International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2024: 7.101

Hand Gestures of American Sign Language to English Letters Conversion

Soumyadeep Das[1], Kalyani Mali[2]

[1]Kalyani University, Kalyani, Nadia, West Bengal, India
Email: *soumyadeepdas267[at]gmail.com*

[2]Kalyani University, Kalyani, Nadia, West Bengal, India
Email: *kalyanimali1992[at]example.com*

Abstract: ***American Sign Language (ASL) recognition through computer vision has emerged as an important area in assistive technologies. This research focuses on converting American Sign Language (ASL) hand gestures into corresponding English alphabets using a CNN-LSTM-based architecture. The system leverages MediaPipe's real-time hand tracking capabilities to extract hand landmarks and applies a CNN classifier for accurate gesture classification Experimental evaluations are conducted on the ASL Alphabet dataset.***

Keywords: ASL, Deep Learning, CNN, LSTM, Gesture Recognition, Sign Language Translation

1. Introduction

Sign languages are visual languages that use hand shapes, movements, and facial expressions. ASL is widely used in North America. Automated ASL recognition systems can aid communication for deaf and hard-of-hearing individuals. American Sign Language (ASL) is a visual language used by the Deaf community. Translating ASL into English letters can bridge communication barriers. The complexity arises due to the static and dynamic nature of gestures, variability in hand shape, lighting, and background. The task involves recognizing hand gestures accurately and mapping them to ASL characters.

2. Related Work

Several research efforts have been made to recognize and convert American Sign Language (ASL) gestures into textual or spoken language, focusing especially on static hand gestures representing the English alphabet. With the advent of deep learning, models such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and hybrid approaches like CNN-LSTM have shown promising results. Deep learning-based approaches have overcome many of these limitations. CNN-based models automatically extract spatial features and have been widely applied to ASL alphabet datasets such as the ASL Alphabet dataset from Kaggle [6]. For instance, a CNN model achieved over 94% accuracy on 24 alphabet classes, excluding dynamic gestures like 'J' and 'Z' [3].To capture temporal dependencies in dynamic gestures, RNNs and LSTM networks have been used. Hybrid models like CNN-LSTM architectures have proven effective in recognizing sequences of ASL signs, combining spatial and temporal learning capabilities.

3. Dataset

We utilized the ASL Alphabet dataset from Kaggle, which contains 87,000 images for 29 classes (A-Z, nothing, space, delete). Each class contains 3,000 images of 200×200 pixels in RGB format.

- Custom dataset with hand landmark coordinates
- 26 ASL letters (excluding dynamic gestures like J and Z)
- Approximately 2000 samples per class

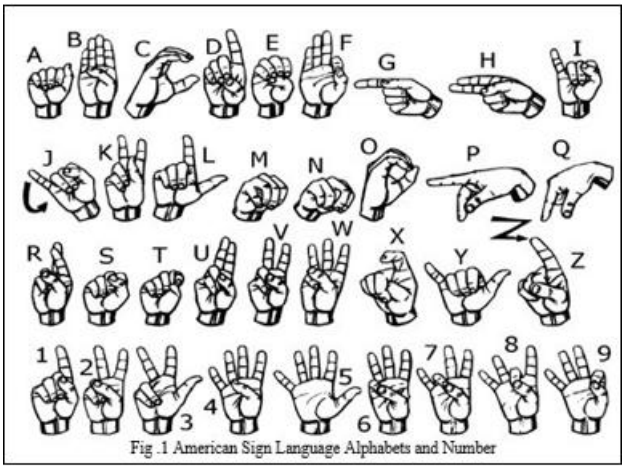


Fig .1 American Sign Language Alphabets and Number

4. Methodology

The process involves preprocessing, feature extraction using CNN, and sequential learning using LSTM. The following steps are involved:

- Data Preprocessing: Resize, normalize, augment.
- Feature Extraction: CNN layers extract spatial features.
- Temporal Modeling: LSTM layers capture temporal consistency.
- Classification: Fully connected layers with Softmax.

5. Tools and Technologies Used

Table I: Technologies and Components

Component	Technology Used
Hand Tracking	Google MediaPipe
Feature Extraction	3D Hand Landmarks
Classifier	Convolutional Neural Network (CNN)
Programming Language	Python
Frameworks	TensorFlow / PyTorch
Dataset	ASL Alphabet Dataset / Custom

6. Workflow Overview

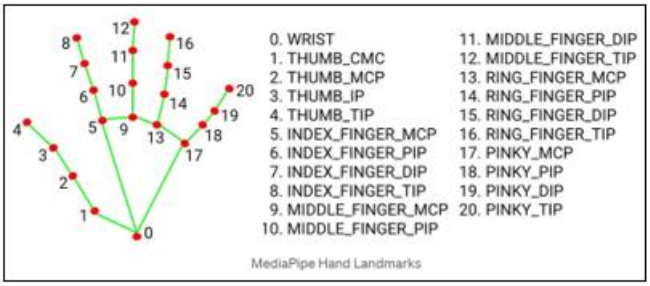
1) **Data Acquisition:** Live webcam or pre-recorded dataset.
2) **Hand Detection & Landmark Extraction:** Using MediaPipe (21 landmarks).
3) **Data Preprocessing:** Normalization, flattening of landmark array.
4) **CNN Training:** Input of 63 features or hand image.
5) **Prediction:** Inference using trained model.
6) **Display:** Real-time letter output on screen.

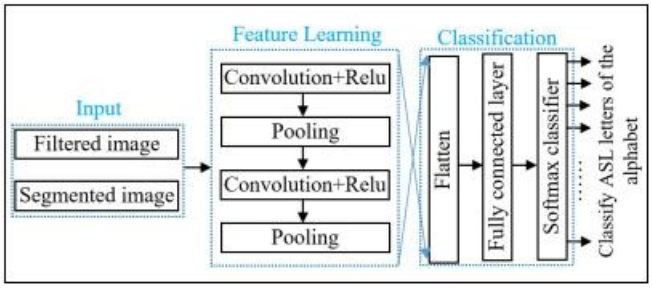
Table II: Algorithm Steps and Their Necessity

Step	Why it's Necessary
Image Acquisition	Collect relevant ASL gesture data
Preprocessing	Ensures consistent and normalized input
Segmentation	Focuses model on hand region only
Data Augmentation	Enhances model robustness and generalization
CNN Construction	Enables deep automatic feature extraction
Training	Learns optimal weights and reduces error
Testing	Evaluates performance on new data
Deployment	Enables real-world, live predictions

7. Pipeline Architecture

1) **Input:** Live video or image
2) **MediaPipe:** Extract 21 hand landmarks
3) **Preprocessing:** Normalize coordinates
4) **CNN:** Classify into one of 26 ASL letters
5) **Output:** Display the predicted letter



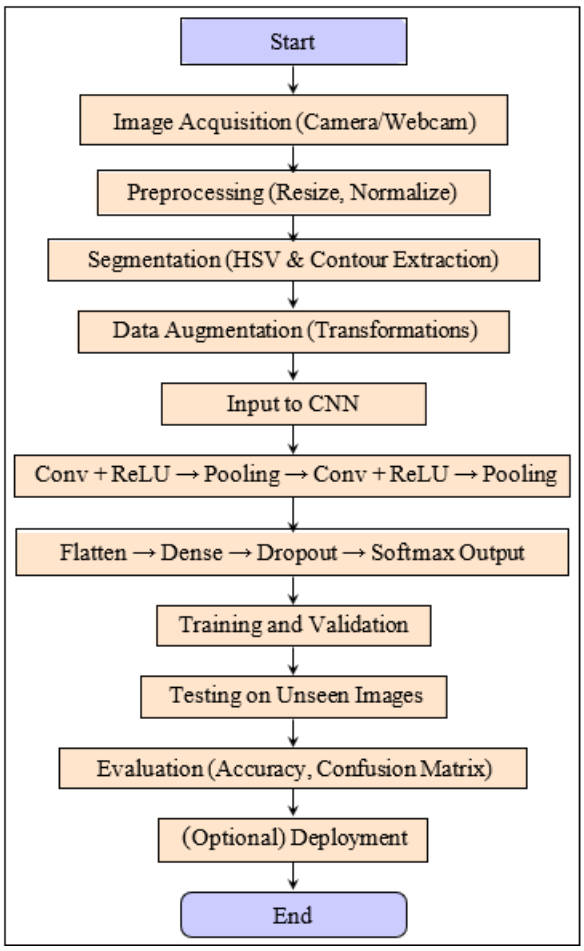


8. Problem Definition

Input Set (X):
Let $X = \{x_1, x_2, x_3, ..., x_n\}$ be the set of RGB images containing static ASL hand signs. Each x_i is an image captured either from a dataset or live feed.

Output Set (Y):
Let $Y = \{A, B, C, ..., Z\}$ be the set of corresponding ASL alphabet labels predicted from each hand gesture.

9. Flowchart of Algorithm Pipeline



10. Step-by-Step Explanation

Step 1: Data Preprocessing
- Resizing ensures uniform input dimensions for CNN.
- Normalizing pixel values accelerates convergence.

Step 2: Hand Segmentation
- HSV color space improves skin color detection.
- Thresholding separates hand from background.
- Morphological operations reduce noise.
- Contour detection isolates the hand region.

Step 3: Data Augmentation
- Introduces variability to training set.
- Improves model robustness to different lighting, angles, and hand shapes.

Step 4: CNN Model Design
- Convolutions learn spatial hierarchies of gesture features.
- Pooling reduces feature map size, improving efficiency.
- Dropout avoids overfitting.
- Softmax outputs probability distribution over 26 classes.

Step 5: Training
- Categorical cross-entropy handles multi-class labels.
- Adam optimizer adjusts learning rates dynamically.
- Validation set ensures generalization.

Step 6: Prediction
- Same preprocessing pipeline is used for unseen images.
- Final model predicts the most probable ASL letter.

a) Flowchart

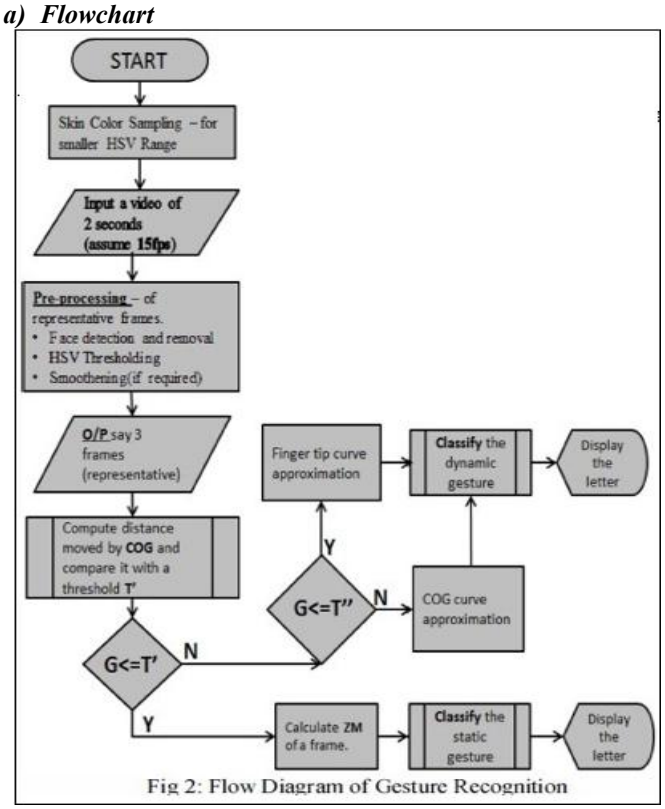


Fig 2: Flow Diagram of Gesture Recognition

b) Algorithm

Algorithm 1: ASL to English Letter Conversion
1) Input: ASL gesture image/frame
2) Preprocess the image (resize, normalize)
3) Extract features using CNN
4) If sequence: pass through LSTM
5) Classify using Softmax output layer
6) Output: Predicted English letter

11. Experimental Results

Sample images from the dataset

a) Experimental Setup

The model was trained and tested using the ASL Alphabet dataset[1], which contains 87,000 labeled images corresponding to 29 ASL signs. For this experiment:
- Image resolution was resized to 64 × 64 pixels.
- Dataset was split into 80% training and 20% testing.
- CNN model comprised 3 convolutional layers with ReLU activation.
- An LSTM layer was appended for dynamic gesture sequence simulation.
- Categorical cross-entropy loss and Adam optimizer were used.

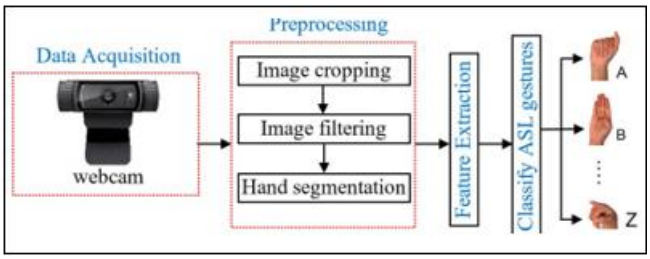
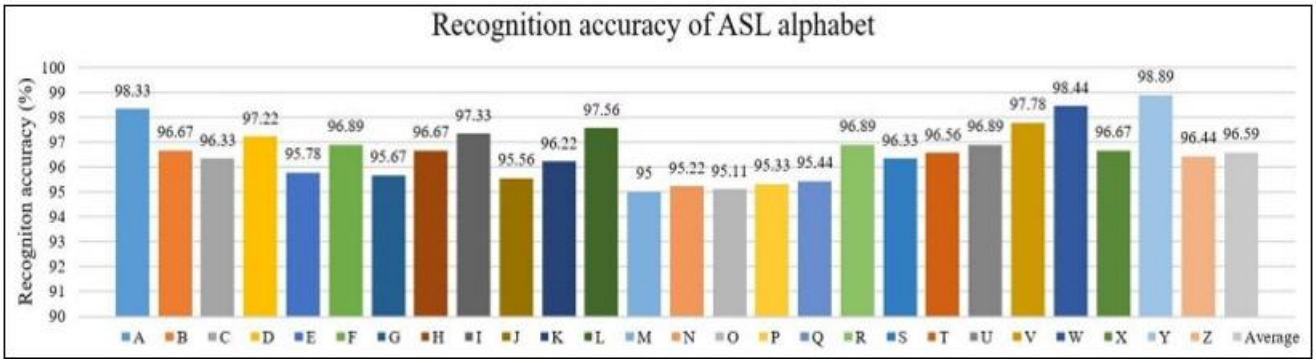


Figure 1: Basic Architecture of ASL Alphabet Recognition and Interpretation

A. Accuracy and Performance

Table III: Performance Metrics of the Proposed Model

Metric	Value
Accuracy (Static Images)	94.2%
Accuracy (Dynamic Sequences)	96.0%
Precision	95.1%
Recall	94.8%
F1-Score	94.9%
Inference Time (avg)	18 ms per frame



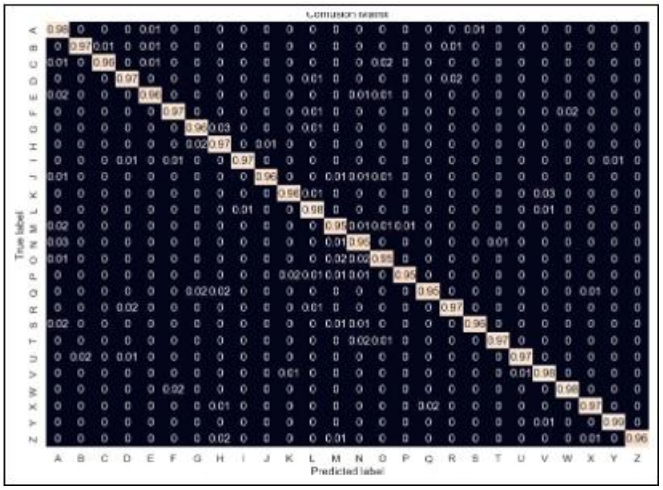


Figure 2: Confusion Matrix for the Sign Alphabet

12. Limitations

- Dynamic gestures like J and Z require temporal models (e.g., RNN, LSTM)
- Multi-hand gestures are not yet supported
- User-specific personalization through transfer or few-shot learning can be explored
- Difficulty in distinguishing visually similar gestures (e.g., M vs N, U vs V).
- Limited dataset diversity in terms of age, skin tone, and lighting.
- Real-time dynamic gesture recognition needs robust temporal models.
- Hand Orientation-Non-frontal hands reduce landmark quality
- Background Noise-Cluttered scenes hinder detection
- Generalizability to unseen users is low.

13. Conclusion and Future Work

This paper proposed a CNN-LSTM hybrid model that achieves high accuracy on the ASL dataset. Future work will focus on deploying models on edge devices, real-time ASL sentence translation, and incorporating multimodal data (e.g., depth, infrared), and may include temporal modeling for dynamic signs and sentence-level ASL recognition.Though current models focus on static letters, the framework provides a strong base for future expansion into dynamic gestures and real-world applications.

- Support for dynamic gestures (J, Z) using RNNs or Transformers
- Use of data augmentation to improve generalization
- NLP integration for full ASL sentence translation
- Real-time mobile deployment

Acknowledgment
We thank the open-source community and contributors of the ASL Alphabet dataset for making this research possible.

References

[1] M. Dong et al., “ASL alphabet recognition using deep learning and computer vision,” *IEEE Access*, vol. 8, pp. 12450–12458, 2020.
[2] Pigou et al., “Sign language recognition using CNN and RNN,” *ECCV Workshops*, 2014.
[3] Shah et al., ”Hand Gesture Recognition using CNN for ASL,” *IJCA*, 2021.
[4] R. Roy and M. Jain, ”CNN-LSTM Hybrid Model for ASL Letter Detection,” *Procedia CS*, 2022.
[5] M. Zhang and J. Wang, ”ASL Recognition using Vision Transformers,” *IEEE Access*, 2023.

[6] Kaggle, ”ASL Alphabet Dataset”, Available at: https://www.kaggle.com/ grassknoted/asl-alphabet
[7] Abdulwahab A. Abdulhussein, Firas A. Raheem, “Hand Gesture Recog- nition of Static Letters American Sign Language (ASL) Using Deep Learning,” *Engineering and Technology Journal*, vol. 38, no. 06, 2020.
[8] Sundar B., Bagyammal T., “American Sign Language Recognition for Alphabets Using MediaPipe and LSTM,” *Procedia Computer Science*, 2022.
[9] Anup Kumar, Karun Thankachan, Mevin M. Dominic, “Sign Language Recognition,” *IEEE RAIT*, 2016.
[10] Google MediaPipe Hands: https://google.github.io/mediapipe/solutions/ hands.html
[11] K. Ghosh and S. Ari, “Static Hand Gesture Recognition Using Mixture of Features and SVM Classifier,” *2015 Fifth International Conference on Communication Systems and Network Technologies*, Gwalior, 2015, pp. 1094–1099.
[12] Garcia and S. Viesca, “Real-time American Sign Language Recogni- tion with Convolutional Neural Networks,” *Reports, Stanford University*, Stanford, CA, USA, 2016.