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 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511191

SignSpeak: Real-Time Sign Language Recognition System

Jayashree Mulimani[1], Nandini N. Kulkarni[2], Neelamma G. Patil[3], Aishwarya L. Khebbani[4], Radhika[5]

Assistant Professor, Department of Computer Science and Engineering, B.V.V. Sangha's Basaveshwar Engineering College, Bagalkot, India[1]

Students, Department of Computer Science and Engineering, B.V.V. Sangha's Basaveshwar Engineering College, Bagalkot, India[2-5]

Abstract: Communication between Indian Sign Language (ISL) users and non-signers remains challenging due to the limited availability of sign language interpreters. To bridge this gap, this paper presents SignSpeak, a web-based platform that facilitates two-way interaction using computer vision and speech technologies. The system employs a MobileNetV2 model to classify 45 static hand signs and integrates MediaPipe hand landmarks with a Long Short-Term Memory (LSTM) network to recognize 30 dynamic ISL phrase sequences, achieving 98.60% and 98.05% test accuracy, respectively, on held-out test sets from custom datasets of 22,500 images and 1,025 video sequences. Beyond gesture recognition, the application incorporates speech-to-text conversion, text-to-speech synthesis, multilingual translation across six languages, and prerecorded ISL video demonstrations. Developed using React and FastAPI and deployed on Vercel and Render, SignSpeak provides an accessible, all-in-one communication tool for everyday use.

Keywords: Indian Sign Language, sign language recognition, MobileNetV2, MediaPipe, LSTM, text-to-speech, multilingual translation.

I. INTRODUCTION

Communication is an essential part of everyday life. However, interaction between Indian Sign Language (ISL) users and people who do not understand sign language can be challenging, particularly when an interpreter is unavailable. This creates a need for accessible technologies that can support communication between both groups [1], [14].

Recent developments in computer vision and deep learning have made automatic sign language recognition more practical. Models such as MobileNetV2 support efficient image classification [11], while MediaPipe Hands enables real-time detection of hand landmarks [12]. These techniques, combined with LSTM networks, can help recognize both individual gestures and signs performed through sequences of movements.

To address this need, **SignSpeak** is developed as a web-based Indian Sign Language recognition and communication support system. It uses MobileNetV2 to recognize 45 static sign classes, including alphabets, digits, and selected words. For dynamic sign recognition, it combines MediaPipe hand landmark extraction with an LSTM model to process video sequences. The application also provides speech-to-text conversion, text-to-speech output, multilingual translation, and prerecorded ISL video demonstrations.

The system is developed using React for the frontend and FastAPI for the backend, with deployment through Vercel and Render. By integrating gesture recognition and speech-based features into one platform, SignSpeak aims to make everyday communication between ISL users and non-signers more convenient and accessible.

Project Purpose

- Interpret both dynamic ISL motion gestures and stationary hand shapes.
- Convert identified hand gestures into legible text.
- Use speech synthesis to produce spoken audio from recognized text.
- Provide integrated multilingual translation to facilitate interlanguage communication.
- Provide an easy-to-use interface that requires little technical knowledge.
- Support everyday communication where a human interpreter is unavailable.

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



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511191

II. AIM AND OBJECTIVES

SignSpeak's main objective is to develop an automated
platform for the recognition and translation of Indian Sign Language that will reduce communication barriers between signers and non-signers.

To achieve this goal, the project focuses on six core objectives:

1. **Static Gesture Classification:** Use a MobileNetV2 deep learning model to precisely identify still hand signs.
2. **Dynamic Gesture Recognition:** Combine MediaPipe Hands with an LSTM network to capture moving hand sequences and complete phrases.
3. **Text Generation:** Convert categorized gestures into legible text strings.
4. **Multilingual Translation:** To increase usability, translate output text into multiple regional languages.
5. **Voice Synthesis:** Produce spoken audio from the recognized or translated text.
6. **Accessible Interface Design:** Create a clear, responsive web application that is easy for any user to operate.

III. LITERATURE OVERVIEW

Indian Sign Language (ISL) recognition has received increasing research attention as computer vision and deep learning techniques have created new possibilities for automated gesture interpretation. Researchers have explored different approaches, ranging from traditional image-processing methods to CNN-based classification and landmark-based sequence recognition. These developments have helped improve the recognition of static gestures and dynamic signs. The following studies examine important contributions in this area and provide the technical background for the proposed SignSpeak system.

A. *Indian Sign Language Recognition System Using SURF with SVM and CNN*

S. Katoch, V. Singh, and U. S. Tiwary [1] proposed an ISL recognition approach using Speeded-Up Robust Features (SURF), Support Vector Machines (SVM), and Convolutional Neural Networks (CNN). Their system recognizes English alphabets and numerical signs from webcam input. The method applies image segmentation and feature extraction before classifying the gestures. It also provides text and speech output through an interactive graphical interface. The study demonstrates how traditional feature extraction and deep learning methods can be applied to static ISL recognition. However, its primary focus is on isolated alphabets and digits rather than continuous sign sequences.

B. *ISL Recognition System Using Integrated Mobile-Net and Transfer Learning Method*

S. Sharma and S. Singh [2] presented an Indian Sign Language recognition system based on MobileNetV2 and transfer learning. Their approach uses a lightweight CNN architecture to identify hand gestures while reducing the computational requirements associated with larger networks. The study highlights the usefulness of transferring previously learned visual features to a custom ISL dataset. This approach is particularly relevant to applications requiring efficient image classification. However, the work primarily addresses gesture recognition and does not establish a complete communication platform combining static signs, continuous phrases, and speech-based interaction.

C. *An Integrated MediaPipe-Optimized GRU Model for Indian Sign Language Recognition*

B. Subramanian et al. [3] developed an ISL recognition approach that combines MediaPipe-based hand tracking with a Gated Recurrent Unit (GRU) network. The system uses landmark information to represent hand movements and applies a sequential learning model to recognize gestures. By using extracted features rather than depending entirely on raw image data, the approach provides an alternative for reducing the complexity of gesture processing. The research also discusses challenges such as hand occlusion, tracking accuracy, and computational requirements. Although the study demonstrates the potential of landmark-based recognition, its use of GRU differs from the LSTM-based sequence model implemented in SignSpeak.

D. *Indian Sign Language Recognition Using OpenCV and the MediaPipe: A Step Towards Inclusion*

B. Narasimha Swamy et al. [4] introduced a two-way ISL communication system using OpenCV, MediaPipe Holistic, and deep learning models. The system extracts landmark information to recognize static and dynamic gestures. A Fully Connected Neural Network (FCNN) is used for static sign classification, while an LSTM model processes dynamic gestures. The authors reported 99.1% accuracy for static recognition and 96% for dynamic recognition. The system also supports the conversion of recognized signs into text and the presentation of corresponding sign videos from text input. However, its recognition is based on a predefined set of gestures, and the study focuses on a limited collection of dynamic signs.

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



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511191

E. ***A Comprehensive Approach to Indian Sign Language Recognition: Leveraging LSTM and MediaPipe Holistic for Dynamic and Static Hand Gesture Recognition***

P. Rawat, P. Kumar, V. K. Tamta, and A. Kumar [5] proposed an ISL recognition framework using MediaPipe Holistic and a Sequential LSTM model. Their system extracts facial, hand, and body landmarks from video frames to represent the spatial and temporal characteristics of gestures. The extracted information is processed through a network containing two LSTM layers and a dense output layer. The authors evaluated their approach using a customized dataset containing 11 gestures and reported an accuracy of 96.97%. This study demonstrates the usefulness of combining landmark extraction with sequential learning for dynamic sign recognition. However, its dataset contains a relatively small number of gesture categories.

F. ***GestureSpeak: A Real-Time Sign to Speech Translation***

O. Tipare, S. Pathre, P. Pathak, and D. Karia [6] presented GestureSpeak, a real-time sign-to-speech translation approach. The work focuses on interpreting sign gestures through a camera-based system and making the recognized information available as spoken output. Its objective is to make communication easier between sign language users and people who do not understand signing. The study is relevant to applications that connect visual gesture recognition with speech technologies. However, sign-to-speech systems remain dependent on the range of gestures supported by their recognition models, and a restricted vocabulary can limit their use in everyday conversations.

G. ***Real-Time American Sign Language Recognition from Video Using Hidden Markov Models***

T. Starner and A. Pentland [7] demonstrated the possibility of recognizing American Sign Language (ASL) from ordinary video input without requiring specialized sensor gloves. Their work uses Hidden Markov Models (HMMs) to interpret hand movements and recognize signs at sentence level. The experiments considered a vocabulary of 40 words and reported different recognition results depending on whether colored gloves were used. This research established an important foundation for camera-based sign language recognition. However, it focused on ASL and a limited vocabulary, rather than Indian Sign Language.

H. ***Continuous Sign Language Recognition: Towards Large Vocabulary Statistical Recognition Systems Handling Multiple Signers***

O. Koller, J. Forster, and H. Ney [8] investigated continuous sign language recognition across multiple signers using statistical recognition techniques. Their approach considers hand shape, hand position, movement, facial information, and signer adaptation. The experiments used the SIGNUM and RWTH-PHOENIX-Weather datasets to study recognition under both controlled and more realistic conditions. Their work highlights the difficulties involved in recognizing continuous signing when different individuals perform the same signs in different ways. It also shows the importance of larger datasets and language modelling for continuous recognition systems.

I. ***Neural Sign Language Translation***

N. C. Camgoz et al. [9] presented a neural machine translation approach that converts sign language video into spoken-language text. Rather than treating recognition as the classification of individual gestures alone, their work considers the grammatical relationship between signs and the target spoken language. The proposed sequence-to-sequence approach uses attention mechanisms and was evaluated using the RWTH-PHOENIX-Weather 2014T dataset for German Sign Language. The research demonstrates the importance of linguistic structure in sign language translation. However, such end-to-end translation methods require extensive annotated datasets and more complex training procedures than fixed-vocabulary recognition systems.

J. ***Vision-Based Hand Gesture Recognition for Human–Computer Interaction: A Survey***

S. S. Rautaray and A. Agrawal [10] reviewed vision-based hand gesture recognition methods used in human–computer interaction. Their survey explains the major stages of a gesture recognition pipeline, including hand detection, segmentation, tracking, feature extraction, and classification. It also discusses practical difficulties such as background interference, lighting variations, differences in hand appearance, and processing speed. Although the survey is not limited to ISL, its discussion provides a useful foundation for understanding the challenges involved in camera-based gesture recognition.

K. ***An Integrative Survey on Indian Sign Language Recognition and Translation***

R. Damdoo and P. Kumar [14] presented a review of research on Indian Sign Language recognition and translation. Their study examines machine learning, deep
learning, and gesture-based methods, along with the development of ISL datasets and dictionaries. It also discusses existing research challenges and possible directions for future work. The review highlights the importance of improving recognition systems beyond isolated gestures and developing more practical communication solutions. This broader

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

DOI: 10.17148/IJARCCE.2026.1511191

analysis provides useful context for identifying the limitations of existing ISL recognition approaches.

Background Technologies

MobileNetV2: Inverted Residuals and Linear Bottlenecks

M. Sandler et al. [11] introduced MobileNetV2, a lightweight convolutional neural network architecture designed to improve computational efficiency. The architecture uses inverted residual blocks, linear bottleneck layers, and depthwise convolutions to reduce the number of operations while retaining useful image features. The authors evaluated the model on image classification, object detection, and segmentation tasks. Its efficient structure makes it suitable for applications where processing resources are limited. SignSpeak adopts MobileNetV2 as the foundation of its static gesture classification model and adapts it to the project's custom ISL dataset through transfer learning.

MediaPipe Hands: On-Device Real-Time Hand Tracking

A. Vakunov et al. [12] introduced MediaPipe Hands, a real-time hand-tracking pipeline designed to work using a single camera. The method consists of a palm detection stage followed by hand landmark estimation, allowing the system to identify 21 key points on each detected hand. These landmarks represent important hand positions, including finger joints and fingertips. The framework is designed to provide efficient tracking on commonly available devices. SignSpeak uses MediaPipe hand landmarks as input features for its dynamic sign recognition model, allowing hand movement information to be processed as a sequence.

Long Short-Term Memory

S. Hochreiter and J. Schmidhuber [13] introduced Long Short-Term Memory (LSTM) networks to address the difficulty of learning long-term dependencies in conventional recurrent neural networks. LSTM uses memory cells and gating mechanisms to control the information retained and passed through a sequence. This makes it useful for learning patterns in data where the order of inputs is important. Since dynamic sign language involves movements that change across video frames, LSTM provides a suitable foundation for sequential gesture classification. SignSpeak applies this concept to a sequence of 30 hand-landmark frames for recognizing predefined sign phrases.

Research Gap

The reviewed studies show that computer vision and deep learning have contributed significantly to the development of sign language recognition systems. CNN architectures are useful for identifying static hand configurations, while landmark-based methods combined with recurrent networks can process gestures that involve movement over time. However, several challenges remain, including limited gesture vocabularies, variations between signers, difficulties in recognizing continuous signs, and the computational requirements of complex recognition models. In addition, many existing approaches concentrate on a particular recognition task rather than combining multiple communication features in one application.

To address these limitations, SignSpeak combines MobileNetV2-based static gesture recognition with MediaPipe landmark extraction and LSTM-based dynamic phrase recognition. The system also integrates speech-to-text conversion, text-to-speech output, multilingual translation, and prerecorded ISL video demonstrations for supported inputs. By bringing these features together in a web-based application, SignSpeak aims to provide a more accessible communication platform for ISL users and non-signers.

The main contributions of this paper are: (i) an integrated web-based ISL communication platform that combines static and dynamic sign recognition with speech output, multilingual translation, and ISL video demonstrations; (ii) custom ISL datasets of 22,500 static images across 45 classes and 1,025 dynamic sequences across 30 phrases; and (iii) an evaluation showing 98.60% and 98.05% test accuracy on static and dynamic signs, respectively.

IV. SYSTEM REQUIREMENTS

SignSpeak accepts image, webcam, and video input; recognizes static hand gestures and dynamic ISL phrases; converts the recognized output to text; translates it into English, Hindi, Kannada, Telugu, Marathi, and Tamil; and produces speech output. The system is expected to respond promptly, to handle invalid or unsupported inputs gracefully, and to keep the frontend, backend, and recognition models modular so that new signs and languages can be added.

The system is implemented in Python and JavaScript.
TensorFlow/Keras is used for model development; OpenCV, CVZone, and MediaPipe for computer vision; NumPy for numerical processing; FastAPI for the backend; and React.js with HTML and CSS for the frontend. The application runs on Linux or Windows and is accessed through Microsoft Edge or Google Chrome.

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



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511191

Each hand is represented by 21 landmarks with three coordinates (X, Y, Z), giving 126 features per frame for two hands, and the dynamic model processes sequences of 30 frames. Accurate landmark detection requires the hands to be clearly visible, and recognition results depend on the training dataset, hand placement, lighting, and input quality.

V. METHODOLOGY

Language barriers, trouble identifying hand gestures, and a shortage of interpreters are just a few of the communication issues that Indian Sign Language (ISL) users and non-signers face. These issues are addressed by the methodology used to develop the SignSpeak system. To identify both static and dynamic sign language gestures, translate them into intelligible text, and produce speech and translation, the development process takes a methodical and sequential approach. The process is intended to yield a system that is effective, usable, and suitable for regular communication.

A. *System Design and Modeling*

During this stage, the frontend, backend, and recognition modules make up the overall architecture of the SignSpeak system. FastAPI is used for backend processing and communication between application components, while React.js is used to create an interactive user interface on the frontend.
The two main recognition models in the system are an LSTM-based model for dynamic sign recognition and a MobileNetV2-based model for static hand gesture recognition. Landmark extraction, hand detection, and image processing are done with OpenCV, CVZone, and MediaPipe.

B. *Image and Video Processing*

Prior to being sent to the recognition models, the input photos and videos undergo processing.
Processing Static Gestures:

- The user takes or uploads the image input.
- CVZone HandDetector is used for hand detection.
- Bounding box coordinates are used to extract the hand region.
- The extracted hand image is positioned on a uniform background after being resized.
- Prior to classification, the image is normalized and resized to 224 × 224 pixels.

Processing Gestures Dynamically:

- The user records or uploads video input.
- OpenCV is used to extract video frames.
- MediaPipe Hands uses every frame to identify hand landmarks.
- Each hand extracts a total of 21 landmarks.
- The wrist is used to normalize the landmark coordinates.
- To facilitate recognition, the extracted features are organized in a sequential format.

C. *Model Implementation*

Two distinct deep learning techniques are used to implement the recognition process.
MobileNetV2 for Static Sign Recognition:
The static recognition module recognizes distinct hand gestures using a MobileNetV2-based classification model. The extracted hand image is sent to the trained model following image preprocessing. The model produces a confidence score and predicts the matching sign label. The user is then shown the identified outcome.

LSTM-Based Dynamic Sign Recognition
The dynamic recognition module employs an LSTM network for sequential gesture recognition and MediaPipe Hands for landmark extraction. With 21 landmarks and three coordinate values for each hand, each frame produces
126 features that represent two hands. The extracted sequence is processed by resampling it into 30 frames and eliminating superfluous static frames. The resulting sequence is passed to the LSTM model to predict the corresponding sentence or phrase.

D. *System Integration*

The recognition modules are integrated with the frontend and backend to create a full application after they are implemented.

- The React frontend offers options for user input and interaction.
- Requests for image and video recognition are handled by the FastAPI backend.
- Static gesture classification is carried out by a model based on MobileNetV2.
- Dynamic sentence recognition is carried out by MediaPipe and LSTM.

 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 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511191

- Text that has been recognized can be translated into the chosen language.
- Audio output is produced by text-to-speech functionality.

Through a single application, users can access various SignSpeak features thanks to this integration.

VI. PROPOSED MODEL

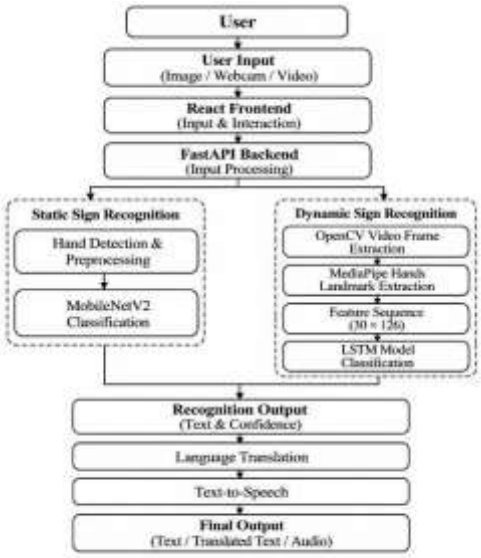


Fig. 1. Architecture and data flow of the proposed SignSpeak system.

A. *Data Flow Design*

The proposed SignSpeak system's operational sequence is depicted in Fig. 1. It explains how users input images, videos, or webcams, how the system interprets both static and dynamic sign gestures, and how text and speech are produced from the recognized output. The process combines deep learning and computer vision methods to facilitate multilingual communication, language translation, and sign recognition.

I. User:

The user is the system's main participant. To perform sign recognition and translation, users can input sign language using pictures, webcam footage, or video files.

II. Input from Users:

Images of static hand gestures or videos with dynamic sign language expressions are used by users as input. The frontend receives the chosen input and processes it further.

III. React Frontend

The React frontend provides the user interface for interacting with the system. It accepts user input, shows recognition results, and gives access to the available functionalities.

IV. FastAPI Backend

The FastAPI backend processes requests from the frontend and connects the input to the appropriate recognition module. It manages the processing required to identify both dynamic and static signs.

V. Static Sign Recognition

Using pictures or webcam input, the static sign recognition module recognizes specific hand gestures. Hand detection and preprocessing are done using CVZone HandDetector. To predict the corresponding sign label, the detected hand region is cropped, resized to 224 × 224 pixels, and fed into the MobileNetV2-based classification model.

VI. Dynamic Sign Recognition

To recognize sign language expressions, the dynamic sign recognition module analyzes video input. Video frames are extracted using OpenCV, and hand landmarks are detected using MediaPipe Hands. When two hands are detected, the



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511191

system creates a feature sequence with 126 features per frame and extracts 21 landmarks per hand.

VII. LSTM Classification and Feature Sequence
The extracted landmark features are arranged into a 30-frame sequence after being normalized. The Long Short-Term Memory (LSTM) model receives this sequence, classifies the dynamic sign, and predicts the matching predefined sentence or phrase.

VIII. Recognition Output
The predicted sign label or sentence and its confidence score are produced by the static and dynamic recognition modules. Through the application interface, the identified outcome is shown as text.

IX. Language Translation
English, Hindi, Kannada, Telugu, Marathi, and Tamil are among the supported languages into which the recognized text can be translated. This feature facilitates cross-linguistic communication.

X. Text-to-Speech Conversion
The recognized or translated text is transformed into audio output by the text-to-speech module. This makes it possible for speakers who do not use sign language to comprehend the message.

XI. Final Output
The final output is displayed by the system as audio, translated text, or recognized text. This facilitates better communication between non-sign language users and Indian sign language users.

B. *User Interface*
The user interface of the proposed SignSpeak system was designed to support communication between Indian Sign Language (ISL) users and non-signers. The interface covers the layout and interaction flow of the system's main modules: static and dynamic sign recognition, ISL-to-text and ISL-to-speech conversion, speech-based features, multilingual translation, and application management.

1) *Home Page:* The home page gives an overview of the SignSpeak platform and acts as the main navigation hub. From it, users can access the static sign recognition, dynamic recognition, speech-to-ISL, and speech-to-text modules, as well as the translation and application management features. The layout was kept simple so that every service is easy to reach.

2) *Sign Recognition Portal*
a) Static Sign Recognition Page: This page recognizes individual hand gestures from an uploaded image or a live webcam feed. The hand region is first localized using the CVZone HandDetector and then classified by a MobileNetV2-based model. The interface displays the predicted sign label along with its confidence score and also provides the recognized output as speech.

b) Dynamic Recognition Page: This page accepts video input containing dynamic sign language expressions. OpenCV extracts the frames, and MediaPipe Hands detects the hand landmarks in each frame. The resulting landmark sequences are passed to an LSTM model, which recognizes predefined sentences or phrases. The recognized output is displayed as text and also converted into speech.

3) *Speech and Translation Portal*
a) Speech-to-ISL Page: This page accepts spoken input and displays the corresponding ISL video clip, so that non-signers can communicate with ISL users.
b) Speech-to-Text Page: This page converts spoken input into text and displays the transcription to the user, supporting speech-based communication.
c) Language Translation: Recognized text can be translated into English, Hindi, Kannada, Telugu, Marathi, and Tamil, so that users can read the recognized message in their preferred language.

4) *Application Management*
a) ***Recognition History Page:*** This page lets users view and review previous recognition activity. The current version uses sample history data to show how recognition records are presented.
b) ***Settings Page:*** This page lets users configure application preferences and interface options to customize their experience.

 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 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511191

C. *System Implementation*

SignSpeak is implemented with React.js for the frontend and Python with FastAPI for the backend, and the two communicate through REST APIs. The system combines computer vision and deep learning techniques for both static and dynamic sign recognition. Static recognition uses a MobileNetV2-based classification model. Dynamic recognition uses OpenCV for frame extraction, MediaPipe Hands for landmark detection, and an LSTM network for sequence classification. The recognized output is presented as text and converted to speech using text-to-speech. The system also provides multilingual translation, which together form an integrated platform for ISL recognition and communication.

D. *Module Development*

1) User Module

- Provides access to the SignSpeak application.
- Allows users to interact with different sign recognition and translation functionalities.
- Provides a common interface for accessing available services.

2) *Frontend Module*

- Developed using React.js to provide the user interface.
- Accepts image, webcam, and video inputs.
- Displays recognition results and provides navigation between different functionalities.

3) *Backend Processing Module*

- Developed using Python and FastAPI.
- Handles requests received from the frontend through REST APIs.
- Connects user inputs to the appropriate recognition and processing modules.

4) *Static Sign Recognition Module*

- Recognizes individual static hand gestures using image or webcam input.
- Uses CVZone HandDetector for hand detection and preprocessing.
- Resizes the detected hand region to 224 × 224 pixels.
- Uses the MobileNetV2-based model to classify the hand gesture and generate the predicted sign label.

5) *Dynamic Sign Recognition Module*

- Recognizes dynamic sign language expressions from video input.
- Uses OpenCV for video frame extraction and MediaPipe Hands for landmark detection.
- Extracts hand landmark coordinates and generates 126 features per frame when two hands are detected.
- Uses a sequence of 30 frames as input to the LSTM model for sentence or phrase classification.

6) *Text Conversion Module*

- Converts recognized static gestures and dynamic sign expressions into understandable text.
- Displays the predicted sign labels or sentences.
- Provides recognition output along with confidence information.

7) *Language Translation Module*

- Translates recognized text into supported languages.
- Supports English, Hindi, Kannada, Telugu, Marathi, and Tamil.
- Helps users understand the recognized message in different languages.

8) *Text-to-Speech Module*

- Converts recognized or translated text into speech output.
- Generates audio to communicate the recognized message.
- Helps non-sign language users understand the output through speech.

9) *Speech-to-ISL Module*

- Provides an interface for speech-related input and sign language communication.
- Supports the speech-to-ISL functionality available in the application.

10) *Learning Module*

- Provides learning-related content for understanding Indian Sign Language gestures.
- Helps beginners learn and practise sign language.
- Presents learning information through the application interface.

11) *Recognition History and Settings Module*

- Provides an interface for accessing recognition-related history and application settings.
- Displays sample recognition history information.
- Provides available application preference options.

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



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511191

E. *Testing and Deployment*

The system was tested at several levels. Unit testing covered the static and dynamic recognition modules, landmark extraction, text conversion, language translation, and text-to-speech. Integration testing verified the communication between the React frontend, the FastAPI backend, the REST APIs, and the recognition models. Validation testing checked image and video inputs, hand detection, landmark extraction, and feature sequences, and user-interface testing covered navigation, input handling, and the display of recognition, translation, and speech output. The recognition performance of the two models is reported in Section VII.

The React frontend is deployed on Vercel, and the FastAPI backend, together with the trained MobileNetV2 and LSTM models and their class-label files, is deployed on Render. The two communicate through REST APIs, so users can provide image, webcam, or video input and access the recognition and translation functions from a web browser.

VII. RESULTS AND DISCUSSION

F. *Experimental Setup and Dataset*

The proposed SignSpeak recognition models were trained and evaluated on custom Indian Sign Language (ISL) datasets captured under varied lighting and background conditions. The static sign dataset comprises **22,500 images** distributed evenly across **45 classes** (500 images per class, covering alphabets, digits, and common words). The dynamic sign dataset consists of **1,025 video sequences** across **30 predefined ISL phrase classes** (including *Be careful*, *Call me later*, *Help me*, *How are you*, *Thank you*, and *No Gesture*), with each sequence resampled into 30 frames containing 126 hand-landmark features per frame. Both datasets were partitioned using an **80% training and 20% testing** split, resulting in 18,000 training and 4,500 testing samples for static gestures, and 820 training and 205 testing samples for dynamic sequences, as detailed in Table I.

TABLE I: DATASET COMPOSITION AND TRAIN–TEST SPLIT

Recognition Module	Classes	Total Samples	Training Set (80%)	Testing Set (20%)	Input Representation
Static Sign (MobileNetV2)	45	22,500	18,000	4,500	224 × 224 RGB Image
Dynamic Sign (MediaPipe + LSTM)	30	1,025	820	205	30 Frames × 126 Landmarks

G. *Quantitative Performance Evaluation*

Both models were evaluated on their respective unseen test sets to assess classification accuracy, precision, recall, F1-score, and average inference speed. The quantitative results are summarized in Table II.

TABLE II: CLASSIFICATION PERFORMANCE ON UNSEEN TEST DATA

Model	Test Samples	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Avg. Inference Time
MobileNetV2 (Static)	4,500	98.60	97.00	97.00	97.00	7.43 ms/image
MediaPipe + LSTM (Dynamic)	205	98.05	98.00	98.00	98.00	< 50 ms/sequence

H. *Discussion and Error Analysis*

As presented in Table II, the MobileNetV2 classifier achieved 98.60% classification accuracy across 4,500 unseen static test images with an average inference latency of 7.43 ms per image, suggesting suitability for interactive use. The MediaPipe + LSTM sequential model achieved 98.05% test accuracy (0.0934 test loss) and 0.98 macro and weighted F1-scores across 205 unseen dynamic phrase sequences, with 26 out of 30 phrase classes achieving 100% precision and recall. Confusion matrix analysis of the dynamic model indicated that only 4 out of 205 test sequences were

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



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511191

misclassified, occurring between signs with similar hand trajectories (such as *How are you* vs. *Thank you*, and *I am deaf* vs. *I am happy*). Representing hand motion with 126 normalized landmark coordinates per frame is far more compact than raw video frames, which keeps the computational cost of the sequence model low.

For context, Swamy et al. [4] report 99.1% (static) and 96% (dynamic) accuracy, and Rawat et al. [5] report 96.97% on a set of 11 gestures. These figures are not directly comparable with ours because the datasets, vocabularies, and evaluation protocols differ.

I. *Limitations*

The reported results should be interpreted within the scope of the study. Both models are trained on closed vocabularies (45 static classes and 30 dynamic phrases) and cannot recognize signs outside them. Recognition relies on hand images and hand landmarks only, so facial expressions and other non-manual markers are not captured, and accuracy depends on lighting, hand visibility, and input quality. The results were obtained on held-out test splits of custom datasets; performance on unseen signers and in unconstrained environments remains to be validated.

VII. CONCLUSION

This paper presented **SignSpeak**, an integrated web-based platform developed to bridge the communication gap between Indian Sign Language (ISL) users and non-signers. By combining a transfer-learned MobileNetV2 classifier for static gestures with MediaPipe hand-landmark extraction and a Long Short-Term Memory (LSTM) network for dynamic sequences, the system recognizes **45 static sign classes** and **30 dynamic ISL phrases**. Experimental evaluation on unseen test sets demonstrated strong performance, achieving **98.60% accuracy** (7.43 ms average inference latency) on static signs and **98.05% accuracy** on dynamic phrase sequences. Beyond gesture recognition, SignSpeak integrates speech-to-text transcription, text-to-speech synthesis, prerecorded ISL demonstrations, and multilingual translation across six languages within a responsive React and FastAPI web architecture. These combined capabilities can supplement human interpreters in everyday interactions where none is available. Future work will focus on expanding the dynamic sign vocabulary, incorporating facial expression and non-manual marker tracking, and scaling the dataset across a broader group of signers to handle complex conversational ISL.

REFERENCES

[1] S. Katoch, V. Singh, and U. S. Tiwary, "Indian Sign Language recognition system using SURF with SVM and CNN," *Array*, vol. 14, Art. no. 100141, Jul. 2022, doi: 10.1016/j.array.2022.100141.

[2] S. Sharma and S. Singh, "ISL recognition system using integrated mobile-net and transfer learning method," *Expert Systems with Applications*, vol. 221, Art. no. 119772, 2023, doi:10.1016/j.eswa.2023.119772.

[3] B. Subramanian, B. Olimov, S. M. Naik, S. Kim, K.-H. Park, and J. Kim, "An integrated mediapipe-optimized GRU model for Indian sign language recognition," *Scientific Reports*, vol. 12, Art. no. 11964, 2022, doi: 10.1038/s41598-022-15998-7.

[4] B. Narasimha Swamy, T. Sri Lakshmi, K. Sowjanya, M. Himabindu, K. S. V. V. Lakshmi, and S. V. M. Varma, "Indian Sign Language Recognition Using OpenCV and the MediaPipe: A Step Towards Inclusion," in *Proc. 2025 2nd Int. Conf. on Networks and Soft Computing (ICNSoC)*, 2025, pp. 497–502, doi: 10.1109/ICNSOC66817.2025.00097.

[5] P. Rawat, P. Kumar, V. K. Tamta, and A. Kumar, "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*, vol. 4, no. 1, 2025, doi: 10.4108/airo.8693.

[6] O. Tipare, S. Pathre, P. Pathak, and D. Karia, "GestureSpeak: A Real-Time Sign to Speech Translation," in *Proc. 2025 3rd Int. Conf. on Inventive Computing and Informatics (ICICI)*, 2025, pp. 120–125, doi:10.1109/ICICI65870.2025.11069898.

[7] T. Starner and A. Pentland, "Real-Time American Sign Language Recognition from Video Using Hidden Markov Models," in *Proc. Int. Workshop on Automatic Face and Gesture Recognition (IWAFGR)*, Zurich, Switzerland, Jun. 1995, pp. 189–196.

[8] O. Koller, J. Forster, and H. Ney, "Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers," *Computer Vision and Image Understanding*, vol. 141, pp. 108–125, Dec. 2015, doi: 10.1016/j.cviu.2015.09.013.

[9] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, "Neural Sign Language Translation," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7784–7793, doi: 10.1109/CVPR.2018.00812.

[10] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: A survey," *Artificial Intelligence Review*, vol. 43, pp. 1–54, 2015, doi: 10.1007/s10462-012-9356-9.



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511191

[11] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018,pp. 4510–4520,doi:10.1109/CVPR.2018.00474.

[12] A. Vakunov, C.-L. Chang, F. Zhang, G. Sung, M. Grundmann, and V. Bazarevsky, "MediaPipe Hands: On-device Real-time Hand Tracking," *arXiv preprint arXiv:2006.10214*, 2020.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[14] R. Damdoo and P. Kumar, "An integrative survey on Indian sign language recognition and translation," *IET Image Processing*, vol. 19, no. 1, Art. no. e70000, 2025, doi: 10.1049/ipr2.7000