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 4, April 2026

DOI: 10.17148/IJARCCE.2026.15428

Deep Learning Based Real-Time Sign Language Recognition

Prof. Minal Patil[1], Rhushabh Gaikwad[2], Rushikesh Ghogare[3], Shekhar Khandale[4], Roshan Avhad[5]

Professor, Computer Engineering, Indira College of Engineering and Management, Pune, India[1]

Students, Computer Engineering, Indira College of Engineering and Management, Pune, India[2]

Students, Computer Engineering, Indira College of Engineering and Management, Pune, India[3]

Students, Computer Engineering, Indira College of Engineering and Management, Pune, India[4]

Students, Computer Engineering, Indira College of Engineering and Management, Pune, India[5]

Abstract: Sign language is a vital medium of communication for individuals with hearing and speech impairments, but the lack of knowledge among non-signers creates barriers. This project proposes a real-time sign language recognition system that can detect alphabets (A-Z) and numerics (0-9) from webcam video. The system combines modern deep learning techniques with web technologies to provide an accurate, fast, and user-friendly solution. The frontend uses WebRTC to capture video streams directly in a browser, making the system platform-independent and usable with any standard laptop or external camera. The backend uses FastAPI with WebSockets to enable real-time communication between the browser and the deep learning model, ensuring low-latency predictions.

The recognition model integrates EfficientNet (transfer learning) for feature extraction, combined with a CNN+RNN to capture spatial and temporal patterns. An attention mechanism enhances performance by focusing on the most informative frames, while a GRU classifier predicts the final alphabet or number with high accuracy. Training and validation are carried out using benchmark datasets along with self-collected samples to ensure adaptability in real-world conditions. The system prototype displays recognized signs as text beneath the video feed, with emphasis on accuracy, robustness, and real-time performance for applications in education, healthcare, and accessibility services.

Keywords: Sign Language Recognition, Deep Learning, Computer Vision, Real-Time System, Gesture Detection, Neural Networks

I. INTRODUCTION

Sign language is a powerful and expressive mode of communication primarily used by individuals with hearing and speech impairments. Unlike spoken languages, which rely on sound, sign language uses gestures, facial expressions, and hand movements to convey meaning. However, a communication gap persists between the hearing-impaired community and the rest of the population, as most people do not understand sign language fluently. This challenge motivates the development of technological solutions that can interpret sign language gestures in real time and translate them into text or speech.

In this project, we propose a real-time sign language recognition system based on deep learning techniques. The system captures live video using a webcam, processes hand gestures through neural network models, and displays the recognized alphabets or numerics on the screen. The core concept is to integrate computer vision, deep learning, and web technologies to create an assistive tool that bridges the communication gap between signers and non-signers.

The system uses a web-based frontend built with WebRTC for real-time video capture and a Python backend (FastAPI) for deep learning inference. The model architecture combines EfficientNet (for spatial features), CNN+RNN layers (for temporal analysis), an attention mechanism to focus on key frames, and a GRU (Gated Recurrent Unit) classifier for final prediction. The system is lightweight, fast, and easily accessible through any modern web browser.

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 4, April 2026

DOI: 10.17148/IJARCCE.2026.15428

II. LITERATURE SURVEY

Sign Language Recognition (SLR) has evolved significantly in recent years, driven by advancements in computer vision and deep learning. Initially, recognition systems relied on hardware-based sensors such as gloves, accelerometers, and infrared devices to track finger positions and hand movements. While these methods provided accurate motion tracking, they were intrusive, costly, and impractical for daily communication. The focus has since shifted toward vision-based approaches using standard cameras, which are non-invasive and suitable for real-time applications.

Recent deep learning architectures have greatly improved the accuracy and efficiency of SLR systems. EfficientNet has been employed to extract rich spatial features from input frames, while hybrid CNN-RNN architectures capture temporal dependencies in gesture sequences. Attention mechanisms are used to highlight key frames and reduce redundancy, leading to enhanced recognition accuracy even under challenging lighting or background conditions [1].

Various transfer learning approaches have also been explored to optimize model performance. Pretrained models such as VGG16, VGG19, ResNet50, and MobileNetV2 have shown remarkable success in recognizing static hand gestures from Indian Sign Language (ISL). Among these, MobileNetV2 stands out for its compact size, reduced computation time, and high validation accuracy, demonstrating its suitability for lightweight, real-time systems [2].

Comprehensive reviews of over a hundred research studies on deep-learning-based sign language recognition reveal a common focus on CNN-based feature extraction and temporal modelling using RNN, LSTM, and GRU networks. These studies highlight key challenges such as dataset limitations, occlusion, signer dependency, and environmental variations, which hinder the generalization of models across diverse users and conditions [3].

For Indian Sign Language, CNN-based recognition systems have achieved impressive results, with models reaching up to 99.93% accuracy when tested on well-structured datasets. These systems demonstrate that deep CNNs can effectively recognize isolated alphabets and numeric gestures, setting a strong foundation for further development of dynamic and real-time recognition architectures [4].

Other research on regional sign languages, such as Nigerian and Kazakh Sign Language, has extended these findings. CNN-based systems have achieved around 95% accuracy for Nigerian Sign Language, whereas recent approaches using YOLOv8 combined with optimized 2DCNN models for Kazakh Sign Language achieved over 98% accuracy in continuous recognition. These results demonstrate that combining real-time object detection with temporal modelling can significantly improve performance and robustness in continuous video-based recognition [5] [6].

III. ARCHITECTURE

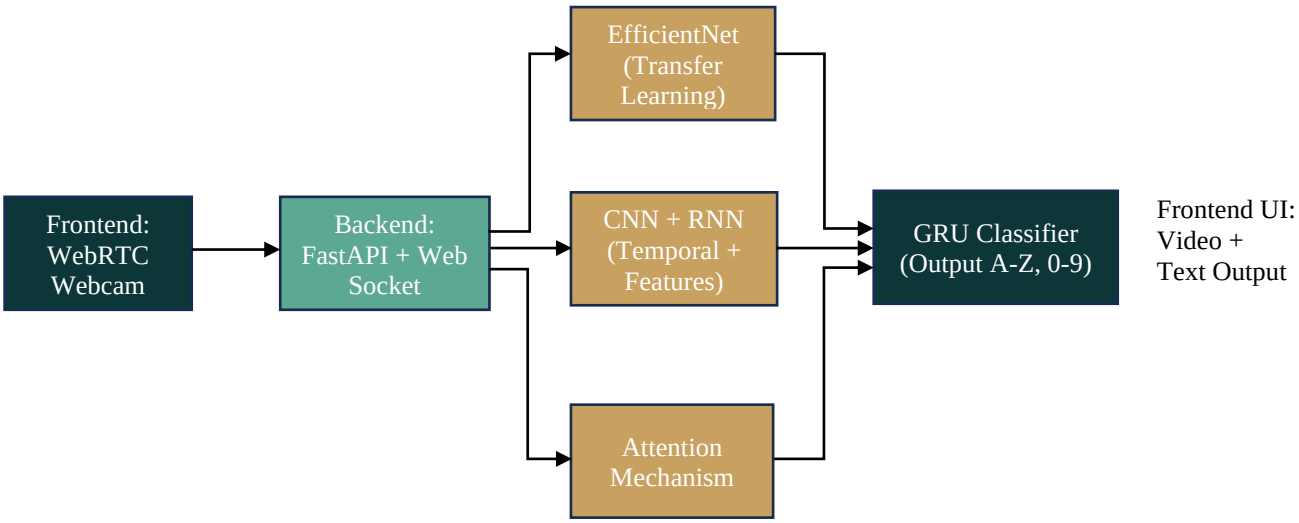

Fig. 1 System Architecture

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.15428

The architecture of the proposed system, titled Deep Learning Based Real-Time Sign Language Recognition, follows a modular and layered design integrating both web technologies and deep learning components to achieve efficient real-time inference.

The overall architecture is illustrated in Figure 1, consisting of five major components:

A. Frontend (WebRTC Webcam)
B. Backend (FastAPI with WebSocket)
C. Deep Learning Modules (EfficientNet, CNN+RNN, and Attention Mechanism)
D. GRU Classifier
E. Frontend Output Interface (Video + Text Display)

A. Frontend: WebRTC Webcam

The frontend is a web-based interface implemented using WebRTC, allowing live video streaming directly through the browser. It captures hand gesture frames in real time from a webcam and sends them to the backend server via WebSocket protocol. This eliminates the need for any additional software installation, making the system platform independent and accessible through standard browsers.

B. Backend: FastAPI + WebSocket

The backend server, built using FastAPI, acts as the communication bridge between the frontend and the deep learning model. It establishes a persistent WebSocket connection for low-latency streaming of image frames. Each frame or sequence of frames received from the frontend is pre-processed, normalized, and forwarded to the model for prediction. The asynchronous handling capability of FastAPI ensures efficient real-time communication.

C. Deep Learning Model: EfficientNet, CNN+RNN, and Attention Mechanism

The model architecture is designed as a hybrid pipeline combining spatial, temporal, and contextual feature extraction:

- EfficientNet (Transfer Learning):

Extracts high-level spatial features from individual frames using pretrained weights, improving accuracy and reducing training time.

- CNN + RNN (Temporal + Feature Learning):

The CNN component captures short-term spatial dependencies, while the RNN component (using GRU units) models the temporal evolution of gestures across multiple frames.

- Attention Mechanism:

Dynamically assigns weights to key frames, enabling the network to focus on the most informative hand shapes and motions while suppressing irrelevant transitions.

These three modules work collaboratively to produce a rich, temporally-aware feature representation of the input video sequence.

D. GRU Classifier

The fused feature vector obtained from the above modules is passed through a GRU based classifier that predicts the corresponding sign label. The classifier outputs one of the 36 possible categories (A-Z alphabets and 0-9 numerics). The final output is accompanied by a confidence score to indicate prediction reliability.

E. Frontend Output Interface

After prediction, the backend sends the recognized sign label back to the frontend interface via the same WebSocket channel. The recognized text is displayed immediately below the live video feed, providing users with visual feedback in real time. This feedback loop ensures smooth user interaction with minimal delay.

IV. RESULTS

This chapter presents the results obtained from the developed real-time sign language recognition system. It evaluates the performance of the system in terms of accuracy, latency, and real-time responsiveness. The results are analysed using both training metrics and practical testing through live webcam input.

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 4, April 2026

DOI: 10.17148/IJARCCE.2026.15428

The objective of this chapter is to validate the effectiveness of the proposed deep learning model and to demonstrate its capability in recognizing hand gestures accurately in real time scenarios.

A. Accuracy

Accuracy measures how many sign sequences are correctly classified out of the total validation samples. In this project, each sample is a sequence of 8 hand-cropped frames, and the model predicts 36 classes (A–Z, 0–9). The dataset is split into 80% training and 20% validation, and accuracy is evaluated on the validation set after each epoch.

During training on the full dataset, the model achieved a best validation accuracy of 99.99% (0.9999) at Epoch 6, and the best performing weights were automatically saved for deployment. This indicates the model can reliably distinguish between all 36 sign classes in real-time conditions.

B. Latency / Frame Rate

The system is configured to capture frames at approximately 10 frames per second (100 ms interval in the frontend) and perform predictions every 2 frames (around 5 predictions/sec). This provides smooth real-time recognition while keeping inference stable. Prediction latency is measured on the backend as model inference time in milliseconds, and the average latency recorded during testing was 70 ms (min 20 ms, max 120 ms).

C. Observations

- The system achieves real-time performance, capturing frames at 10 fps and generating predictions every 2 frames (5 predictions/sec), which is smooth enough for continuous gesture recognition.
- With the full dataset, the model converged quickly and reached very high validation accuracy (99.99%), showing strong class separation for both alphabets and numerals.
- Backend inference latency remained low, typically between 20–120 ms, allowing responsive real-time output.
- The model's accuracy improved rapidly in early epochs, after which improvements became marginal, indicating that early stopping is effective to prevent overfitting and reduce training time.
- Automatic data collection successfully saves high-confidence, good-quality hand frames into the correct class folders, enabling incremental dataset growth.
- GPU usage significantly reduced training time compared to CPU, confirming that CUDA acceleration is necessary for large-scale training on this architecture.

D. Challenges Faced

- **High training time:** Due to the large dataset and sequence-based input (8 frames per sample), each epoch required significant processing time, even with GPU acceleration.
- **GPU memory limits:** The GTX 1650 (4 GB VRAM) restricts batch size; larger batch sizes caused memory errors and required careful tuning.
- **Class imbalance:** Numeral folders (0–9) initially contained fewer samples compared to alphabet folders, which can bias training if not managed.
- **WebSocket stability:** Continuous frame transmission required careful handling to avoid disconnections and invalid frame errors.
- **Permission and browser issues:** Webcam access depended on browser permissions and local server setup, which needed troubleshooting during deployment.
- **UI data consistency:** Ensuring history/logs updated in real time without lag required lightweight logging and optimized frontend polling.

E. Screenshots

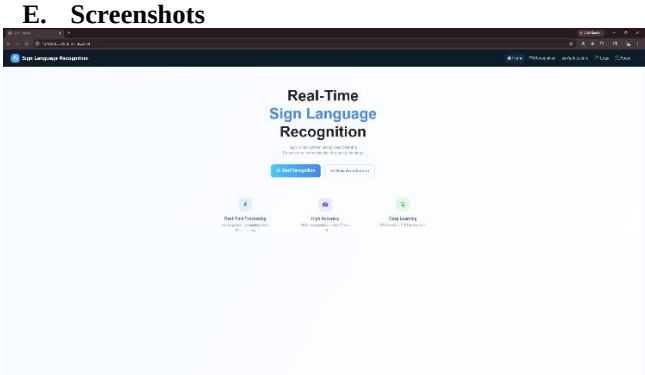

Fig. 2. Home Page

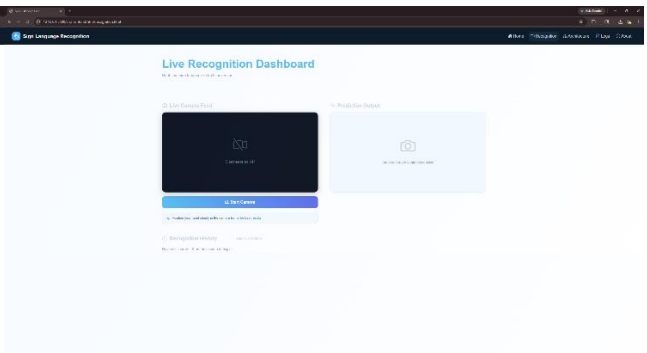

Fig. 3. Recognition Dashboard

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 4, April 2026

DOI: 10.17148/IJARCCE.2026.15428

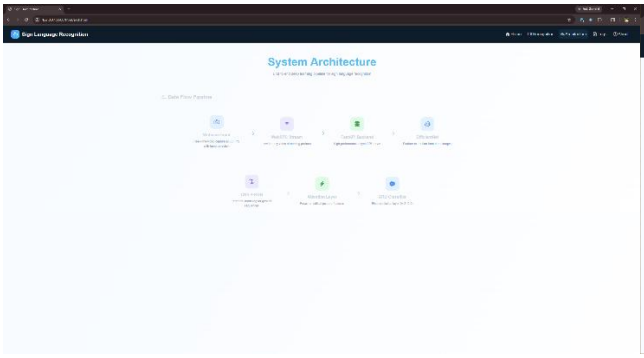

Fig. 4. System Architecture Page

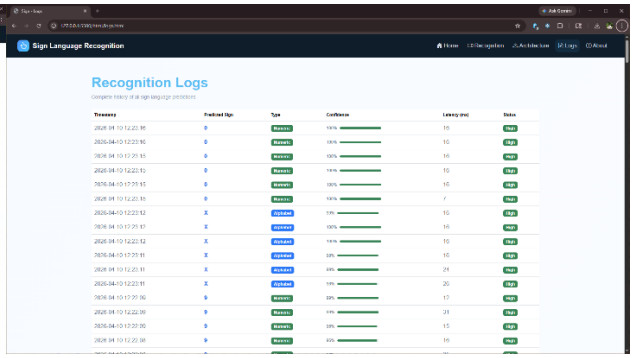

Fig. 5. Recognition Logs

V. CONCLUSION

The project "Deep Learning Based Real-Time Sign Language Recognition" has been successfully implemented and evaluated in Stage-II. This phase focused on the development, integration, and testing of the complete system, transforming the proposed design into a fully functional application. The system integrates a WebRTC-based frontend for real-time video capture with a FastAPI backend using WebSocket communication for low-latency data transmission. A hybrid deep learning architecture consisting of EfficientNet for feature extraction, CNN RNN for temporal modeling, attention mechanism for feature prioritization, and GRU for final classification was implemented as per the proposed design.

The model was trained using a structured dataset of hand gesture images representing alphabets (A–Z) and numerics (0–9). The system achieved high accuracy in real-time prediction, with performance optimized to maintain low latency suitable for live applications. Additionally, a dynamic dataset collection mechanism was implemented to automatically store high-quality frames, enabling continuous system improvement.

The frontend interface successfully displays live camera feed, prediction output, recognition history (last 10 predictions), and detailed logs including confidence and latency. Extensive testing, including manual and automated testing, confirmed that all modules function correctly and meet system requirements.

The results demonstrate that the system effectively recognizes sign language gestures in real time with reliable accuracy and responsiveness. This project contributes toward improving accessibility and communication for the hearing-impaired community.

Overall, the successful completion of Stage-II validates the feasibility and effectiveness of the proposed system. The developed solution provides a strong foundation for further enhancements such as continuous sign recognition, mobile deployment, and multi-language support.

VI. FUTURE SCOPE

While the current phase focuses on design and planning, several key tasks are scheduled for the implementation phase and beyond:

A. Model Implementation and Training

- Develop and train the hybrid deep learning architecture using real and augmented sign datasets.
- Fine-tune EfficientNet and GRU layers to balance accuracy and speed for real time inference.

B. Real-Time System Integration

- Connect the WebRTC frontend with the FastAPI backend through WebSockets for low-latency frame streaming.
- Implement a responsive web interface for smooth camera control and output display.

C. Continuous Sign Recognition

- Extend from isolated alphabet/number recognition to continuous gesture interpretation for complete sentence translation.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.15428

- Employ advanced sequence modeling methods such as Transformers or Connectionist Temporal Classification (CTC).

D. Optimization and Deployment

- Apply model compression and quantization for faster inference on CPU and edge devices.
- Create a deployable web and mobile version with a user-friendly interface.

E. Voice Output and Multilingual Support

- Integrate text-to-speech modules to vocalize recognized signs.
- Provide multilingual support for translated text and speech output.

F. User Testing and Feedback Integration

- Conduct testing with real users, including hearing-impaired participants, to assess usability and accuracy.
- Use feedback to improve gesture clarity, interface accessibility, and model robustness.

G. Data Expansion and Generalization

- Continuously expand the dataset with varied lighting conditions, skin tones, and backgrounds.
- Implement domain adaptation to improve generalization across users.

REFERENCES

[1]. Yenisari, E., & Yavuz, S. (2025). "Deep Learning Based Sign Language Recognition Using Efficient Multi-Feature Attention Mechanism". *Digital Object Identifier 10.1109/ACCESS.2024. Doi Number*.

[2]. Ingoley, S., & Bakal, J. (2025). "Efficient Models Based on Deep Learning Technique for Indian Sign Language". *Procedia Computer Science, 258(2025) 318-331*

[3]. Abd Al-Latief, S. T., Yussof, S., Ahmad, A., & Khadim, S. (2024). "Deep Learning for Sign Language Recognition: A Comparative Review". *Journal of Smart Internet of Things (JSIoT), VOL 2024, No.01 | 77-116 | 2024 DOI:10.2478/jsiot-2024-0006.*

[4]. Chikkamath, S., Budiha, S.V., & S. R. Nirmala. et al. (2024). "Deep Learning based Indian Sign Language Recognition for People with Speech and Hearing Impairment". *Conference Paper March 2024 DOI: 10.1109/InC460750.2024.10649094*.

[5]. Adewole, D. B., Adesugba, A., Agbelusi, O., & Olatunde, O. V. (2024). "Sign Language Recognition Using Deep Learning: Advancements and Challenges". *INTERNATIONAL JOURNAL OF LATEST TECHNOLOGY IN ENGINEERING, MANAGEMENT & APPLIED SCIENCE (IJLTEMAS), ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIII, Issue XII, December 2024.*

[6]. Zholshiyeva, L., et al. (2025). "Deep Learning-Based Continuous Sign Language Recognition". *Journal of Robotics and Control (JRC), Volume 6, Issue 3, 2025 ISSN: 2715-5072, DOI: 10.18196/jrc.v6i3.25881*.

