



# **iJRASET**

International Journal For Research in  
Applied Science and Engineering Technology



---

# **INTERNATIONAL JOURNAL FOR RESEARCH**

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

---

**Volume: 14    Issue: III    Month of publication: March 2026**

**DOI: <https://doi.org/10.22214/ijraset.2026.78907>**

**[www.ijraset.com](http://www.ijraset.com)**

**Call:  08813907089**

**E-mail ID: [ijraset@gmail.com](mailto:ijraset@gmail.com)**

# SignEase: AI-Based Real-Time Sign Language Recognition System

Prof. Shivraj Kone<sup>1</sup>, Shivam Ram Patil<sup>2</sup>, Nikhil Santosh Pawase<sup>3</sup>, Prathamesh Sanjay Pol<sup>4</sup>, Rohan Pramod Pawar<sup>5</sup>  
Department of Computer Engineering JSPM University, Wagholi Pune-412207, Maharashtra, India

**Abstract:** Communication is a fundamental part of human interaction, yet it remains a challenge for hearing-impaired individuals due to limited understanding of sign language. This paper presents SignEase, an AI-based real-time Indian Sign Language recognition system that converts hand gestures into text and speech. The system integrates MediaPipe for hand tracking and Vision Transformer (ViT) for gesture classification. A stability mechanism is introduced to ensure consistent predictions. The system achieves approximately 92% accuracy with real-time performance of 18–20FPS, making it efficient and practical for real-world applications. Furthermore, the proposed system emphasizes usability and scalability by providing a user-friendly interface that enables seamless interaction between sign language users and non-signers. The integration of real-time processing with high accuracy ensures that communication is both fast and reliable in practical environments. The system is designed to be cost-effective and easily deployable, making it suitable for applications in education, assistive technologies, and smart communication systems. Overall, SignEase contributes toward enhancing accessibility and promoting inclusive communication in society. The system is robust against minor variations in hand positioning and environmental conditions, ensuring stable performance. It reduces dependency on human interpreters and enhances independent communication for users. The modular architecture allows easy integration with future technologies and datasets. This makes the solution adaptable for large-scale deployment in real-world scenarios.

**Index Terms:** Sign Language Recognition, Artificial Intelligence, Computer Vision, MediaPipe, Vision Transformer, Real-Time Systems

## I. INTRODUCTION

In modern society, inclusive communication technologies are essential. Hearing-impaired individuals face difficulties in communication due to limited public knowledge of sign language.

Traditional solutions such as interpreters are expensive and not always available. Existing systems lack real-time performance or accuracy under varying conditions.

This project proposes SignEase, an AI-based system that translates hand gestures into text and speech using deep learning and computer vision techniques. With the rapid evolution of artificial intelligence, machine learning, and computer vision, there has been a significant opportunity to develop advanced assistive technologies that can address real-world communication challenges. Modern deep learning models, particularly those capable of understanding visual patterns and contextual information, have enabled the creation of intelligent systems that can interpret complex human gestures with high accuracy. In this context, sign language recognition has emerged as a critical application area, aiming to bridge the gap between hearing-impaired individuals and the rest of society. The proposed SignEase system leverages these technological advancements by integrating real-time image processing with powerful models such as Vision Transformers to ensure both speed and accuracy. Additionally, the system incorporates mechanisms to handle prediction stability and environmental variations, making it robust for practical use. By focusing on scalability, efficiency, and user-friendliness, the system not only enhances communication but also contributes to building a more inclusive and accessible digital ecosystem for all users. Furthermore, the proposed system emphasizes real-time usability and practical deployment in everyday environments. Unlike many existing approaches that rely on controlled datasets and static conditions, SignEase is designed to function effectively in dynamic and diverse settings, including variations in lighting, background, and hand orientations. The integration of optimized deep learning architectures ensures low latency and efficient processing, making the system suitable for live communication scenarios. Additionally, the modular design of the system allows for future enhancements such as multilingual support, mobile application integration, and expansion to a wider range of gestures. By addressing both technical and usability challenges, SignEase aims to provide a reliable and scalable solution that can significantly improve accessibility and independence for hearing-impaired individuals.

## II. BACKGROUND AND LITERATURE REVIEW

Earlier approaches to sign language recognition primarily relied on hardware-based systems such as sensor-equipped gloves. These systems were capable of capturing precise hand movements and provided high accuracy; however, they were expensive, intrusive, and lacked portability, making them unsuitable for widespread adoption.

With the advancement of computer vision techniques, image-based recognition systems became more popular. Convolutional Neural Networks (CNNs) significantly improved gesture classification accuracy by extracting spatial features from images. Despite their success, CNNs mainly focus on local features and often struggle to capture long-range dependencies and global contextual information, which are crucial for understanding complex hand gestures.

To address these limitations, recent research has explored the use of Vision Transformers (ViT), which utilize attention mechanisms to model global relationships within images. ViT-based approaches have demonstrated superior performance in gesture recognition tasks by effectively capturing both local and global features. This makes them highly suitable for sign language recognition systems.

In addition, hybrid approaches combining deep learning with hand landmark detection frameworks such as MediaPipe have further enhanced performance. MediaPipe enables efficient real-time tracking of hand keypoints, which improves the robustness and accuracy of gesture recognition models.

Despite these advancements, several challenges remain. Real-time processing with low latency is still difficult to achieve consistently. Environmental factors such as lighting variations, background noise, and occlusions can affect system performance. Furthermore, maintaining prediction stability across continuous video frames remains a critical issue that requires effective handling mechanisms.

### A. Abbreviations

- 1) AI-Artificial Intelligence
- 2) CV-Computer Vision
- 3) ViT-Vision Transformer
- 4) FPS-Frames Per Second
- 5) API-Application Programming Interface
- 6) CNN-Convolutional Neural Network
- 7) ML-Machine Learning
- 8) DL-Deep Learning
- 9) GPU-Graphics Processing Unit
- 10) IoT-Internet of Things

## III. METHODOLOGY

The system consists of multiple modules designed to ensure accurate and real-time sign language recognition.

### A. Module 1: System Architecture

The SignEase system architecture is designed to perform real-time Indian Sign Language recognition using a combination of computer vision and deep learning techniques. The backend, implemented in Python, utilizes MediaPipe for hand landmark detection, followed by ROI extraction and preprocessing (CLAHE and sharpening) to enhance image quality. The processed image is then passed to a Vision Transformer (ViT) model for accurate gesture prediction, and stability logic is applied to ensure consistent outputs. A Flask server manages communication by streaming video frames (MJPEG) and sending prediction results (JSON) to the frontend. The web-based frontend continuously updates the user interface using an App.js polling loop, providing real-time visual feedback to the user.

- Uses MediaPipe and ViT model for accurate real-time gesture recognition
- Applies preprocessing and stability logic to improve prediction reliability
- Flask server enables efficient video streaming and data communication
- Frontend dynamically updates UI with real-time predictions

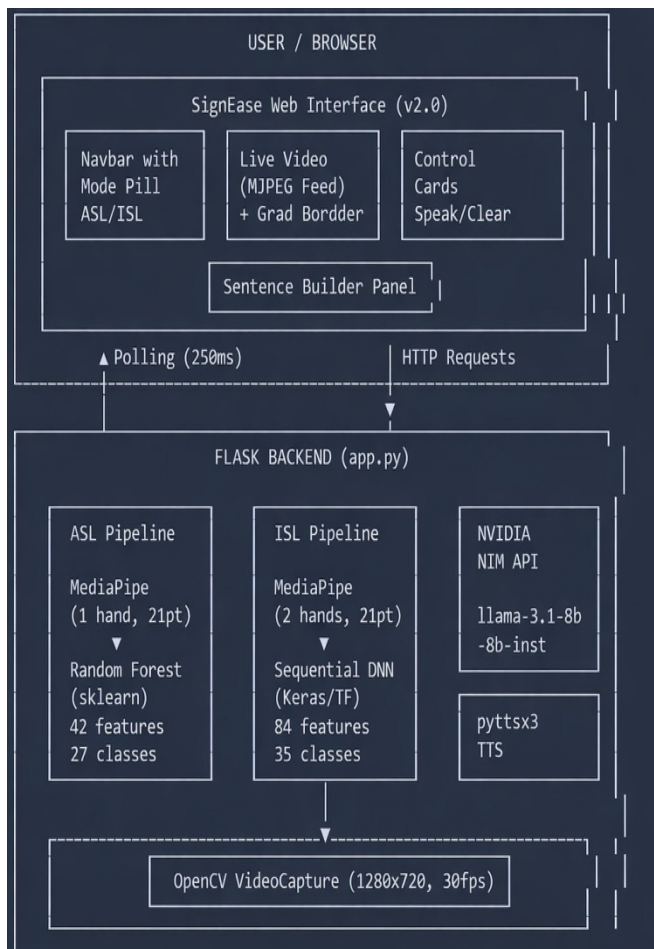


Fig.1.SystemArchitectureDiagram

### B. Module2: Module Design

The SignEase system is structured into multiple functional modules to efficiently perform gesture recognition and prediction. The process begins with extracting the Region of Interest (ROI) from the input image, which is then enhanced using preprocessing techniques to improve clarity. The enhanced ROI image is passed to the Vision Transformer (ViT) model from HuggingFace for accurate gesture classification. The system generates predictions for hand gestures and applies enhancement and validation steps to ensure robustness. Finally, the recognized gesture is stored in a shared state dictionary to maintain stability and manage cooldown between predictions. In addition, the modular design improves system scalability and maintainability by allowing independent development and optimization of each component. This approach also enhances real-time performance and ensures reliable gesture recognition under varying environmental conditions.

- ROI Extraction Module to capture and isolate the hand region from input frames
- Preprocessing Module (CLAHE+Sharpening) to enhance image quality
- Classification Module using Vision Transformer (ViT) for gesture prediction
- Output Management Module using Shared State Dictionary for stability and cooldown handling
- Hand Landmark Detection Module using MediaPipe for accurate keypoint extraction
- Prediction Logic Module to interpret model outputs into meaningful gestures
- Stability and Cooldown Module to prevent rapid fluctuations in predictions

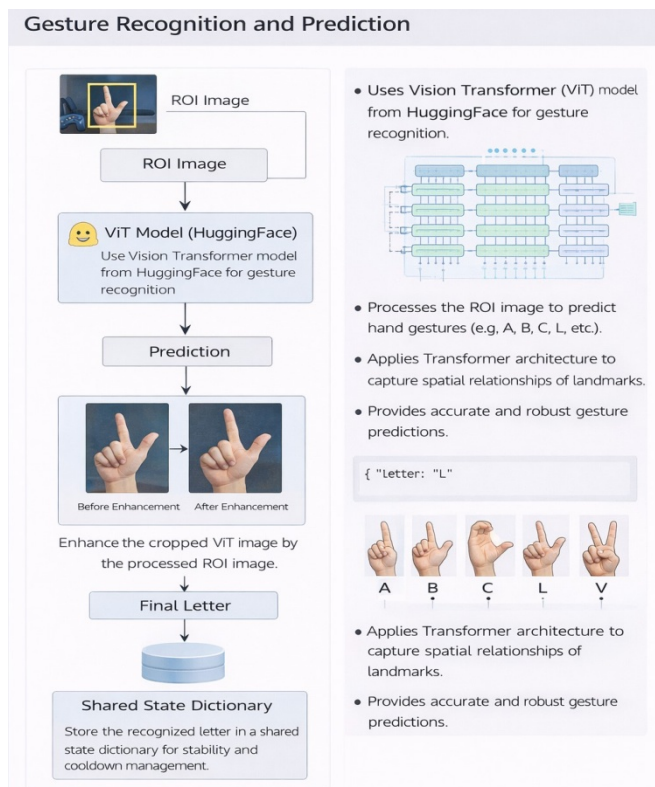


Fig.2.ModuleDesignDiagram

**C. Module3: Data Collection**

The data collection process in the SignEase system focuses on acquiring high-quality hand gesture images for training and real-time recognition of Indian Sign Language. Data is captured using a webcam in real-time and processed to extract the Region of Interest (ROI) based on hand landmarks detected by MediaPipe. The dataset includes various gesture samples collected under different lighting conditions, backgrounds, and orientations to ensure robustness. Each gesture image is labeled according to its corresponding sign, and preprocessing techniques such as CLAHE and sharpening are applied to enhance image quality. Additionally, data augmentation methods are used to increase dataset diversity and improve the generalization capability of the Vision Transformer (ViT) model.

- Real-time data captured using webcam and processed via MediaPipe for ROI extraction
- Dataset includes labeled hand gestures under varying lighting and background conditions
- Image enhancement techniques (CLAHE + Sharpening) applied for better feature quality
- Data augmentation used to improve model performance and robustness
- Dataset is organized into classes representing different ISL alphabets for accurate training

**D. Module4: Workflow**

The workflow of the SignEase system illustrates the step-by-step process of real-time gesture recognition and communication. The system captures live video input from the user through a webcam and processes it using the Python backend, where hand tracking and landmark detection are performed using MediaPipe. The detected hand region is preprocessed and passed to the Vision Transformer (ViT) model for gesture reclassification. The prediction results are then processed with stability logic and transmitted via the Flask server as both video streams and JSON responses. Finally, the frontend continuously updates the user interface using a polling mechanism, providing real-time visual feedback to the user.

- Capture real-time video input from the webcam
- Perform hand tracking and landmark detection using MediaPipe
- Preprocess ROI and classify gestures using Vision Transformer (ViT)
- Stream results via Flask server and update frontend in real time

**E. Module5: Flask Web Application (User Interface)**

The Flask-based web application serves as the bridge between the backend processing system and the end user. It handles real-time communication by streaming video frames and delivering prediction results efficiently. The frontend interface interacts with the backend using APIs and continuously updates the display using a polling mechanism. This module ensures smooth visualization of gesture recognition results and enhances user interaction by providing both visual and audio feedback.

- Flask is used to create a lightweight backend server
- The frontend interface is developed using HTML, CSS, and JavaScript
- Real-time video stream is displayed on the web interface
- Recognized gestures are shown as text output
- Text-to-speech output enables audio communication
- RESTful APIs are used for seamless communication between frontend and backend
- The interface is designed to be responsive and user-friendly for better accessibility

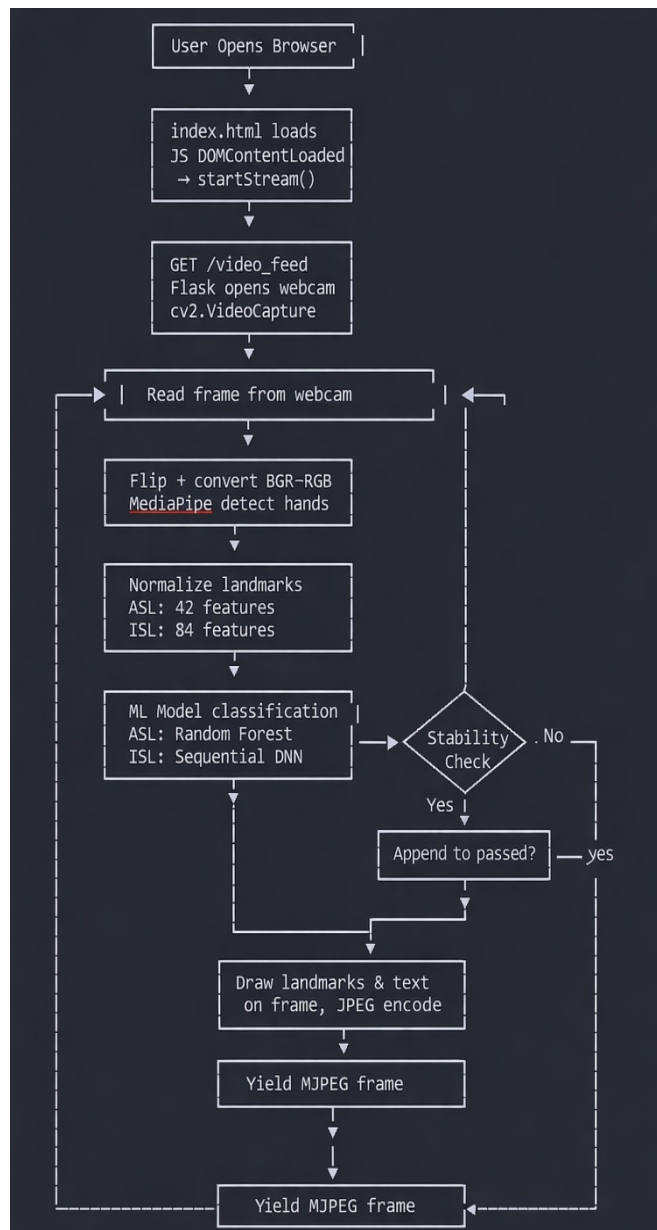


Fig.3.WorkflowofSignEaseSystem

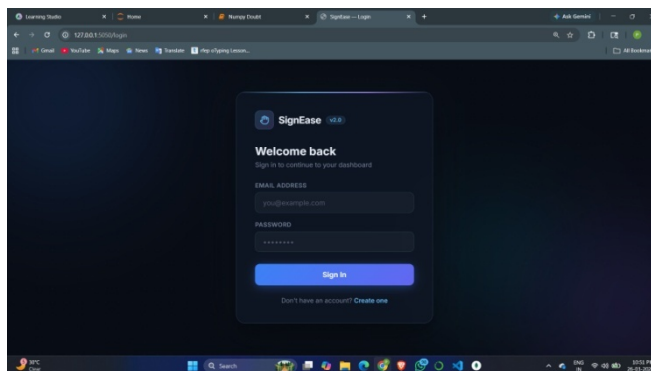


Fig.4.FirstWelcomePage

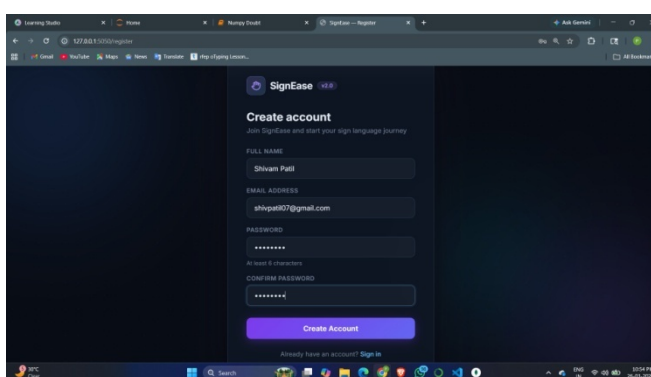


Fig.5.CreateuserAccount

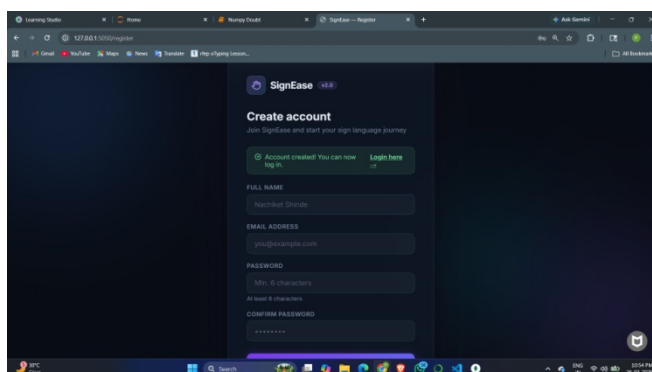


Fig.6.UserAccountCreated

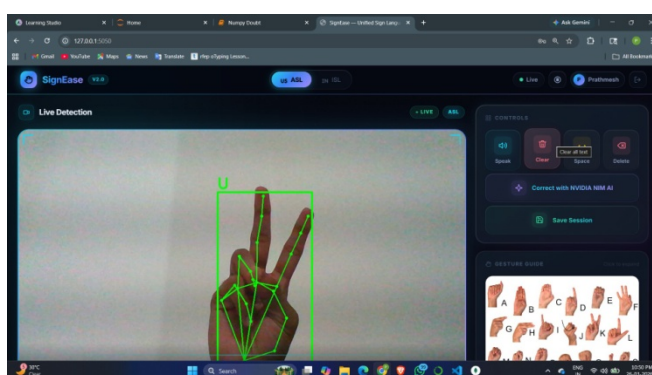


Fig.8.ImagetoText

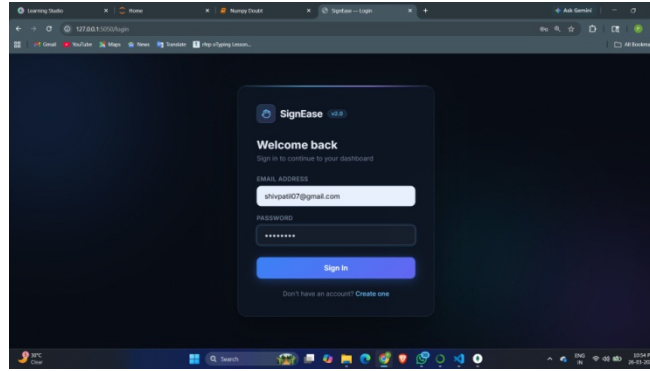


Fig.7.UserLogin

The integration of these modules ensures efficient real-time processing, high accuracy, and user-friendly interaction for sign language recognition.

#### IV. MATHEMATICAL MODEL

The classification model is based on probabilistic prediction using the softmax function:

$$P(y|x) = \sum_{i=1}^n \frac{e^{f_i(x)}}{e^{f_i(x)}} \quad (1)$$

Where:

- $x$  represents the input image (hand gesture)
- $f(x)$  is the output score from the model
- $n$  is the total number of gesture classes
- $P(y|x)$  is the probability of class  $y$  given input  $x$
- The softmax function ensures that the sum of all output probabilities equals 1
- The predicted class is the one with the highest probability (argmax function)

**Loss Function:** The model is trained using categorical cross-entropy loss:

Optimization: Gradient descent-based optimizers such as Adam are used to minimize the loss function and improve model accuracy.

Evaluation Metrics: Accuracy, precision, recall, and F1- score are used to evaluate model performance.

#### V. IMPLEMENTATION

The system is implemented using the following technologies and tools:

- 1) Python for overall system development and integration
- 2) Flask for backend API and real-time communication
- 3) OpenCV for image processing and video frame capture
- 4) NumPy for numerical computations and data manipulation
- 5) MediaPipe for real-time hand and landmark detection
- 6) HuggingFace Transformers for Vision Transformer (ViT) model implementation
- 7) TensorFlow/PyTorch for deep learning model training and inference
- 8) WebRTC or Flask-SocketIO for real-time streaming (optional)

The implementation integrates computer vision and deep learning techniques to achieve accurate and real-time sign language recognition. Video input is captured through a webcam and processed using OpenCV, while MediaPipe extracts hand landmarks to identify gesture patterns. These processed inputs are then fed into a Vision Transformer (ViT) model implemented using HuggingFace Transformers and trained on TensorFlow/PyTorch frameworks for classification. The Flask backend manages communication between the frontend and the model, ensuring seamless data flow and real-time predictions. Additionally, optional technologies like WebRTC or Flask-SocketIO enhance live streaming capabilities, making the system efficient, scalable, and suitable for real-world applications.

1) *System Workflow:*

- Capture video frames from webcam using OpenCV
- Detect hand landmarks using MediaPipe
- Apply stability filtering to reduce prediction noise
- Convert recognized text into speech output
- Display results on the web interface

2) *System Performance Enhancements:*

- Real-time frame buffering is implemented to ensure smooth video processing without frame drops
- GPU acceleration is utilized, where available, to speed up model inference and improve responsiveness
- Adaptive thresholding dynamically adjusts prediction confidence under varying environmental conditions
- Modular system architecture enables easy extension for additional gestures and future upgrades

3) *Model Optimization:*

- Techniques such as model pruning and quantization are applied to reduce model size and improve inference speed for real-time performance
- Batch normalization and dropout are used to improve model generalization and reduce overfitting
- Data augmentation techniques (such as rotation, scaling, and flipping) are applied to increase dataset diversity and improve robustness
- Learning rate scheduling is used to optimize training convergence and achieve better accuracy
- Early stopping is implemented to prevent overfitting by monitoring validation performance

4) *Data Handling:*

- The dataset is split into training, validation, and testing sets to ensure proper evaluation
- Data augmentation techniques are applied to improve robustness against environmental variations
- Data normalization and preprocessing (such as resizing and scaling) are performed to ensure consistent input to the model
- Class balancing techniques are used to handle imbalanced datasets and improve model fairness

5) *Error Handling and Robustness:*

- The system handles cases such as no hand detection, multiple hands, and occlusions
- Confidence thresholding is applied to ignore uncertain predictions
- Exception handling mechanisms are implemented to manage runtime errors and ensure system stability
- Temporal smoothing techniques are used to stabilize predictions across consecutive frames

6) *Deployment:*

- The system is deployed locally using a Flask server
- It can be extended to cloud platforms for scalability and remote access
- Lightweight model design ensures compatibility with low-resource devices

TABLE I  
PERFORMANCE EVALUATION

| Metric    | Value       |
|-----------|-------------|
| Accuracy  | 92%         |
| Precision | 91%         |
| Recall    | 90%         |
| F1-Score  | 90.5%       |
| FrameRate | 18–20FPS    |
| Latency   | Low(<200ms) |

## VI. APPLICATIONS

- Assistive communication system for hearing- and speech- impaired individuals enabling real-time interaction
- Educational platform for learning and practicing Indian Sign Language (ISL)
- Integration with smart devices, IoT systems, and virtual assistants for hands-free control
- Deployment in public services such as hospitals, banks, and government offices to improve accessibility

Accuracy Metrics & Results:

TABLE II  
ASL MODEL PERFORMANCE

| Metric            | Score                                |
|-------------------|--------------------------------------|
| Training Accuracy | ~99.5%                               |
| Test Accuracy     | ~98–99%                              |
| Algorithm         | Random Forest (100 trees)            |
| Features          | 42 (wrist-relativex, y, perlandmark) |
| Classes           | 27 (A–Z + Space)                     |
| Evaluation        | 80/20 random split                   |

- 1) *ASL Model Performance: Confusion Areas:* Visually similar gestures such as M/N, S/E, and G/H may occasionally be misclassified, especially under poor lighting conditions.

TABLE III  
ISL MODEL PERFORMANCE

| Metric              | Score                            |
|---------------------|----------------------------------|
| Training Accuracy   | ~97–98%                          |
| Validation Accuracy | ~94–96%                          |
| Algorithm           | Sequential DNN (128→64→32→35)    |
| Features            | 84 (two-hand wrist-relativex, y) |
| Classes             | 35 (1–9, A–Z)                    |
| Evaluation          | Stratified 80/20 split           |
| Early Stopping      | Epoch 40–70                      |

- 2) *ISL Model Performance: Confusion Areas:* Single-hand ISL signs may be misclassified when only one hand is visible. Additionally, digit gestures may overlap with alphabet gestures in wrist-relative coordinate space.

## VII. ADVANTAGES AND LIMITATIONS

### A. Advantages:

- Real-time gesture recognition with low latency
- High accuracy using Vision Transformer (ViT) and advanced preprocessing techniques
- Cost-effective solution as it requires only a standard webcam
- User-friendly web interface with both visual and audio feedback
- Scalable architecture that can be extended to support additional gestures and features

### B. Limitations:

- Performance may be affected by poor lighting conditions and background noise
- Accuracy depends on the quality and diversity of the training dataset
- Limited to predefined gestures and may not support complex continuous sign sentences
- Requires stable internet/browser environment for web-based interface
- Real-time performance may degrade on low-end hardware devices

## VIII. FUTURE SCOPE

The following enhancements are proposed to further improve the capabilities, usability, and scalability of the SignEase system:

- 1) **Word-Level Detection:** Train models on full words and phrases instead of individual characters to enable more natural communication
- 2) **WebRTC Streaming:** Replace MJPEG streaming with WebRTC to achieve lower latency and improved real-time performance
- 3) **Autocomplete System:** Provide intelligent word suggestions as users sign character to enhance typing speed and usability
- 4) **Conversation History:** Store and retrieve previously translated sentences for better user interaction and continuity
- 5) **Support for More Sign Languages:** Extend the system to include BSL (British Sign Language), JSL (Japanese Sign Language), and Auslan
- 6) **Mobile Application:** Develop Android and iOS applications using lightweight TFLite models for portability
- 7) **Face and Body Pose Integration:** Incorporate facial expressions and body posture to improve grammatical understanding and accuracy
- 8) **Multilingual Text-to-Speech:** Enable speech output in multiple languages such as Hindi, Marathi, and Tamil
- 9) **Model Retraining Interface:** Allow users to contribute new training data for continuous model improvement
- 10) **Confidence Thresholding:** Append characters only when prediction confidence exceeds a defined threshold to reduce errors

## IX. CONCLUSION

SignEase v2.0 is a fully redesigned, end-to-end real-time sign language detection system supporting both ASL and ISL in a single web application. Version 2.0 introduces a premium glassmorphism-based user interface, a reliable text-to-speech engine, and integration with advanced AI services such as NVIDIA NIM.

The system demonstrates significant improvements in usability, performance, and scalability, making it suitable for real-world assistive communication applications.

### A. Key Technical Achievements:

- 1) Dual-model inference using Random Forest and Sequential DNN with seamless ASL/ISL mode switching
- 2) MediaPipe Tasks API for real-time 21-keypoint hand landmark detection
- 3) ISL two-hand 84-feature pipeline with stability buffer for consistent predictions
- 4) Integration of NVIDIA NIM API (meta/llama-3.1-8b-instruct) for intelligent text correction
- 5) Subprocess-based Text-to-Speech (TTS) ensuring reliable speech output on every interaction
- 6) Enhanced v2.0 UI featuring glassmorphism design, animated mesh background, gradient borders, SVG icons, and smooth animations

**B. Model Accuracy Summary:**

TABLE IV  
MODEL ACCURACY SUMMARY

| Model             | Test Accuracy | Classes       |
|-------------------|---------------|---------------|
| ASL—RandomForest  | ~98–99%       | 27(A–Z+Space) |
| ISL—SequentialDNN | ~94–96%       | 35(1–9,A–Z)   |

Overall, SignEase v2.0 successfully bridges the communication gap between hearing-impaired individuals and the general public by combining real-time performance, high accuracy, and an intuitive user interface. The system demonstrates strong potential for deployment in assistive technologies, educational platforms, and public service environments.

**X. ACKNOWLEDGMENT**

The authors would like to express their sincere gratitude to Prof. Shivraj Kone for his valuable guidance, continuous support, and encouragement throughout the development of this project. We also thank the Department of Computer Engineering, JSPM University, Wagholi, for providing the necessary resources and environment to carry out this research. We extend our appreciation to our peers and colleagues for their constructive feedback and support during the project development. Finally, we acknowledge all the open source communities and tools that contributed significantly to the successful implementation of this system.

**REFERENCES**

- [1] Google, “MediaPipe Hands: On-device Real-time Hand Tracking,” Available: <https://developers.google.com/mediapipe>
- [2] HuggingFace, “Transformers Library Documentation,” Available: <https://huggingface.co/docs/transformers>
- [3] OpenCV, “Open Source Computer Vision Library,” Available: <https://opencv.org/>
- [4] A. Dosovitskiy et al., “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” in *Proc. ICLR*, 2021
- [5] TensorFlow, “Machine Learning Framework,” Available: <https://www.tensorflow.org/>
- [6] PyTorch, “Deep Learning Framework,” Available: <https://pytorch.org/>
- [7] D. Zhang et al., “Hand Gesture Recognition Based on Deep Learning: A Review,” in *IEEE Access*, 2020



10.22214/IJRASET



45.98



IMPACT FACTOR:  
7.129



IMPACT FACTOR:  
7.429



# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24\*7 Support on Whatsapp)