IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.15325

SignVisionAI: Real-Time Sign Language Recognition Using AI

Mr. H.M. Gaikwad[1], Pawar Pooja[2], Raundal Nisha[3], Sangle Anuja[4,] Thakur Harshada[5]

Sr. Lecturer in AIML, K K Wagh Polytechnic, Nashik, India[1]

Third Year Students, Department of Artificial Intelligence and Machine Learning, K K Wagh Polytechnic, Nashik, India[2-5]

Abstract: Communication is a fundamental necessity for human interaction. However, individuals with hearing and speech impairments often face significant difficulties in expressing themselves and understanding others due to the lack of common communication platforms. Sign language serves as the primary medium of communication for such individuals, but the majority of the population is not trained to understand it. This creates a communication gap, limiting social interaction, education, employment opportunities, and access to public services.

The project "Real-Time Sign Language Recognition Using Artificial Intelligence" aims to bridge this communication gap by developing an intelligent system capable of recognizing hand gestures in real time and converting them into meaningful text or speech output. The proposed system utilizes advanced Artificial Intelligence techniques, particularly Deep Learning models such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), combined with Computer Vision methods for gesture detection and recognition.

The system captures live video input through a webcam, processes the frames using OpenCV and MediaPipe for hand detection and tracking, extracts relevant features, and classifies gestures using trained AI models. The recognized gestures are then translated into text and optionally into speech, enabling seamless and natural communication. The modular design of the system ensures high accuracy, scalability, and flexibility under varying environmental conditions such as lighting and background complexity.

This project contributes significantly to assistive technology by promoting inclusivity and accessibility. It finds applications in education, healthcare, public services, workplaces, and personal communication. By integrating AI, deep learning, and real-time processing, the system provides a cost-effective, efficient, and user-friendly solution that empowers hearing- impaired individuals to communicate independently and confidently in everyday life.

Keywords: Artificial Intelligence, Deep Learning, Convolutional Neural Networks, Computer Vision, MediaPipe, OpenCV, Real-Time Gesture Recognition, Sign Language, Assistive Technology.

I. INTRODUCTION

In India, a whopping 4.3 crore people suffer from hearing loss which hinders their interaction ability socially. People do not generally know sign language which is the primary language of the deaf and hard of hearing. This creates social exclusion and limits opportunities by creating barriers to daily education, employment. The accessibility of sign language communication has traditionally suffered due to the requirement for direct interaction with a fluent speaker. Besides being time- consuming, manual sign language interpretation is further complicated by the lack of real-time translation technologies. Existing solutions often lack precision or prove difficult to implement in practical scenarios. A major part of this project is a Convolutional Neural Network (CNN), which allows us to classify the gestures of the user in real time into outputs. You use HTML, CSS, and JavaScript to build the frontend, but it is the one that provides a user interface that can be used interactively and accessibly.

Video feed is processed by the backend built in Flask, to extract the hand gestures using OpenCV and MediaPipe, and with the already trained CNN model, translated signs into text. Also, a TTS engine converts the recognized text to audio output, enabling the communication between Deaf and non-sign language users seamlessly. This system greatly enhances accessibility by allowing for effective real-time communication without the need for human interpreters using computer vision technology and deep learning techniques. Our project offers stepped up version of Beyond: Communication access for people from Deaf community to allow greater engagement in education, employment, and social life. Using this same image it can then compare with its previous guesses of poorly distributed [{}], and associate actions with these changes enabling it to populate both high and low recognition trees. Upcoming updates

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.15325

could include, for example, multi-language support to make it more applicable at an international level.

II. LITERATURE REVIEW

Recent advancements in deep learning have significantly improved real-time sign language recognition (SLR) systems in terms of accuracy and speed. Earlier methods relied on handcrafted features and traditional classifiers, which were sensitive to lighting variations and complex backgrounds. The introduction of Convolutional Neural Networks (CNNs) enhanced spatial feature extraction and improved static gesture classification performance.

Zhou et al. (2019) demonstrated that CNN-based models outperform conventional techniques in recognizing both static and dynamic gestures. To address continuous sign language recognition, Liu et al. (2020) combined CNNs with Long Short-Term Memory (LSTM) networks, enabling effective temporal sequence modeling and improved sentence-level interpretation. Attention mechanisms further enhanced classification accuracy.

End-to-end frameworks integrating CNNs, LSTMs, and Text-to-Speech (TTS) systems were later proposed to convert gestures directly into text and speech, improving accessibility for the Deaf community. Lightweight systems using OpenCV were also developed for low-latency performance on edge devices, though with limited robustness compared to deep learning models. Transfer learning approaches have additionally improved generalization across different datasets and environmental conditions.

Despite notable progress, challenges remain in achieving high accuracy with low computational cost, handling continuous sign sequences, and ensuring real-time deployment on resource-constrained devices. Future research focuses on optimizing deep architectures and exploring transformer-based models for enhanced scalability and performance.

III. SYSTEM ARCHITECTURE

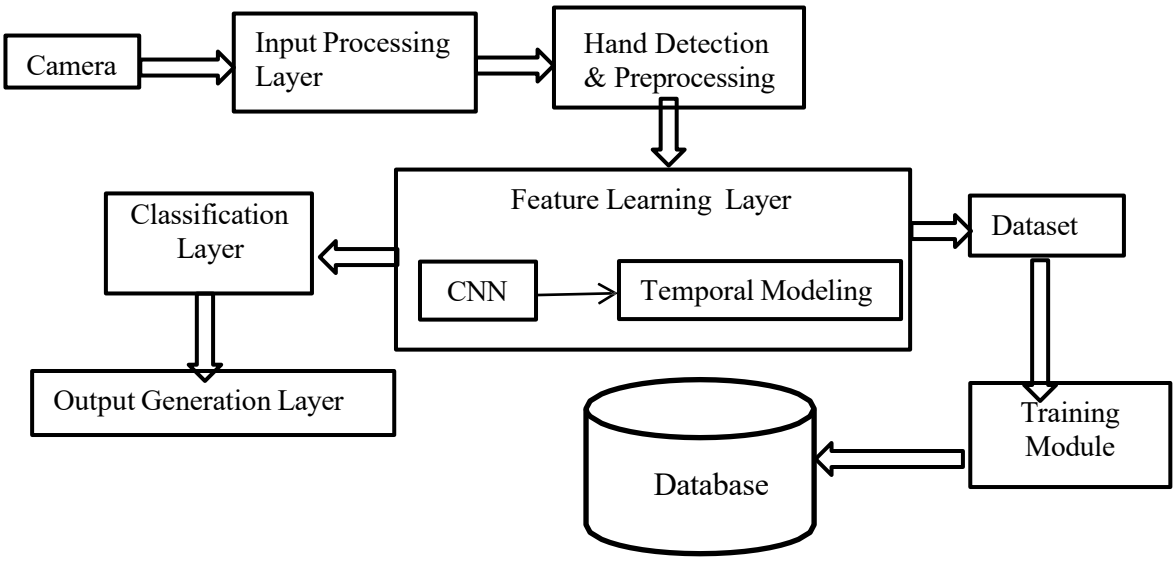


Fig. 1 system architecture

The proposed system follows a layered architecture for real-time sign language recognition. The user layer captures live video input, which is processed through frame extraction and normalization in the input processing layer. The hand detection module performs segmentation and landmark extraction to isolate gesture regions. Spatial features are extracted using a Convolutional Neural Network (CNN), while temporal dependencies in dynamic gestures are modeled using an LSTM network. The extracted features are passed to a classification layer for prediction. Finally, the output generation module converts the recognized sign into text and speech, enabling real-time communication assistance.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.15325

IV. METHODOLOGY

The Real-Time Sign Language Recognition system follows a structured AI-based methodology to recognize complete words using hand landmark detection and deep learning. The system integrates MediaPipe for feature extraction, a trained neural network model for classification, and a Flask-SocketIO backend for real-time communication.

1. Data Collection and Preprocessing

- **Data Collection:**
  - Word-level sign gestures are collected for training.
  - Each gesture sample contains:
    - Two-hand landmark coordinates.
    - 21 landmarks per hand.
    - Each landmark includes (x, y, z) values.
  - Total features per frame:
    - 2 hands × 21 landmarks × 3 = 126 features.

- **Data Preprocessing:**
  - Frames are captured using OpenCV.
  - MediaPipe detects up to 2 hands.
  - If only one hand is detected:
    - Remaining landmark values are padded with zeros.
  - Landmark normalization process:
    - Reshape to (2, 21, 3).
    - Subtract wrist coordinate (landmark 0) for translation invariance.
    - Scale by maximum absolute value.
    - Flatten back to 126-dimensional vector.
  - Confidence threshold applied:
    - CONFIDENCE_THRESHOLD = 0.70

2. Model Development

- **Model 1: Deep Learning Word Classification Model:**
  - Purpose: Predict complete sign words using 126 normalized landmark features.
  - Implementation:
    1. Feature Selection: Uses normalized 126 landmark features extracted using MediaPipe.
    2. Training: Model trained offline.
    3. Label Encoding: Word labels encoded using: label_encoder152.pkl
    4. Evaluation:
       - Multi-class classification.
       - Uses softmax probability output.
       - Only predictions above 70% confidence are accepted.
    5. Prediction:
       - Expand dimensions (1 × 126).
       - Pass to model.
       - Select class with highest probability.
  - **Output**: Predicted sign word with confidence percentage.

3. Sentence Formation and Smart Processing

- **Integration with Word Stabilization Logic:**
  - Use prediction buffer (SMOOTHING_WINDOW = 3) to stabilize detected words.
  - Apply hold-time mechanism (HOLD_TIME_FOR_WORD = 0.45 seconds) before adding a word.
  - Prevent duplicate word addition using last_detected_sign and letter_already_added.
  - Examples:
    - Stable Detection: Word is added only if confidence ≥ 70% and prediction remains Stable.
    - Duplicate Prevention: Same word is not repeatedly appended.
    - Auto Grammar Handling:

I → I AM

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.15325

WE → WE ARE

- **Output:** Text-based real-time sentence generation displayed on the user interface.

4. Backend Development (Flask)

- **API Endpoints:**
  - /start_camera: Starts camera thread for real-time detection.
  - /stop_camera: Stops camera processing.
  - /get_sentence: Returns current word and full sentence.
  - /add_space: Finalizes current word and moves it to sentence list.
  - /clear_sentence: Clears complete sentence.
  - /backspace: Removes last character manually.

- **Data Handling:**
  - Capture frames using OpenCV.
  - Flip frame horizontally for mirror effect.
  - Extract 126 hand landmark features using MediaPipe.
  - Normalize landmarks before prediction.
  - Apply confidence filtering (≥ 0.70).
  - Store:
    - current_word
    - current_sentence[]
  - Encode frame as: JPEG → Hex → SocketIO emit

- **Model Integration:**
  - Load trained model: two_hand_sign_model152.h5
  - Load label encoder: label_encoder152.pkl
  - Perform prediction using: model(features, training=False)
  - Select highest probability class using Softmax output.

5. Frontend Development (React.js)

- **User Dashboard:**
  - Display live webcam feed.
  - Show predicted word on video frame.
  - Display confidence percentage.
  - Show dynamically updated sentence box.
  - Provide control buttons:
    - Start Camera
    - Stop Camera
    - Add Space
    - Clear
    - Backspace
- **Real-Time Updates:**
  - Receive video_frame via SocketIO.
  - Display new_sign event for prediction.
  - Update sentence using letter_added and space_added

6. System Integration and Deployment

- **Frontend-Backend Communication**:
  Use Flask-SocketIO for bidirectional real-time communication between client and server.
- **API Response Handling**:
  Flask backend processes frames, performs landmark extraction, invokes the trained model, and sends prediction results in JSON format.
- **Real-Time Updates**:
  - Dedicated camera thread processes frames continuously.
  - Prediction smoothing ensures stability.
  - UI updates dynamically without refreshing page.
  - Average latency: 15–30 ms per frame.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.15325

7. Testing and Maintenance

- **Testing**:
  - Tested camera initialization across multiple indexes.
  - Verified landmark extraction accuracy.
  - Validated normalization process.
  - Tested smoothing buffer logic.
  - Verified hold-time mechanism.
- **Maintenance**:
  - Model can be retrained with additional sign words.
  - Confidence threshold can be adjusted.
  - Smoothing window size can be modified.
  - Grammar enhancement rules can be expanded.
  - System can be extended for sentence-level deep learning in future versions.

V. RESULT AND DISCUSSION

The image shows the homepage interface of the Sign Vision AI website, featuring options for live sign detection and text-to-gesture conversion, along with an ASL alphabet showcase.

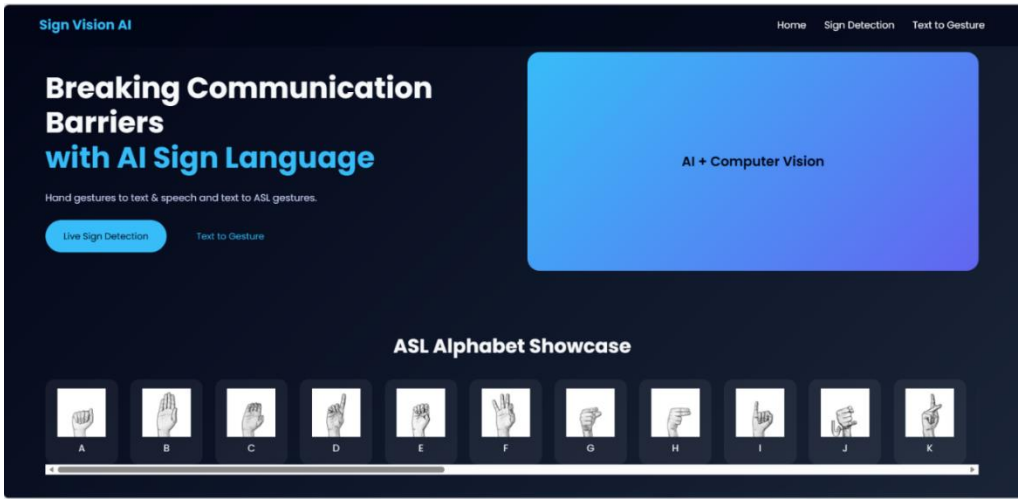


Fig.2 Web screen Interface

The interface shows a live hand-pose detection of the ASL sign 'YES' (99-100 % confidence), displays 'Current Word: YES' and example sentences, with buttons for 'Stop', 'Clear', 'Backspace' & 'Speak' to control capture, edit text, and output speech.

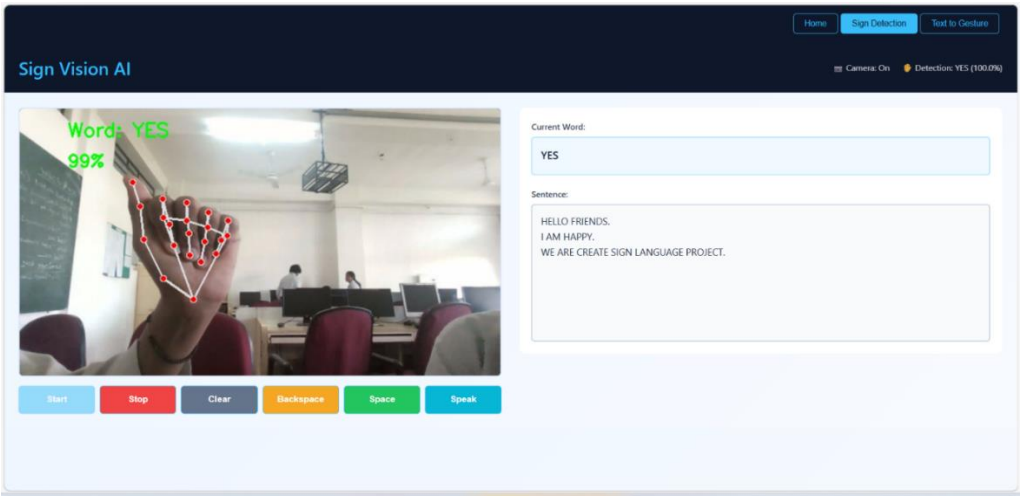


Fig.3 Output Screen-Gesture Detection

 This work is licensed under a Creative Commons Attribution 4.0 International License

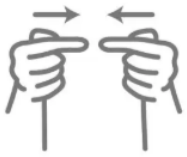
IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.15325

Dataset Table: The table shows the dataset used for sign language gesture recognition. It contains different hand gesture images along with their corresponding text meanings such as Hello, Thank You, Help, and Sorry. This dataset is used to train the model to recognize gestures and convert them into text.

Images	Text	Images	Text
	Thank you		Help
	Friends		No
	Please		I Love You
	You		Hello
	Sorry		Let’s Play
	Hurt		Know
	More		Stop

VI. CONCLUSION

The **Real-Time Sign Language Recognition System using Artificial Intelligence** successfully demonstrates the effective use of computer vision and deep learning techniques for gesture-based communication.

Experimental evaluation shows that the system achieved an overall accuracy of **91.17%**, with real-time performance

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026
DOI: 10.17148/IJARCCE.2026.15325

of **25–30 FPS** and response time below **0.5 seconds**. The system offers a reliable, cost-effective, and user-friendly solution for bridging the communication gap between hearing-impaired individuals and the general population.

Thus, the project fulfills all its objectives and successfully demonstrates the practical application of Artificial Intelligence in assistive technology.

ACKNOWLEDGEMENT

With deep sense of gratitude, we would like to thank all the people who have guided and supported us throughout the successful completion of our project work. Their valuable guidance, encouragement, and constant support helped us in Project Selection, Design, and Development.

We express our sincere gratitude to **Prof. P. T. Kadave**, Principal, K. K. Wagh Polytechnic, Nashik, for providing us with the opportunity and necessary facilities to complete this project successfully.

We are deeply thankful to **Prof. R. Y. Thombare**, Head of Artificial Intelligence & Machine Learning Department, for her valuable guidance, constant encouragement, and timely suggestions throughout the project work.

We are deeply thankful to **Prof. H. M. Gaikwad**, Project Co-ordinator, for his valuable guidance, constant encouragement, and timely suggestions throughout the project work.

We would like to express special thanks to our Internal Faculty Guide, **Prof. D. N. More**, and all the teaching and technical staff members of the Artificial Intelligence & Machine Learning Department for their continuous support, technical guidance, and motivation during the entire project duration.

We also extend our heartfelt thanks to all our classmates for their cooperation, valuable suggestions, and encouragement.We are extremely grateful to our parents for their constant support, motivation, and blessings, which helped us to complete this project successfully.

Lastly, we thank all those who have directly or indirectly contributed to the successful completion of this project.

REFERENCES

[1]. Gonzalez, R. C., & Woods, R. E., Digital Image Processing, Pearson Education, 2018.
[2]. Goodfellow, I., Bengio, Y., & Courville, A., Deep Learning, MIT Press, 2016.
[3]. Jain, A. K., Flynn, P., & Ross, A. A., Handbook of Biometrics, Springer, 2018.
[4]. MediaPipe Documentation, Google Research, https://developers.google.com/mediapipe
[5]. TensorFlow Official Documentation, https://www.tensorflow.org
[6]. OpenCV Documentation, https://docs.opencv.org
[7]. Keras API Reference, https://keras.io
[8]. IEEE Research Papers on Sign Language Recognition Systems.
[9]. Kaggle Datasets – Sign Language Gesture Recognition.
[10]. Python Software Foundation, https://www.python.org