Assignment

Comparative Analysis of Traditional and Deep Learning-Based Feature Extraction Methods for Hand-Based Gestures

Deep Learning(22AIE304)

Voota Koushik CH.SC.U4AIE23062

AMRITA VISHWA VIDYAPEETHAM | School of Engineering
AMRITAPURI | BENGALURU | CHENNAI | COIMBATORE

Dataset Used: American Sign Language (ASL) Hand Gesture Dataset

- Number of Classes: 36 (26 alphabets A–Z + 10 digits 0–9)
- Total Samples: ~2,500+ labeled images

Why ASL Recognition is Important

Bridge for communication: Helps people with hearing or speech impairments communicate more easily with the wider community.

Accessibility: Enables inclusive technologies (like apps, translators, or smart devices) that make daily life smoother.

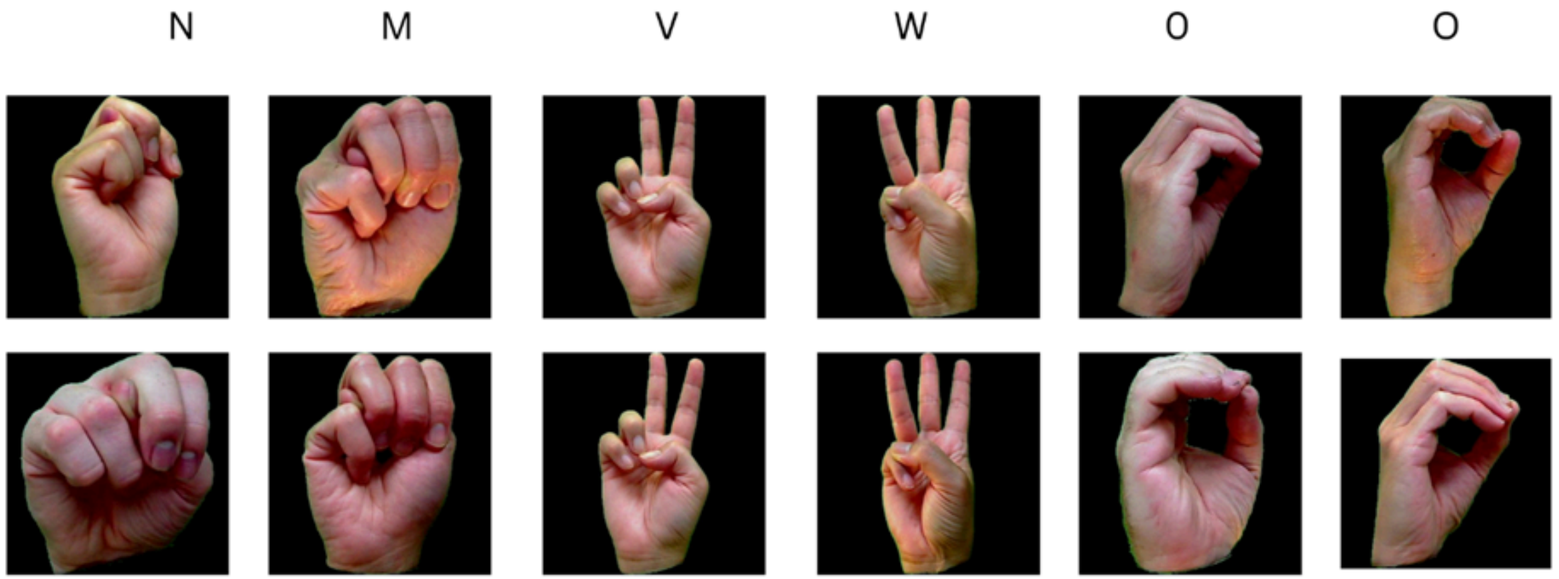
Real-time translation: Can convert ASL gestures into text or speech instantly, breaking language barriers.

Social impact: Promotes equal opportunities in education, workplaces, and public services.

Challenges in ASL Recognition

Similar-looking signs: **Many letters (e.g., M, N, S) have only subtle differences,** making classification difficult.

Need for strong features: Simple pixel-level features are not enough robust handcrafted or learned features are required.



How the Model Tackled Subtle & Similar Gestures

HOG + Logistic Regression

Why it worked well:

HOG captures edge orientations and gradient distributions across cells.

For M vs N, the gradient histograms differentiate based on the number of visible finger folds (two for M vs one for N).

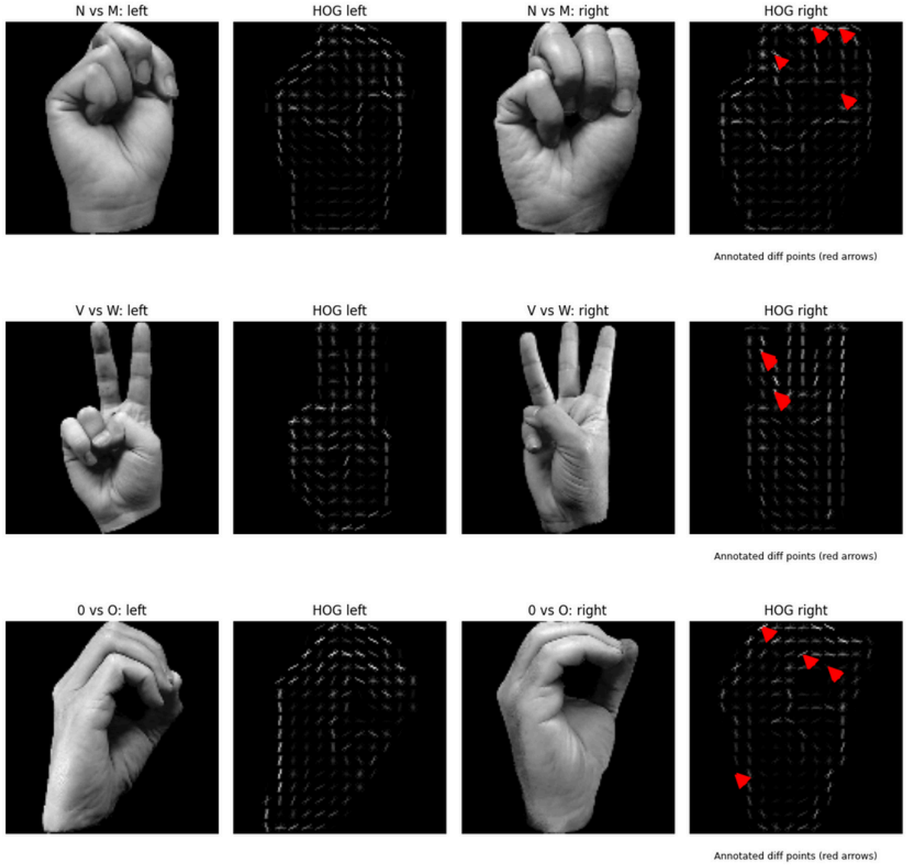
For V vs W, HOG picks up the additional finger edge in W, which changes the gradient orientation density.

For 0 vs O, the contour of the closed loop differs slightly (circular vs oval), which HOG detects in edge curvature.

How it classified: By projecting these high-dimensional descriptors into a nearly linearly separable space, Logistic Regression drew clean decision boundaries.

Limitations:

Misclassifications still occurred when finger overlaps reduced edge clarity (e.g., tilted hands).



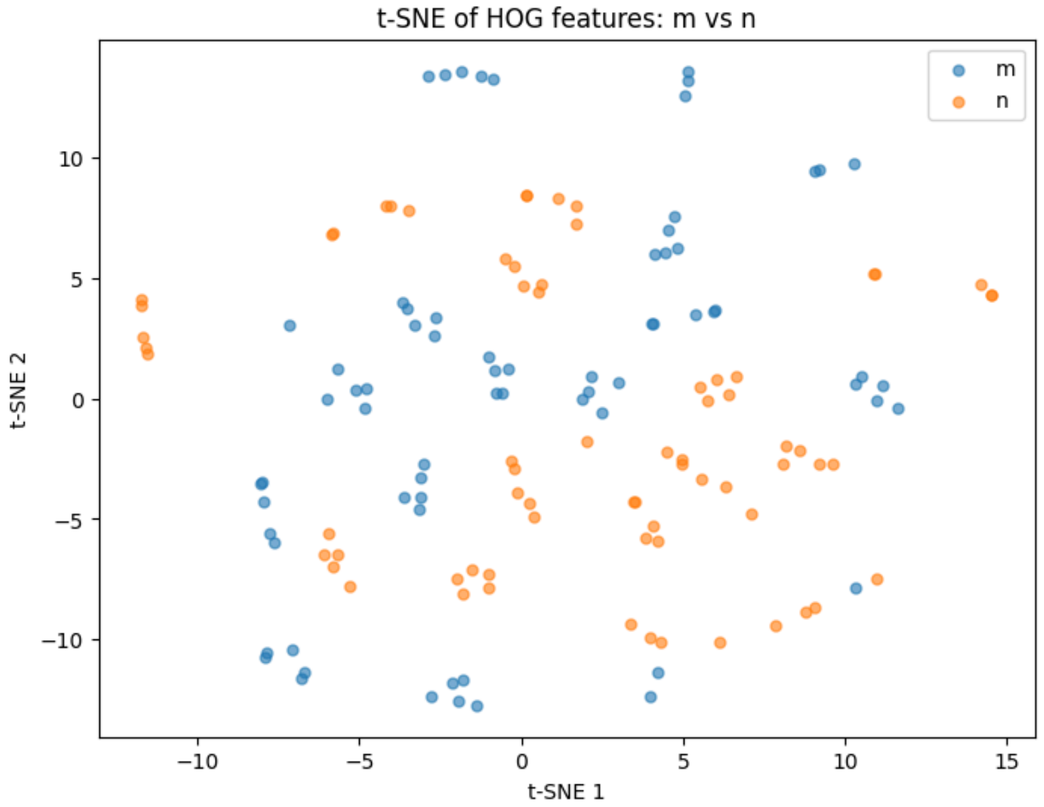
N vs M:
Both are closed-fist shapes.
HOG highlights extra vertical gradient components near the finger folds in “M”.
This difference allows Logistic Regression to separate them.

V vs W:
Gestures differ by one extra extended finger.
HOG shows additional diagonal gradients in “W” where the third finger adds new edge structure.
These extra orientations become the discriminative signal.

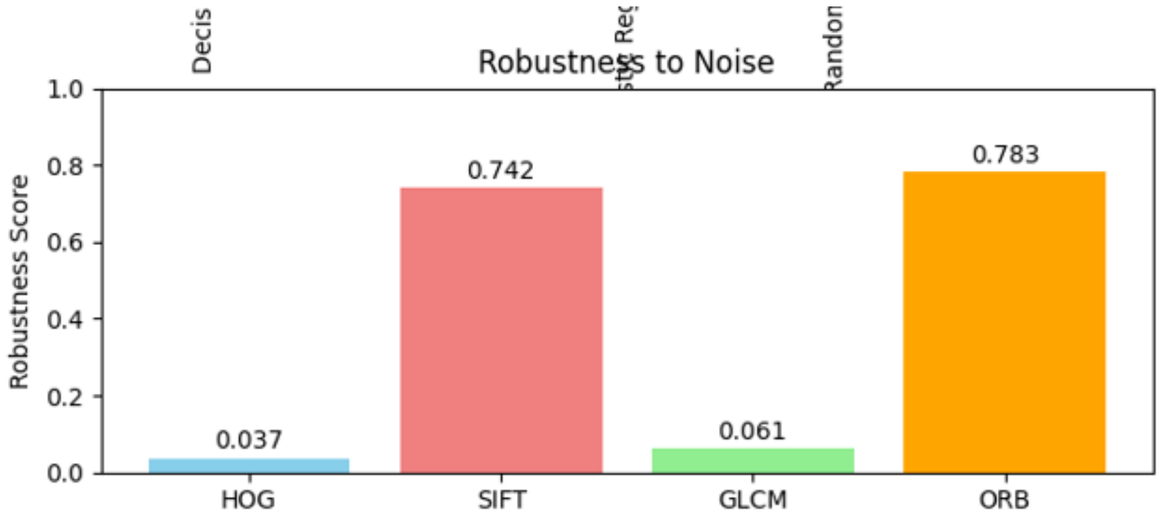
0 vs O:
Both are circular shapes.
Subtle curvature differences in the contour edges generate distinct gradient patterns, marked by red arrows.
While hard for humans, HOG captures these orientation shifts effectively.

t-SNE Visualization of HOG Features (M vs N)

Shows how HOG descriptors separate gestures, but with overlapping clusters causing confusion.



Each dot is one ASL image represented in HOG feature space, reduced to 2D using t-SNE.
Blue = “M”, Orange = “N”.
Ideally, the two classes should form distinct clusters.
Here we see partial overlap → explains why the model sometimes misclassifies “M” as “N”.
Cause: both gestures have similar closed-finger shapes; gradient differences are subtle.
Takeaway: HOG works well overall, but for very similar gestures, deeper features (CNN, MobileNet) achieve better separation.



HOG & GLCM collapse under noise because they rely directly on pixel intensity gradients or co-occurrences. SIFT & ORB remain stable because their keypoint-based, local descriptors are naturally less affected by pixel-level noise.

HOG (0.037)
HOG relies heavily on edge gradients. Gaussian noise adds random pixel intensity variations that disrupt gradients, so edge orientation histograms get corrupted.
Even slight noise makes HOG lose its structure → huge accuracy drop.
GLCM (0.061)
GLCM measures pixel co-occurrence statistics. Noise completely alters pixel intensity distributions, so the calculated co-occurrence matrix becomes meaningless.
This explains the near-zero robustness.
SIFT (0.742)
SIFT detects keypoints at stable extrema in scale-space (Difference-of-Gaussians).
Many of these keypoints are relatively resistant to mild Gaussian noise, so SIFT retains useful descriptors.
That's why robustness is much higher than HOG/GLCM.
ORB (0.783)
ORB uses FAST keypoint detection + binary BRIEF descriptors.
Binary descriptors are computed from pixel intensity comparisons in patches → less sensitive to global intensity fluctuations caused by noise.
This makes ORB surprisingly noise-resistant despite being lightweight.

To address the limitations of traditional feature extraction methods, we developed a custom Convolutional Neural Network (CNN) architecture specifically designed for ASL gesture recognition. The network consists of **three convolutional layers followed by a flatten operation, a dense layer with 128 neurons, dropout regularization, and a final dense layer with 36 outputs corresponding to the gesture classes, totaling approximately 2.45 million trainable parameters.** This architecture enables the model to learn hierarchical representations, progressing from low-level edge detection through intermediate finger shape recognition to complete hand gesture understanding.

The CNN achieved exceptional performance on the test set with an accuracy of 97.6%, macro-F1 score of 0.9759, Cohen's Kappa of 0.9755, and a near-perfect ROC-AUC score of 0.9998 using one-vs-rest evaluation. Most gesture classes achieved perfect classification accuracy, with errors primarily occurring between visually similar gesture pairs such as M vs N, V vs W, and 0 vs O—the same challenging cases identified in traditional feature extraction approaches. Specific letters including 'w', 'm', and 'z' showed slightly lower recall rates, indicating these gestures present inherent recognition challenges regardless of the classification method employed.

The CNN approach offers significant advantages over traditional methods, combining high accuracy with robustness to moderate noise levels while generating compact 128-dimensional feature embeddings. These embeddings can serve as powerful representations for hybrid classification systems, providing a learned feature space that captures the essential characteristics of ASL gestures more effectively than handcrafted features. The model's ability to automatically learn relevant features eliminates the need for manual feature engineering while achieving superior performance across all evaluation metrics.

Method	Accuracy	F1-score (Macro)	Cohen's Kappa	ROC-AUC (OvR)	Key Observations
Custom CNN (end-to-end)	0.9761	0.9759	0.9755	0.9998	Learns hierarchical features; robust but slightly confused on similar pairs (M/N, V/W, O/0).
CNN Features + Logistic Regression	**0.9781**	**0.978**	**0.9775**	0.9996	Best CNN-based performance; embeddings are linearly separable, LR handles them effectively.
CNN Features + Random Forest	0.9761	0.9758	0.9755	**0.9999**	Matches end-to-end CNN; highly stable, strong ROC-AUC.
CNN Features + KNN	0.9642	0.964	0.9632	0.9948	Good but sensitive to outliers; lower recall in tricky gestures.
CNN Features + Decision Tree	0.9145	0.9158	0.9121	0.956	Overfits in high-dimensional embedding space; weakest among CNN-based classifiers.

MobileNetV2 for ASL Gesture Recognition

- To leverage pretrained knowledge, MobileNetV2 (trained on ImageNet) was fine-tuned for ASL recognition. Training was performed in two stages: (1) freezing the backbone while training the classification head, and (2) unfreezing the top layers for fine-tuning.

- The fine-tuned model achieved 96.4% accuracy, with a macro F1-score of 0.964 and a ROC-AUC of 0.9997. While effective, its robustness to noise was limited: performance dropped sharply from 95.2% (σ=0.05) to 29.6% (σ=0.15). This reflects the ImageNet bias toward natural images, making the network more sensitive to perturbations in structured gesture data.

- MobileNetV2 embeddings (1280-D features) proved highly effective when paired with classical classifiers. Logistic Regression achieved 98.2% accuracy, surpassing both the fine-tuned MobileNetV2 and the custom CNN. Random Forest and KNN also performed strongly (≈96%), while Decision Trees struggled (82.5%) due to overfitting in high dimensions.

Method	Accuracy	F1-score (Macro)	Cohen's Kappa	ROC-AUC (OvR)	Robustness to Noise	Notes
Fine-tuned MobileNetV2	0.9642	0.964	0.9632	0.9997	Drops from 95.2% (σ=0.05) → 29.6% (σ=0.15)	Good baseline transfer learning, but fragile under noise
MobileNetV2 Features + Logistic Regression	**0.9821**	**0.9821**	**0.9816**	**0.9999**	Degrades under heavy noise (≈29% at σ=0.15)	Best overall accuracy; embeddings highly separable
MobileNetV2 Features + Random Forest	0.9622	0.9624	0.9611	0.9999	Moderate robustness	Stable, strong ROC-AUC
MobileNetV2 Features + KNN	0.9602	0.9601	0.9591	0.9956	Sensitive to perturbations	Good, but less stable
MobileNetV2 Features + Decision Tree	0.825	0.826	0.82	0.9101	Very poor under noise	Overfits in high-dim space