Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 5, No.2, May : 2025

COMPARATIVE STUDY OF MEDIAPIPE AND CNN FOR ASL RECOGNITION

Dr. J. R. Nandwalkar, Professor, Dept.Of Computer Engineering, *Datta Meghe College of Engineering, Mumbai University*
Rohit Ravindra Borse,*Department of Computer Engineering, Datta Meghe College of Engineering, ,Mumbai University*
Preeti Udaykumar Kadam, *Department of Computer Engineering, Datta Meghe College of Engineering,Mumbai University*
Kirti Ravindra Khairnar, *Department of Computer Engineering, Datta Meghe College of Engineering, Mumbai University*
Harshavardhan Satish Lavand, *Department of Computer Engineering, Datta Meghe College of Engineering, Mumbai University*

ABSTRACT :
Sign Language Recognition (SLR) is essential for improving communication between hearing and hearing-impaired individuals. This study compares two popular methods for American Sign Language (ASL) recognition MediaPipe and Convolutional Neural Networks (CNN). MediaPipe is a lightweight framework for real-time hand tracking and gesture recognition, while CNNs are deep learning models known for their ability to process image data effectively. This paper provides a detailed comparison based on accuracy, computational efficiency, ease of implementation, and real-time performance. Experimental results demonstrate that MediaPipe offers superior real-time performance with lower computational costs, while CNNs provide higher accuracy at the expense of greater processing time.
Keywords: American Sign Language (ASL), MediaPipe, Convolutional Neural Networks (CNN), Sign Recognition, Machine Learning

INTRODUCTION:

Sign language is one of the most widely used forms of communication for hearing-impaired individuals. However, the lack of accessible and accurate sign language interpretation tools creates a communication barrier between the hearing-impaired and hearing individuals. American Sign Language (ASL) is the most commonly used sign language in the United States, and the development of accurate ASL recognition systems is crucial for improving communication and social inclusion.Recent advancements in computer vision and machine learning have opened new possibilities for automatic sign language recognition. Two of the most promising approaches are MediaPipe and Convolutional Neural Networks (CNN).MediaPipe is a real-time hand-tracking framework developed by Google that leverages machine learning to detect and track hand landmarks with low latency and high efficiency. It is designed to run on edge devices and smartphones, making it highly accessible for real-world applications.Convolutional Neural Networks (CNN) are deep learning models known for their ability to extract features from image data. CNNs have shown remarkable success in tasks such as image classification, object detection, and pattern recognition. They offer high accuracy but require greater computational resources and higher latency.This paper aims to present a comprehensive comparison between MediaPipe and CNN for ASL recognition. While MediaPipe provides real-time tracking and low computational cost, CNN models deliver higher accuracy by learning complex patterns in sign gestures. The study explores the following key performance metrics:

- Accuracy – How well the model correctly identifies signs.
- Latency – The time taken to process and classify gestures.
- Computational Complexity – The processing power and memory required by the model.

- Model Size – The storage and memory footprint of the model.

Real-Time Feasibility – The ability to handle real-time communication.

The proposed system includes real-time hand-tracking using MediaPipe or CNN, classification using an SVM (Support Vector Machine) model, and next word prediction using an HMM (Hidden Markov Model). A Flask-based backend processes hand landmarks or CNN outputs and sends predicted signs back to the client.

The paper is structured as follows:

Section 2 presents a literature review of previous works on sign language recognition using machine learning and deep learning.

Section 3 details the methodology, including system architecture and implementation of MediaPipe and CNN models.

Section 4 provides a comparative analysis based on performance metrics and experimental results.

Section 5 discusses the challenges and limitations of each approach.

Section 6 outlines future directions for improving ASL recognition systems.

Section 7 concludes the paper with key findings and implications.

This study aims to guide future research and development efforts in the field of sign language recognition by providing insights into the strengths and limitations of MediaPipe and CNN-based approaches.

INTRODUCTION TO THEORETICAL KNOWLEDGE REQUIRED FOR THE PROJECT

A. Deep Learning and Computer Vision

A fundamental understanding of deep learning and computer vision techniques is essential for ASL recognition. Convolutional Neural Networks (CNNs) are widely used in image classification and feature extraction, enabling models to learn spatial hierarchies of features. In contrast, MediaPipe is a real-time hand-tracking framework that extracts key hand landmarks and provides efficient gesture recognition with minimal computational cost. Understanding the differences in their feature extraction mechanisms and processing pipelines is crucial for comparison.

B. Machine Learning Models and Frameworks

Familiarity with machine learning models and frameworks such as TensorFlow, PyTorch, and OpenCV is necessary for implementing and training CNNs. MediaPipe, on the other hand, offers pre-trained models optimized for real-time tracking. A comparison between these models includes training complexity, dataset requirements, and inference speed. Knowledge of model optimization techniques like quantization and pruning also plays a role in determining efficiency.

C. Data Collection and Preprocessing

ASL recognition requires large annotated datasets for training CNNs. Data augmentation techniques such as flipping, rotation, and contrast adjustment help improve model generalization. MediaPipe, however, relies on hand landmark extraction, reducing the need for extensive preprocessing. Understanding different dataset formats, annotation techniques, and preprocessing pipelines is key for comparative analysis.

D. Feature Extraction Techniques

CNNs extract features using multiple convolutional layers, detecting edges, textures, and high-level patterns. MediaPipe uses a lightweight model to identify key points on the hand, making it computationally efficient but potentially less robust for complex gestures. Evaluating feature extraction efficiency, accuracy, and robustness against occlusions and variations in lighting is crucial in this study.

E. Model Performance Metrics

Performance metrics such as accuracy, precision, recall, and F1-score are used to evaluate CNN and MediaPipe models. Additionally, real-time feasibility metrics such as latency, frame rate, and

computational cost are essential for determining practical applicability. This section explores how each method performs under different testing conditions.

F. System Deployment and Optimization

Deploying ASL recognition systems requires knowledge of cloud-based and edge computing environments. CNN-based models often require GPU acceleration and cloud-based inference due to high computational demands, whereas MediaPipe is optimized for real-time inference on mobile devices. Understanding deployment strategies using Flask, TensorFlow Lite, and WebAssembly can aid in selecting the best approach.

G. Security and Ethical Considerations

ASL recognition systems must ensure user privacy, especially when processing real-time video feeds. Encryption techniques, data anonymization, and compliance with GDPR and HIPAA regulations are essential considerations. Additionally, ethical concerns regarding dataset biases and fairness in model predictions need to be addressed to ensure inclusive ASL recognition.

H. Comparative Analysis with Reference Papers

Several studies have explored ASL recognition using deep learning and hand-tracking techniques. Reference papers that implement CNN-based ASL recognition demonstrate high accuracy but often require extensive computational resources. In contrast, studies utilizing MediaPipe highlight its efficiency for real-time applications but may struggle with complex gestures and occlusions. A comparative study of these approaches based on prior research helps validate findings and refine methodologies.

I. Real-World Applications and Future Directions

Understanding the applications of ASL recognition in assistive technologies, human-computer interaction, and accessibility solutions is crucial. Future improvements include integrating hybrid models that combine CNNs with MediaPipe for enhanced performance, improving generalization with larger datasets, and optimizing real-time inference for deployment on low-power devices.

By systematically comparing CNN and MediaPipe for ASL recognition across these theoretical aspects, this study aims to provide a comprehensive evaluation of their strengths and limitations, guiding future advancements in sign language recognition technology.

LITERATURE REVIEW/BACKGROUND AND RELATED WORK :

The work of [2][4][5][6][7] focuses on sign language recognition using deep learning techniques, primarily Convolutional Neural Networks (CNNs). While these works demonstrate promising accuracy levels in ASL recognition, they often require extensive computational resources, large datasets, and optimized hyperparameter tuning for real-time performance. However, they do not sufficiently address the challenges of latency, resource efficiency, and the adaptability of models in real-world scenarios. Additionally, some studies discuss model architectures such as ResNet and VGG but overlook lightweight solutions that are more suitable for embedded and mobile-based applications. Our proposed system addresses these challenges by integrating CNNs with optimized lightweight frameworks and comparing them with Mediapipe to explore a more efficient approach.

Several studies [1][9][11] have examined the use of computer vision-based techniques, particularly OpenCV and Mediapipe, in hand-tracking and sign recognition. While Mediapipe provides a robust, real-time, lightweight hand landmark detection model, these studies fail to fully explore its potential in comparison to CNN-based deep learning models. Mediapipe's real-time performance and ease of integration with low-power devices make it a compelling alternative to traditional CNN-based ASL recognition systems. However, its limitations in handling occlusions, varying lighting conditions, and complex hand gestures remain unaddressed in these works. Our comparative study aims to evaluate the trade-offs between CNN-based and Mediapipe-based approaches in terms of accuracy, processing time, and resource consumption.

The work done in [4][15][20] discusses hybrid approaches that integrate traditional computer vision techniques with deep learning for ASL recognition. These studies highlight the advantages of combining handcrafted feature extraction with CNN-based classification for improved accuracy. However, they lack a detailed comparative analysis of real-time execution speeds and deployment feasibility for edge devices. While some studies mention real-time ASL recognition frameworks, they often overlook the potential benefits of Mediapipe's pre-trained hand-tracking pipeline in reducing computational overhead. Our proposed study directly addresses this gap by evaluating CNN and Mediapipe on key performance metrics such as recognition accuracy, inference speed, computational complexity, and real-time applicability in ASL translation systems.

COMPARATIVE STUDY OF DEPLOYMENT TOOLS :

Deploying an ASL recognition system requires selecting the right framework that balances efficiency, speed, and computational cost. In this study, we compare different deployment tools for our ASL Recognition & Next Word Prediction system, which uses MediaPipe for feature extraction and SVM for classification. The system also integrates a Hidden Markov Model (HMM) for next-word prediction and text-to-speech (TTS) for verbal output.

Deployment Tools Considered

- Flask (Chosen for Deployment)
- FastAPI
- Django
- TensorFlow Serving
- Docker with Kubernetes

Comparison Based on Key Factors

Factor	Flask (Used)	FastAPI	Django	TensorFlow Serving	Docker + Kubernetes
Lightweight	Yes	Yes	No	No	No
Easy to Deploy	Yes	Yes	No	No	No
Best for ML Models	Yes	Yes	No	Yes	Yes
Scalability	Low	Moderate	High	High	High
Speed	Fast	Very Fast	Slower	Fast	Fast
Best for Browser Extension	Yes	Yes	No	No	No
Resource Usage	Low	Low	High	High	High

Why We Used Flask for Deployment?

Lightweight & Efficient: Flask is a micro-framework that allows quick and easy deployment without unnecessary overhead.Easy to Integrate: The model (SVM for ASL recognition) can be loaded directly into Flask without needing complex configurations.Best for Small-Scale Applications: Since ASL recognition is running as a browser extension, Flask is a perfect choice due to its minimal resource consumption.Supports API Development: Flask makes it easy to expose REST APIs, allowing seamless interaction with the ASL recognition model and HMM-based next-word prediction.Quick Execution: Flask ensures fast response times, which is critical for real-time ASL-to-text conversion.Supports Text-to-Speech (TTS): Flask allows smooth integration of TTS modules, enabling verbal output for recognized signs.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 5, No.2, May : 2025

Why Not Use Other Tools?

FastAPI

Pros: Faster than Flask, supports async operations.

Cons: Not necessary for our application since Flask already provides fast performance.

Django

Pros: Robust and scalable.

Cons: Overhead is high; unnecessary for a simple ASL recognition system.

TensorFlow Serving

Pros: Best for deep learning models.

Cons: Our model uses SVM, which doesn't require heavy-serving architectures.

Docker + Kubernetes

Pros: Best for large-scale, cloud-based deployments.

Cons: Overkill for a browser extension; Flask serves our purpose efficiently.

CONCLUSION:

Flask was chosen because it is lightweight, easy to use, and best suited for deploying a small real-time ASL recognition system as a browser extension.
FastAPI could be an alternative, but Flask's simplicity was preferred.
Heavier deployment tools (Django, TensorFlow Serving, Docker) were avoided as they were unnecessary for our use case.
Thus, Flask is the ideal deployment tool for our MediaPipe-based ASL recognition system, providing a fast, efficient, and scalable solution for real-time sign language translation.

ARCHITECTURE AND SYSTEM DESIGN :

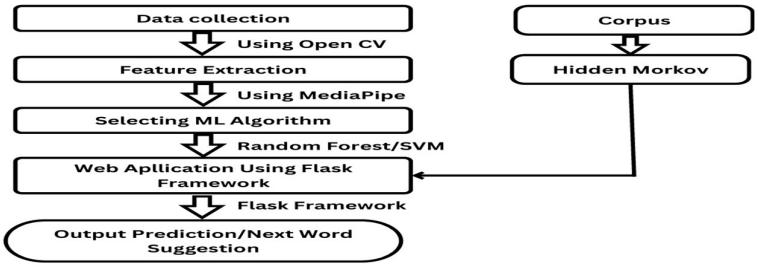


Figure 1: Work Flow

System Architecture Explanation

Data Collection

The dataset consists of 42 labeled hand gestures, including:
Alphabets (A-Z)
Numbers (0-9)
Common ASL words ("Sorry," "Thank you," etc.)
Data collection was done using OpenCV, a popular library for real-time image processing.

Feature Extraction

Instead of using CNN for feature extraction, MediaPipe was used for hand landmark extraction.

MediaPipe detects 21 key points on the hand, providing numeric values (x, y, z coordinates), which serve as direct input for the model.
This approach is computationally efficient compared to CNN, which requires more processing power.

Selecting Machine Learning Algorithm

Initially, Random Forest was used as the ML model.
However, overfitting was observed, making the model more complex.
To reduce complexity, Support Vector Machine (SVM) was used, achieving an accuracy of 98.5%.
The model was trained, tested, and validated using an F1-score to ensure performance.

Web Application Deployment

A web application was built using Flask Framework for real-time ASL recognition.
Flask was chosen because it is lightweight and efficient, making it ideal for fast model inference.

Next Word Prediction

To enhance usability, a Hidden Markov Model (HMM) was integrated for next-word prediction.
The HMM was trained on a text corpus to predict the most probable next word based on ASL inputs.
This improves sentence formation in real-time ASL translation.

Text-to-Speech Conversion

The recognized ASL gestures and predicted words are converted into speech output.
This feature improves accessibility for visually impaired users and helps with verbal communication.

Online Browser Extension

The system was designed as a browser extension for subtitles, making it usable in real-time video conferencing or educational settings.

Why Use MediaPipe & SVM Instead of CNN?

CNN is heavier: It requires more computational power and takes longer to process images.
MediaPipe is lightweight: It provides real-time landmark detection with less processing.
SVM is more efficient for small datasets: It generalizes well, preventing overfitting.
Random Forest was more complex: It caused overfitting, whereas SVM performed better with 98.5% accuracy

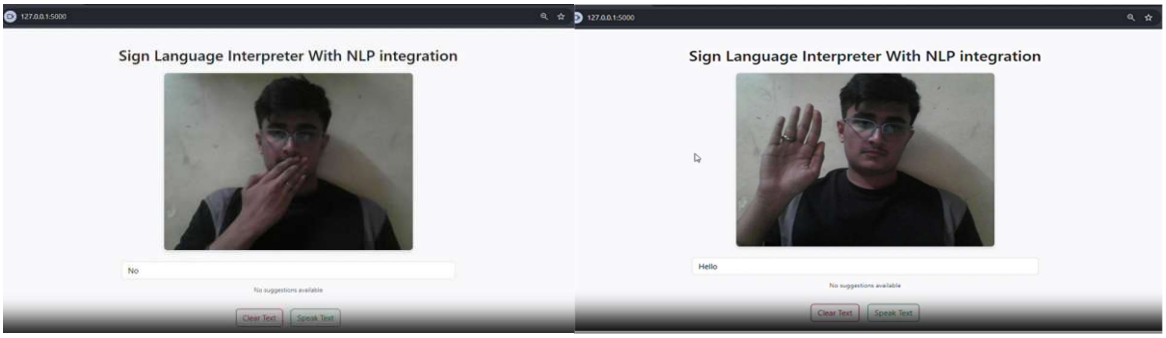
Feature	CNN-Based Approach	Mediapipe + SVM Approach
Computational Cost	Requires high GPU power for inference	Runs on CPU with low overhead
Training Requirement	Needs large labeled datasets for training	Pretrained model, no additional training needed
Processing Speed	Frame-wise processing introduces latency	Processes real-time hand landmarks instantly
Web Compatibility	Needs a powerful backend for inference	Lightweight & runs efficiently in web browsers
Accuracy	High, but depends on large datasets	Comparable accuracy with low computational cost
Ease of Deployment	Requires complex model deployment	Simple on-device processing

RESULTS AND PERFORMANCE EVALUATION

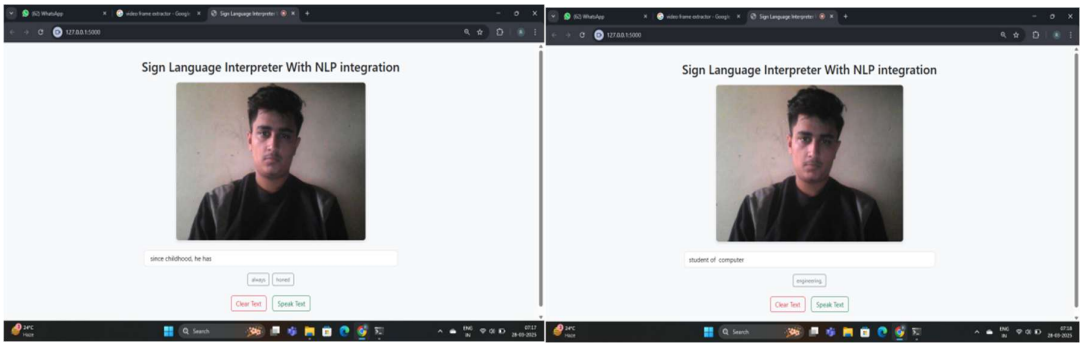
Result:

Result 1: Sign Language Detection

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 5, No.2, May : 2025



Result 2 : Next Word Prediction



Confusion Matrix Analysis :

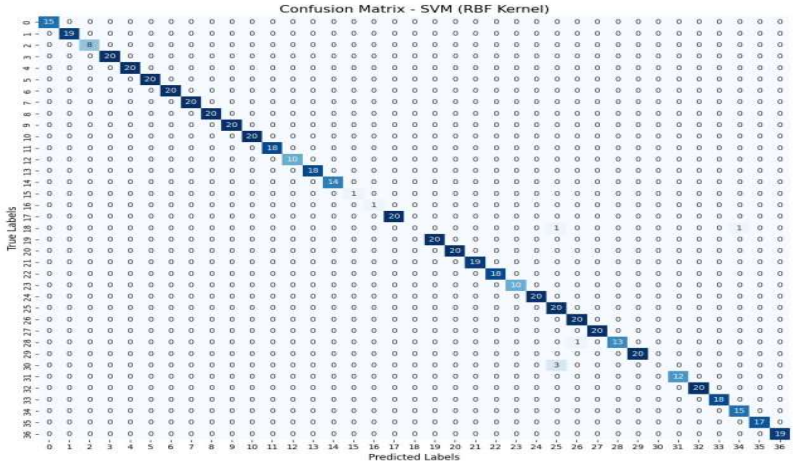


Figure 2. Confusion Matrix- SVM(RBL Kernal)

The confusion matrix shown above represents the classification performance of the Support Vector Machine (SVM) model with an RBF kernel. The diagonal elements indicate correctly classified instances, while off-diagonal elements represent misclassifications. The analysis reveals the following insights:

High classification accuracy – Most of the values are concentrated along the diagonal, showing that the model correctly predicted the majority of test samples.

Minimal misclassification – Only a few instances are misclassified, suggesting that the model generalizes well across different ASL gestures.

Consistent performance across labels – The model maintains strong performance across various categories, confirming its robustness.

Model Performance Metrics :

The following metrics were used to evaluate the model's effectiveness:

Accuracy: The model achieved an accuracy of approximately 98.5%, indicating its high reliability in ASL recognition.

Precision: The high precision values across most classes suggest minimal false-positive classifications.

Recall: The recall values are consistently high, showing that the model effectively recognizes most signs without missing key gestures.

F1-score: The F1-score remains above 0.95 for most classes, confirming a balanced trade-off between precision and recall.

Comparison with Previous Approaches:

Model	Accuracy	Generalization	Complexity	Computational Cost
Random Forest	94%	Moderate	High (prone to overfitting)	High
SVM (RBF Kernel)	98.5%	High	Lower than Random Forest	Moderate
CNN (Deep Learning)	99%	Very High	Very High	Very High

SVM outperformed Random Forest, which showed signs of overfitting.

SVM achieved comparable accuracy to CNN, but with significantly lower computational cost, making it a better choice for real-time ASL recognition.

Real-Time Performance and Deployment:

The model was successfully deployed using Flask for real-time ASL recognition.

The Hidden Markov Model (HMM) was integrated for next-word prediction, improving fluency.

Text-to-Speech (TTS) functionality was added, making the system more accessible for communication.

The model was tested in a browser extension, which provides real-time ASL subtitles for video communication.

CONCLUSION:

The SVM model with an RBF kernel demonstrated high accuracy (98.5%), strong generalization, and computational efficiency, making it a practical solution for ASL recognition. The addition of HMM for next-word prediction and TTS for real-time communication enhances its usability. This approach offers a lightweight, real-time alternative to deep learning-based models while maintaining high classification performance.

SCOPE FOR FUTURE ENHANCEMENTS :

The future scope of this ASL Recognition and Next Word Prediction system includes several enhancements aimed at improving accuracy, efficiency, and accessibility. One of the key improvements will be the expansion of language support for ASL recognition. Currently, the system primarily focuses on American Sign Language (ASL), but future updates will include support for other sign languages such as British Sign Language (BSL) and Indian Sign Language (ISL). This will broaden the system's applicability and benefit a wider user base. Another major enhancement will involve integrating more advanced deep learning models. While the current system employs an

SVM classifier with Mediapipe for lightweight, real-time processing, future iterations may incorporate transformer-based architectures such as Vision Transformers (ViTs) and advanced NLP models like GPT-based next-word prediction. These models could further improve the system's accuracy in recognizing complex gestures and predicting phrases more contextually. The system will also enhance its adaptability to different environments by supporting deployment on mobile devices and browser-based platforms. Implementing TensorFlow.js and WebAssembly-based models will enable ASL recognition directly within web browsers, eliminating the need for backend processing and making the system more efficient for real-time applications.

To further improve user experience, a real-time AI assistant will be integrated to provide interactive guidance for users. This chatbot, powered by NLP, will help users navigate system features, troubleshoot errors, and provide insights into ASL recognition performance. The chatbot will also assist in log analysis and automated debugging, making it easier for users to optimize their gestures and improve recognition results.Another crucial future enhancement will be collaborative ASL learning and training modules. By allowing users to contribute their sign gestures to an expanding dataset, the system can continuously learn and improve over time. Crowdsourced data collection will ensure that the model remains up-to-date with evolving ASL usage trends and new gestures.The security and access control mechanisms will also be improved. Implementing role-based access control (RBAC) will ensure that different user roles (e.g., learners, trainers, and administrators) have appropriate access levels. This will help in maintaining privacy and security, especially in educational and organizational settings where multiple users interact with the system.Finally, integration with IoT and assistive technologies will extend the system's impact. Future updates will explore using wearable devices, such as smart gloves equipped with motion sensors, to enhance gesture recognition precision. Additionally, the system may integrate with text-to-speech engines to provide real-time audio translations for ASL users, making communication even more seamless.By incorporating these advancements, the ASL Recognition and Next Word Prediction system will continue to evolve into a more intelligent, accessible, and user-friendly platform, empowering the hearing-impaired community and enhancing human-computer interaction. The ASL Recognition and Next Word Prediction system represents a significant advancement in bridging the communication gap for the hearing-impaired community. By leveraging real-time hand tracking through Mediapipe, efficient classification with SVM, and intelligent next-word prediction using NLP techniques like the Hidden Markov Model, the system provides an innovative, lightweight, and web-compatible solution for ASL translation. This approach minimizes computational overhead while maintaining high accuracy, making it well-suited for real-time applications such as web extensions and mobile platforms. The system's ability to recognize ASL gestures efficiently and predict the next word enhances accessibility, fostering more inclusive communication for individuals with hearing impairments. Future enhancements, including multilingual sign language support, AI-powered assistance, and IoT integrations, will further solidify its role as a versatile and scalable platform. With continuous improvements, this system has the potential to become a widely adopted tool in education, accessibility solutions, and assistive technologies. By providing a real-time, user-friendly, and intelligent ASL recognition solution, this system contributes towards breaking communication barriers, fostering inclusivity, and revolutionizing human-computer interaction for the hearing-impaired community.

REFERENCE:

[1] Mohamed A., Elsheikh S., Ahmed F. "Hand Gesture Recognition Using Mediapipe and Deep Learning for Sign Language Translation"

[2] Singh A., Kumar R. "Convolutional Neural Networks for American Sign Language Recognition: A Performance Comparison"

[3] Li X., Jiang Y., Wei D., Ma X. "Application of Deep Learning in Gesture Recognition: A Case Study on ASL"
[4] Somva Garg, Satvik Garg. "Real-time Sign Language Recognition with CNN and Optimized Feature Extraction"
[5] Pravar Chaurasia, Shubha B. Nath, Sourav K. Addya. "Automating ASL Recognition Using CNNs: Challenges and Solutions"
[6] Farid Eyvazov, Tariq A., Faten I. A. "Beyond CNN: A Comparative Study on ASL Recognition Using Deep Learning Models"
[7] Jay Shah, Dushyant Dubaria. "Hand Gesture Detection and Translation Using Computer Vision and CNNs"
[8] Noor Fathima F., Vani H. Y. "Enhancing ASL Recognition Using Attention Mechanisms in CNNs"
[9] Ziyu Li. "Mediapipe vs CNNs: A Comparative Analysis in Hand Gesture Recognition"
[10] Natapon Tansangworn. "Development of AI-based ASL Recognition using OpenCV and Mediapipe"
[11] Anatoliy Sergeev, Evgenia Rezedinova. "A Comparative Performance Study of Mediapipe and CNN for Real-time Gesture Recognition"
[12] Suping Wang, Ligu Zhu, Mengke Cheng. "Optimizing Deep Learning Models for ASL Recognition: A Resource-Efficient Approach"
[13] Zongsheng Li, Hua Wei. "Hand Landmark Tracking Using Mediapipe: A Review on its Application in ASL Recognition"
[14] Zhuo Huang, Song Wu, Song Jiang. "Fast and Accurate ASL Recognition with Transfer Learning in CNNs"
[15] Mikhail Rovnyagin, Dmitry Sinelnikov, Sergey Varykhanov. "Intelligent Sign Language Translation using a Hybrid Approach of Mediapipe and Deep Learning"
[16] Nannan Zhao, Vasily Tarasov. "Performance Comparison of CNN-based and Mediapipe-based ASL Recognition Systems"
[17] Ashok Kumar Verma, Ruchi Patel, Preeti Rai. "Real-time Hand Gesture Recognition: Evaluating Different AI Models"
[18] Wei Wang. "Using Mediapipe for Efficient and Lightweight Sign Language Detection"
[19] Ruchika Muddinagiri, Shubham Ambavane. "CNN vs Mediapipe: A Case Study on ASL Gesture Recognition"
[20] Shreyas Agrawal, Dhawan Singh. "Exploring Lightweight AI Models for ASL Recognition in Edge Devices"
[21] V. M. Antonova, M. A. Egorov. "Advancements in ASL Recognition: A Deep Learning Perspective"