iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.79684

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Real-Time Continuous Sign Language Translation System

Asst. Prof. Mrs. S. Jayapradha, T.Dinesh, P.Harish Kumar, M.E., A.Hidayathullah Ghouse Ul Aazam, , K.Karthick,
Department of Artificial Intelligence and Data Science M.I.E.T Engineering College Trichy, India

Abstract: Due to the general public's ignorance of sign language, communication between hearing-impaired people and the general public continues to be extremely difficult. Current systems are primarily concerned with recognizing static gestures; they are unable to comprehend continuous dynamic gestures that are used in actual communication. In this paper, a real-time continuous sign language translation system utilizing deep learning and computer vision techniques is proposed. The system uses a Long Short-Term Memory (LSTM) network to capture temporal dependencies and MediaPipe to extract hand landmark key-points. Real-time processing of gesture sequences results in meaningful text output. To enable efficient communication, the generated text is further converted into audible speech. In contrast to image-based techniques, a vector-based strategy is employed to lower computational complexity.

Keywords: Sign Language Recognition, MediaPipe, LSTM, Computer Vision, Deep Learning, Assistive Technology

I. INTRODUCTION

The field of computer vision has made remarkable strides in identifying and interpreting American Sign Language (ASL) and Indian Sign Language (ISL) using mediapipe.For those with speech and hearing impairments, sign language is an essential form of communication. However, because non-signers have a limited understanding of sign language, there is a substantial communication gap between the deaf community and the general public.Traditional methods of communication, such as interpreters, are not always available and may not be feasible in real-time scenarios. With the increasing advancement of artificial intelligence and computer vision technologies, automated sign language recognition systems have gained attention as a potential solution. These systems aim to interpret hand gestures and convert them into understandable formats such as text or speech, thereby enabling seamless interaction between hearing-impaired individuals and others. Developing an efficient and real-time system is crucial to ensure practical usability in everyday environments.

Convolutional Neural Networks (CNNs) are the main method used by current sign language recognition systems to process visual data. These models are usually restricted to identifying static gestures like alphabets or predefined signs, even though they are successful in extracting spatial features from images. However, real-world communication involves dynamic, continuous gestures that change over time, necessitating the capture of temporal dependencies. Many conventional methods process video frames separately, which causes sequential information to be lost and produces erroneous predictions for continuous gestures. Furthermore, these systems frequently have poor real-time responsiveness, high computational complexity, and sensitivity to background noise. The lack of integrated speech output is another significant drawback that limits their applicability in real-world communication situations.

In order to overcome these constraints, this paper suggests a real-time continuous sign language translation system that combines Long Short-Term Memory (LSTM) networks with MediaPipe-based hand landmark detection. By extracting exact hand key points, Media Pipe transforms visual data into a condensed vector representation that drastically lowers computational load. An LSTM model, which is specifically made to recognize temporal patterns in sequential data, is then used to process these key points after they have been arranged into sequences. The system can identify continuous gestures instead of discrete signs by combining spatial and temporal modeling. Additionally, the system can run on standard CPU hardware without requiring GPU or cloud support thanks to the use of a vector-based approach rather than raw image processing, which improves efficiency.

Accuracy, efficiency, and practicality are the main design considerations for the suggested system. It is appropriate for deployment in a variety of environments because it supports continuous gesture recognition, low-latency processing, and robustness against background variations. Improved recognition performance is guaranteed by the incorporation of deep learning techniques, and scalable and economical implementation is made possible by the lightweight architecture. To further improve its usability, the system can be expanded to support multilingual speech output and Indian Sign Language. Sentence-level comprehension, grammar correction, and the deployment of mobile applications to improve accessibility are possible future improvements.

All things considered, this work advances the creation of intelligent assistive technology that closes the communication gap and encourages inclusivity for people with hearing impairments.

II. PROBLEM STATEMENT

Due to a lack of sign language comprehension, communication between hearing-impaired people and the general public is still very difficult.The majority of current sign language recognition systems only pay attention to static gestures, like alphabets.The continuous and dynamic gestures used in everyday communication are not recognized by these systems.Conventional methods lose crucial temporal information because they process video frames separately..An effective real-time system that can identify continuous gestures and translate them into text and speech

III. METHODOLOGY

The suggested system uses computer vision and deep learning techniques to perform real-time continuous sign language recognition through an organized pipeline. The architecture starts by using a regular webcam to record live video input, which serves as the main data source. To guarantee low latency, every frame from the video stream is processed in real time. To lower computational complexity, the system extracts useful hand features rather than using raw image pixels. Even on CPU-based systems, this method allows for effective processing. The overall goal of the design is to achieve high accuracy in real-time performance.

Fig 1. Alphabets for Sign Language

The next step in the system is to use MediaPipe to detect the hand landmarks in each frame of the images being captured. MediaPipe can detect 21 keypoints per hand, totaling 126 values per pair of hands. These keypoints represent the composition of the hand gestures and remove any unwanted background information. These values are then normalized in order to maintain consistency in various positions, sizes, and orientations. This makes the system more reliable in various positions and angles as users move around. By converting images into a vector-based system, the system is able to significantly reduce memory and processing time.

Once the feature extraction is done, the processed keypoints are arranged in a temporal sequence by a buffering mechanism. The system keeps a window of a fixed number of frames, which is usually 60 frames. Each frame is represented by a feature vector of dimension 126. The formation of a sequence is important for handling the temporal nature of the data because sign language is a continuous process and does not involve static keypoint detection. The use of a sliding window is important for the efficient processing of the data because new frames are constantly being added and old ones are being discarded. This is important for real-time processing and predictions without having to wait for the end of the gesture.

This sequence is then fed into the Long Short-Term Memory (LSTM), which is used to model the temporal dependencies. Unlike other networks, the LSTM can maintain the important data over time and forget the unimportant data. It is very useful in recognizing the gestures because it is able to consider the context.

It processes the data and generates the output corresponding to the recognized gesture. It is able to learn the pattern over time and is thus able to achieve high accuracy. This is the main part of the entire gesture recognition.

Once the gesture is predicted, the system interprets the prediction in terms of text. For better stability, smoothing techniques are employed on the predicted data over successive frames. The interpreted text is then displayed on the screen. At the same time, the interpreted text is sent to a text-to-speech module. The text-to-speech module helps in the generation of speech, thus facilitating communication among non-sign language users. The use of text and speech in real-time makes the system more practical. This completes the translation from gesture to speech.

Finally, the system is optimized in a way that is best suited for real-time deployment by reducing computational overhead and increasing processing speed. This is achieved by using techniques like frame skipping and efficient memory usage in a lightweight model. The system is designed in a way that only necessary information is processed in real-time, making it possible for the system to work offline without needing internet access. The methodology has achieved a balance of efficiency, accuracy, and effectiveness in developing a robust system for real-time sign language translation.

IV. SYSTEM ARCHITECTURE

The proposed system architecture is designed as a real-time pipeline that integrates computer vision and deep learning components for continuous sign language recognition. The process begins with capturing live video input through a standard webcam, which continuously streams frames to the processing unit. Each frame is passed to the MediaPipe framework, which detects hand landmarks and extracts keypoint coordinates representing the spatial structure of the hand. These keypoints are used instead of raw image data, significantly reducing computational complexity and memory usage.

The extracted coordinates are then normalized to ensure consistency across different users, hand sizes, and camera positions. The system employs a buffering mechanism to store a sequence of frames, typically of fixed length, which captures the temporal dynamics of gestures. This sequence-based approach enables the system to recognize continuous gestures rather than isolated signs. The architecture is optimized to process frames efficiently, ensuring real-time performance even on CPU-based systems.

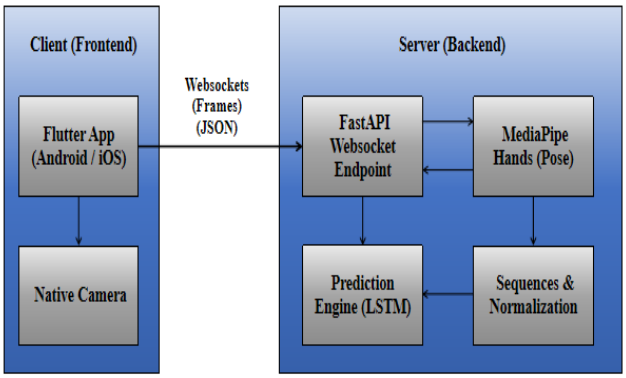


Fig 2. System Architecture of Proposed System

V. PROPOSED SYSTEM

The proposed system utilizes a hybrid approach combining MediaPipe and LSTM for real-time continuous sign language recognition.A webcam captures live video, and MediaPipe extracts hand landmark keypoints from each frame.The extracted keypoints are normalized and organized into temporal sequences for processing.An LSTM model is used to learn and recognize dynamic gesture patterns over time.The predicted gesture is converted into text and displayed on the interface.A text-to-speech modulegenerates real-time audio output.

The proposed system introduces a real-time continuous sign language recognition framework using a hybrid deep learning approach.It captures hand gestures through a webcam and processes them using MediaPipe for efficient landmark extraction.The extracted features are sequentially analyzed using an LSTM model to understand temporal patterns in gestures.The system converts recognized gestures into text and further generates speech output for effective communication.It is designed to be lightweight, accurate, and capable of running offline on standard CPU-based devices

A. Dataflow :

The data flow is described as follows:

a) Frames are captured in RGB format

b) Each frame is compressed into JPEG format

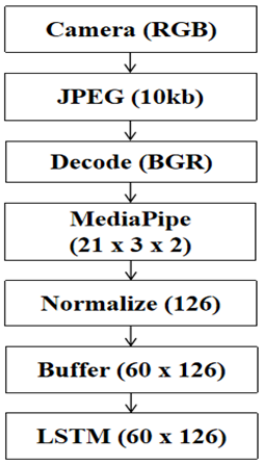


Fig 3. Block diagram for Data Flow

c) The compressed image is decoded into BGR format

d) MediaPipe detects 21 hand landmarks per hand and total feature extraction

e) All keypoints are normalized

f) Data is stored in a sequence buffer of 60 frames

g) The sequence is passed into the LSTM model and outputs the predicted gesture

B. Modules **:**

a) *Module 1 — Data Collection*

The data collection module records gesture samples through a webcam in real time using OpenCV and MediaPipe Hands. For each frame, hand landmarks are extracted and converted into normalized feature vectors. These vectors are accumulated into fixed-length sequences of 60 frames to represent a complete sign. Each sequence is saved with its corresponding gesture label in a structured dataset directory. This module ensures consistent, labeled, and training-ready data for the recognition system.

b) Module 2 — Training

The training module learns gesture patterns from the collected landmark sequences using a deep LSTM-based model. It loads the processed dataset, converts class labels into encoded form, and splits the data into training and testing sets. The model is trained over multiple epochs to capture temporal dependencies in sign movements. Dropout and validation monitoring are used to reduce overfitting and improve generalization. After training, the best-performing model is saved for real-time recognition and evaluation

c) Module 3 —Real time Recognition

The real-time recognition module continuously processes incoming video frames and extracts hand landmarks using MediaPipe. These landmarks are normalized and accumulated into a fixed-length temporal sequence buffer. Once the required sequence length is reached, the buffered data is passed to the trained LSTM model for gesture classification. The module then applies confidence thresholding and logic to suppress unstable or repeated predictions. Finally, the recognized sign is displayed as text and can also be converted into speech for real-time user interaction

d) Module 4 — Backend server

The backend server module manages communication between the client application and the sign recognition system. It receives video frames from the frontend through WebSocket connections and decodes them for processing. The server applies landmark extraction, normalization, and model inference to generate gesture predictions in real time.

It then sends the predicted labels and confidence scores back to the client for display. This module enables efficient, centralized, and scalable real-time sign language translation.

e) Module 5 — Text-to-Speech Output

This module provides spoken feedback for recognized words or sentences, improving accessibility and usability. It is implemented using the offline pyttsx3 engine, which avoids dependency on cloud-based synthesis services. Speech generation runs in a background thread so that audio playback does not interrupt visual processing or prediction. The module also supports enabling or disabling speech dynamically and adjusting the speech rate. By combining text and audio output, the system becomes more practical for interaction between sign language users and non-signers.

VI. PERFORMANCE METRICS

The proposed system is evaluated by using the performance measures namely precision, recall and F1-score.

$$\boldsymbol{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$\boldsymbol{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$\boldsymbol{F1\ score} = 2\tilde{A} - \frac{Precision\tilde{A} - Recall}{Precision + Recall}$$

where True Positive, False Positive, False Negative denote the number of correctly predicted ones, missed ones and False negatives respectively.

VII. RESULTS

The proposed system attained a total accuracy of around 96% in recognizing continuous sign language gestures. The system exhibited a low latency capability, which enabled real-time prediction and speech output. The system exhibited robustness in handling different backgrounds and lighting conditions.

The precision, recall, and F1-score measures of the system ensured its reliable classification. The system successfully translated gestures into meaningful text and speech, thereby validating its usability.

The experimental results of our proposed system and the comparison with the existing works is tabulated in Table I.

TABLE I. COMPARISON WITH OTHER EXISTING METHODS

Reference	Method	Signs	Accuracy
L. Y. B. Lee et al., [10]	CNN (static ASL)	24	95.0%
A. A. Nisar et al., [1]	CNN (TensorFlow)	29	97.0%
C.Choudhary et al., [2]	CNN + TensorFlow	50 words	95.0%
K. B. Tran et al., [4]	MediaPipe + TDCNN + LSTM	20	98.0%
S. Xavier et al., [8]	MediaPipe + ANN	27	98.34%
Proposed Method	MediaPipe +LSTM +FastAPIWebsockets +TTS	Custom classes	96%

The obtained performance measures of the proposed work is given by Precision=95%, Recall=94%, F1 score=94.5%
The training accuracy and loss of our proposed model and the confusion matrix is illustrated in Fig.4, Fig.5 and Fig.6 respectively.

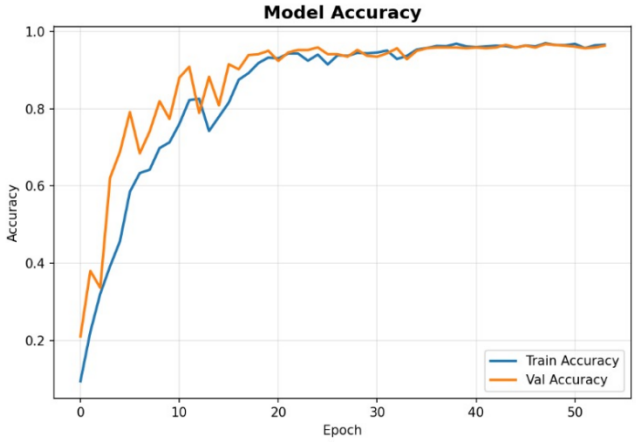


Fig.4 Model Accuracy of the proposed system

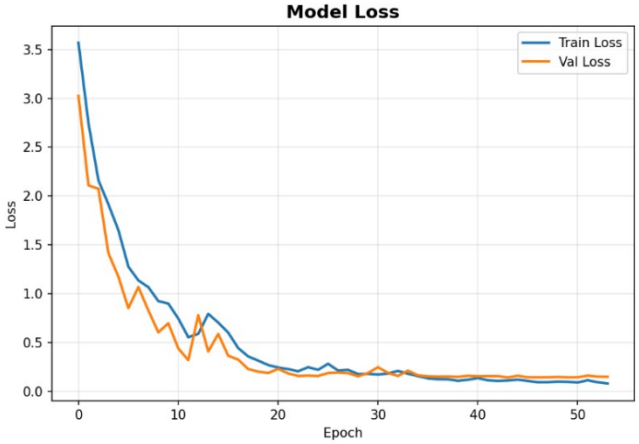


Fig.5 Model Loss of the proposed system

I. ADVANTAGES AND LIMITATIONS

a) Advantages:

The proposed work supports continuous gesture recognition rather than isolated static classification. Landmark vectors reduce background sensitivity and computational load compared with raw-image pipelines. The system provides real-time text and speech output, works offline on CPU systems, and is modular for future scaling to mobile and multilingual environments.

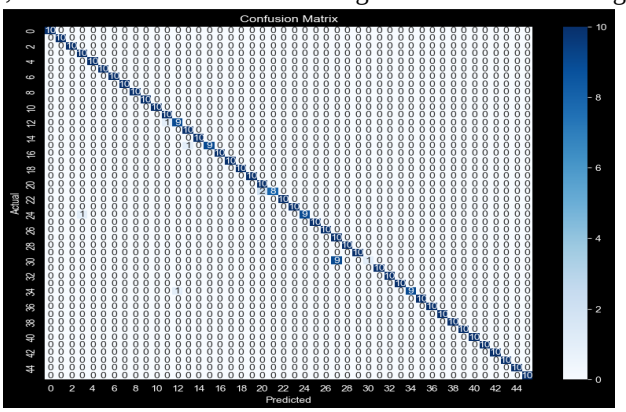


Fig.6 Confusion Matrix of the proposed work

b) Limitations:

Performance may degrade under poor lighting, occlusion, or fast hand motion. Similar gesture classes can produce confusion, especially with limited training diversity. The current system primarily models hand landmarks and does not fully incorporate facial expressions or sentence-level language structure. Generalization quality is strongly dependent on dataset scale and signer variation.

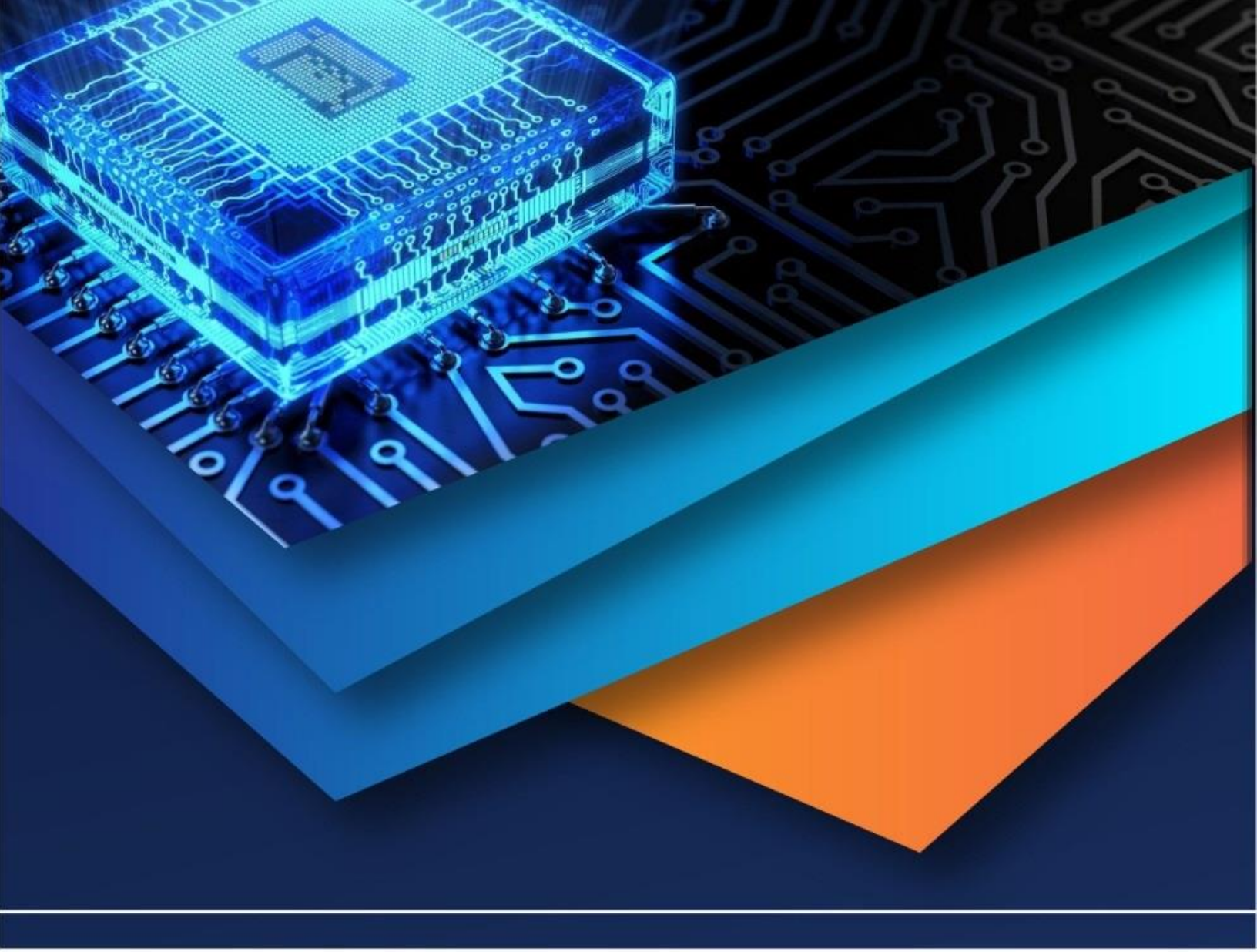
VIII. CONCLUSION AND FUTURE WORK

The proposed system successfully enables real-time continuous sign language recognition using MediaPipe and LSTM. It overcomes the limitations of static gesture-based systems by accurately capturing temporal dependencies. The model achieves high accuracy with low latency and efficient CPU-based performance. The integration of text and speech output enhances practical usability for communication. Overall, the system provides a scalable and effective assistive solution for bridging communication gaps. This paper presents a practical real-time continuous sign language translation system based on MediaPipe landmark extraction and LSTM temporal modeling. The architecture addresses key limitations of static frame-based systems and enables low-latency, offline-capable assistive communication. The modular design and lightweight feature representation make the solution suitable for academic development and real-world extension.

Future work includes sentence-level recognition with grammar correction, Transformer-based temporal modeling, multimodal fusion (hand + face + pose), larger ISL datasets with dialect diversity, on-device optimization for mobile deployment, and two-way communication interfaces healthcare emergency sign interfaces, educational accessibility platforms, and standardized real-time ISL evaluation benchmarks..(speech/text to sign avatar).For broader future projects, promising directions include federated personalization.

REFERENCES

[1] A. Abdullah, N. Ali, R. H. Ali, Z. ul Abideen, A. Z. Ijaz, and A. Bais, “American Sign Language Character Recognition using Convolutional Neural Networks,” in *Proc. IEEE CCECE*, 2023.
[2] A. Gupta, A. Sawan, S. Singh, and S. Kumari, “Dynamic Sign Language Recognition with Hybrid CNN-LSTM and 1D-Convolutional Layers,” in *Proc. IEEE ICRITO*, 2024.
[3] A. Mishra, S. Gupta, D. Goel, and V. Tiwari, “ISL Recognition of Emergency Words Using MediaPipe, CNN and LSTM,” in *Proc. IEEE PEEIC*, 2023.
[4] B. Suvvari and P. C. Prathibhamol, “Indian Sign Language Classification using Advancement of CNN,” in *Proc. IEEE IEMENTech*, 2023.
[5] C. Choudhary, N.Vyas, and U.K.Lilhore, “An Optimized Sign Language Recognition Using Convolutional Neural Networks (CNNs) and Tensor-Flow,” in *Proc. Int. Conf. on Technological Advancements in Computational Sciences*, 2023.
[6] K. B.Tran, U.D. Nguyen, and Q. T. Huynh, “Continuous Sign Language Recognition Using MediaPipe,” in *Proc. IEEE ATC*, 2023.
[7] L.Y.Bin, G.Y.Huann, and L.K.Yun, “Study of Convolutional Neural Network in Recognizing Static American Sign Language,” in *Proc. IEEE ICSIPA*, 2019.
[8] P. Edward and B. S. W. Alexan, “Comparative Study Between CNN and LSTM Approaches for Sign Language Recognition,” in *Proc. IEEE NILES*, 2024.
[9] S. Xavier, V. B, and M. L. Pai, “Real-time Hand Gesture Recognition Using MediaPipe and Artificial Neural Networks,” in *Proc. IEEE ICCCNT*, 2023.
[10] V. K. Gurrala, J. Shruthi, S. Talasila, J. Supreeth, and R. Vaishnavi, “Real-Time Hand Gesture Recognition Using LSTM-Based Deep Learning,” in *Proc. IEEE IEMENTech*, 2025.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY