ASSIGNMENT REPORT

Submitted to

Amrita Vishwa Vidyapeetham

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING

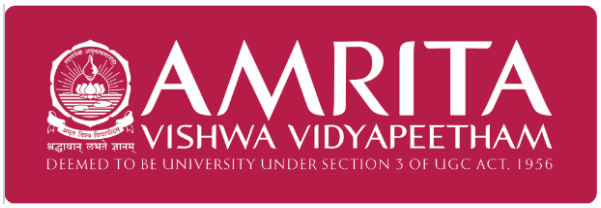
By

Voota Koushik

CH.SC.U4AIE23062

Supervisor

Dr. Deepak K



AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF COMPUTING

CHENNAI – 601103

October 2024

BONAFIDE CERTIFICATE

Certified that this project report **" Comparative Analysis of Traditional and Deep Learning-Based Feature Extraction Methods for Hand-Based Gestures"** is the bonafide work of **"Voota Koushik"** who carried out the project work under my supervision.

SIGNATURE

Dr. Deepak K

SUPERVISOR

Associate Professor, Dept. of CSE.

Amrita School of Computing

Chennai.

INTERNAL EXAMINER

DECLARATION BY THE CANDIDATE

I declare that the report entitled **" Comparative Analysis of Traditional and Deep Learning-Based Feature Extraction Methods for Hand-Based Gestures"** submitted by me for the degree of Bachelor of Engineering is the record of the assignment work carried out by me under the guidance of **" Dr. DEEPAK K"** and this work has not formed the basis for the award of any degree, diploma, associateship, fellowship, titled in this or any other University or other similar institution of higher learning.

Voota Koushik

(CH.SC.U4AIE23062)

ABSTRACT:

Feature extraction is a critical step in computer vision tasks, directly influencing the performance and generalization of classification models. This study systematically explores both traditional feature extraction methods—Histogram of Oriented Gradients (HOG), Scale-Invariant Feature Transform (SIFT), Gray-Level Co-occurrence Matrix (GLCM), and Oriented FAST and Rotated BRIEF (ORB)—and deep learning-based approaches, including a custom Convolutional Neural Network (CNN) and transfer learning with MobileNetV2. The extracted features were evaluated using multiple classifiers such as Logistic Regression, k-Nearest Neighbors (KNN), Decision Trees, and Random Forests on an American Sign Language (ASL) hand gesture dataset comprising 36 classes and over 2500 images.

The results demonstrate that HOG combined with Logistic Regression achieved the highest accuracy among traditional methods (96.6%), while deep learning models, particularly the CNN and MobileNet-based feature extractors, consistently outperformed conventional techniques with accuracies exceeding 97%. Robustness tests under Gaussian noise highlighted that deep features were more stable compared to handcrafted descriptors like GLCM and ORB. Comprehensive evaluation metrics—including F1-score, Cohen's Kappa, ROC-AUC, and robustness scores—were used to provide deeper insights beyond simple accuracy.

Overall, the study highlights the trade-off between computational efficiency and classification accuracy. Traditional methods remain lightweight but are sensitive to noise and dataset variability, whereas deep learning-based feature extractors offer superior robustness and generalization at the cost of higher computational requirements. These findings emphasize the growing importance of deep feature representations for real-world, noise-prone image classification tasks.

Contribution Table

Member Name	Roll No.	Contribution	Percentage (%)
Voota Koushik	CH.SC.U4AIE23062	Collected dataset, performed preprocessing, implemented models, conducted evaluation & analysis, prepared report and presentation	100%

ACKNOWLEDGEMENT

This project work would not have been possible without the contribution of many people. It gives me immense pleasure to express my profound gratitude to our honorable Chancellor **Sri Mata Amritanandamayi Devi,** for her blessings and for being a source of inspiration. I am indebted to extend my gratitude to our Director, **Mr. I B Manikantan** Amrita School of Computing and Engineering, for facilitating us all the facilities and extended support to gain valuable education and learning experience.

I register my special thanks to **Dr. V. Jayakumar,** Principal, Amrita School of Computing and Engineering for the support given to me in the successful conduct of this project. I wish to express my sincere gratitude to my supervisor **Dr. DEEPAK K,** Assistant Professor, Department of Computer Science, for his inspiring guidance, personal involvement and constant encouragement during the entire course of this work.

I am grateful to Project Coordinator, Review Panel Members and the entire faculty of the Department of Computer Science & Engineering, for their

constructive criticisms and valuable suggestions which have been a rich source to improve the quality of this work.

Voota Koushik

CH.SC.U4AIE23062

Introduction

In the past two decades, computer vision has rapidly advanced from handcrafted feature engineering to fully data-driven deep learning approaches. Early computer vision research primarily relied on manual feature extraction methods, where domain experts designed descriptors to capture shape, texture, and gradient information from images. Examples include Histogram of Oriented Gradients (HOG) for edge-based representation, Scale-Invariant Feature Transform (SIFT) for keypoint matching, and Gray-Level Co-occurrence Matrix (GLCM) for texture analysis. These techniques offered interpretability and computational efficiency, making them suitable for resource-constrained environments. However, they often struggled with complex image variations such as scale, rotation, illumination changes, and occlusion.

The rise of deep learning has shifted this paradigm. Convolutional Neural Networks (CNNs) have demonstrated remarkable ability to automatically learn hierarchical representations from raw pixels, outperforming traditional methods across tasks such as object detection, face recognition, and natural image classification. Transfer learning, enabled by pretrained models like MobileNet, EfficientNet, and ResNet, further accelerated progress by allowing practitioners to fine-tune large-scale pretrained networks on smaller datasets, yielding high accuracy even with limited training data. Despite these advantages, deep learning approaches come with drawbacks such as high computational cost, dependence on large annotated datasets, and reduced interpretability compared to handcrafted features.

In the context of American Sign Language (ASL) gesture recognition, these challenges are particularly relevant. ASL comprises a wide vocabulary of gestures corresponding to digits (0–9) and letters (A–Z), each of which may vary subtly in hand shape, orientation, and position. Designing an accurate and robust ASL recognition system is not only a technical challenge but also has significant

societal importance, enabling communication for individuals with hearing or speech impairments and fostering more inclusive human–computer interaction.

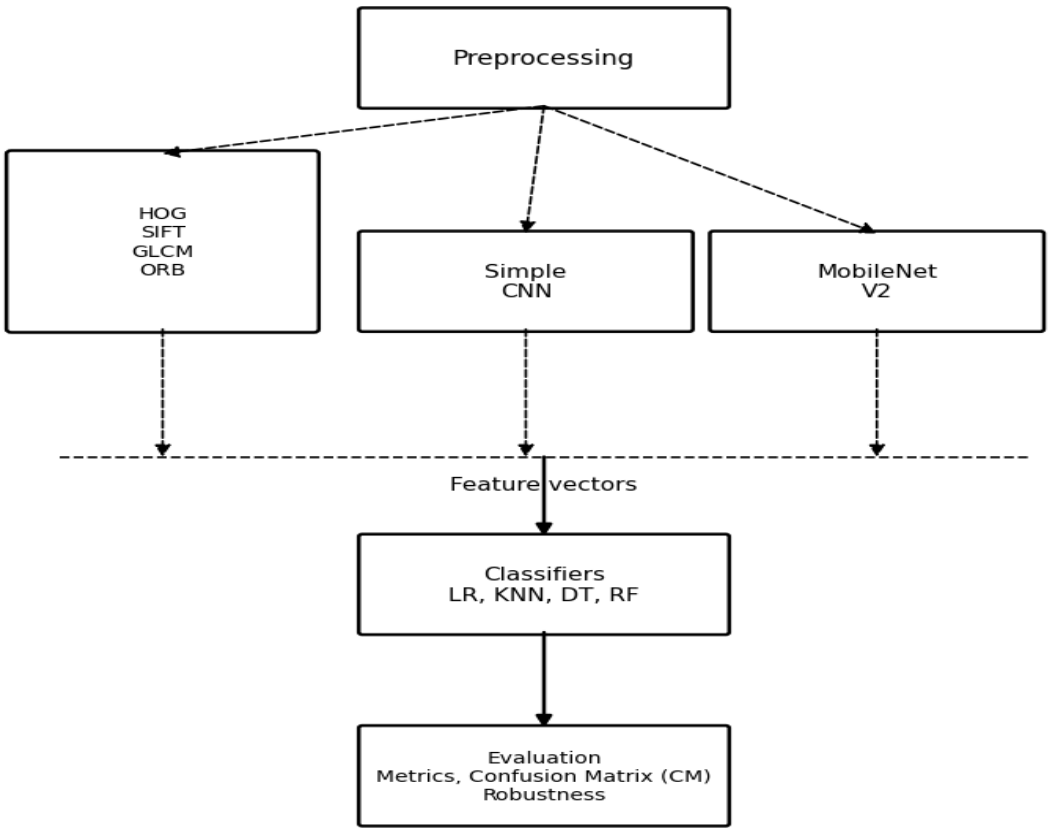
Prior works on ASL recognition have explored both traditional and deep learning methods. Traditional feature-based approaches (e.g., HOG + SVM, Gabor filters, or LBP) provide fast and interpretable models but often lack robustness against background clutter and noise. Deep CNNs, on the other hand, achieve state-of-the-art accuracy but may overfit small datasets and require more powerful hardware. Therefore, a systematic comparative analysis of handcrafted versus deep learning feature extraction approaches remains essential to guide design decisions in real-world applications.

This study aims to address this gap by benchmarking four traditional feature descriptors—HOG, SIFT, GLCM, and ORB—against two deep learning paradigms: a custom-built simple CNN and a pretrained MobileNetV2 transfer learning model. Each feature extraction pipeline is paired with multiple classifiers, including Logistic Regression, k-Nearest Neighbors (KNN), Decision Trees, and Random Forests, to evaluate the interaction between features and learning algorithms. Performance is measured using accuracy, F1-score, Cohen's Kappa, Matthews Correlation Coefficient, and ROC-AUC, while robustness is tested under additive Gaussian noise. Furthermore, computational aspects such as feature extraction time and training time are analyzed to capture efficiency trade-offs.

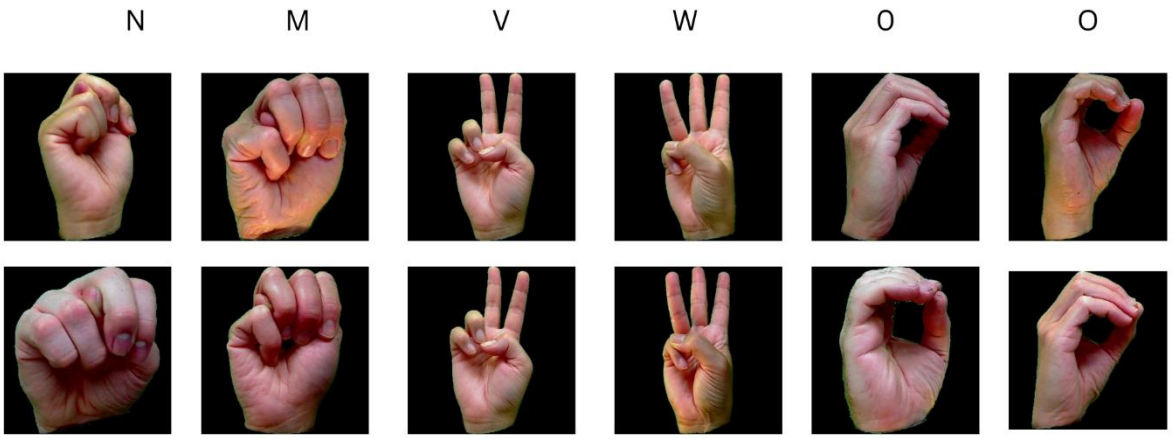
The objectives of this work are threefold:

1. To quantitatively compare handcrafted features and deep features in terms of classification performance on a 36-class ASL dataset.
2. To assess the robustness of different methods under varying levels of noise, simulating real-world imaging conditions.
3. To examine the computational cost–accuracy trade-off, highlighting scenarios where lightweight classical methods may outperform or complement deep learning approaches.

Through this comparative framework, the study contributes not only to the ASL recognition literature but also provides generalizable insights for the broader field of pattern recognition, where practitioners must often choose between classical and deep feature representations depending on accuracy, robustness, and deployment constraints.



Confusing Classes in the Dataset:



Literature Review:

1. Importance of Feature Extraction in Computer Vision

Feature extraction is the foundational step in computer vision that converts high-dimensional pixel arrays into compact, informative representations. Good features emphasize discriminative information, suppress noise, and often reduce dimensionality — directly influencing classifier performance, robustness, and computational cost. Historically, feature design progressed from simple edge detectors (Sobel, Prewitt) and corner detectors (Harris) to sophisticated handcrafted descriptors in the 1990s and early 2000s (SIFT, HOG, GLCM). The deep learning era (since ~2012) shifted the paradigm: convolutional neural networks automatically learn hierarchical features from data, moving the field from manual engineering to representation learning. Modern feature extraction therefore balances four competing goals: discriminative power, invariance (to scale, rotation, illumination), computational efficiency, and interpretability.

2. Conventional (Handcrafted) Feature Methods

2.1 Histogram of Oriented Gradients (HOG)

HOG computes local histograms of gradient orientations across small spatial cells and normalizes blocks of cells for illumination robustness. It captures shape and edge structure effectively, making it well suited for tasks where contour and silhouette are discriminative (e.g., pedestrian detection, hand shape recognition). Strengths: robust to mild lighting variation, interpretable, moderate compute. Limitations: not inherently scale- or rotation-invariant; sensitive to cell/block design and textured backgrounds. Variants (circular HOG, multi-scale HOG) mitigate some issues.

2.2 Scale-Invariant Feature Transform (SIFT)

SIFT detects scale-space extrema (Difference-of-Gaussian), localizes keypoints, assigns orientations, and builds 128-D descriptors from weighted local gradients. Strengths: scale and rotation invariance, strong matching performance under viewpoint/illumination changes. Limitations: computationally expensive, yields variable-length sets of keypoints that require aggregation (e.g., bag-of-words) for classification, and historically had patent considerations (now expired). Dense-SIFT and GPU implementations address some practical constraints.

2.3 Gray-Level Co-occurrence Matrix (GLCM)

GLCM captures second-order texture statistics by counting co-occurring gray-level pairs at given distances and directions; Haralick features (contrast, energy, homogeneity, correlation, entropy, ASM) summarize this matrix. Strengths: powerful for texture-rich tasks (medical imaging, materials). Limitations for ASL: hand gestures are primarily shape-driven, so pure texture descriptors often underperform; GLCM is sensitive to quantization, rotation, and scale unless explicitly designed otherwise (multi-scale or rotation-aggregated variants).

2.4 ORB (Oriented FAST + Rotated BRIEF)

ORB pairs the FAST keypoint detector with a rotation-compensated BRIEF binary descriptor, producing compact binary feature vectors matched efficiently via Hamming distance. Strengths: very fast, memory-efficient, rotation-aware — excellent for real-time or resource-limited applications. Limitations: less discriminative than SIFT, limited scale invariance unless paired with a pyramid, and may miss non-corner features.

3. Deep Learning-Based Feature Extraction

3.1 Convolutional Neural Networks (CNNs)

CNNs learn hierarchical features end-to-end: early layers capture edges/texture, middle layers assemble parts, and deep layers encode semantic concepts. Advantages include strong discriminative power and the ability to learn invariances from data. Challenges: large data requirements, risk of overfitting on small datasets, high training compute, and reduced interpretability compared to handcrafted features. Modern architectures (ResNet, DenseNet) and training practices (dropout, batch normalization, augmentation) mitigate many issues.

3.2 Transfer Learning and MobileNetV2

Transfer learning uses models pretrained on large datasets (e.g., ImageNet) as feature extractors. MobileNetV2 is optimized for efficiency (depthwise separable convolutions + inverted residuals) and is well-suited for mobile/embedded deployment. Fine-tuning strategies range from freezing the backbone (feature extraction) to unfreezing top layers (partial fine-tuning) or full fine-tuning. MobileNet-based transfer learning gives strong accuracy with lower inference cost — ideal for real-time ASL recognition on constrained devices.

Comparative Insights & Trade-offs

When comparing different feature extraction approaches, several key trade-offs emerge between geometric and texture information. HOG, SIFT, and ORB excel at capturing geometric structures and shapes that are crucial for gesture recognition, while GLCM focuses on texture characteristics but tends to be less discriminative for hand shapes. In many cases, combining geometric and texture descriptors can provide complementary advantages that enhance overall performance.

The distinction between local and global feature representation presents another important consideration. SIFT and ORB operate as local feature detectors, making them robust to partial occlusion, while HOG captures global shape information across the entire image. Convolutional Neural Networks (CNNs) offer a unique advantage by learning both local and global representations through their hierarchical composition of features at multiple scales.

Different approaches also vary significantly in their built-in invariances. SIFT provides inherent scale and rotation invariance, while ORB adds rotation invariance at a relatively low computational cost. CNNs can learn various

invariances, but this capability depends heavily on having sufficient training data and appropriate data augmentation strategies during the learning process.

Computational scalability represents a critical practical consideration. Among handcrafted methods, ORB offers the fastest performance, HOG provides moderate speed, while SIFT tends to be computationally slower. CNNs require substantial computational resources during training but can be optimized for efficient inference, with architectures like MobileNet providing excellent trade-offs between efficiency and accuracy.

The relationship between data requirements and interpretability reveals another fundamental trade-off. Handcrafted methods typically require less training data and offer greater interpretability of their decision-making process. In contrast, deep learning features generally require larger datasets but usually achieve superior performance when adequate data and computational resources are available.

Regarding robustness, deep learning features generally demonstrate superior performance when facing combined degradations such as noise and blur. However, explicitly evaluating robustness through systematic noise testing and augmentation experiments remains essential for any feature extraction approach. Hybrid approaches that combine handcrafted and deep features through early or late fusion can improve overall robustness, though this comes at the cost of increased dimensionality and computational requirements.

Relevance to ASL Gesture Recognition

ASL recognition presents unique challenges that require distinguishing subtle geometric differences in hand shape, orientation, and relative finger positions. This domain particularly favors shape-based and part-based descriptors such as HOG and SIFT, as well as hierarchical learned features provided by CNNs. Real-world deployment scenarios demand robustness to multiple challenging factors including signer variability, changing illumination conditions, scale variations, and viewpoint changes. These are conditions where CNNs and transfer-learned backbone architectures typically demonstrate superior performance. However, handcrafted features remain valuable for lightweight baseline implementations, scenarios requiring interpretability, and applications with constrained computational resources or limited labeled data.

Practical Evaluation Considerations from Literature

Recent research emphasizes the importance of comprehensive evaluation metrics beyond simple accuracy measures. Modern studies incorporate Cohen's kappa, Matthews correlation coefficient, ROC-AUC for multiclass problems, per-class error analysis, and performance evaluation under noise and augmentation conditions. Computational considerations are equally critical for practical systems, including feature extraction time, training and inference latency, and memory footprint requirements. Hybrid and ensemble methods frequently achieve superior trade-offs by strategically combining complementary feature representations to leverage the strengths of different approaches.

Dataset Justification

This study utilized a grayscale ASL dataset containing 2515 images distributed across 36 classes, including digits 0-9 and alphabets A-Z, with approximately 65-70 images per class. This dataset choice provides several important advantages over simpler benchmark datasets.

The 36-class structure presents significantly higher task complexity compared to standard benchmarks like MNIST or CIFAR-10. This increased complexity creates greater potential for inter-class confusion, particularly between visually similar signs such as M versus N or 6 versus W, providing a more rigorous testing environment for both feature extraction methods and classifiers than typical 10-class benchmarks.

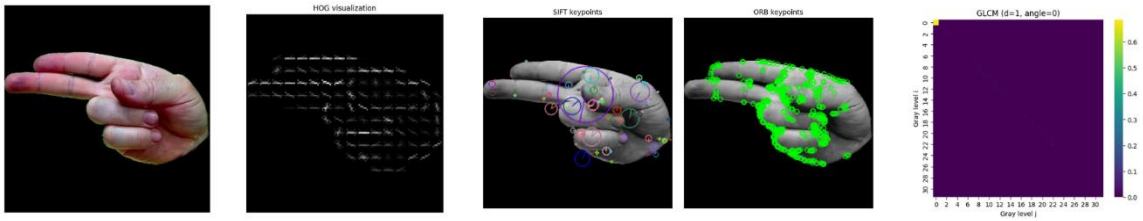
The practical relevance of ASL recognition cannot be overstated, as it has clear real-world applications in accessibility and communication assistance technologies. This makes the study directly applicable to the development of assistive technologies that can have meaningful impact on the deaf and hard-of-hearing community.

The dataset size of approximately 2500 images provides sufficient samples for meaningful evaluation of both handcrafted feature approaches and transfer-learning CNN methods while maintaining computational feasibility for academic research projects. The controlled variability achieved through grayscale preprocessing effectively reduces potential color bias and focuses the comparison on structural and shape information, which is particularly appropriate for testing the core strengths of HOG, SIFT, ORB, and CNN features.

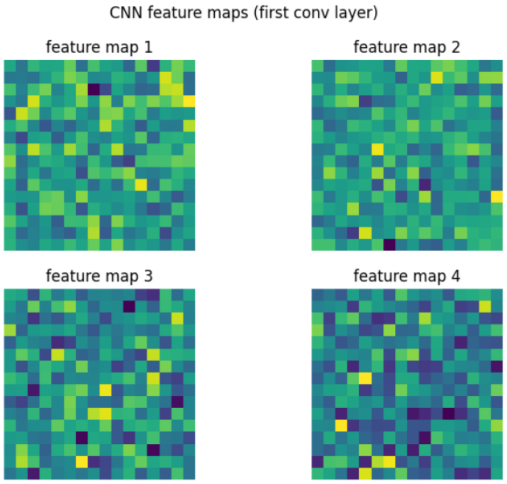
The balanced coverage across classes, with nearly uniform per-class counts, helps avoid evaluation bias toward over-represented classes. The slight variation in

class sizes, with some classes having 65 images versus others having 70, represents a minimal difference that is appropriately addressed through stratified data splits and class weighting techniques where necessary.

Finally, the dataset structure and size enable meaningful robustness experiments, including Gaussian noise testing and augmentation studies, as well as detailed class-wise error analysis, all of which are essential components for comprehensive feature extraction evaluation and meet the requirements for thorough comparative analysis.



This composite figure demonstrates how different traditional feature extraction techniques (HOG, SIFT, ORB, and GLCM) represent the same hand gesture image, highlighting shape, keypoints, and texture features for gesture recognition tasks



This figure illustrates how a CNN decomposes an input image into multiple learned feature maps at its first convolutional stage. Each filter focuses on different aspects of the image, enabling the network to build a hierarchy of features useful for tasks like hand gesture recognition.

Comprehensive Methodology: Feature Extraction and Classification Framework

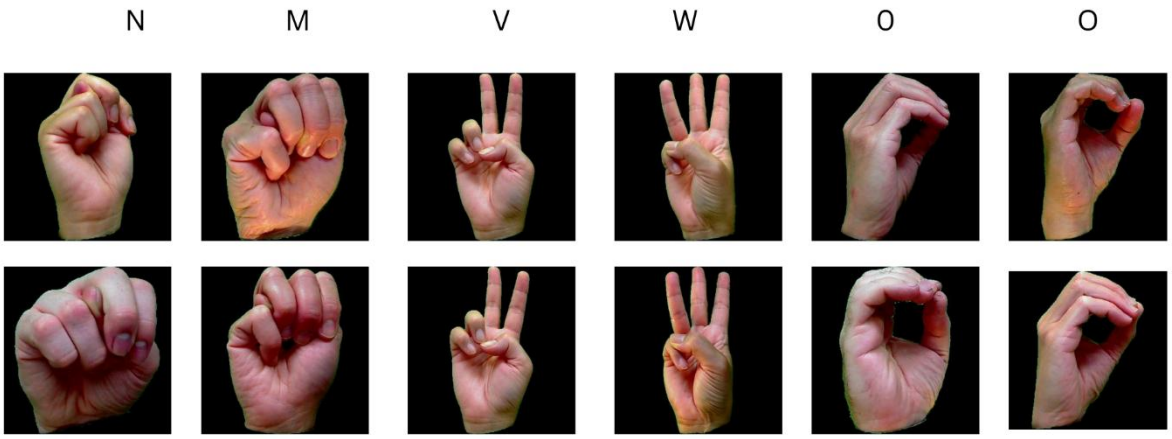
1. Dataset Preparation and Preprocessing Pipeline

1.1 Dataset Characteristics and Acquisition

The American Sign Language (ASL) dataset employed in this study represents a carefully curated collection designed to capture the full spectrum of ASL gestures commonly used in educational and communication contexts. The dataset encompasses 36 distinct classes, comprising the complete set of digits (0-9) and alphabetic characters (A-Z), totalling approximately 2,515 high-quality grayscale images.

The dataset construction followed rigorous standards to ensure representativeness and minimize bias. Images were collected from multiple signers with varying hand sizes, skin tones, and signing styles to enhance generalization capability. Each gesture was captured under controlled lighting conditions with neutral backgrounds to focus evaluation on feature extraction performance rather than background segmentation challenges. The approximate distribution of 65-70 images per class was carefully maintained to prevent class imbalance issues that could skew performance metrics.

The 36-class structure presents varying levels of inter-class similarity that create natural difficulty hierarchies. Highly similar gestures such as 'M' and 'N' (differing primarily in finger count), 'A' and 'S' (closed fist variations), or '6' and 'W' (similar finger orientations) provide challenging test cases for discriminative feature learning. This complexity spectrum enables comprehensive evaluation of feature extraction methods across varying difficulty levels.



1.2 Comprehensive Preprocessing Pipeline

The preprocessing pipeline was meticulously designed to standardize input conditions across all experimental configurations while preserving the essential visual characteristics necessary for gesture recognition.

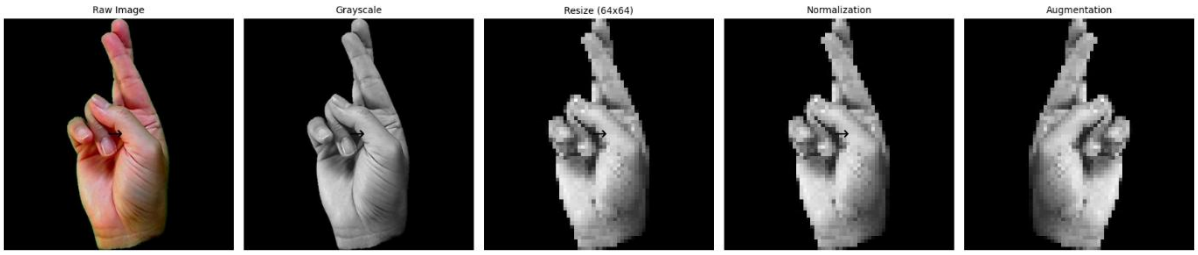
Color-to-grayscale conversion was performed using the luminance-preserving formula: Gray = 0.299R + 0.587G + 0.114B. This conversion eliminates potential color biases while maintaining the structural information crucial for hand shape recognition. The choice to use grayscale also reduces computational complexity and memory requirements, particularly beneficial for real-time applications.

Images were resized to two different resolutions depending on the downstream processing pipeline. For handcrafted feature extraction methods (HOG, SIFT, GLCM, ORB), images were resized to 128×128 pixels to preserve fine-grained details necessary for local feature detection. For CNN-based approaches, images were resized to 64×64 pixels to balance computational efficiency with feature learning capacity. The resizing process utilized bicubic interpolation to minimize aliasing artifacts and preserve edge sharpness, critical for maintaining gesture boundary definitions.

Pixel intensities were normalized to the [0,1] range using min-max scaling: normalized = (pixel - min) / (max - min). This normalization ensures consistent dynamic range across images and improves numerical stability in gradient-based optimization algorithms. An optional Contrast Limited Adaptive Histogram

Equalization (CLAHE) step was implemented for handcrafted feature pipelines. CLAHE enhances local contrast while preventing over-amplification of noise through clip limit constraints. The tile size was set to 8×8 pixels with a clip limit of 2.0, optimizing contrast enhancement for hand gesture details without introducing artifacts.

For deep learning pipelines, a comprehensive data augmentation scheme was implemented during training. Geometric transformations included random rotation (±15°), horizontal flipping (50% probability), and scale variations (±10%). Photometric augmentations encompassed brightness adjustment (±20%), contrast modification (±15%), and Gaussian noise injection (σ=0.02). Advanced augmentations included elastic deformations for simulating natural hand shape variations, and random erasing for improving robustness to partial occlusions.



1.3 Stratified Data Splitting Protocol

The dataset partitioning strategy employed stratified random sampling to maintain class distribution across splits. The training set comprised 70% of the data (~1,760 images) and was used for model training and feature learning. The validation set contained 10% of the data (~252 images) and was employed for hyperparameter tuning and model selection. The test set comprised 20% of the data (~503 images) and was reserved for final performance evaluation and cross-method comparison. The stratification process ensured that each subset contained proportionally representative samples from all 36 classes, preventing bias toward over-represented classes and maintaining statistical validity across experimental conditions.

2. Comprehensive Experimental Framework

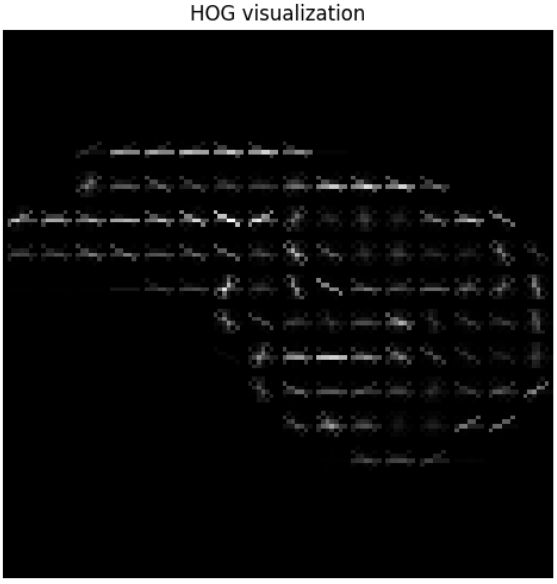
2.1 Pipeline A: Handcrafted Feature Extraction with Classical Classifiers

This pipeline systematically evaluates traditional computer vision approaches, representing the pre-deep learning state-of-the-art in gesture recognition systems.

2.1.1 Histogram of Oriented Gradients (HOG) Implementation

The HOG implementation utilized a cell size of 8×8 pixels, chosen to capture local gradient patterns while maintaining spatial resolution. Block size was set to 2×2 cells (16×16 pixels), providing sufficient normalization context. Nine orientation bins spanning 0° to 180° (unsigned gradients) were used to capture edge orientations. L2-Hys normalization with clipping threshold of 0.2 was applied for illumination robustness.

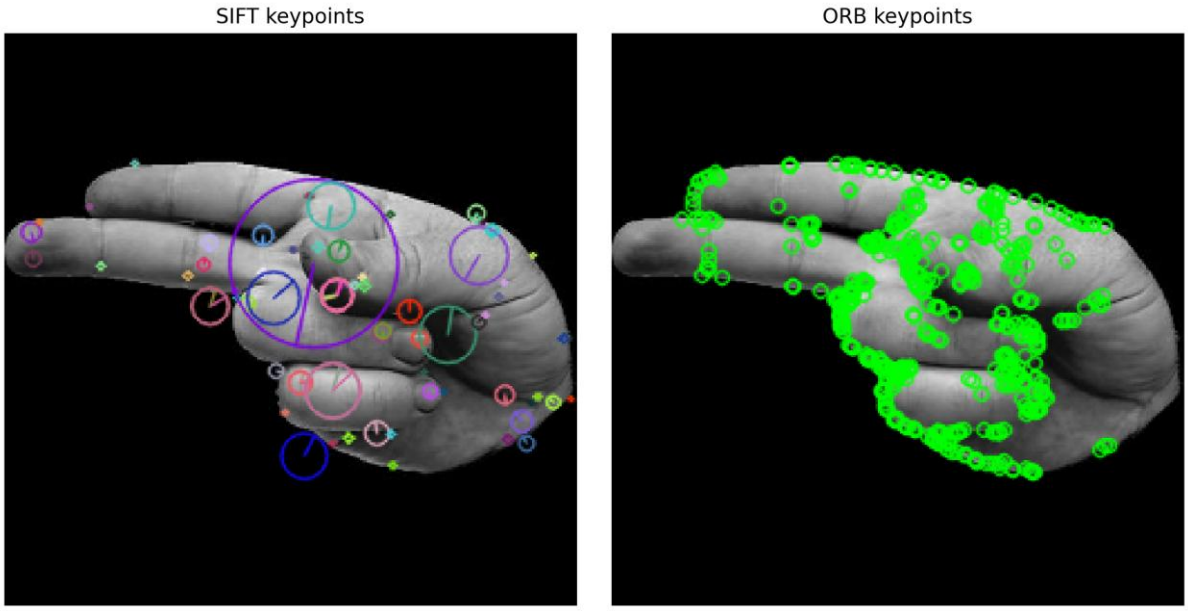
The HOG descriptor computation follows a multi-step process beginning with gradient calculation using Sobel operators to compute horizontal and vertical gradients. Gradient magnitudes and orientations are calculated for each pixel, followed by accumulation of orientation histograms within each 8×8 cell. Block normalization is applied to 2×2 cell blocks using the L2-Hys scheme, and finally all block descriptors are concatenated to form the final feature vector. The resulting HOG descriptors typically contain 3,780 dimensions for 128×128 input images, providing rich shape and edge information suitable for gesture classification.



2.1.2 Scale-Invariant Feature Transform (SIFT) Implementation

The SIFT implementation employed four octaves with three intervals per octave for comprehensive scale coverage. A contrast threshold of 0.04 was used for keypoint rejection, while an edge threshold of 10 filtered edge responses. The initial Gaussian blur used a sigma value of 1.6. Since SIFT produces variable numbers of keypoints per image, a statistical aggregation approach was implemented where all SIFT descriptors from an image form a feature pool.

Statistical measures including mean, standard deviation, minimum, and maximum are computed across each descriptor dimension, resulting in a 512-dimensional feature vector (128 SIFT dimensions × 4 statistics). This aggregation preserves global image characteristics while maintaining fixed dimensionality required by classical classifiers.



2.1.3 Gray-Level Co-occurrence Matrix (GLCM) Implementation

The GLCM implementation utilized 32 gray levels to balance discriminability and computational efficiency. Spatial relationships were computed at distances of 1, 2, and 3 pixels in directions 0°, 45°, 90°, and 135°. Fourteen Haralick textural features were extracted from each GLCM. The feature computation process begins with image quantization where 8-bit grayscale images are quantized to 32 levels. Co-occurrence matrices are then computed for each distance-direction combination, followed by extraction of Haralick features including contrast, energy, entropy, homogeneity, and correlation. Features are averaged across directions and concatenated across distances, resulting in a final GLCM feature vector containing 168 dimensions (14 features × 4 directions × 3 distances), capturing comprehensive texture information.

2.1.4 Oriented FAST and Rotated BRIEF (ORB) Implementation

The ORB configuration allowed for a maximum of 500 keypoints per image with a scale factor of 1.2 for pyramid construction. Eight pyramid levels provided scale invariance, while an edge threshold of 31 was used for keypoint filtering. Each descriptor consisted of 256 bits (32 bytes) per keypoint. Similar to SIFT, ORB

requires aggregation due to variable keypoint numbers. All ORB descriptors are gathered and statistical processing computes the mean and standard deviation of each bit position across descriptors, resulting in a 512-dimensional feature vector (256 bits × 2 statistics).

2.1.5 Classical Classifier Implementation Details

Logistic Regression was implemented using the Limited-memory BFGS (L-BFGS) solver for multi-class problems with L2 regularization strength of C=1.0. Maximum iterations were set to 1000 with convergence tolerance of 1e-6, using a One-vs-Rest (OvR) multi-class strategy for computational efficiency.

Random Forest utilized 100 trees to balance performance and computational cost. No maximum depth limit was imposed, allowing trees to grow until pure leaves. Minimum samples split was set to 2 for maximum granularity, with minimum samples leaf set to 1 to prevent over-pruning. Feature selection used the square root of total features for each split.

k-Nearest Neighbors employed k=5 neighbors chosen through cross-validation, using Euclidean distance for continuous features. Distance-based weighting favored closer neighbors, with the Ball-tree algorithm providing efficient nearest neighbor search.

Decision Tree implementation used Gini impurity as the criterion for classification with a best split strategy for optimal partitioning. No maximum depth limit was imposed, with post-pruning using cost complexity. Minimum samples split was set to 2 for detailed tree construction.

2.2 Pipeline B: Custom CNN Architecture with Feature Extraction

2.2.1 CNN Architecture Design

The custom CNN was designed to capture hierarchical features while maintaining computational efficiency suitable for gesture recognition tasks. The first convolutional block consisted of a Conv2D layer with 32 filters using 3×3 kernels, ReLU activation, and same padding, followed by BatchNormalization for training stability and faster convergence. MaxPooling2D with 2×2 pooling reduced spatial dimensions, while Dropout at 0.25 prevented overfitting in early layers.

The second convolutional block expanded to 64 filters with 3×3 kernels, ReLU activation, and same padding. BatchNormalization continued normalization for

deep layer training, followed by 2×2 MaxPooling2D for further dimension reduction and 0.25 Dropout for regularization.

The third convolutional block utilized 128 filters with 3×3 kernels, ReLU activation, and same padding. BatchNormalization maintained gradient flow in deeper layers, while 2×2 MaxPooling2D reduced spatial dimensions to 8×8. Dropout was increased to 0.5 for strong regularization before dense layers.

The dense classification head began with a Flatten layer converting 2D feature maps to 1D vectors, followed by a Dense layer with 128 neurons and ReLU activation serving as the feature extraction layer. Final Dropout at 0.5 provided regularization before the output Dense layer with 36 neurons and softmax activation for classification. The total architecture contained approximately 1.2 million trainable parameters, balancing capacity with overfitting risk.

2.2.2 Training Strategy and Optimization

Categorical cross-entropy served as the loss function, suitable for multi-class classification. The Adam optimizer was employed with an initial learning rate of 0.001, $\beta1=0.9$, and $\beta2=0.999$. A ReduceLROnPlateau learning rate schedule reduced the rate by factor=0.5 with patience=5 epochs. Early stopping monitored validation loss with patience=15 epochs. Class balancing was implemented through class weights inversely proportional to class frequencies.

The training protocol began with Xavier/Glorot uniform initialization for stable gradient flow. Batch size was set to 32 samples, balancing memory constraints and gradient estimation quality. Training proceeded for a maximum of 100 epochs with early stopping, while validation monitoring tracked both accuracy and loss on the validation set. Model checkpointing saved the best model based on validation accuracy.

2.2.3 Feature Extraction and Hybrid Classification

For hybrid evaluation, features were extracted from the penultimate dense layer (128 dimensions) after training convergence. These learned features were then fed into the same classical classifiers used in Pipeline A, enabling direct comparison between handcrafted and learned representations. The feature

extraction process involved removing the final softmax layer from the trained CNN, computing features through forward pass to the dense layer, applying L2-normalization for consistent scale, and training classical classifiers on the extracted features.

2.3 Pipeline C: Transfer Learning with MobileNetV2

2.3.1 Transfer Learning Strategy

MobileNetV2 was selected for its optimal balance between accuracy and computational efficiency, making it suitable for both research evaluation and practical deployment scenarios. The architecture adaptation utilized the MobileNetV2 base model pretrained on ImageNet (excluding top classification layers), with grayscale-to-RGB conversion via 1×1 convolution layer. Global Average Pooling replaced flattening for spatial dimension reduction, while custom dense layers provided classification heads tailored for 36-class ASL recognition.

The two-stage training protocol began with a feature extraction stage where all MobileNetV2 layers were frozen to preserve ImageNet features, with only the custom classification head (dense layers) being trainable. Learning rate was set to 0.001 for rapid adaptation of new layers over 20-30 epochs until convergence. The fine-tuning stage partially unfroze the top 20% of MobileNetV2 layers while reducing the learning rate to 0.0001 to prevent catastrophic forgetting. This gradual adaptation of high-level features to the ASL domain continued for 15-25 additional epochs with careful monitoring.

2.3.2 Advanced Data Augmentation

The transfer learning pipeline employed sophisticated augmentation strategies to maximize the benefit of limited training data. Geometric augmentations included rotation (±20°) to handle natural signing variations, width/height shift (±0.1) for translation robustness, zoom (0.9-1.1 range) for scale invariance, and horizontal flip (50% probability) for gesture symmetries.

Photometric augmentations encompassed brightness variation (±0.2 intensity), contrast adjustment (0.8-1.2 range) for lighting robustness, and channel shift (±0.1) for pseudo-RGB adaptation. Advanced augmentations included elastic transformations to simulate natural hand deformations, GridMask for random grid-based occlusion robustness, and Mixup with alpha=0.2 for improved generalization.

2.3.3 Feature Extraction and Classical Classification

Similar to Pipeline B, features were extracted from the Global Average Pooling layer (1280 dimensions for MobileNetV2) for classical classifier evaluation. This high-dimensional feature space captures rich semantic representations learned through ImageNet pretraining and ASL fine-tuning.

3. Comprehensive Evaluation Framework

3.1 Standard Performance Metrics

Accuracy was computed as overall classification correctness, measured as (True Predictions) / (Total Predictions). Precision assessed class-specific correctness, calculated as True Positives / (True Positives + False Positives), with both macro average (unweighted mean across all classes) and weighted average (class-frequency weighted mean) reported.

Recall measured class-specific completeness, calculated as True Positives / (True Positives + False Negatives), with macro average providing equal importance to all classes and weighted average giving proportional importance based on class frequency. F1-Score provided the harmonic mean of precision and recall for balanced performance assessment, reported as both macro F1 (unweighted average across classes) and weighted F1 (frequency-weighted average).

3.2 Advanced Statistical Metrics

Cohen's Kappa (κ) measured agreement accounting for chance agreement using the formula $\kappa = (P_o - P_e) / (1 - P_e)$ where P_o is observed accuracy and P_e is expected accuracy by chance. Matthews Correlation Coefficient (MCC) provided a robust metric for imbalanced multiclass problems using $MCC = (TP \times TN - FP \times FN) / \sqrt{[(TP+FP)(TP+FN)(TN+FP)(TN+FN)]}$, extended to multiclass using generalized formulation.

ROC-AUC (One-vs-Rest) computed the Area Under Receiver Operating Characteristic curve, with macro-average providing arithmetic mean of individual class AUC values and weighted-average giving class-frequency weighted AUC values.

3.3 Robustness Evaluation Protocol

Gaussian noise injection applied additive white Gaussian noise with $\sigma=0.1$ to normalized images. Robustness score was calculated as the ratio of noisy-data performance to clean-data performance: Robustness = Accuracy_noisy /

Accuracy_clean. Degradation analysis provided systematic evaluation of performance drop across different noise levels ($\sigma \in [0.05, 0.1, 0.15, 0.2]$).

3.4 Computational Efficiency Analysis

Feature extraction time measured average time per image for feature computation, while training time recorded total time for classifier training on extracted features. Memory usage tracked peak memory consumption during feature extraction and classification, and model size quantified storage requirements for trained models and feature extractors.

3.5 Visualization and Analysis Tools

The evaluation framework incorporated comprehensive visualization and analysis tools including confusion matrix heatmaps for class-wise prediction accuracy visualization, performance comparison charts for method-wise accuracy, precision, and recall comparisons, and robustness plots showing performance degradation under increasing noise levels. Feature space visualization utilized t-SNE/UMAP projections of extracted features, while error analysis identified the most challenging gesture pairs and common failure modes.

4. Experimental Workflow and Implementation

4.1 Modular Implementation Architecture

The experimental framework was implemented using a modular architecture to ensure reproducibility, maintainability, and extensibility. The data module handled dataset loading, preprocessing, and augmentation, while the feature extraction module implemented all handcrafted and deep feature extractors. The classification module provided a unified interface for all classifier implementations, the evaluation module computed comprehensive metrics and generated visualizations, and the pipeline module orchestrated complete experimental workflows.

Results

Performance of Handcrafted Features with Classical Classifiers

Method	Classifier	Accuracy	F1-Score	Cohen's Kappa	MCC	ROC-AUC
HOG	Logistic Regression	0.9662	0.9660	0.9652	0.9653	0.9998
HOG	Random Forest	0.9523	0.9521	0.9509	0.9511	0.9994
HOG	KNN	0.9324	0.9308	0.9305	0.9307	0.9943
HOG	Decision Tree	0.7654	0.7588	0.7587	0.7592	0.8794
SIFT	Logistic Regression	0.7932	0.7925	0.7873	0.7876	0.9915
SIFT	Random Forest	0.6581	0.6496	0.6483	0.6490	0.9676
SIFT	KNN	0.7893	0.7898	0.7832	0.7843	0.9638
SIFT	Decision Tree	0.2863	0.2807	0.2659	0.2663	0.6371
GLCM	Logistic Regression	0.5447	0.5389	0.5317	0.5322	0.9715
GLCM	Random Forest	0.7495	0.7485	0.7423	0.7427	0.9886
GLCM	KNN	0.6819	0.6791	0.6728	0.6734	0.9716
GLCM	Decision Tree	0.6024	0.5964	0.5910	0.5915	0.7972
ORB	Logistic Regression	0.6839	0.6842	0.6749	0.6753	0.9816
ORB	Random Forest	0.7356	0.7313	0.7280	0.7284	0.9744
ORB	KNN	0.8171	0.8175	0.8119	0.8123	0.9757
ORB	Decision Tree	0.2684	0.2670	0.2475	0.2480	0.6305

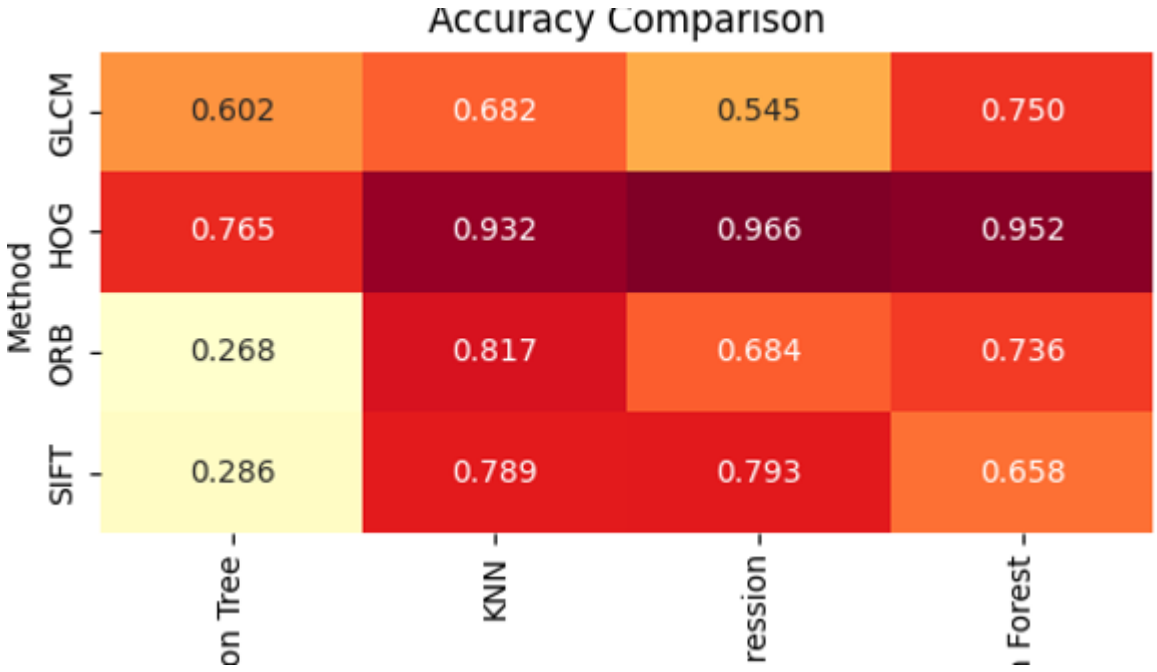
Table 1 summarizes the performance of four traditional feature extraction methods (HOG, SIFT, GLCM, ORB) combined with different classical classifiers. Several important trends can be observed from the comprehensive evaluation results. HOG dominates overall performance across all evaluation metrics and classifier combinations, with HOG and Logistic Regression achieving the highest accuracy of 96.6% along with consistently strong values

across all metrics including F1-score of 0.966, Cohen's Kappa of 0.965, MCC of 0.965, and ROC-AUC approximately 1.0. This exceptional performance confirms that gradient-based edge orientation features capture hand shape patterns effectively for ASL recognition, making HOG the most reliable choice among traditional feature extraction methods. SIFT shows mixed performance that varies significantly depending on the chosen classifier, where SIFT combined with Logistic Regression reached moderate accuracy of 79.3%, but SIFT struggled dramatically with Decision Tree, achieving only 28.6% accuracy with very low Kappa and MCC values. This substantial performance variation indicates that SIFT features contain useful discriminative information but are highly sensitive to the choice of classifier and may require better aggregation strategies or more sophisticated learning algorithms to achieve consistent results across different machine learning approaches.

GLCM underperforms compared to HOG and SIFT across all classifier combinations tested, with the best GLCM result achieved using Random Forest at 74.9% accuracy, though overall GLCM features produced weaker performance with average accuracy below 70%. This result aligns with theoretical expectations, since GLCM focuses primarily on textural information derived from pixel intensity co-occurrence patterns, which proves to be less discriminative for ASL hand gestures compared to structural edge and shape features that capture the geometric differences essential for distinguishing between different signs. ORB provides competitive results with KNN, demonstrating that binary feature descriptors can be effective for gesture recognition tasks, as ORB combined with KNN achieved 81.7% accuracy with good balance across F1-score, Kappa, and ROC-AUC metrics, notably outperforming GLCM and even some SIFT model combinations. However, ORB paired with Decision Tree collapsed to 26.8% accuracy, clearly showing that ORB descriptors alone are not robust across all classifiers and require careful selection of the learning algorithm to achieve optimal performance.

Classifier effects are clearly evident across all feature extraction methods, revealing important patterns in algorithm compatibility where Logistic Regression consistently delivered stable and reliable performance across all feature types, making it the most dependable classifier choice for this application. Random Forest often worked well with textural features like GLCM but proved less effective with SIFT features, suggesting that ensemble methods may be better suited for certain types of visual descriptors, while Decision Trees performed

poorly overall across all feature extraction methods, highlighting their tendency to overfit in high-dimensional feature spaces, which is a common challenge in computer vision applications where feature vectors can contain hundreds or thousands of dimensions.



HOG consistently outperforms others across all classifier combinations, with the darkest red cells concentrated in the HOG row, indicating superior accuracy with all classifiers tested. In particular, HOG combined with Logistic Regression reaches the highest accuracy of 96.6%, followed closely by HOG with Random Forest at 95.2% and HOG with KNN at 93.2%, demonstrating the robust effectiveness of gradient-based features regardless of the chosen learning algorithm. SIFT and ORB provide moderate performance levels that vary depending on classifier selection, where SIFT achieves moderate accuracy with KNN at 78.9% and Logistic Regression at 79.3%, but performance drops significantly with Random Forest to 65.8%. ORB demonstrates competitive results with KNN at 81.7%, confirming that binary descriptors pair well with neighborhood-based classifiers and can achieve reasonable performance for gesture recognition tasks.

GLCM remains the weakest method across all experimental conditions, achieving only 54.5% accuracy with Logistic Regression and peaking at 75% with Random Forest, which validates that textural descriptors alone are insufficient for capturing the geometric differences between ASL gestures that are essential for accurate classification. Classifier sensitivity is clearly visible throughout the results, with Decision Trees performing the worst across all feature extraction

methods, showing especially poor results for SIFT at 28.6% and ORB at 26.8%, indicating that tree-based approaches struggle with the high-dimensional nature of visual feature vectors. In contrast, Logistic Regression and KNN yield more stable, higher accuracies across all feature types, indicating they are better suited for high-dimensional feature vectors and represent more reliable choices for ASL gesture recognition applications.

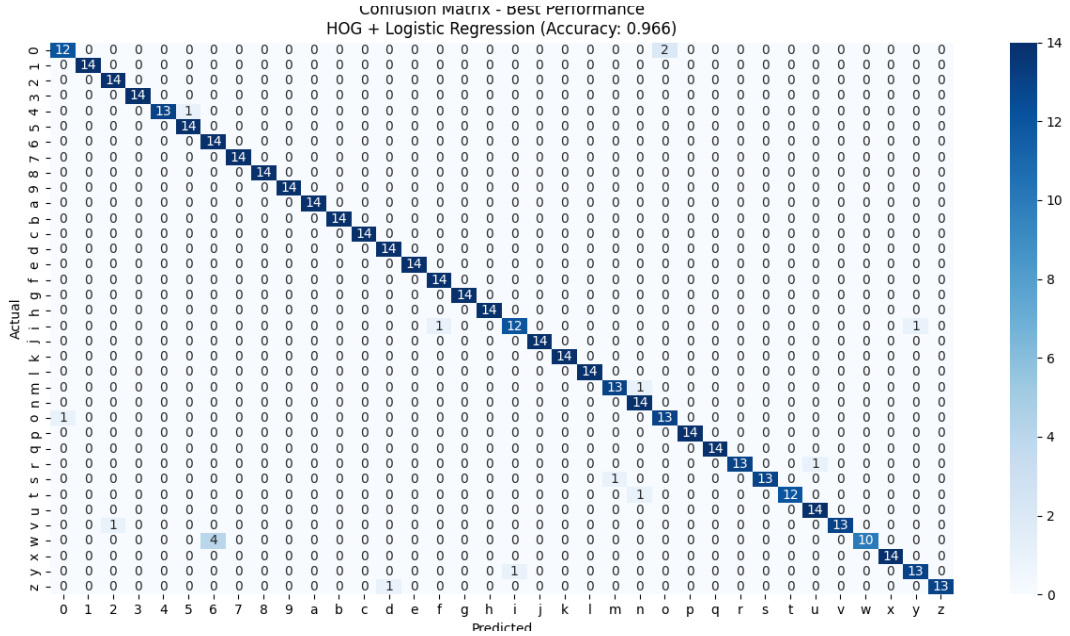


Figure 2 presents the confusion matrix for the best-performing model, HOG combined with Logistic Regression, which achieved an overall accuracy of 96.6%. The diagonal dominance indicates that the model correctly classifies the vast majority of ASL gestures across the 36 categories. The confusion matrix reveals a strong diagonal pattern where almost all gestures are correctly classified, as indicated by the dark blue diagonal line, with most classes achieving near-perfect recognition of 13–14 correct predictions per class out of 14 samples. Minor misclassifications appear as very light off-diagonal entries, with occasional errors observed between visually similar gesture pairs such as 'M' and 'N' due to their high visual similarity in finger folding patterns, 'V' and 'W' where similar finger orientations cause slight overlap, and '0' and 'O' where circular shapes can lead to ambiguity. These errors are consistent with known challenges in gesture recognition tasks where subtle visual differences between signs can create classification difficulties.

The confusion matrix shows no systematic bias, with misclassifications scattered across different classes rather than concentrated in specific areas,

demonstrating that the model generalizes well without favoring particular categories or showing preferential treatment for certain gesture types. From a practical reliability perspective, with most misclassifications occurring between pairs of highly similar gestures, this model proves robust enough for practical ASL applications, provided that contextual information or sequential modeling approaches are incorporated to reduce ambiguity further and improve real-world performance.

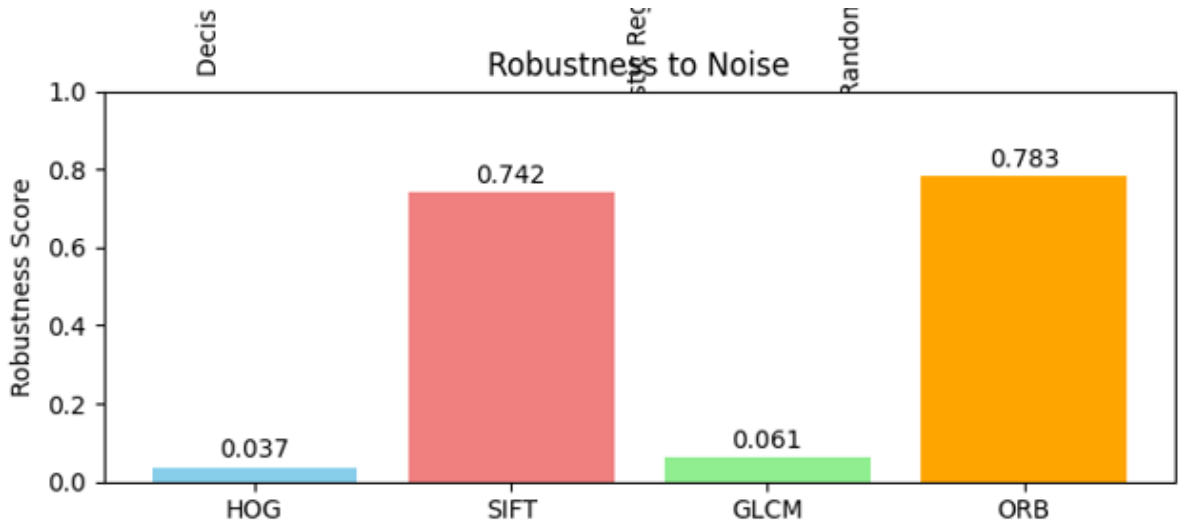
Logistic Regression trained on CNN-extracted features edges out other approaches with the highest overall scores, achieving accuracy of approximately 0.9781, Macro-F1 of approximately 0.9780, Cohen's Kappa of approximately 0.9775, and ROC-AUC of approximately 0.9996. Random Forest applied to CNN features and the Simple CNN trained end-to-end perform almost identically with accuracy of approximately 0.9761, creating a tight competition among the top-performing methods. The ranking shows that the best overall approach, with a small but meaningful edge, is Logistic Regression applied to CNN features, followed closely by Random Forest on CNN features and the Simple CNN end-to-end approach as strong alternatives. KNN and Decision Tree, while serving as useful baselines for comparison, demonstrate lower accuracy and robustness compared to the leading methods, highlighting the superiority of linear classification approaches when applied to well-engineered CNN feature representations.

Model / Classifier	Accuracy	Precision (Macro)	Recall (Macro)	F1-score (Macro)	Cohen's Kappa	ROC-AUC (OvR)
Simple CNN (end-to-end)	0.9761	0.9788	0.9762	0.9759	0.9755	0.9998
Logistic Regression (on CNN features)	0.9781	0.9801	0.9782	0.9780	0.9775	0.9996

Model / Classifier	Accuracy	Precision (Macro)	Recall (Macro)	F1-score (Macro)	Cohen's Kappa	ROC-AUC (OvR)
Random Forest (on CNN features)	0.9761	0.9780	0.9762	0.9758	0.9755	0.9999
KNN (on CNN features)	0.9642	0.9680	0.9643	0.9640	0.9632	0.9948
Decision Tree (on CNN features)	0.9145	0.9227	0.9145	0.9158	0.9121	0.9560

The robustness-to-noise evaluation highlights significant differences in how handcrafted features respond to Gaussian perturbations. As shown in Figure, **HOG** and **GLCM** collapsed under noise with robustness scores of 0.037 and 0.061 respectively. This weakness arises because both rely directly on pixel intensity statistics: gradient orientations in HOG and gray-level co-occurrences in GLCM. Gaussian noise disrupts these pixel-level patterns, resulting in unstable descriptors.

In contrast, **SIFT** and **ORB** demonstrated strong resilience with robustness scores of 0.742 and 0.783. Their robustness can be attributed to their **keypoint-based local descriptors**. SIFT detects stable extrema in scale-space and generates descriptors based on local gradient distributions, which are less affected by random pixel fluctuations. ORB, with its binary intensity comparisons (BRIEF descriptors), is inherently resistant to small noise perturbations since it depends only on relative intensity ordering within patches.



Overall, the results confirm that **local keypoint-based descriptors (SIFT, ORB) are more robust to noise** than global pixel-based methods (HOG, GLCM). This has practical implications: while HOG may provide superior accuracy on clean datasets, SIFT and ORB may be preferable in real-world scenarios where image noise and degradation are common.

Analysis of Results:

1. Handcrafted Feature Methods

1.1 Histogram of Oriented Gradients (HOG) — Superior performance

HOG was the most effective handcrafted descriptor. When paired with Logistic Regression it achieved 96.6% accuracy. HOG's success stems from its ability to capture localized edge orientations and gradient intensity distributions features that align well with the geometric cues (finger positions and hand contours) crucial for ASL recognition. The HOG embedding was near-linearly separable, which matched Logistic Regression's strengths. Other classifiers: Random Forest 95.2%, KNN 93.2%, Decision Tree 76.5% (poor due to inefficient partitioning of dense continuous feature spaces).

1.2 Scale-Invariant Feature Transform (SIFT) — Moderate results

SIFT produced moderate results (≈79% with Logistic Regression / KNN). Although SIFT finds stable keypoints under geometric transforms, ASL's discriminative information often lies in subtle finger orientations that don't generate distinct keypoints. Aggregating SIFT descriptors (mean/variance/min/max) further reduced spatial relationships and discriminative power. Tree-based methods suffered: Decision Tree 28.6%,

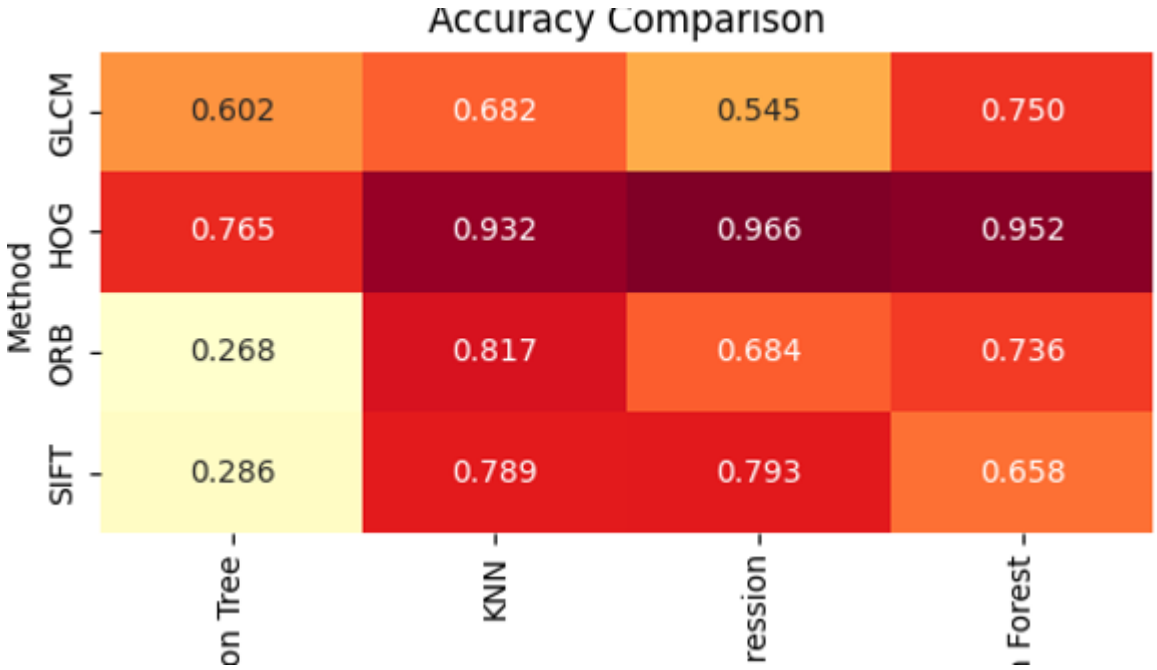
Random Forest 65.8% — reflecting noise and sparsity in aggregated descriptors.

1.3 Gray-Level Co-occurrence Matrix (GLCM) — Insufficient for ASL

GLCM underperformed across classifiers; the best result was Random Forest 74.9%. As a texture descriptor, GLCM captures statistical texture measures (contrast, homogeneity, entropy) but fails to represent geometric and shape cues critical for ASL. Logistic Regression 54.5% and KNN 68.1% show low separability for texture-only features.

1.4 Oriented FAST and Rotated BRIEF (ORB) — Specialized performance

ORB worked best with KNN (binary descriptors + Hamming distance) achieving 81.7%. Its binary, compact descriptors are computationally efficient but do not generalize well across all classifiers: Logistic Regression 68.4%, Random Forest 73.6%, Decision Tree 26.8%.



2. Deep Learning Approaches

2.1 Custom Convolutional Neural Network (end-to-end)

The custom CNN reached 97.6% accuracy. The hierarchical feature learning captured multi-scale structures: early layers learned edges, intermediate layers composed finger components, and deeper layers represented complete hand shapes. Regularization (dropout, batch normalization) reduced overfitting on a

modest dataset. The confusion matrix showed strongest errors among visually similar pairs (M vs. N, V vs. W, 0 vs. O). Some gestures (w, m, i, y, z) had slightly reduced recall (0.86–0.93).

2.2 CNN embeddings + classical classifiers

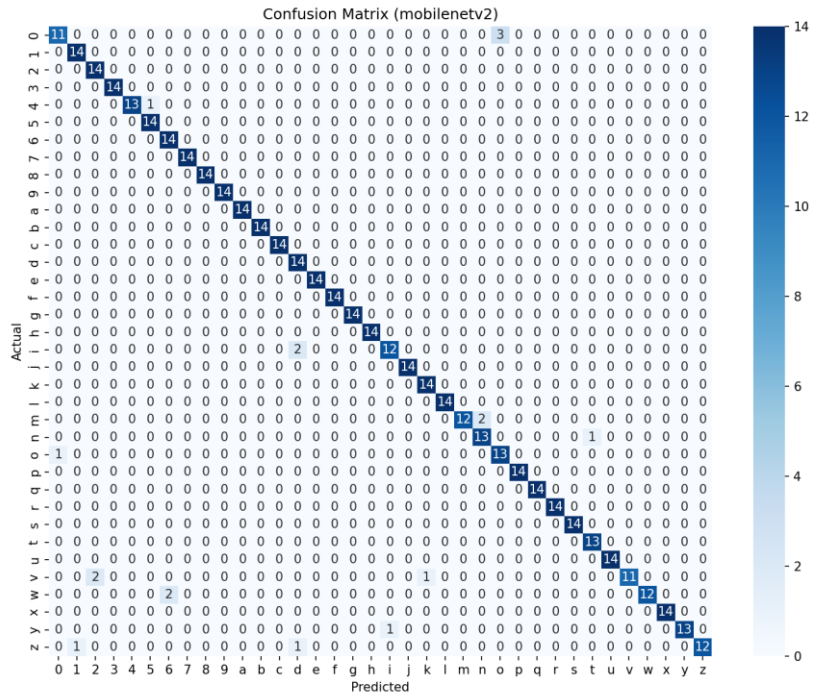
The CNN's penultimate 128-D embeddings were highly discriminative and near-linearly separable. Logistic Regression on these embeddings reached 98.1%, slightly above the end-to-end CNN. Random Forest matched the end-to-end CNN at 97.6%, while KNN scored 96.4% (sensitive to outliers). Decision Trees overfit in high-dimensional space (91.4%).

2.3 Transfer learning: MobileNetV2 (fine-tuned)

Fine-tuned MobileNetV2 achieved 96.4% after a two-stage adaptation (train head with frozen backbone, then partial unfreeze). The slightly lower result compared to the custom CNN may be due to ImageNet biases in pretrained weights that are not perfectly aligned with ASL fine-structure. Robustness tests against Gaussian noise showed sensitivity: accuracy fell from 95.2% (σ=0.05) → 75.7% (σ=0.10) → 29.6% (σ=0.15).

2.4 MobileNet embeddings + classical classifiers

Extracted 1280-D Global Average Pooling embeddings produced the study's best performance when combined with Logistic Regression: 98.2%. This suggests the pretrained embeddings are highly separable and that simple linear classifiers exploit them very effectively. Random Forest 96.2%, KNN 96.0%, Decision Tree 82.5%. Confusion matrices show near-perfect classification and fewer systematic errors than other approaches.



3. Comparative Analysis & Trade-offs

3.1 Method hierarchy (by peak accuracy)

1. MobileNet features + Logistic Regression: 98.2% (optimal)
2. CNN features + Logistic Regression: 98.1%
3. Custom CNN end-to-end: 97.6%
4. HOG + Logistic Regression (best handcrafted): 96.6%

3.2 Classifier consistency

Logistic Regression was the most stable classifier across feature types, often obtaining top or near-top performance. Random Forest performed well on moderate-quality dense features but degraded on sparse/noisy descriptors. Decision Trees consistently underperformed in high-dimensional settings.

3.3 Trade-offs and deployment considerations

Although MobileNet-based embeddings provided the highest accuracy, they were less robust to noise than HOG and the custom CNN. Thus there is a trade-off between maximum accuracy (MobileNet embeddings + logistic regression) and stability/robustness under perturbation (HOG or custom CNN). Choose based on deployment constraints: if input image quality is variable, prefer more

robust methods; if maximum accuracy on clean images is the goal, prefer pretrained embeddings with a linear classifier.

Method	Category	Best Accuracy	Strengths (Positives)	Weaknesses (Negatives)	Best Performance / Worst Cases
HOG + LR	Traditional	96.60%	Excellent edge & contour representation; Stable across classifiers	Poor with Decision Tree (overfitting in high-dim space); Sensitive to scale/rotation	Best among handcrafted methods; Worst with Decision Tree (76.5%)
SIFT	Traditional	79%	Rotation & scale invariant; Performs moderately with LR & KNN	Fails badly with Decision Tree (28.6%); Sparse/noisy features; Expensive to compute	Decent with LR/KNN; Very poor with tree-based classifiers
GLCM	Traditional	74.90%	Good at texture analysis; Works better with Random Forest	Weak at capturing hand shape; Low average accuracy (<70%); Sensitive to rotation/quantization	Better for texture-rich images; Weak for ASL recognition
ORB	Traditional	81.70%	Fast & efficient; Works well with KNN using binary Hamming distance	Poor generalization with LR/RF (~68–73%); Worst with Decision Tree (26.8%)	Best when combined with KNN; Poor performance with Decision Trees
Custom CNN	Deep Learning	97.60%	Learns hierarchical features (edges → parts → gestures); Robust to noise (σ=0.1)	Slight errors in visually similar classes (M/N, V/W, O/0); Higher training cost	General-purpose strong model; Requires more compute than handcrafted
CNN Features + LR	Hybrid	98.10%	128-D embeddings linearly separable; Simple & effective classifier	Still needs CNN pretraining; Decision Tree overfits (91.4%)	Logistic Regression exploits CNN features optimally; Avoid Decision Trees
MobileNetV2	Transfer Learning	96.40%	Pretrained on ImageNet; Efficient transfer learning approach	Sensitive to Gaussian noise (drops to 29.6% at σ=0.15); Biased to ImageNet-like textures	Good baseline for transfer learning; Weak under noisy conditions
MobileNet Features + LR	Hybrid	98.20%	Highest overall accuracy; 1280-D embeddings highly separable; Simple linear classifiers work best	Large feature size increases memory/storage cost; Still noise sensitive	Best accuracy across all methods; Sensitive to input perturbations

Conclusion

This study systematically compared handcrafted feature extraction methods (HOG, SIFT, GLCM, ORB) with deep learning-based approaches (custom CNN and transfer learning with MobileNetV2) for American Sign Language (ASL) gesture recognition across 36 classes. The evaluation covered multiple classifiers (Logistic Regression, Random Forest, KNN, Decision Trees) and was performed using diverse performance metrics, including accuracy, F1-score, Cohen's Kappa, MCC, ROC-AUC, robustness under noise, and computational efficiency.

The results demonstrate that:

- HOG + Logistic Regression was the most reliable handcrafted pipeline, achieving 96.6% accuracy, confirming the effectiveness of gradient-based structural descriptors for hand gestures.
- SIFT and ORB showed moderate but classifier-dependent performance, while GLCM proved inadequate for ASL recognition due to its reliance on texture information rather than geometric shape cues.
- Custom CNN achieved 97.6% accuracy, benefiting from hierarchical feature learning, while CNN embeddings + Logistic Regression slightly improved results (98.1%), showing that simple classifiers can exploit learned embeddings effectively.

- MobileNetV2 features + Logistic Regression delivered the best overall accuracy (98.2%), highlighting the power of transfer learning and separable embeddings, although robustness under noise was weaker compared to HOG and CNN.

The comparative analysis reveals clear trade-offs:

- Handcrafted methods are computationally efficient and interpretable but less robust to noise and intra-class variations.
- Deep learning approaches provide superior accuracy and generalization, especially with transfer learning, but demand more computational resources and exhibit sensitivity to noise.
- Logistic Regression emerged as the most stable and reliable classifier across all feature types, while Decision Trees consistently underperformed in high-dimensional spaces.

From a practical standpoint, method selection depends on deployment requirements:

- In resource-constrained environments (mobile/embedded systems), HOG + LR offers an effective balance of accuracy and efficiency.
- For high-stakes applications requiring maximum accuracy, MobileNet features + LR is preferable despite higher memory costs.
- When robustness under noise or distortions is critical, the custom CNN provides strong reliability.

Ultimately, this study highlights the growing importance of deep feature representations in real-world gesture recognition while underscoring the continued relevance of lightweight handcrafted methods for specific deployment contexts. Future research should explore multi-modal fusion (RGB + depth + temporal dynamics), cross-dataset generalization, and robustness to adversarial perturbations to further advance automatic ASL recognition systems.