Real-Time Sign Language Interpretation

CV End-Semester Evaluation Report

CV 419/619 – Computer Vision

Tarun Balaji A (210002001)
Amirthan Arul (210005005)
Nitheeshvar M (210005026)

1. Overview

Gestures are messages conveyed non verbally and are interpreted through vision. The nonverbal communication used by deaf and mute individuals is known as sign language.

Sign language is one of the oldest and most natural form of language for communication, but since most people do not know sign language and interpreters are very difficult to come by, there is a need for real time sign language decoding using neural networks.

Sign language is a visual mode of communication that comprises three key components (Figure 1).

Fingerspelling	**Word level sign vocabulary**	**Non-manual features**
Used to spell words letter by letter .	Used for the majority of communication.	Facial expressions and tongue, mouth and body position.

Figure 1: Three components of sign language

The American Sign Language (ASL) contains predefined signs for each letter and numbers. An excerpt of the signs is given in Figure 2.

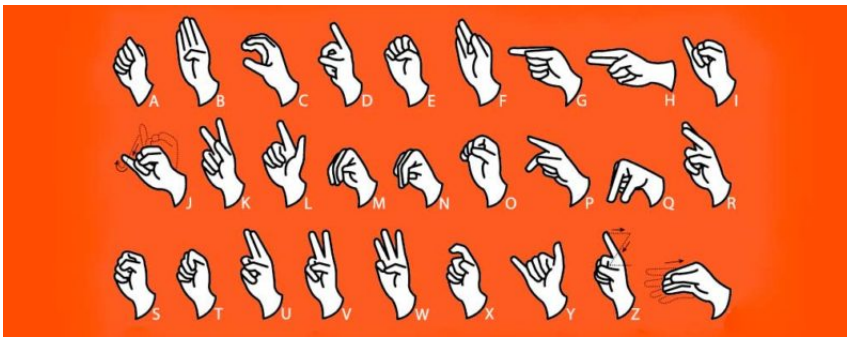


Figure 2: Fingerspelling in ASL

2. Objectives

The objectives of the project are:

- To train various models that can accurately decode American Sign Language using the specified datasets and compare the accuracy of said models.
- To identify sign language using a webcam and interpret it in real-time.

3. Scope

The system can serve as an assistive technology, enabling real-time communication between deaf and hearing individuals through smart devices, mobile apps, and even smart glasses. In education, such a model can help non-signers learn sign language interactively while also enhancing virtual learning experiences for hearing-impaired students.

The healthcare sector can also benefit, as doctors, nurses, and emergency responders can use it to communicate with deaf patients, especially in telemedicine. Additionally, in human-computer interaction (HCI), sign language recognition can allow gesture-based control of smart home devices, robots, and virtual assistants, making technology more accessible. AI-powered interpretation and translation can further bridge communication gaps in workplaces and public spaces by converting sign language into text or speech in real-time. The model can also be integrated into security and surveillance systems to recognize distress signals or emergency signs.

4. Traditional Methods

4.1. Convolutional Neural Network (CNN)

Approach: CNNs are deep learning models designed to extract spatial hierarchies of features from images. In the context of ASL sign recognition, CNNs typically take raw images or hand landmark heatmaps as input and learn filters to capture edge, shape, and gesture patterns.

Strengths:

- Automatically learns rich spatial and hierarchical features.
- Performs well with large, labeled datasets.
- Effective when using raw image input (i.e., no need for hand-crafted features).

Weaknesses:

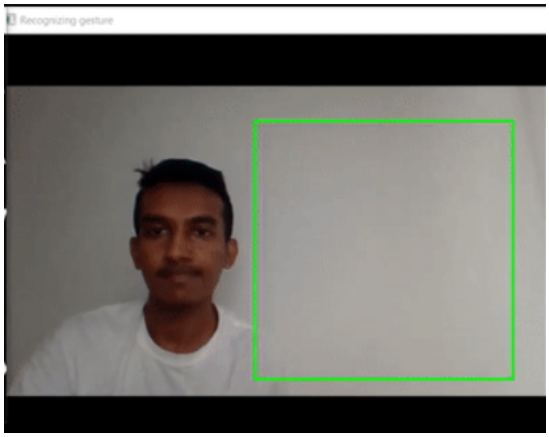
- *Position and scale sensitivity*: Without normalization, CNNs can misinterpret the same gesture at different positions or scales.
- *Data hungry*: Requires a large and diverse dataset to generalize effectively.
- *Overfitting risk*: Can overfit on training data if not properly regularized or augmented.
- *Computationally intensive*: Training and inference require substantial computational resources (e.g., GPUs).

4.2. Random Forest Classifier

Approach: A Random Forest is an ensemble of decision trees that operates on structured, tabular data. In our project, we use extracted features—such as normalized landmark coordinates—as input to classify ASL gestures.

Strengths:

- Fast training and inference on smaller datasets.
- Interpretable and allows analysis of feature importance.
- Robust to overfitting when appropriately tuned.
- Performs well when given clean, engineered features (e.g., relative landmark coordinates).



(a) Co-ordinates are taken globally [1]

(b) New approach: Co-ordinates taken relative to the hand size

Figure 3: Different methodologies for detecting hand landmarks

Weaknesses:

- *Lacks spatial awareness*: Unlike CNNs, Random Forests do not inherently capture spatial structure unless explicitly encoded.
- *Scalability issues*: May struggle with high-dimensional data or large feature spaces.
- *Limited generalization*: May not handle unseen variations in hand shapes or occlusions well without invariant features.

5. Our Methodology

5.1. Position and Scale Invariance through Relative Hand Landmark Coordinates

Traditional image classification methods using raw pixels or absolute landmark coordinates are sensitive to the hand's position and size within the frame, often leading to poor generalization. To address this, we normalize the hand landmark coordinates relative to a bounding box enclosing the hand. This transformation achieves both translation and scale invariance. The difference between the two methods of marking hand landmark co-ordinates are given in Figure 3.

Mathematical Justification Let the set of detected 2D hand landmarks be:

$$L = \{(x_i, y_i)\}_{i=1}^{N}$$

where (x_i, y_i) are the coordinates of the i^{th} landmark.

We define the bounding box using:

$$x_{\min} = \min_i x_i, \quad x_{\max} = \max_i x_i$$

$$y_{\min} = \min_i y_i, \quad y_{\max} = \max_i y_i$$

Each landmark is then normalized to obtain its relative coordinate:

$$x_i' = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}, \quad y_i' = \frac{y_i - y_{\min}}{y_{\max} - y_{\min}}$$

The resulting coordinates $(x_i', y_i') \in [0, 1]^2$ represent the hand gesture independent of its absolute position and size in the image.

5.2. Generated Dataset

To create our dataset, we utilized a pretrained hand tracking model provided by MediaPipe, which detects 21 key hand landmarks per hand. For this project, we restricted our analysis to a single hand, resulting in 21 landmarks. Each landmark provides *(x, y)* coordinates, which are normalized relative to the bounding box enclosing the hand. Consequently, our dataset contains 42 feature columns representing the relative spatial configuration of the hand.

In addition, we constructed a confusion matrix to analyze the classifier's performance. The matrix highlights patterns of similarity and confusion between specific ASL letters—for instance, signs such as 'M' and 'N' show high inter-class similarity, making them harder to distinguish. Conversely, gestures that are visually distinct (e.g., 'L' or 'Y') are more reliably classified. This insight helps us identify which signs benefit from our feature representation and which may require additional disambiguating features or preprocessing.

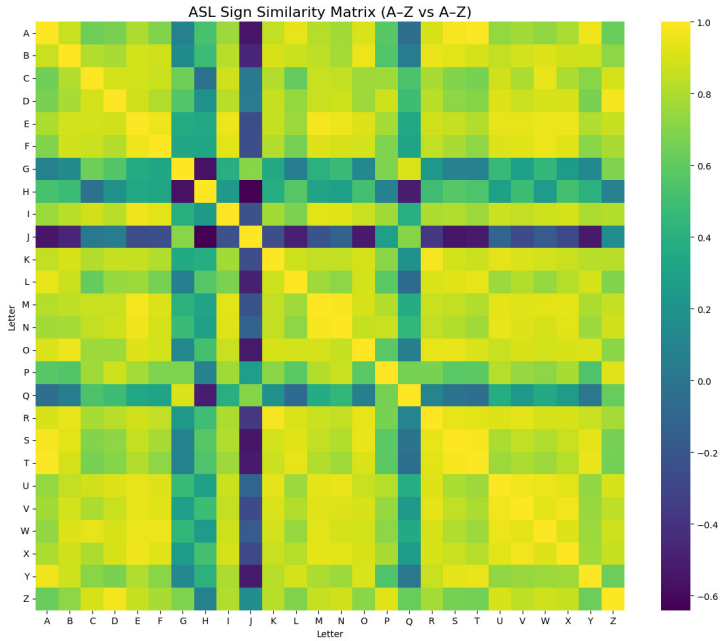


Figure 4: **Confusion Matrix:** Visualization of classifier performance across ASL letters. Darker squares indicate higher classification accuracy.

Advantages Over Direct Classification

- **Position Invariance**: The gesture's location in the frame does not affect the normalized coordinates.

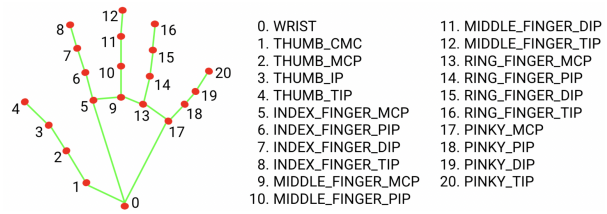


Figure 5: Hand landmarks detected by Mediapipe

- **Scale Invariance**: Gestures of different hand sizes or image scales yield similar relative patterns.
- **Feature Compactness**: Focuses only on the geometry of the hand, removing background and positional noise.
- **Improved Generalization**: Reduces variability in the input space, allowing the classifier to generalize across diverse samples with fewer examples.

This normalization approach follows the principles of affine invariance and robust feature engineering, enabling our classifier to learn gesture semantics without being distracted by irrelevant visual variations.

5.3. Pre-Trained Model: Mediapipe

For models other than CNN based methods, we use Mediapipe [2] to detect hand landmarks. The model detects the keypoint localization of 21 hand-knuckle coordinates (Figure 5) within the detected hand regions. The model was trained on approximately 30K real-world images, as well as several rendered synthetic hand models imposed over various backgrounds.
outputs precise x and y coordinates for each of the 21 points, allowing for detailed tracking of hand pose and gesture. After detection, MediaPipe uses tracking in subsequent frames to maintain smooth and real-time landmark detection without having to run the full palm detector every time.

6. Datasets used

- Fingerspelling sign language [3]
- Real Time American Sign Language Recognition Using Yolov6 Model [4]
- Realtime Sign-Language Detection Using LSTM Model [5]

6.1. Model Summary

We trained an Artificial Neural Network (ANN) classifier that takes a 42-dimensional input vector representing the normalized hand landmark coordinates (21 landmarks, each with x and y values).ANNs are a class of machine learning models inspired by the biological neural networks of the human brain. They are composed of interconnected layers of nodes (neurons), where each connection carries a weight adjusted during training. For our use case, a fully connected (dense) network was sufficient due to the structured and

compact nature of the input features. The model architecture is relatively lightweight and incorporates dropout layers to reduce overfitting and improve generalization.

The motivation behind using a shallow network is based on the premise that our features are already highly informative due to normalization relative to the hand's bounding box. This reduces the complexity required from the classifier itself, as the pretrained landmark extractor (via MediaPipe) provides robust and consistent features across varying contexts.

The model was implemented using a sequential architecture, as summarized below:

Layer (type)	**Output Shape**	**Param #**
Dense (256 units, ReLU)	(None, 256)	11,008
Dropout (rate=0.3)	(None, 256)	0
Dense (128 units, ReLU)	(None, 128)	32,896
Dropout (rate=0.3)	(None, 128)	0
Dense (64 units, ReLU)	(None, 64)	8,256
Dense (Output: 26 classes, Softmax)	(None, 26)	1,690
Total Parameters		**53,850**
Trainable Parameters		53,850
Non-trainable Parameters		0

Table 1: Summary of the ANN architecture used for ASL gesture classification.

7. Outcomes and Performance Metrics

We have trained 3 models, two to showcase the traditional methods using CNN and Random-Forest based classifiers, and the third model being the ANN based approach that uses both relative hand co-ordinates and Mediapipe to detect hand landmarks. We label these Model-1, Model-2 and Model-3 respectively

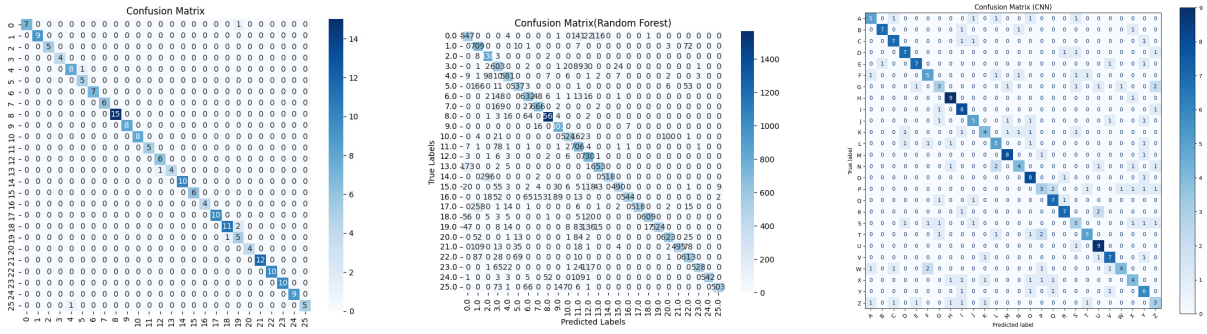
Metric	**Model-1 (CNN)**	**Model-2 (Random Forest)**	**Model-3 (ANN)**
Accuracy	55.4%	76.2%	**98.5%**
Precision	60.4%	70.5%	**96.1%**
Recall	62.3%	69.9%	**94.0%**
F1 Score	85.9%	70.2%	**95.5%**

Table 2: Comparison of classification performance for the three trained models.

The confusion matrix for the three models are given in Figure 6. The metrics are further summarized in Table 2.

8. Conclusion

In this project, a real-time sign language detection system was successfully developed using MediaPipe for efficient hand landmark extraction and an Artificial Neural Network



(a) Model-1: CNN Based (b) Model-2: Random Forest (c) Model-3: ANN Based

Figure 6: Confusion Metrics of different methods

(ANN) for gesture classification. MediaPipe's lightweight and accurate hand tracking framework enabled precise detection of 21 hand landmarks in real time, providing robust input features for the ANN model. The neural network effectively learned to classify different sign gestures based on these landmarks, allowing the system to recognize signs quickly and with good accuracy. This combination of MediaPipe and ANN resulted in a system that is both computationally efficient and capable of running on real-time platforms such as webcams or mobile devices. Overall, the project demonstrates a practical and scalable solution for sign language recognition, paving the way for improved communication tools for the hearing and speech impaired community.

References

[1] Harshbg, "Sign language interpreter using deep learning," 2020, version 1.0. [Online]. Available: https://github.com/harshbg/Sign-Language-Interpreter-using-Deep-Learning

[2] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, and M. Grundmann, "Mediapipe: A framework for building perception pipelines," 2019. [Online]. Available: https://arxiv.org/abs/1906.08172

[3] A. Raj, "Fingerspelling sign language dataset," 2023, accessed: 2025-03-23. [Online]. Available: https://www.kaggle.com/datasets/ayuraj/asl-dataset

[4] A. Gomez and E. Arzuaga, "Real time american sign language recognition using yolov6 model," pp. 343–353, 08 2024.

[5] R. Kumar, A. Bajpai, and A. Sinha, "Mediapipe and cnns for real-time asl gesture recognition," 2023. [Online]. Available: https://arxiv.org/abs/2305.05296