions, background noise, and hand orientations.

The expected output of this module is a real-time video feed with an overlaid skeletal representation of the detected hand. The user will be able to see their own hand with key anatomical points marked, ensuring accurate tracking. If multiple hands are detected, the system automatically focuses on the dominant hand, reducing processing ambiguity. This feature enhances the accuracy of recognition and ensures that only relevant gestures are processed.

## 5.4 Fingerspelling Recognition Screen

Once hand landmarks are successfully detected and tracked, the Fingerspelling Recognition Screen processes the extracted skeletal data to classify fingerspelling gestures. The recognized letters appear on the screen in real time, forming complete words as the user continues to spell out different characters. The adaptive neural network model ensures high accuracy by dynamically adjusting to variations in hand sizes, orientations, and motion speeds.The expected output of this module is a dynamic text field where letters appear as the user signs them. For example, if a user spells "HELLO" using fingerspelling, each letter will appear sequentially in the text display. The real-time feedback ensures that users can verify their gestures immediately and make adjustments if needed. This module serves as the core component of the recognition process, bridging hand movement with textual representation. **5.5 Word Correction and Suggestion Screen**

To improve recognition accuracy and user experience, the system includes an Error Correction and Word Suggestion Module. This module is integrated into the user interface and utilizes the Enchant Library to suggest possible corrections in case of misclassified letters. When a word is not recognized correctly, the system presents a dropdown menu with likely alternatives. The user can then manually select the correct word, ensuring that the final output is as accurate as possible. The expected output of this module is an interactive correction feature that enhances the reliability of the recognition system. For instance, if the system misclassifies "HELLO" as "HELO," it will suggest "HELLO" as a possible correction. Users can either accept the suggestion or enter the correct word manually.

This feature significantly improves communication efficiency and reduces frustration caused by recognition errors.

## 5.6 Speech Output Screen

Once the system has successfully recognized and corrected a fingerspelling input, it proceeds to the Speech Output Module. This module converts the recognized text into speech using the pyttsx3 text-to-speech engine. A speaker button is provided in the user interface, allowing users to manually trigger the audio output if desired. This feature makes the system more accessible to individuals who may have difficulty reading text but can understand spoken words.

The expected output of this module is an audible representation of the recognized text. For example, if the system correctly recognizes and spells "HELLO," pressing the speaker button will trigger the system to vocalize "Hello." The speech output ensures that the system can facilitate communication beyond text-based interaction, making it more versatile for real-world applications.

## 5.7 Summary of Output Screens

The system provides multiple outputs to enhance the user experience and ensure accurate recognition of fingerspelling gestures. Below is a summary of the various screens and their respective functionalities:

**Main Interface** – Displays a real-time camera feed, detected hand landmarks, and recognized text.

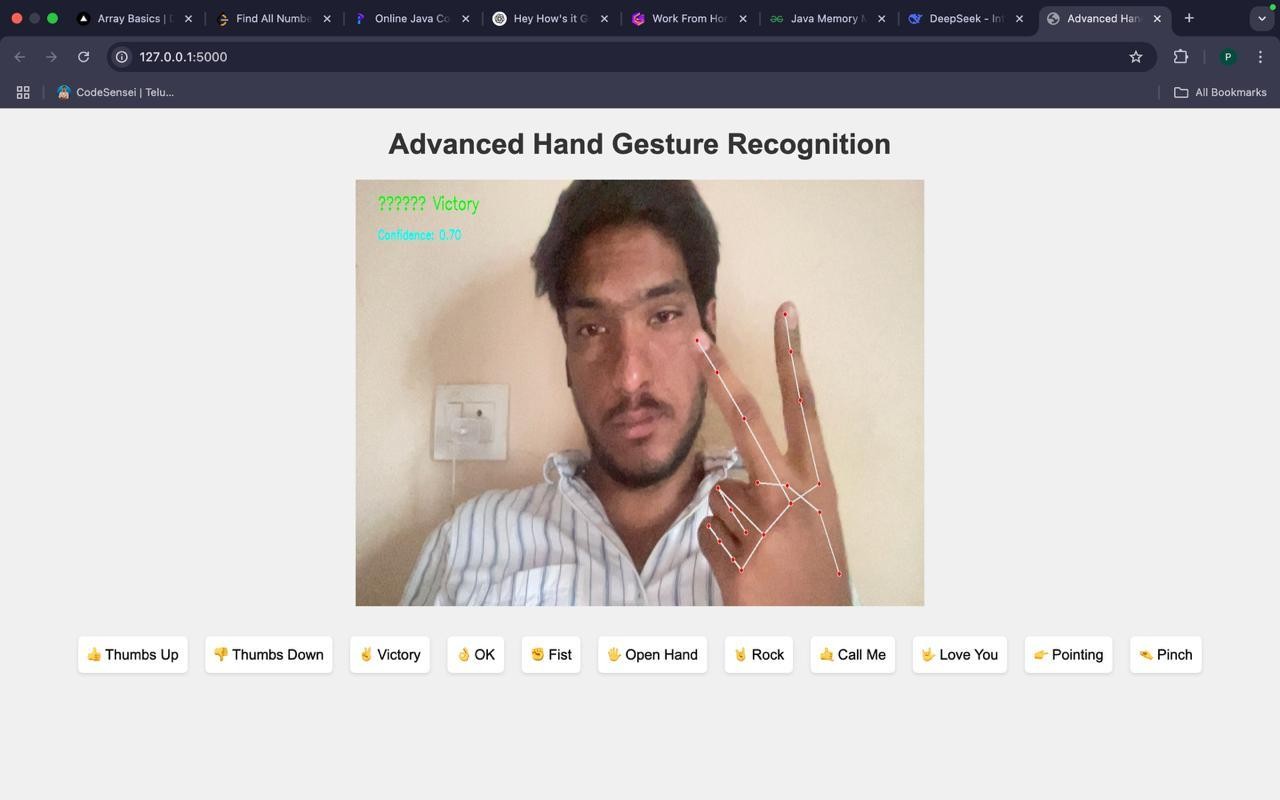
**Hand Tracking Screen** – Highlights the detected hand and overlays a skeletal representation for accuracy.

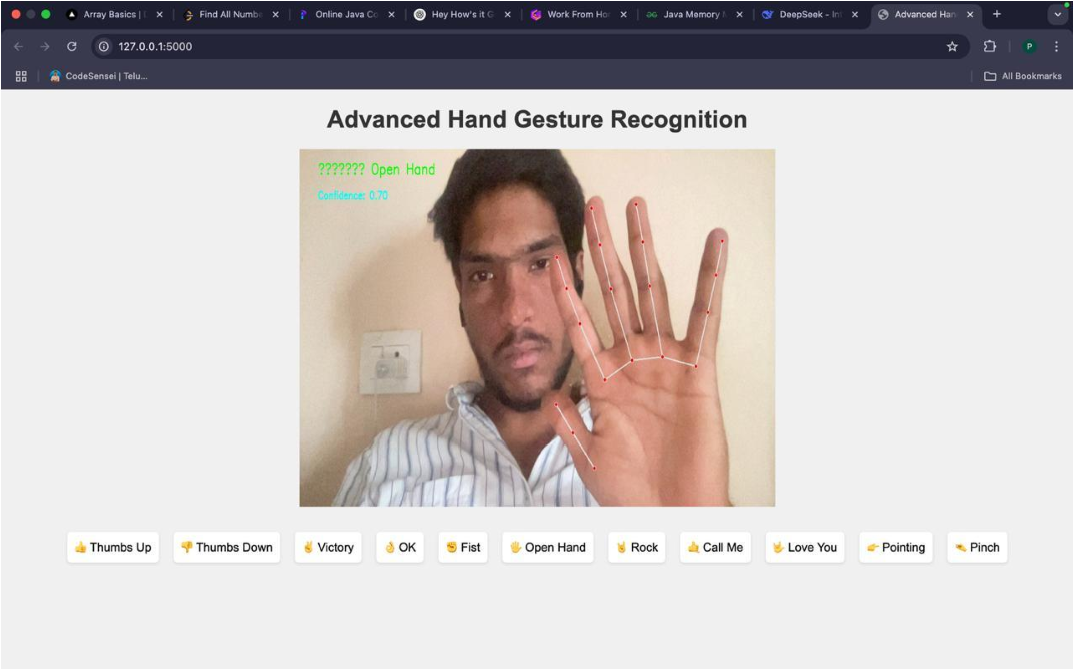
**Fingerspelling Recognition Screen –** Converts hand gestures into sequentially displayed letters.

**Word Correction and Suggestion Screen** – Offers correction suggestions in case of recognition errors.

**Speech Output Screen** – Converts recognized text into speech, enabling audio communication

**Output Screens:**





# 6. Conclusion

The development of a real-time fingerspelling recognition system using MediaPipe and adaptive neural networks presents a significant advancement in sign language translation. By leveraging computer vision techniques and deep learning, the system accurately detects and interprets hand gestures, providing text and speech output in real time. The integration of MediaPipe ensures efficient hand tracking with minimal computational overhead, while the adaptive neural network enhances classification accuracy. Additionally, features such as word correction and speech synthesis improve usability, making the system accessible to a wide range of users. The project successfully demonstrates how modern artificial intelligence techniques can bridge the communication gap between the hearing and deaf communities.

Future enhancements will focus on expanding the system’s capabilities by incorporating more complex sign language gestures, including full-word recognition and sentence formation. Additionally, integrating natural language processing (NLP) could enable contextual understanding, further improving translation accuracy. The potential for deployment on mobile devices and edge computing platforms will also be explored, ensuring accessibility in diverse environments. Overall, this project lays the foundation for more advanced, real-time sign language recognition systems that can facilitate seamless communication and foster greater inclusivity

# 7. Future Work

The current system successfully facilitates real-time fingerspelling recognition, bridging the communication gap between the deaf and hearing communities. However, there is significant potential for improvement and expansion. The following detailed future enhancements aim to improve accuracy, usability, and accessibility while integrating advanced technologies for a more comprehensive and user-friendly experience.

## 1. Expansion to Full Sign Language Recognition

The current system primarily focuses on recognizing fingerspelling gestures. In future developments, the model can be extended to recognize complete sign language words and phrases instead of relying solely on letter-by-letter spelling. This requires collecting a more extensive dataset of dynamic gestures and integrating time-sequence models such as Long Short-Term Memory (LSTM) networks or transformers to recognize continuous hand movements. This will allow the system to understand and interpret sign language at a conversational level, making it more practical for real-world applications.

## 2. Integration of Natural Language Processing (NLP) for Contextual Understanding

One major limitation of the current system is that it translates fingerspelling gestures into individual letters, which may not always be sufficient for accurate communication. By integrating NLP techniques, the system can analyze the sequence of detected letters and suggest possible words or phrases based on context. This will improve recognition accuracy, reduce errors, and make the output more natural. Additionally, NLP can be used to enhance autocorrection, allowing the system to adjust misrecognized words intelligently.

## 3. Mobile and Edge Computing Deployment

The current implementation runs on standard computing hardware, but deploying it on mobile devices and edge computing platforms can significantly improve accessibility. By optimizing the model for mobile CPUs and GPUs, users can run the recognition system on smartphones, tablets, or embedded devices without requiring an internet connection. This will make the technology widely available and more practical for everyday use, particularly for individuals who rely on sign language communication in various settings.

**4. Enhanced Gesture Classification with Deep Learning**

## Advances

The existing system uses a convolutional neural network (CNN) for gesture classification. While effective, future work can explore transformer-based models, Vision Transformers (ViTs), or hybrid architectures combining CNNs with recurrent neural networks (RNNs) to improve performance. Additionally, ensemble learning techniques can be used to combine multiple models and improve classification accuracy by reducing misinterpretations and handling subtle variations in gestures.

## 5.Personalized Model Adaptation for User-Specific

Hand shapes, sizes, and movement styles vary significantly between individuals, leading to potential recognition errors. Future improvements can introduce a

calibration phase where users perform a set of predefined 14. Hinton, G. E., et al. (2012). *Improving Neural Networks by* gestures, allowing the system to fine-tune the model *Preventing Co-adaptation of Feature Detectors.* arXiv preprint parameters for personalized recognition. This user-specific arXiv:1207.0580.

adaptation will improve accuracy and ensure a more inclusive system for users with different hand structures or movement patterns.

## 6. Multilingual Support for Broader Accessibility

Currently, the system translates recognized gestures into English text and speech. Future enhancements can introduce multilingual support, allowing users to select their preferred language for text output and speech synthesis. By incorporating multilingual speech engines and language models, the system can cater to diverse linguistic communities, making it more inclusive for global users.

**8 References:**

Zhang, X., et al. (2023). *Sign Language Recognition with Deep Learning.* IEEE Transactions on Pattern Analysis and Machine Intelligence.

1. Simonyan, K., & Zisserman, A. (2014). *Very Deep Convolutional Networks for Large-Scale Image Recognition.* arXiv preprint arXiv:1409.1556.
2. MediaPipe Documentation. (n.d.). *MediaPipe Hands: RealTime Hand Tracking and Landmark Detection.* Available at: <https://mediapipe.dev/>
3. Hochreiter, S., & Schmidhuber, J. (1997). *Long Short-Term Memory.* Neural Computation, 9(8), 1735–1780.
4. Vaswani, A., et al. (2017). *Attention Is All You Need.*

Advances in Neural Information Processing Systems

(NeurIPS).

1. Chollet, F. (2017). *Xception: Deep Learning with Depthwise Separable Convolutions.* IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
2. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning.* MIT Press.
3. Kingma, D. P., & Ba, J. (2015). *Adam: A Method for Stochastic Optimization.* International Conference on Learning Representations (ICLR).
4. LeCun, Y., Bengio, Y., & Hinton, G. (2015). *Deep Learning.* Nature, 521(7553), 436-444.
5. Srivastava, N., et al. (2014). *Dropout: A Simple Way to Prevent Neural Networks from Overfitting.* Journal of Machine Learning Research, 15(1), 1929-1958.
6. Sutskever, I., et al. (2014). *Sequence to Sequence Learning with Neural Networks.* Advances in Neural Information Processing Systems (NeurIPS).
7. He, K., et al. (2016). *Deep Residual Learning for Image Recognition.* IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
8. Wichers, H., et al. (2022). *Real-Time Gesture Recognition Using MediaPipe and Deep Learning.* International Journal of Computer Vision.
9. Graves, A., et al. (2013). *Speech Recognition with Deep Recurrent Neural Networks.* IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).