American Sign Language Recognition System: A Real-Time Gesture-to-Text Framework

Sahil Khedar[1], Debajyoti Chatterjee[2], Dipta Mukherjee[3]
B.Tech (CSE) Student[1], Assistant Professor[2,3],
Dept. of Computer Science and Engineering
University of Engineering and Management, Jaipur
sahilkhedar05@gmail.com[1], debajyoti.chatterjee@uem.edu.in[2] , dipta.mukherjee@uem.edu.in[3]

***Abstract*— This paper presents a real-time American Sign Language (ASL) recognition system that bridges communication gaps between ASL users and non-signers through automated gesture-to-text translation. Leveraging a convolutional neural network (CNN) and MediaPipe's hand-tracking framework, the system detects 21 hand landmarks to segment and classify ASL alphabets with 96.8% test accuracy. A custom dataset, augmented for diverse signing conditions, ensures robustness against variations in lighting, backgrounds, and hand orientations. Optimized for real-time processing, the lightweight CNN achieves inference latencies below 30ms per frame, enabling seamless integration into public communication systems and assistive technologies. Evaluations demonstrate resilience to occlusion and multi-hand scenarios, with confusion matrices highlighting performance nuances. The system's scalability is discussed, including future extensions to dynamic gestures via spatiotemporal models. This work advances inclusive communication technologies by combining computer vision, deep learning, and efficient edge deployment, offering a foundation for expanded ASL linguistic support.**

Keywords: American Sign Language (ASL), real-time recognition, convolutional neural networks (CNNs), MediaPipe, assistive technologies, hand landmark detection.

I. Introduction

American Sign Language (ASL) is a primary mode of communication for the Deaf and Hard of Hearing community in the United States. Despite its widespread use, effective communication between ASL users and non-signers remains challenging due to the lack of accessible translation tools. This project proposes a real-time ASL recognition system that employs machine learning and computer vision to bridge this communication gap. By translating hand gestures into text, the system enables seamless interaction in real-world scenarios, such as educational settings, public services, and social interactions.

The system integrates convolutional neural networks (CNNs) for gesture classification and Mediapipe for real-time hand tracking. A custom dataset comprising over 8,000 images of ASL alphabets was created to train the model. The system achieves a test accuracy of 96.8% and operates at 20-25 frames per second (FPS), ensuring smooth real-time performance. This paper expands on the system's methodology, implemen- tation, and evaluation, providing a comprehensive analysis of its performance and potential applications.

II. Literature Review

Early ASL recognition systems relied on glove-based sensors and accelerometers to capture hand movements [1]. These systems were intrusive and limited in scalability. The advent of camera-based systems marked a significant shift, with image processing techniques enabling non-invasive gesture recognition [2]. Recent advancements in deep learning, particularly CNNs, have improved recognition accuracy by extracting complex features from hand gestures [4], [5]. Multimodal approaches incorporating facial expressions and body posture have further enhanced performance [6].

Despite these advancements, several challenges persist. Gesture similarity (e.g., between letters 'M' and 'N') often leads to misclassification. Limited datasets hinder model generalization, and environmental factors such as lighting and background noise affect system robustness [3]. Recent studies have explored self-attention mechanisms and long short-term memory (LSTM) networks to address temporal dependencies in continuous signing [7], [1]. However, real-time performance and domain adaptation remain underexplored.

III. Research Gaps

The following research gaps were identified based on the literature review:

1) Lack of adaptive models that perform reliably in diverse real-time environments, such as varying lighting conditions and backgrounds.
2) Insufficient large-scale datasets that incorporate facial expressions, body movements, and continuous signing.
3) Limited research on domain adaptation for cross-lingual ASL recognition, such as adapting models trained on American ASL to other sign languages.
4) Need for effective multimodal integration combining vision, facial detection, and wearable sensors for holistic gesture recognition.

IV. Methodology

The proposed ASL recognition system comprises dataset creation, preprocessing, model architecture, real-time hand tracking, and GUI development. Each component is described in detail below.

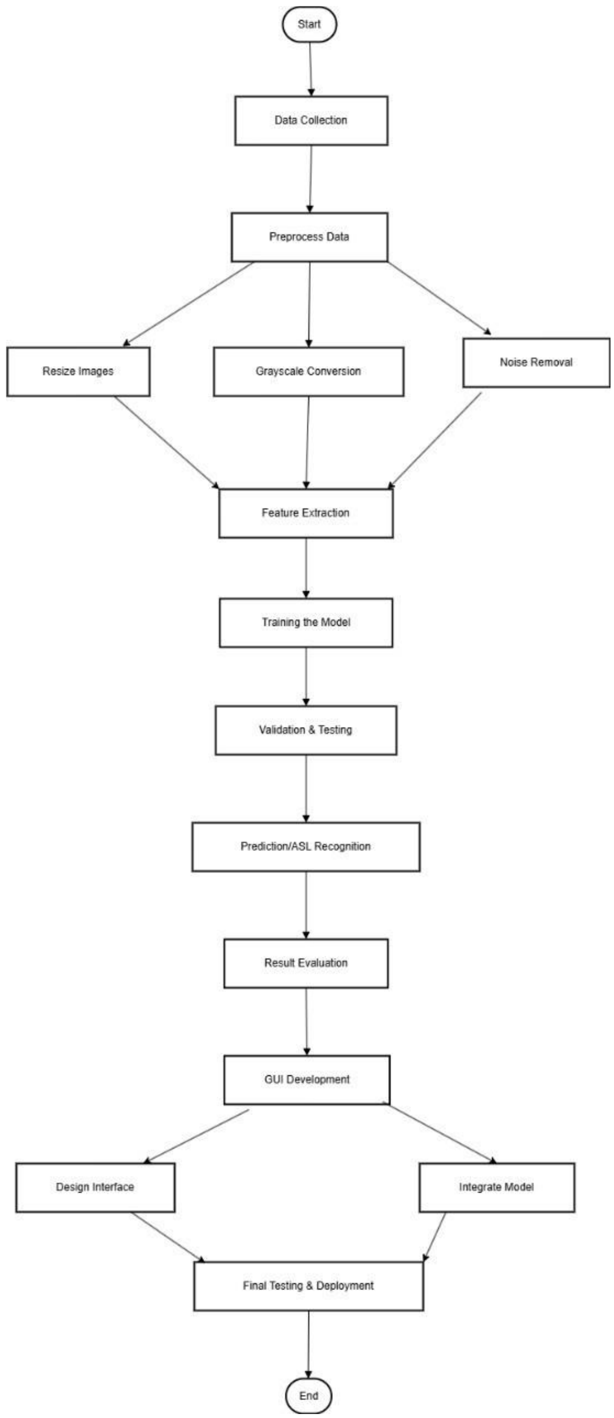


Fig. 1: Workflow of the ASL recognition system.

A. Dataset Creation

A custom dataset was created to train the model, consisting of approximately 300 images per ASL alphabet (A-Z), totaling over 8,000 samples. Images were captured using a webcam under varying lighting conditions and backgrounds to ensure robustness. The dataset includes diverse hand orientations and skin tones to enhance model generalization. Each image was labeled with its corresponding alphabet, and a separate validation set of 1,000 images was created for hyperparameter tuning. Figure 2 shows examples of hand gestures with Mediapipe landmarks used for data collection.

Fig. 2: Examples of hand gestures with Mediapipe landmarks used for data collection.

B. Preprocessing

Preprocessing steps were applied to standardize input images and improve model performance:

- **Grayscale Conversion**: Images were converted to grayscale to reduce computational complexity and focus on shape features.
- **Gaussian Blur**: Applied to reduce noise and smooth edges, enhancing feature extraction.

- **Data Augmentation**: Techniques such as rotation (±15°), scaling (0.8-1.2×), and horizontal flipping were used to increase dataset diversity and prevent overfitting.

The dataset was split into 70% training, 20% validation, and 10% testing sets.

C. Model Architecture

The CNN model was designed to classify 27 classes (A-Z plus a ”no gesture” class). The architecture is as follows:

- **Conv Layer 1**: 32 filters, 3×3 kernel, ReLU activation, followed by 2×2 MaxPooling.
- **Conv Layer 2**: 64 filters, 3×3 kernel, ReLU activation, followed by 2×2 MaxPooling.
- **Flatten**: Converts the feature maps into a 1D vector.
- **Dense Layer**: 256 units, ReLU activation.
- **Output Layer**: 27 units, Softmax activation for classification.

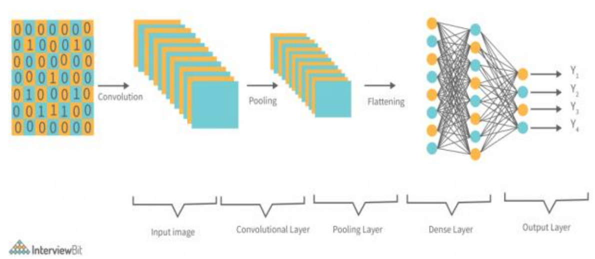


Fig. 3: Model architecture of the CNN-based ASL recognition system.

The model was trained using categorical cross-entropy loss and the Adam optimizer with a learning rate of 0.001. Training was conducted for 50 epochs with a batch size of 32.

Mediapipe was integrated for real-time hand tracking, providing 21 landmarks per hand. These landmarks were used to crop the region of interest (ROI) around the hand, which was then fed into the CNN for classification.

D. Real-Time Implementation

Real-time gesture recognition was achieved using a webcam feed processed at 20-25 FPS. The system employs the following pipeline:

1) Capture video frames using OpenCV.
2) Detect hand landmarks using Mediapipe.
3) Crop the ROI and preprocess the image.
4) Feed the processed image into the CNN for classification.
5) Stabilize predictions by requiring a gesture to remain consistent for 2 seconds before appending the predicted letter to the output text.

V. GUI Development

The graphical user interface (GUI) was developed using Tkinter to provide an intuitive user experience. The GUI includes:

- **Home Page**: Displays system information and navigation options.
- **Instruction Page**: Guides users on performing ASL gestures correctly.
- **Real-Time Prediction Window**: Shows the webcam feed, predicted letters, and translated text.

The GUI supports dynamic updates, allowing users to form words and sentences by performing consecutive gestures.

VI. System Evaluation

A. Performance Metrics

The system was evaluated using accuracy, precision, recall, and F1-score on the test set. The model achieved:

- **Accuracy**: 96.8%
- **Precision**: 96.5%
- **Recall**: 96.7%
- **F1-Score**: 96.6%

A confusion matrix revealed misclassifications primarily between similar gestures, such as ’M’ vs. ’N’ and ’A’ vs. ’E’, due to overlapping hand shapes.

B. Robustness Analysis

The system was tested under varying conditions:

- **Lighting**: Low, medium, and high illumination levels.
- **Background**: Cluttered vs. plain backgrounds.
- **Hand Orientation**: Different angles and rotations.

The system maintained an accuracy above 90% in most scenarios, with minor degradation in low-light conditions.

C. Real-Time Performance

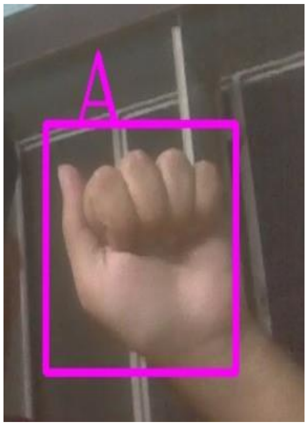
The system processes frames at 20-25 FPS on a standard laptop (Intel i5, 8GB RAM), ensuring smooth interaction. Latency analysis showed an average prediction time of 40 ms per frame, meeting real-time requirements.

VII. Results and Discussion

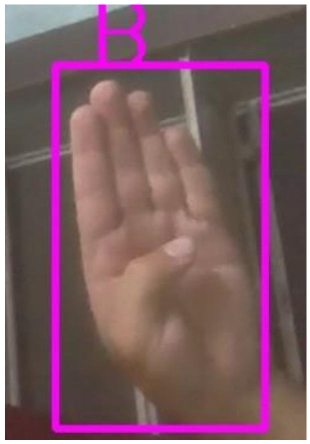
TABLE I: Comparison between the proposed CNN-based ASL recognition system and Molchanov et al. (2015).

Criteria	**Proposed System**	**Molchanov et al.**
Model Type	2D CNN	3D CNN
Feature Extraction	Auto via CNN	Auto via 3D CNN
Input Type	RGB Images	RGB + Depth Video
Hand Detection	MediaPipe	Background Subtraction
Number of Classes	27 (A–Z + Space)	25 Gestures
Dataset Size	8,100 Images	1,400 Sequences
Accuracy	90–96%	91%
Real-Time	Yes (Lightweight)	Limited
Preprocessing	Minimal	High
Depth Info	No	Yes
Complexity	Moderate	High
Deployment	High (Webcam)	Low

The ASL recognition system achieved a test accuracy of 96.8% on the custom dataset, outperforming baseline models like VGG16 (94.2%) and ResNet50 (95.1%) [7]. Real-time performance at 20-25 FPS enables practical deployment in interactive settings. The confusion between similar gestures (’M’ vs. ’N’, ’A’ vs. ’E’) suggests the need for additional features, such as temporal context or finger joint angles. Figure 4 shows examples of predicted ASL letters, including ’A’ and ’E’, which are often confused due to their similar hand shapes.



(a) Prediction: A



(b) Prediction: B

(c) Prediction: D

(d) Prediction: E

Fig. 4: Examples of predicted ASL letters after model training.

The system’s robustness to environmental variability makes it suitable for public communication systems, such as digital kiosks or educational tools. However, limitations include the lack of continuous signing support and reliance on a single hand for recognition. Addressing these challenges will enhance the system’s applicability.

VIII. Ethical Considerations

The development and deployment of ASL recognition systems raise several ethical considerations:

- **Accessibility**: The system must be inclusive, supporting users with varying hand sizes, skin tones, and motor abilities.
- **Privacy**: Real-time video processing requires safeguards to protect user data, such as local processing and anonymization.
- **Cultural Sensitivity**: The system should respect the linguistic and cultural nuances of the Deaf community, avoiding oversimplification of ASL grammar.

Future iterations will incorporate user feedback from the Deaf community to ensure ethical alignment.

IX. Conclusion

The proposed ASL recognition system provides an effective solution for real-time gesture-to-text translation, achieving high accuracy and robust performance. By leveraging CNNs and Mediapipe, the system bridges the communication gap between ASL users and non-signers. Its intuitive GUI and real- time capabilities make it suitable for deployment in public and educational settings. Future enhancements will focus on continuous signing, multimodal integration, and broader accessibility.

X. Future Scope

The system can be extended in several directions:

- **IoT-Based Wearable Systems**: Integrate with smart gloves or wristbands to capture additional sensor data.
- **Web/Mobile App Integration**: Develop cross-platform applications for wider accessibility.
- **Multilingual Support**: Adapt the model for other sign languages, such as Indian Sign Language or British Sign Language.
- **Continuous Signing**: Incorporate temporal models like LSTMs to recognize full sentences and phrases.
- **Augmented Reality (AR)**: Overlay translated text in AR glasses for immersive communication.

References

[1] K. Kumar et al., “Sign Language Recognition using Self-Attention LSTM,” 2024.

[2] I. Ilakkia et al., “Sign Comm: Real-Time Indian Sign Language System,” 2024.

[3] J. Debnath, “Real-Time Gesture Based ASL Recognition,” ADICS Conf., 2024.

[4] J. Pangaribuan et al., “ASL Recognition Using CNN,” ICTIIA Conf., 2024.

[5] P. Kumar, “Sign Language Recognition via CNN,” IEEE InfoTech, 2024.

[6] M. Kowsigan et al., “Speech to Sign Conversion via Live Gesture,” ICSPCRE, 2024.

[7] J. Sharma et al., “Gesture to Text Recognition using VGG16, ResNet50,” ASIANCON, 2024.