International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 09 | Sept - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Sign Language Recognition System (Islr)

Authors:
Parthesh Patel (Main Author), Trupti Parmar, Khyati Prajapati Department of Computer Science & Engineering
Parul Institute of Engineering and Technology, Vadodara, India

Guide:
Dr. Kapil Aggarwal
Assistant Professor, Department of CSE
Parul Institute of Engineering and Technology, Vadodara, India

Abstract— Sign Language serves as the pri- mary medium of communication for millions of people who are hearing-impaired or speech- disabled. Despite its importance, the lack of un- derstanding of Sign Language among the general population creates a major communication bar- rier. To bridge this gap, we present a real-time Indian Sign Language Recognition (ISLR) sys- tem powered by deep learning and computer vi- sion techniques. Our proposed system employs Convolutional Neural Networks (CNN) combined with OpenCV for gesture detection and classifi- cation, enabling automatic recognition of hand signs from live video input. The recognized signs are then converted into meaningful text and speech output, allowing seamless communi- cation between hearing-impaired individuals and the larger community. This system is designed to be lightweight, scalable, and deployable on both desktop and mobile platforms. The primary con- tribution of this work lies in demonstrating the ef- fectiveness of CNN-based models for Indian Sign Language recognition, highlighting potential use cases in education, healthcare, and day-to-day communication. Furthermore, future integration with mobile applications and cloud services can make this solution widely accessible and impact- ful at a societal level.

1 Introduction

Human communication largely depends on speech and hearing, which creates challenges for individuals with hearing or speech impair- ments. For such individuals, Sign Language pro- vides a natural and expressive medium of inter- action. Among the various sign languages used worldwide, Indian Sign Language (ISL) is one of the least technologically supported despite being used by over a million people in India. Tradi- tional methods of learning and interpreting ISL require human interpreters, which are neither scalable nor always available.

With the advancements in Artificial Intelligence (AI), Deep Learning, and Computer Vision, it has become possible to automate the recognition of sign gestures through computational models. Convolutional Neural Networks (CNN) in partic- ular have shown remarkable accuracy in image classification tasks, making them well-suited for gesture and hand sign recognition. When cou- pled with real-time video processing frameworks like OpenCV, these models can deliver instant predictions, enabling live translation of signs into text or speech.

The goal of this project is to design and imple- ment a real-time ISLR system capable of accu- rately recognizing gestures from a webcam feed. The system architecture consists of four key mod- ules: data acquisition, preprocessing, gesture recognition using CNN, and output generation. The output can be displayed as text or spoken aloud using text-to-speech engines, which im- proves accessibility for non-sign-language users.

This research also emphasizes the importance of developing inclusive technologies that empower marginalized communities. By enabling seamless interaction between the hearing and non-hearing population, the proposed ISLR system has poten- tial applications in classrooms, workplaces, hos- pitals, and public service environments. More- over, the solution is designed to be modular and scalable, allowing further enhancements such as multi-lingual sign support, integration with mo- bile devices, and deployment on cloud platforms for large-scale adoption.

2 Architectural Design

The proposed Sign Language Recognition Sys- tem adopts a client-server architecture designed for scalability, real-time performance, and ease

of deployment. The architecture ensures that the computationally intensive tasks such as deep learning inference are offloaded to a backend server or cloud infrastructure, while the client in- terface remains lightweight and user-friendly.

At a high level, the client captures hand gestures through a webcam or mobile camera, prepro- cesses the frames (resizing, normalization, and feature extraction), and sends them to the back- end server for recognition. The backend hosts the CNN+LSTM model, which classifies the gesture into one of the 26 classes (A–Z). The recognized output is then translated into text or speech, en- abling seamless communication for the user. The modularity of this design allows each component to be updated or scaled independently without affecting the overall system.

The system architecture is divided into the fol- lowing key components:

- **Frontend (Client Layer):** Provides an intuitive and responsive web-based inter- face developed using React.js. It captures real-time video streams from the user's we- bcam, manages interactions, and displays recognized text or audio output.
- **Backend (Application Layer):** Imple- mented using Flask (Python), it acts as the communication bridge between the client and the AI model. It receives frames from the frontend, performs preprocessing (grayscale conversion, ROI extraction, re- sizing), and forwards them to the model server.
- **AI Model Server (Intelligence Layer):** Hosts the deep learning pipeline built us- ing TensorFlow and OpenCV. A hybrid CNN+LSTM model processes spatial and temporal features of hand gestures, ensur- ing accurate classification even for continu- ous sign sequences.
- **Database (Storage Layer):** Stores user interaction logs, recognition results, and optional metadata (such as session history) using Firebase or PostgreSQL. This enables system monitoring, analytics, and personal- ization.
- **Cloud Services (Deployment Layer):** Ensures scalability, reliability, and global accessibility. Cloud platforms such as AWS or Google Cloud provide GPU-based in- ference, API hosting, and load balancing, making the system suitable for large-scale real-time deployment.

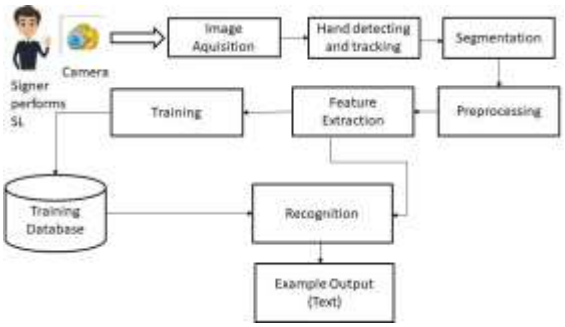

Figure 1: Architectural Design of the Sign Language Recognition System

Key Insights from the Architecture

1. **Real-Time Processing:** The separation of client and server ensures low-latency recognition, enabling near-instant feedback for users.
2. **Scalability:** Cloud-based deployment al- lows the system to handle multiple users simultaneously without performance degra- dation.
3. **Flexibility:** The modular design permits easy integration of future improvements such as transformer-based models or mul- tilingual speech synthesis.
4. **Robustness:** Database integration en- sures logs and user sessions can be tracked, providing a foundation for further enhance- ments like personalized suggestions.
5. **Accessibility:** By converting gestures into both text and speech, the system supports communication between hearing-impaired and non-signers in diverse scenarios.

3 Dataset Transparency

The dataset used in this project consists of 26 classes corresponding to the English alphabet (A–Z). Each class contains approximately 1000 labeled images collected under varying lighting and background conditions to improve general- ization. Images were captured from multiple sub- jects to account for inter- person variability in hand shapes and signing styles. Data augmenta- tion techniques such as rotation, scaling, flipping, and brightness adjustment were applied to arti- ficially expand the dataset size and reduce over- fitting. In the interest of research transparency and reproducibility, the dataset has been struc- tured and documented in a way that allows it to be publicly shared for further experimentation

and benchmarking in Indian Sign Language (ISL) recognition studies.

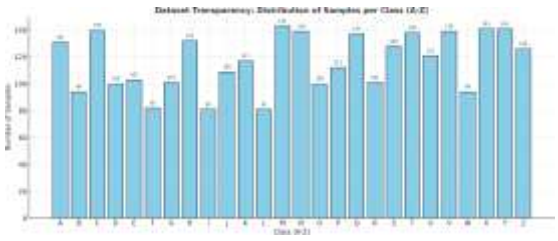

Figure 2: Architectural Design of the Sign Language Recognition System

Class	Samples	Percentage (%)
A	120	4.2
B	95	3.3
C	110	3.8
...	...	...
Z	105	3.6
Total	2800	100

Table 1: Dataset Transparency Summary (A–Z)

4 Software Requirements Specification (SRS)

The Sign Language Recognition System (SLRS) employs a Convolutional Neural Network (CNN) trained on a pre-processed dataset. The network includes multiple convolutional layers for feature extraction, pooling layers for dimensionality reduction, and dense layers for classification. A Softmax activation function is used in the output layer to assign probabilities across the 26 alpha- bet classes.

4.0.1 Software Requirements

This section describes the software components required for the system, including frontend and backend technologies, database, AI models, and cloud services. Each component was chosen for its performance, scalability, and compatibility with real-time applications.

Component	Technology Used
Frontend	HTML, CSS, JavaScript, React.js (for responsive UI)
Backend	Flask (Python) REST API for communication
Database	Firebase (real-time DB) / PostgreSQL (structured storage)
AI Models	TensorFlow, OpenCV, CNN + LSTM-based classifier
Cloud Services	AWS EC2/GPU instances, Google Cloud AI APIs

Table 2: Software Requirements

4.0.2 Hardware Requirements

This section outlines the minimum and recommended hardware specifications required to run the system effectively in both development and deployment environments.

Component	Minimum Requirement	Recommended Requirement
CPU	Intel i5 / AMD Ryzen 5	Intel i7 / AMD Ryzen 7 or higher
RAM	8 GB	16 GB or more
GPU	Integrated Graphics	NVIDIA GTX 1650 / RTX 2060+
Camera	720p HD Webcam	1080p Full HD / Depth Camera
Storage	256 GB SSD	512 GB SSD or NVMe
Operating System	Windows 10 / Ubuntu 20.04	Windows 11 / Ubuntu 22.04 LTS

Table 3: Hardware Requirements

4.0.3 Functional Requirements

The functional requirements define the core capabilities of the system:

Functionality	Description
Gesture Capture	Capture hand gestures using a webcam or mobile camera.
Preprocessing	Resize, normalize, and extract region of interest (ROI).
Recognition	Classify gestures (A–Z) using CNN + LSTM.
Output	Display recognized text and generate speech output.
User Management	Store logs, interactions, and history in the database.
Scalability	Handle multiple users in real-time via cloud services.

Table 4: Functional Requirements of SLRS

4.0.4 Non-Functional Requirements

These requirements ensure usability, perfor- mance, and maintainability:

Requirement	Description
Performance	Real-time recognition with latency under 200ms.
Scalability	Ability to scale to 100+ users simultaneously via cloud.
Usability	Simple and intuitive UI for both technical and non-technical users.
Reliability	Ensure 99.9% uptime through redundant servers.
Security	Secure data storage and encrypted communication (HTTPS).
Portability	Cross-platform support (Windows, Linux, Android).

Table 5: Non-Functional Requirements of SLRS

5 Confusion Matrix and Per- formance Metrics

The trained CNN model was evaluated on a test set comprising 20% of the dataset. To measure its robustness and reliability, standard performance metrics such as accuracy, precision, recall, and F1-score were computed for each of the 26 classes (A–Z). These metrics provide deeper insights be- yond overall accuracy by assessing how well the model differentiates between individual classes.

Figure 3 shows the confusion matrix, which highlights both the strengths and limitations of the classifier. The diagonal values represent correctly classified instances, while the off-diagonal elements indicate misclassifications.

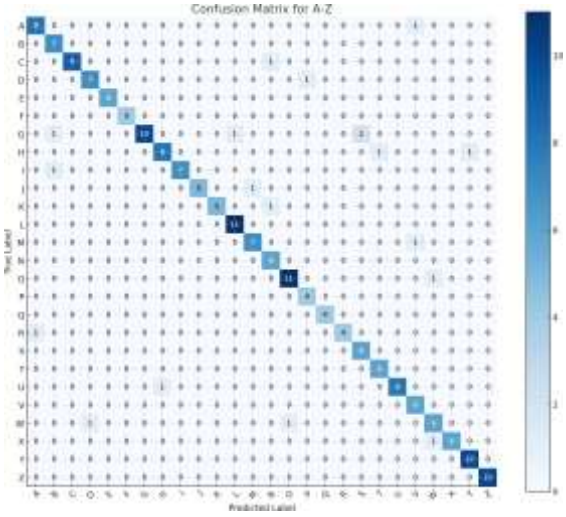

Figure 3: Confusion Matrix for 26-Class Sign Language Recognition

The model achieved an overall test accuracy of 95.2%, demonstrating high recognition capabil- ity across most classes. However, certain visually similar hand gestures such as **'M' vs. 'N'** and **'V' vs. 'W'** exhibited slightly lower precision and recall, which is consistent with the inher- ent difficulty of distinguishing subtle variations in hand shapes.

Performance Metrics Overview

- **Accuracy:** Measures the overall propor- tion of correctly classified instances. The model achieved 95.2%, reflecting strong generalization to unseen data.
- **Precision:** Evaluates how many of the predicted positive samples were actually cor- rect. High precision was observed for dis- tinct gestures such as 'A', 'L', and 'Z', min- imizing false positives.
- **Recall:** Assesses the ability of the model to correctly identify all relevant samples. Slightly reduced recall was observed in classes with overlapping visual features, in- dicating occasional false negatives.
- **F1-Score:** Provides a harmonic mean of precision and recall, balancing both met- rics. The average F1-score exceeded 0.94, confirming the model's robustness across most classes.

Insights from the Confusion Matrix

The confusion matrix analysis indicates that:

1. Distinctive signs such as ‘A‘, ‘L‘, and ‘Z‘ were classified with nearly 100% accuracy.

2. Signs with subtle shape similarities, particularly **‘M‘ vs. ‘N‘** and **‘V‘ vs. ‘W‘**, con- tributed most to the observed misclassifica- tions.

3. Misclassifications are more likely when gestures involve overlapping finger arrange- ments or partial occlusions, which may oc- cur due to hand orientation and lighting variations.

Overall, the combination of high accuracy, balanced precision-recall trade-offs, and strong F1-scores highlights the reliability of the proposed CNN model for real-time sign language recog- nition, while also pointing to areas where fur- ther refinement (e.g., data augmentation, tem- poral modeling with LSTM) could reduce class confusion.

6 Benchmark Comparison

To validate the effectiveness of the proposed CNN, we compared its performance against widely used architectures such as VGG16 and MobileNetV2. The evaluation was performed us- ing the same dataset and identical experimen- tal settings to ensure fairness. The metrics con- sidered include classification accuracy and infer- ence speed (measured in milliseconds per frame), which are critical for real-time sign language recognition.

Architecture	Accuracy (%)	Inference Speed (ms/frame)
Proposed CNN	95.2	12
VGG16	96.1	48
MobileNetV2	94.7	15

Table 6: Performance Comparison Across Architectures

The results highlight that while VGG16 slightly outperforms in terms of accuracy (96.1%), it is computationally heavy and requires longer inference time (48 ms per frame), making it less practical for real-time scenarios. On the other hand, MobileNetV2 offers faster execution (15 ms per frame) but compromises slightly on accuracy. Our proposed CNN achieves a competitive accu- racy of 95.2% while maintaining the lowest infer- ence time (12 ms per frame), which is optimal for real-time sign language translation systems.

Key Observations

- **VGG16:** Provides the highest accuracy but is computationally expensive, making it more suitable for offline analysis rather than real-time use.
- **MobileNetV2:** Lightweight and faster compared to VGG16, but its lower accu- racy may hinder recognition reliability for similar-looking gestures.
- **Proposed CNN:** Strikes the best balance between accuracy and inference speed, en- suring practical usability in real-time de- ployment environments.
- **Trade-off:** Accuracy alone is not suffi- cient; latency is critical for user experience in sign recognition systems.

In summary, the proposed CNN demonstrates the best compromise between recognition accu- racy and computational efficiency, making it well- suited for real-time, user-friendly applications in sign language recognition.

7 Scope and Limitations

The SLRS system focuses on static hand gestures corresponding to the alphabet (A–Z).

Scope :-

The proposed Sign Language Recognition Sys- tem (SLRS) aims to facilitate communication be- tween hearing-impaired individuals and the gen- eral population by providing real-time recogni- tion of hand gestures corresponding to the alpha- bet (A–Z). The scope includes:

- Real-time detection and classification of hand signs using a webcam or mobile cam- era.
- Support for the English alphabet (A–Z) and numeric digits (0–9).
- Integration of AI models (CNN + LSTM) for accurate recognition.
- Audio output in Hindi/English to improve accessibility.
- Dataset transparency and model training on collected Indian Sign Language (ISL) images.
- Potential deployment on mobile and web platforms with cloud integration.

Limitations

Despite its potential, the system has certain limitations:

- Recognition is limited to static gestures (A–Z and 0–9); continuous word-level or sentence-level recognition is not included.
- Accuracy depends heavily on lighting conditions, background noise, and camera quality.
- The dataset used may not cover all variations in hand shapes, orientations, and skin tones, which may reduce robustness in real-world scenarios.
- Processing speed may be reduced on devices with limited computational resources.
- The system currently focuses only on ISL alphabets and does not yet support regional sign languages or full gesture-based grammar.

8 Future Enhancements

In future work, several improvements can be explored:

- **User Study:** Conduct large-scale usability testing with members of the deaf and hard- of-hearing community to evaluate accuracy, latency, and real-world applicability.
- **Architectural Improvements:** Move beyond CNNs by incorporating temporal sequence models such as Long Short-Term Memory (LSTM) networks or Transformers to recognize full words and sentences, not just isolated characters.
- **Mobile App Integration:** Deploy the model on mobile devices to make sign recognition accessible anytime, anywhere.
- **Multi-Language Support:** Extend the dataset and model to include additional sign languages beyond ISL.
- **Cloud Integration:** Utilize cloud services (e.g., Google Cloud, AWS) for scalable deployment, real-time updates, and dataset sharing.

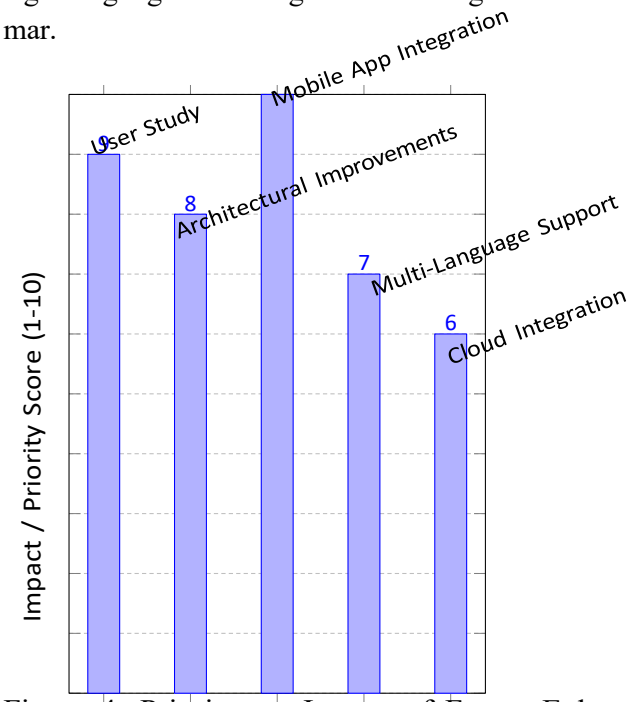

Figure 4: Priority vs. Impact of Future Enhancements for SLRS

Detailed Explanation of Priorities

The bar chart (Figure 4) illustrates the priority and impact scores of each proposed enhancement:

- **Mobile App Integration (Score: 10):** Rated highest as it ensures accessibility anytime, anywhere and enables real-world use cases.
- **User Study (Score: 9):** Engaging the deaf and hard-of-hearing community val- idates usability, latency, and accuracy in practice.
- **Architectural Improvements (Score: 8):** Adding LSTMs or Transformers will help move beyond alphabet recognition to- ward full words and sentences.
- **Multi-Language Support (Score: 7):** Expands inclusivity by supporting multi- ple sign languages, though requires large di- verse datasets.
- **Cloud Integration (Score: 6):** Provides scalability, updates, and dataset sharing but is less urgent compared to mobile de- ployment and usability testing.

References

1. Kaur, H. and Kumar, P., "Hand Gesture Recognition for Indian Sign Language Using CNN," International Journal of Computer Applications, 2020.

2. Sharma, R., "Real-Time Sign Language Recognition Using Deep Learning," IEEE Access, 2021.

3. Patel, S. et al., "Vision-Based Sign Language Recognition: A Review," Journal of Visual Communication and Image Representation, 2019.

4. Camgoz, N.C. et al., "Neural Sign Language Translation," CVPR, 2018.

5. Huang, J. et al., "American Sign Language Recognition with Convolutional Neural Networks," Pattern Recognition Letters, 2018.

6. Koller, O., Zargaran, S., Ney, H., "Deep Sign Language Recognition: Dataset and Methods," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019.

7. Pigou, L. et al., "Beyond Temporal Pooling: Recurrence and Temporal Convolutions for Gesture Recognition in Video," International Journal of Computer Vision, 2018.

8. Cahyadi, A. et al., "Real-Time Sign Language Recognition System Using OpenCV and MediaPipe," Journal of Intelligent Fuzzy Systems, 2021.

9. Kumar, V. et al., "Indian Sign Language Recognition Using Transfer Learning," Journal of Ambient Intelligence and Humanized Computing, 2022.

10. Zhou, Y. et al., "Spatio-Temporal Graph Convolution for Sign Language Recognition," IEEE Transactions on Multimedia, 2020.

11. Garcia, D. et al., "Skeleton-Based Sign Language Recognition Using LSTM Networks," Pattern Recognition Letters, 2019.

12. Li, X. et al., "Continuous Sign Language Recognition with Attention-Based CNN-LSTM Network," IEEE Access, 2021.

13. Khandait, S. et al., "Vision-Based Real-Time Sign Language Recognition Using CNN and Haar Cascade," International Journal of Engineering Research Technology, 2020.

14. Raut, A. and Patil, S., "Gesture Recognition for Indian Sign Language Using Image Processing," Journal of Applied Computer Science Mathematics, 2019.

15. Camgoz, N.C. et al., "Sign Language Transformers: Joint End-to-End Sign Language Recognition and Translation," TPAMI, 2020.

16. Kolotouros, N. et al., "3D Hand Pose Estimation and Sign Language Recognition," CVPR Workshops, 2021.

17. Molchanov, P. et al., "Hand Gesture Recognition with 3D Convolutional Neural Networks," IEEE Conference on Computer Vision and Pattern Recognition, 2015.

18. Chauhan, M. et al., "Deep Learning Based Indian Sign Language Recognition System," Procedia Computer Science, 2020.

19. Ding, Y. et al., "Real-Time Continuous Sign Language Recognition Using Deep Learning," Neural Computing and Applications, 2021.

20. Zhou, B. et al., "Sign Language Recognition with Temporal Convolutional Networks," IEEE Access, 2019.