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 10, October 2026

DOI: 10.17148/IJARCCE.2026.1510133

Sign-Speak: Real-Time Continuous Indian Sign Language Translator

Rashmi N D[1], Yashaswini H L[2], Sinchana M S[3], Madhushree M[4], Lavanya N L[5]

Student, Dept. of Computer Science and Engineering, East West College of Engineering, Bengaluru, India[1]

Student, Dept. of Computer Science and Engineering, East West College of Engineering, Bengaluru, India[2]

Student, Dept. of Computer Science and Engineering, East West College of Engineering, Bengaluru, India[3]

Project Guide, Dept. of Computer Science and Engineering, East West College of Engineering, Bengaluru, India[4]

Project Guide, Dept. of Computer Science and Engineering, East West College of Engineering, Bengaluru, India[5]

Abstract: Communication between users of Indian Sign Language (ISL) and people who do not understand sign language can be challenging. Computer vision and machine learning provide opportunities to develop assistive systems that recognize visual gestures and present their predicted meanings in accessible formats. This paper presents Sign-Speak, a web-based sign-language recognition and translation application that integrates real-time camera input, hand-landmark visualization, gesture classification, text output, speech controls, and uploaded-video processing. The application provides a user interface for viewing the camera feed, observing the predicted sign and confidence score, and accessing translation results. It also supports uploading video files in MP4, AVI, and MOV formats, subject to the interface's displayed 50 MB limit. The system uses a trained machine-learning model within a Python-based application. The implementation combines computer-vision processing and a web interface to provide an accessible way to experiment with sign recognition. The application was launched and its interface and workflows were observed during project testing. Quantitative accuracy, processing speed, and generalization performance must be established through controlled experiments before they can be reported as measured results. The proposed system serves as a prototype for assistive communication and can be extended with a larger vocabulary, improved temporal recognition, and more comprehensive evaluation.

Keywords: Indian Sign Language, Sign Language Recognition, Computer Vision, Deep Learning, Hand Gesture Recognition, LSTM, Real-Time Processing, Text-to-Speech, Assistive Technology.

I. INTRODUCTION

Communication is an essential part of everyday life. People who use sign language may face communication barriers when interacting with individuals who have limited knowledge of sign language. Sign language recognition systems aim to reduce these barriers by interpreting visual signs and converting their predicted meanings into text or speech.

Indian Sign Language is used by members of the Deaf community in India. Recognizing signs through a computer system is challenging because gestures can vary with hand orientation, movement, camera position, lighting conditions, and the person performing the sign. Systems must process visual information and classify gestures while providing useful feedback to the user.

Recent developments in computer vision and deep learning have made camera-based gesture recognition increasingly practical. A camera can capture hand movements without requiring specialized wearable sensors. Hand-landmark representations can provide structured information about hand shape and position, while temporal models can learn patterns across video frames.

This paper presents Sign-Speak, a web-based application developed to demonstrate sign-language recognition and translation. The application contains a real-time camera interface and a separate video-upload interface. Its live interface displays the camera feed, detected hand landmarks, a predicted sign label, a confidence value, and translation text. Speech-related controls are also provided to support audio output.

The video-upload interface allows a user to submit a supported video file and view the translation result. These features bring camera-based recognition and video-based processing together in a single application.

 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 10, October 2026

DOI: 10.17148/IJARCCE.2026.1510133

The current system is a prototype. Its demonstrated interface and functions should not be interpreted as evidence that it recognizes every Indian Sign Language sign or translates arbitrary continuous sign-language sentences accurately. Its actual vocabulary and recognition performance depend on the trained model and the implemented processing pipeline.

II. BACKGROUND CONCEPTS

A. Indian Sign Language Recognition

Sign-language recognition uses visual information to identify signs performed by a signer. Depending on the task, a system may classify an isolated sign, recognize a sequence of signs, or translate continuous signing into natural-language text. These are different levels of difficulty and should be evaluated separately.

An isolated-sign classifier predicts a label for a gesture or gesture sequence. Continuous sign-language recognition must additionally identify sign boundaries and distinguish meaningful sequences. Sign-Speak provides a recognition interface, but claims about continuous sentence-level translation require evidence from its implementation and testing.

B. Computer Vision and Hand-Landmark Detection

Computer vision enables software to process images and video captured by a camera. A video stream is made up of successive frames, each of which can be processed to detect visual features.

Hand-landmark detection represents a hand using key points around its joints and fingertips. These points can describe hand shape and finger arrangement. The live camera screenshot of Sign-Speak shows landmark points and connecting lines over the detected hand.

Landmark-based processing can reduce dependence on raw image appearance, although recognition can still be affected by occlusion, poor lighting, fast movement, and inaccurate landmark detection.

C. Deep Learning and Temporal Models

Deep-learning models learn patterns from examples during training. For gesture recognition, the input may consist of image features or sequences of features extracted from video frames.

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network designed to learn dependencies in sequential data. They can be used for gesture recognition when the model receives a sequence of frame-level features.

The project includes trained model files associated with sign recognition. The exact input dimensions, network layers, training configuration, and feature representation should be documented from the model and source code rather than assumed.

D. Text-to-Speech

Text-to-speech (TTS) converts text into spoken audio. It can improve accessibility by allowing recognized text to be heard instead of only read. Sign-Speak includes a TTS toggle and a “Speak All” control in its interface.

The presence of these controls does not by itself establish which speech engine or languages work successfully. Supported voices and languages should be verified through testing.

E. Web-Based Application Design

A web application provides a user interface through a browser. It can organize camera controls, translation panels, file-upload components, and output controls into a single interface.

Sign-Speak displays a real-time camera section and a separate video-upload section. This design enables users to access different input workflows without needing to interact directly with model files or command-line tools.

III. LITERATURE REVIEW

The Existing research investigates sign-language recognition using handcrafted image features, convolutional neural networks, landmark extraction, and temporal deep-learning models. The following studies provide background for the design of Sign-Speak.

A. Indian Sign Language Recognition Using SURF, SVM, and CNN

Katoch, Singh, and Tiwary studied an Indian Sign Language recognition system using Speeded-Up Robust Features (SURF), Support Vector Machines (SVM), and Convolutional Neural Networks (CNN). Their work addressed recognition of ISL alphabets and digits from live video, with predicted outputs presented as text and speech [1].

The study demonstrates the use of computer-vision features and classification techniques for sign recognition. However, recognizing alphabets and digits does not automatically solve the more complex problem of interpreting continuous signing.



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026
DOI: 10.17148/IJARCCE.2026.1510133

B. ISL Recognition Using MediaPipe and LSTM

A 2023 conference paper titled Indian Sign Language (ISL) Recognition and Translation using MediaPipe and LSTM investigated combining MediaPipe-based landmark extraction with LSTM-based recognition. The work also discussed speech input and sign-related output as part of a communication-oriented application [2].

This research is relevant because it illustrates how structured landmark features and sequential learning can be combined in a sign-language system. Its complete capabilities and results should be compared with Sign-Speak only after checking the methods and evaluation details of both systems.

C. Word-Level Recognition Using MediaPipe and LSTM-GRU

A 2024 IEEE conference paper, Word Level Sign Language Recognition Using MediaPipe and LSTM-GRU Network, proposed a hybrid sequential model for word-level ISL recognition. The authors reported a test accuracy of 89.50% for their evaluated dataset [3].

The work highlights the importance of temporal information and structured visual features. Its reported accuracy belongs to that study's dataset and experimental setup; it must not be attributed to Sign-Speak.

D. Real-Time Sign-Language Interpretation

Mariappan, Murugesan, and Selvan presented a real-time interpreter for short sentences in Indian Sign Language using MediaPipe and deep learning. Their study used a custom video dataset and investigated the recognition of meaningful short sentences [4].

This research shows the difference between classifying individual signs and interpreting sequences intended to convey a sentence. Sentence-level interpretation generally requires appropriate data, temporal modelling, and evaluation of the complete output.

E. Research Gap

The reviewed studies demonstrate several approaches to sign recognition, but their datasets, vocabularies, models, and evaluation methods differ. A practical application must also combine recognition with a usable interface and accessible output.

Sign-Speak addresses the application-integration aspect by providing a browser-based interface for live camera input, predicted labels, confidence display, video upload, translation results, and speech-related controls. Further testing is needed to establish the reliability of its recognition, the quality of its uploaded-video output, and its performance across different users and environments.

IV. COMPARATIVE ANALYSIS AND RESEARCH GAP

Different sign-recognition approaches have different strengths and limitations. Table I compares their general characteristics with the intended functionality of Sign-Speak.

TABLE I. COMPARATIVE ANALYSIS OF REVIEWED SYSTEMS

Approach	Main characteristics	Limitations
Manual interpretation	Human interpreter understands signs in context	Requires access to a suitable interpreter
Handcrafted image features	Uses manually designed visual descriptors and classifiers	May be sensitive to changes in appearance and movement
CNN-based recognition	Learns visual patterns from image data	A frame-based model may not fully represent temporal movement
Sequential deep learning	Learns patterns across sequences of features	Depends on appropriate sequential training data and model design
SignSpeak prototype	Combines a web interface, live camera, landmark display, trained recognition model, video upload, and output controls	Vocabulary coverage and recognition quality must be established experimentally

A. Identified Research Gaps

- **Communication accessibility:** Recognition output should be easy for users to view and, where supported, hear.
- **Real-time processing:** Camera-based systems must process incoming frames without excessive delay.
- **Gesture variability:** Differences in hand orientation, lighting, background, and signing style can affect predictions.



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026
DOI: 10.17148/IJARCCE.2026.1510133

- **Video interpretation:** Repeated frame-level predictions can produce repetitive output if temporal segmentation and duplicate suppression are insufficient.
- **Integrated usability:** Recognition, file upload, text output, and speech-related features should be accessible from one application.

B. Contribution of Sign-Speak

Sign-Speak provides a prototype that combines live camera recognition and uploaded-video processing in one browser-based interface. It displays the predicted sign and confidence value and provides controls for camera operation, translation output, and speech-related functionality.

The contribution of the project is its integrated application workflow. Claims of superior accuracy, improved performance over other systems, or reliable continuous sentence translation require controlled comparative experiments and are not assumed in this paper.

V. PROPOSED SYSTEM ARCHITECTURE

The proposed architecture describes the main processing stages of Sign-Speak. The precise internal implementation of feature extraction and model inference must be aligned with the project source code before final submission.

A. System Overview

The system accepts visual input through two workflows: a live camera stream or an uploaded video file. The input is processed by the application's video-processing and recognition pipeline. The predicted result is then presented in the web interface.

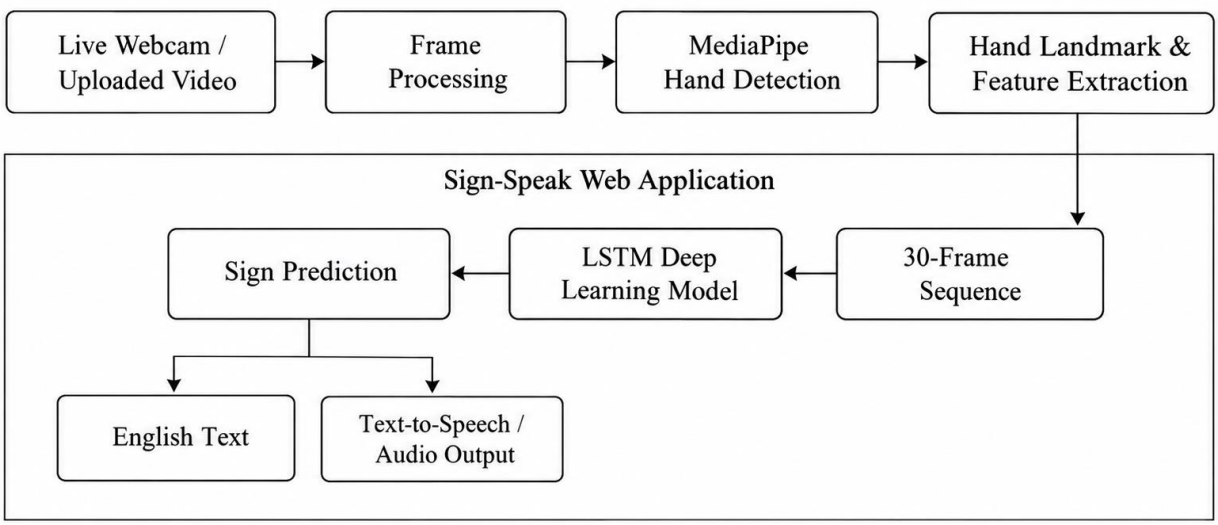


Figure 1. Proposed Sign-Speak system architecture

B. Live Camera Input

The live-camera interface enables the user to start and stop camera processing. The camera feed is displayed in the left panel of the application. When a hand is detected, landmarks can be visualized over the camera image.

The camera workflow allows the user to observe the input and the system's predicted output. Its behaviour depends on camera availability, lighting conditions, hand visibility, and the model's supported classes.

C. Video Upload

The upload interface accepts MP4, AVI, and MOV files and displays a maximum file size of 50 MB. The user can choose a file through the “Browse Files” control or use the drag-and-drop area.

After processing, the application presents a translation result. The quality of this result depends on the recognition pipeline and its ability to process the temporal sequence of the uploaded video.

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 10, October 2026

DOI: 10.17148/IJARCCE.2026.1510133

D. Recognition and Output

The recognition component uses the project's trained model to generate a predicted class. The application displays the predicted sign and confidence score in the live interface. The translation panel displays text output, and the interface provides speech-related controls.

The confidence score is the model or application confidence value displayed by the interface. It should not be interpreted as a guarantee that the prediction is correct.

VI. METHODOLOGY

A. A. Data Collection and Preprocessing

A supervised sign-recognition system requires examples associated with known sign labels. Video-based recognition may involve extracting frames or other visual representations from recorded gestures.

The Sign-Speak application uses trained model files for recognition. The original dataset, number of samples per class, participant details, train-validation-test split, and data augmentation procedure must be obtained from the training code or project documentation. These details should not be guessed.

During inference, the application processes live camera input or an uploaded video according to its implemented pipeline. Preprocessing may include resizing, frame selection, or feature extraction if those operations are present in the source code.

B. Feature Extraction and Representation

The live interface shows points and connections over a detected hand, indicating that hand-landmark visualization is part of the displayed workflow. A landmark-based recognition system represents hand structure using coordinates or other derived features.

The actual feature vector, coordinate normalization, number of frames per prediction, and missing-landmark handling should be verified from the implementation. Only the features used by the trained model should be described as its input.

C. Model Training and Gesture Classification

The project contains trained model files and a label-encoder file. During recognition, the model produces an output that the application associates with a sign label.

If the final implementation uses an LSTM model, its purpose is to learn temporal dependencies between sequential input features. The exact number of LSTM layers, hidden units, loss function, optimizer, and training epochs must be confirmed from the training script or saved model configuration.

The predicted class and confidence value are displayed by the live interface. Reliable classification depends on the quality of the training examples and the similarity between the training data and the user's current input.

D. Translation and Speech Output

After classification, the application presents the predicted result in the translation panel. It also provides a “Speak All” button and a TTS toggle for speech-related output.

The video-upload workflow generates a translation-results display after processing the uploaded file. The generated text should be evaluated for correctness, repeated labels, and meaningful sequencing before the system is described as a reliable sentence translator.

E. User Interaction

The application provides the following principal controls:

- Real-time Camera: Opens the live-recognition section.
- Start Camera: Starts camera processing.
- Stop Camera: Stops camera processing.
- Text-to-Sign: Opens the video-upload and translation-results section shown in the screenshots. The actual functionality should be described according to the implementation.
- Speak All: Requests speech output for the displayed text, where supported.
- Clear: Clears displayed translation content, according to the implemented behaviour.
- Browse Files: Selects a video for upload and processing.

VII. IMPLEMENTATION DETAILS

A. Application Development

The Sign-Speak system is developed as a web-based Indian Sign Language recognition and translation application using Python. The application provides a user-friendly interface through which users can access real-time camera recognition

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 10, October 2026

DOI: 10.17148/IJARCCE.2026.1510133

and video-upload functionality. The web interface displays the camera feed, recognition results, confidence score, and translation text. It also provides controls to start and stop the camera, upload video files, clear displayed results, and access speech output.

B. Video Processing and Hand Landmark Detection

The system accepts visual input through a live camera or an uploaded video file. Video frames are processed to identify hand gestures for recognition. Hand-landmark visualization helps display important points and connections on the detected hand. OpenCV is used for computer-vision and video-processing tasks, while MediaPipe supports hand-landmark processing where integrated into the recognition pipeline. These features help represent hand gestures for subsequent classification.

C. Sign Recognition and Classification

The application uses a trained machine-learning model to predict the sign represented by the processed input. TensorFlow and Keras support model loading and inference in the project environment. The model's predicted output is mapped to a corresponding sign label using the project's label-encoding information. The application displays the predicted sign and its confidence score in the real-time recognition interface. The recognition performance depends on the training data, supported sign classes, and quality of the input video.

D. Translation and Speech Output

After sign recognition, the application displays the predicted label and associated text in the translation panel. It also provides text-to-speech controls, including a TTS toggle and a “Speak All” button, to support spoken output. The uploaded-video workflow processes the selected video and presents the generated translation result. The correctness of the displayed text depends on the model's predictions and the application's processing logic.

E. Web Interface and User Interaction

The Sign-Speak interface organizes its features into real-time camera and video-upload sections. Users can start or stop camera processing, view hand landmarks, observe recognition results, and select supported video files for processing. The upload interface accepts MP4, AVI, and MOV formats and displays a maximum file size of 50 MB. The interface combines input selection, recognition feedback, translation display, and speech-related controls in a single application, making the system easier to demonstrate and use.

VIII. RESULTS AND EVALUATION

The Sign-Speak application was successfully launched and tested through a web browser. The system provides a real-time camera interface for sign recognition and a separate interface for uploading videos and displaying translation results. During operation, the application displays the camera feed, detected hand landmarks, predicted sign label, confidence score, and translation text. It also provides speech-related controls to support audio output.

The video-upload feature allows users to upload supported video files in MP4, AVI, and MOV formats. After processing, the application displays the generated translation result. The user interface includes controls such as Start Camera, Stop Camera, Speak All, and Clear to support interaction with the system.

The observed results demonstrate the integration of camera input, sign recognition, translation display, and video-upload functionality in a single web application. However, the correctness of the predictions and the quality of the generated translations require further systematic testing.

The system can be evaluated using sign-recognition accuracy, precision, recall, F1-score, processing time, and translation correctness. These metrics help measure how effectively the application identifies signs and processes visual input.

Recognition accuracy can be calculated by comparing the number of correctly predicted signs with the total number of test samples. Precision and recall can be used to assess classification performance for individual sign classes. The F1-score provides a combined measure of precision and recall. Processing time can be measured separately for real-time camera recognition and uploaded-video processing.

Functional testing can also be performed to verify camera operation, hand-landmark visualization, sign prediction, video upload, translation display, and text-to-speech output. The application should be tested under different lighting conditions, hand positions, and backgrounds to identify possible recognition errors.

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 10, October 2026

DOI: 10.17148/IJARCCE.2026.1510133

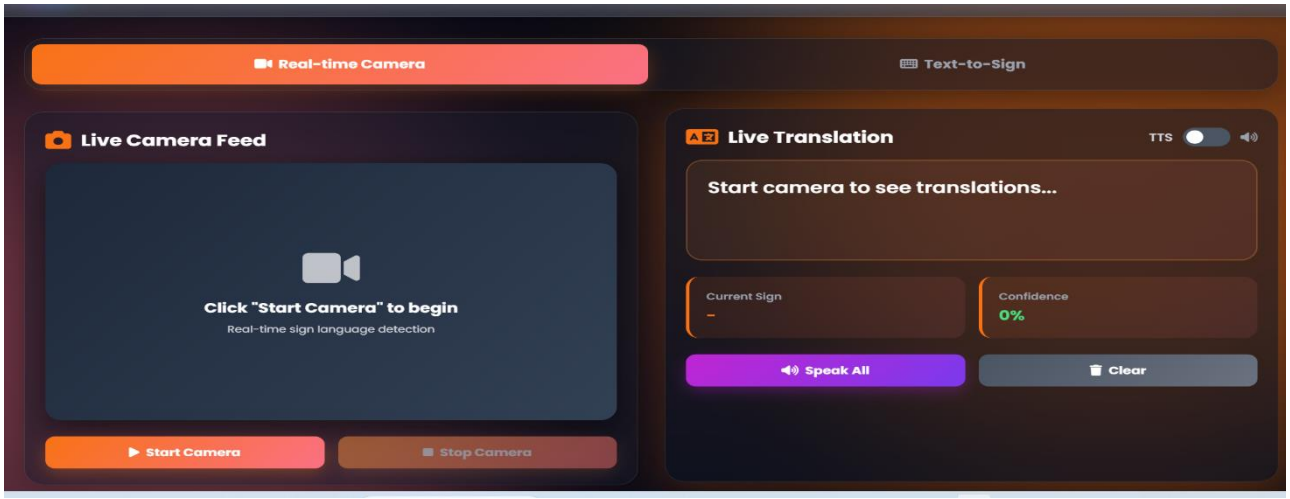


Fig. 2. Live camera Input

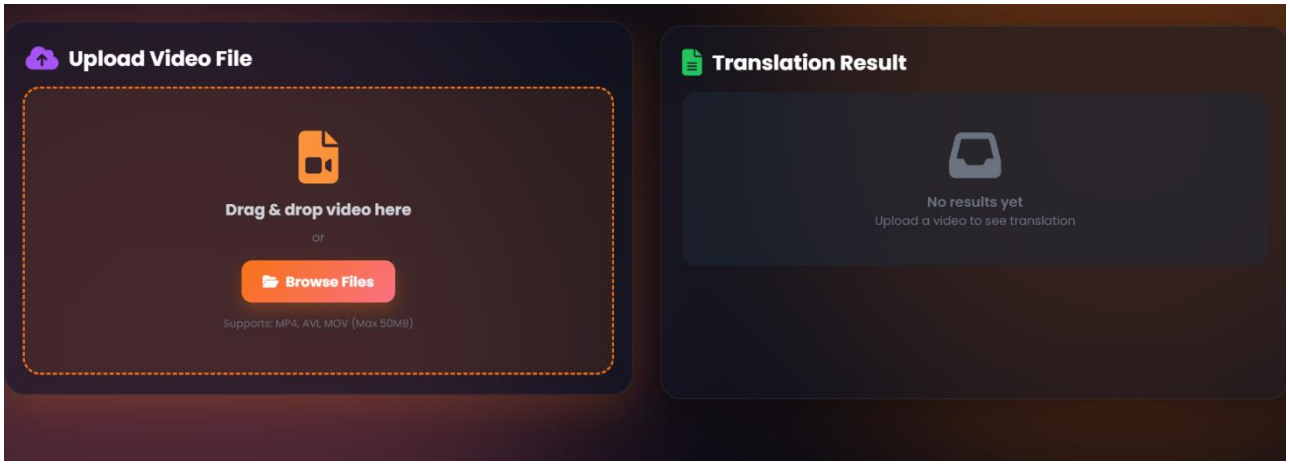


Fig. 3. Recorded Video Input

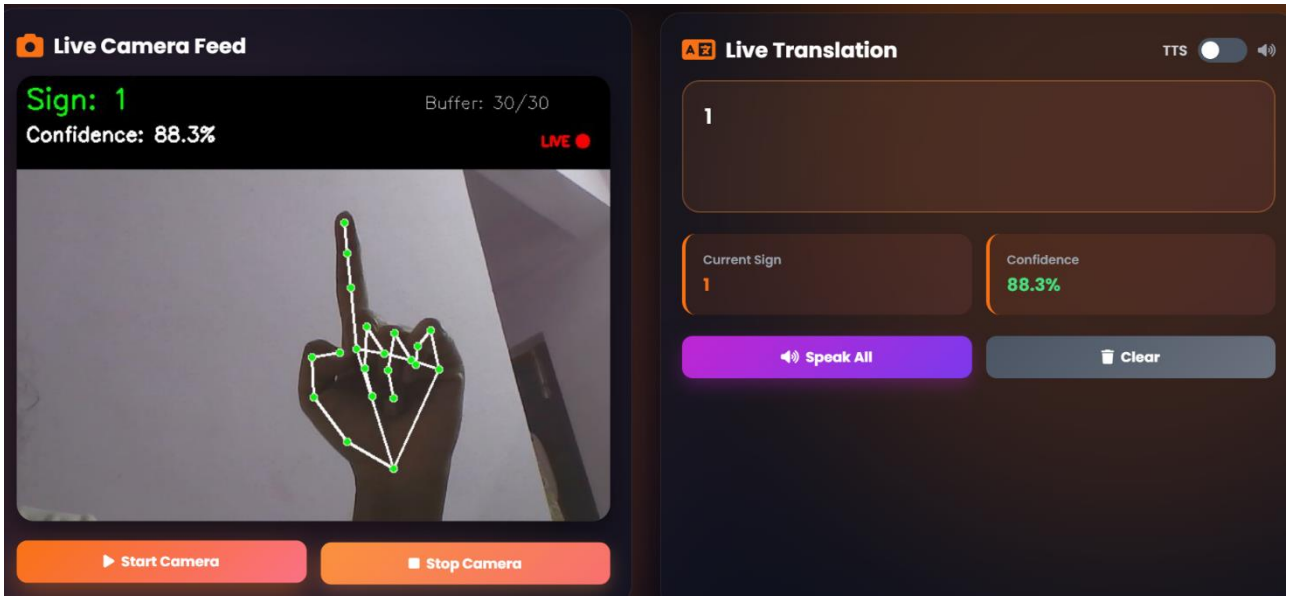


Fig. 4. Prediction of sign

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 10, October 2026

DOI: 10.17148/IJARCCE.2026.1510133

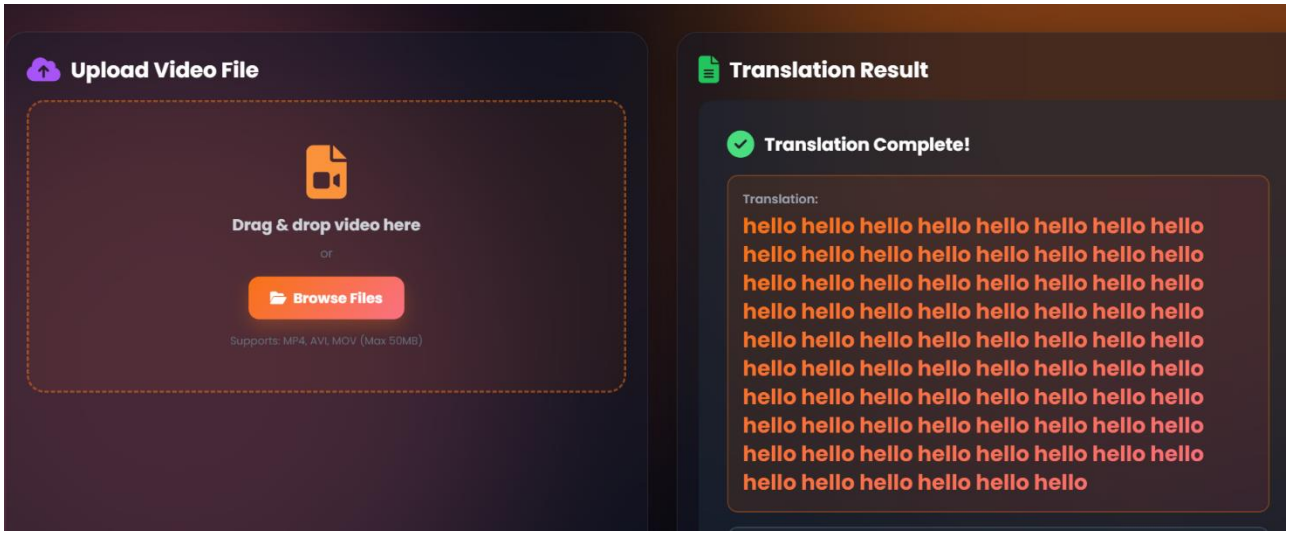


Fig. 5. Recorded video output

IX. DISCUSSION

The Sign-Speak system demonstrates the integration of computer vision, machine learning, and a web-based interface for Indian Sign Language recognition. The application provides two input methods: real-time camera input and video upload. This allows users to perform sign recognition through a live camera or submit a previously recorded video for processing.

The real-time recognition interface displays the camera feed, detected hand landmarks, predicted sign label, confidence score, and translation text. These features help users observe how the system processes visual input and presents recognition results. The confidence score provides an indication of the model's confidence in its prediction, but it does not guarantee that the predicted sign is correct.

The video-upload feature extends the application beyond live camera input by allowing users to process supported video files and view the generated translation. However, the quality of the translation depends on the recognition model and the processing of signs across video frames. Repeated predictions and incorrect labels may affect the readability and usefulness of the output.

The integration of text-to-speech controls provides an additional way to access translation output. When speech functionality operates correctly, it can help users listen to the displayed text rather than relying only on visual output. The web-based interface also brings the major functions together in one place, making the system easier to operate and demonstrate.

Despite these advantages, the system has limitations. Recognition may be affected by lighting, hand orientation, background conditions, and differences in signing styles. Its capabilities are also limited by the vocabulary and training data of the model. Further evaluation using labelled test videos, multiple users, and different environmental conditions is necessary to measure recognition accuracy and reliability.

Overall, Sign-Speak provides a practical prototype for exploring camera-based sign recognition and accessible communication. Future improvements can focus on expanding the supported vocabulary, reducing repeated predictions, improving continuous sign recognition, and conducting systematic performance evaluations. These improvements could make the application more reliable and useful for real-world communication assistance.

X. ADVANTAGES OF THE PROPOSED SYSTEM

- Real-Time Recognition: The system supports camera-based sign recognition and displays the predicted sign during operation.
- Hand-Landmark Visualization: It displays detected hand landmarks, helping users observe the hand features used in the recognition process.

 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 10, October 2026

DOI: 10.17148/IJARCCE.2026.1510133

- User-Friendly Interface: The web-based application provides simple controls for camera operation, video uploading, and viewing translation results.
- Video Upload Support: Users can upload video files in supported formats, including MP4, AVI, and MOV, for sign recognition and translation.
- Text Output: The system displays predicted sign labels and translation text, making the results easier to understand.
- Speech Output: Text-to-speech controls allow users to access spoken output when the speech functionality is available.
- Integrated Application: Camera recognition, video processing, translation display, and speech-related controls are provided within a single application.
- Potential for Future Expansion: The system can be extended with additional sign classes, improved recognition techniques, and more comprehensive language support.

XI. CONCLUSION

The Sign-Speak project presents a web-based Indian Sign Language recognition and translation system that integrates computer vision, machine learning, and an interactive user interface. The system supports real-time camera input and video uploading to recognize signs and display the predicted labels, confidence scores, and translation text. It also provides hand-landmark visualization and text-to-speech controls to support accessible communication.

The application brings camera-based recognition, video processing, and translation output together in a single platform. Its simple interface allows users to interact with the recognition system and observe the results. However, the effectiveness of the system depends on the trained model, supported sign vocabulary, input quality, and accuracy of the generated predictions.

In conclusion, Sign-Speak provides a foundation for developing assistive sign-language recognition applications. Future enhancements can include expanding the vocabulary, improving recognition accuracy, reducing repeated predictions, supporting continuous sign-language translation, and conducting extensive performance testing. These improvements could increase the reliability and practical usefulness of the system for communication assistance.

XII. FUTURE SCOPE

The Sign-Speak system can be further improved to enhance recognition accuracy, usability, and accessibility. The following enhancements can be considered for future development.

- Expanded Sign Vocabulary: The system can be trained to recognize a larger number of Indian Sign Language signs, enabling users to communicate a wider range of words and expressions.
- Continuous Sign Language Translation: The system can be enhanced to recognize sequences of signs and convert them into meaningful sentences rather than producing repeated or isolated predictions.
- Improved Recognition Accuracy: Training with a larger and more diverse dataset can help improve recognition performance under different lighting conditions, backgrounds, hand positions, and signing styles.
- Multilingual Support: The application can be extended to provide translation text and speech output in additional languages, such as Kannada, Hindi, and other regional languages.
- Improved Video Processing: The uploaded-video workflow can be enhanced to reduce repeated predictions, identify sign boundaries, and generate more readable translation results.
- Mobile Application Development: A mobile version can be developed to make sign-language recognition more accessible on smartphones and tablets.
- Performance Evaluation: Further testing can be conducted using a labelled dataset to measure accuracy, precision, recall, F1-score, and processing time.
- Enhanced User Interface: The interface can be improved with clearer feedback, better error handling, and additional accessibility features.

In the future, these enhancements could make Sign-Speak a more reliable and accessible tool for supporting communication between Indian Sign Language users and people who do not understand sign language.

ACKNOWLEDGMENT

The authors thank Madhushree M and Lavanya N L for their guidance and continuous support throughout this project, and the Department of Computer Science and Engineering, East West College of Engineering, for providing the resources necessary to carry out this work.



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026
DOI: 10.17148/IJARCCE.2026.1510133

DECLARATIONS

- **Funding Source:** No funding was received for this work.
- **Acknowledgement:** The authors thank East West College of Engineering and the Department of Computer Science and Engineering for their support.

- **Data Availability:** Data is available from public sources or upon reasonable request.
- **Authors' Contribution:** All authors contributed to the research, implementation, analysis, and manuscript preparation.
- **Use of AI and AI-Assisted Technologies:** AI tools were used for language improvement and grammar checking. The authors verified the content.
- **Conflict of Interest:** The authors declare no conflict of interest.
- **Copyright Permissions:** Copyright permissions were obtained for all figures and tables used.

REFERENCES

[1]. Geetha M., Aloysius N., Somasundaran D. A., Raghunath A., Nedungadi P. (2025). "Toward Real Time Recognition of Continuous Indian Sign Language: A Multi-Modal Approach Using RGB and Pose." IEEE Access, vol. 13, pp. 60270-60283. DOI: 10.1109/ACCESS.2025.3554618

[2]. Rawat P., Kumar P., Tamta V. K., Kumar A. (2025). "A Comprehensive Approach to Indian Sign Language Recognition: Leveraging LSTM and MediaPipe Holistic for Dynamic and Static Hand Gesture Recognition." EAI Endorsed Transactions on AI and Robotics.

[3]. Anithadevi N. et al. (2025). "MediaPipe-LSTM-Enhanced Framework for Real-Time Dynamic Sign Language Recognition in Inclusive Communication Systems." Engineering Reports.

[4]. Subramanian B. et al. (2022). "An integrated mediapipe-optimized GRU model for Indian sign language recognition." Scientific Reports (widely cited baseline work).

[5]. Dhanasree G. et al. (2026). "Real-Time Sign Language Recognition Using Mediapipe and LSTM Based Deep Learning." (Recent work on real-time implementation).

[6]. A. Bhadouria, P. Bindal, N. Khare, D. Singh, and A. Verma, "LSTM-Based Recognition of Sign Language," Proc. 16th International Conference on Contemporary Computing, 2024. :contentReference[oaicite:2]{index=2}

[7]. "SLRMPCMC: Sign Language Recognition Using MediaPipe and Cross-Model Comparison," Proc. IEEE ICEECT, 2024. :contentReference[oaicite:3]{index=3}

[8]. M. Al-Qurishi, T. Khalid, and R. Souissi, "Deep Learning for Sign Language Recognition: Current Techniques, Benchmarks, and Open Issues," IEEE Access, vol. 9, pp. 126917–126951, 2021.

[9]. S. Hochreiter and J. Schmidhuber,"Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[10]. A. Graves and J. Schmidhuber, "Framewise Phoneme Classification with Bidirectional LSTM Networks," Proc. International Joint Conference on Neural Networks, 2005.

[11]. M. Geetha, N. Aloysius, D. A. Somasundaran, A. Raghunath, and P. Nedungadi, “Toward Real-Time Recognition of Continuous Indian Sign Language: A Multi-Modal Approach Using RGB and Pose,” *IEEE Access*, 2025. doi: 10.1109/ACCESS.2025.3554618.

[12]. R. Damdoo, P. Kumar, and R. Gogoi, “End-to-end sentence-level Indian sign language translation with ISH-NEWS dataset and transformer model,” *Scientific Reports*, 2026. doi: 10.1038/s41598-026-60893-0.