Archives available at journals.mriindia.com

International Journal on Advanced Electrical and Computer Engineering

ISSN: 2349-9338

Volume 15 Issue 01s, 2026

Comparison of Machine Learning Models for Hand Sign Recognition

[1]Abhishek Shrikant Patil, [2]Ranjana Sitaram Zinjore

[1,2] MTES's Smt. G. G. Khadse College, Muktainagar, Maharashtra, India

Emails: [1] abhishek.patil.1987@gmail.com, [2] rszinjore14@gmail.com

Peer Review Information

Submission: 05 Dec 2025

Revision: 25 Dec 2025

Acceptance: 10 Jan 2026

Keywords

Indian Sign Language (ISL), Support Vector Machine (SVM),K-Nearest Neighbors (KNN), Radial Basis Function(RBF) Kernel, Histogram of Gradients (HOG), Hand Sign Recognition, Feature Extraction.

Abstract

Hand sign recognition plays a crucial role in the development of assistive technologies for human–computer interaction and communication support for persons with hearing and speech impairments. This paper presents a comparative study of machine learning models for static hand sign recognition using a publicly available Indian Sign Language (ISL) dataset downloaded from Kaggle. A unified preprocessing pipeline was applied to all images, followed by extraction of three complementary feature sets: Hu Moments for capturing global shape properties, contour-based shape features for structural hand geometry, and Histogram of Oriented Gradients (HOG) for local gradient-based texture information. These features were combined into a single feature vector and used to train two major classifier families: K-Nearest Neighbors (KNN) with different values of *k*, and Support Vector Machines (SVM) with multiple kernels, including Linear, Polynomial, and Radial Basis Function (RBF). The performance of each model was evaluated using accuracy, precision, recall, and F1-score to identify the most effective classifier for multi-class hand sign recognition. Experimental results show that SVM with the RBF kernel consistently achieves superior classification accuracy compared to KNN and other SVM variants, demonstrating its suitability for high-dimensional feature representations in hand gesture recognition systems. The findings provide useful insights for selecting efficient machine learning models for real-time sign recognition applications.

Introduction

Indian Sign Language (ISL) is very important for communication among people who have hearing and speech disabilities in India. With the growing use of digital technology, automatic hand gesture recognition systems are becoming more useful, as they can help reduce communication problems. However, recognizing ISL hand gestures correctly is still difficult. This is because hand shapes, hand movements, lighting conditions, and backgrounds change from image to image. Recent progress in image processing and machine learning has made it possible to improve gesture recognition systems. In this research work, an effective preprocessing and feature extraction method is developed, and different classifiers are tested for recognizing ISL alphabets. The main goal of this study is to achieve high accuracy and to develop a system that can be used in real and practical sign language applications.

Literature Review

Research in hand sign recognition has progressed significantly with the integration of image processing, feature engineering, and machine learning approaches. Most systems begin with preprocessing operations such as image resizing, Gaussian smoothing, HSV/YCbCr-based skin segmentation, background subtraction, and binarization to isolate the hand region and reduce noise prior to feature extraction. Traditional studies emphasize the use of handcrafted features to represent gesture shape and structure. Widely adopted techniques include Histogram of Oriented Gradients (HOG), which captures fine-grained edge orientations and local gradients; Local Binary Patterns (LBP) for encoding texture variations; Hu Moments, which provide global shape descriptors invariant to scale and rotation; and contour-based features, such as convex hull, centroid distances, and bounding geometry, which effectively represent hand silhouettes. These handcrafted features have been employed in several earlier works for gesture representation due to their simplicity and robustness in controlled environments. For classification, researchers have used a variety of machine learning algorithms, with Support Vector Machines (SVM) and K-Nearest Neighbors (KNN) being among the most common choices. SVM has been used effectively for gesture classification because of its ability to handle high-dimensional feature sets and non-linear decision boundaries, particularly when using polynomial and RBF kernels [1], [7]. Similarly, KNN has been applied for recognizing static hand gestures, benefiting from its simplicity and strong performance on feature-rich datasets [8]. Several earlier works also employed Artificial Neural Networks (ANNs), especially for fingertip-based and geometric features [10], while others used multilayer perceptron networks for continuous gesture sequences [8]. More recent studies have transitioned towards deep learning architectures, predominantly Convolutional Neural Networks (CNNs), due to their capability to automatically learn hierarchical features directly from raw images. Works such as those by Meshram et al. [3], Deshpande et al. [4], and Flores et al. [5] demonstrate that CNN-based models achieve significantly higher accuracy compared to handcrafted methods. Transfer learning approaches using pretrained models like GoogLeNet and VGG16 further improve performance through feature reuse and enhanced generalization [2], [3]. These methods consistently report accuracies above 95%, highlighting the superiority of deep learning for large and diverse gesture datasets. Despite these advancements, literature also indicates certain gaps. Several studies rely on controlled environments with uniform backgrounds and lighting conditions [4], which limits model robustness. Many datasets contain a limited number of gestures or exclude dynamic signs such as "J" and "Z," reducing the applicability of these systems in real-world ISL contexts. Furthermore, motion-based features and temporal modeling remain underexplored in comparison to static gesture recognition. Overall, the literature shows that robust preprocessing, discriminative feature extraction, and the selection of an appropriate classifier are critical for achieving high recognition accuracy. While CNN-based systems currently dominate performance benchmarks, handcrafted feature approaches using HOG, LBP, Hu Moments, and contour-based descriptors combined with SVM or KNN remain relevant for lightweight, low-cost implementations. Mengmeng Han et al. [9] employed background subtraction, a Gaussian Mixture Model, and binarization for preprocessing. Using a CNN on a dataset of 10 hand gestures, their approach achieved an accuracy of 93.8%. A. Sharmila Konwar et al. [11] created their own dataset and resized the images. Skin detection was performed using the HSV color model along with a skin pixel detection algorithm. Noise was reduced using a Gaussian filter, and edges were detected using Canny and Sobel operators. Their method successfully recognized five ASL alphabets, while other gestures could not be detected due to variations in hand shape, uneven backgrounds, and lighting conditions. Felix Zhan [13] created a dataset of 9 hand gestures and used skin pixel extraction, grayscale conversion, and image resizing for preprocessing. A CNN model trained with batch normalization, spatio-temporal augmentation, and SGD optimization achieved 98.76% test accuracy and 98.2% validation accuracy, though the study was limited to only nine gestures.

Proposed Architecture

The architecture of the proposed hand sign recognition system, as depicted in Figure. 1, follows a structured pipeline designed to ensure robust and accurate gesture classification. The process begins with the **input dataset**, consists of hand gesture images captured under different lighting, pose, and background conditions. These images are carefully preprocessed through steps like resizing, converting to grayscale, removing the background, segmenting skin areas, filtering out noise, and enhancing contrast to make the input more consistent and reduce differences within the same class.

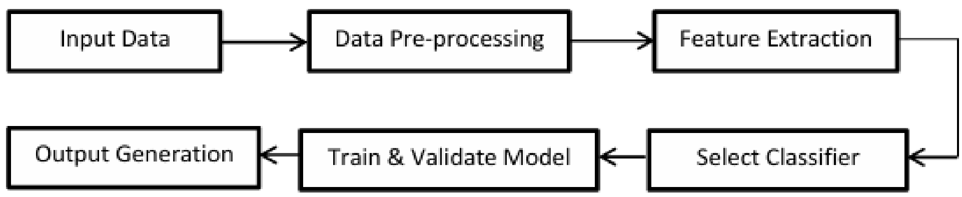

Figure 1. System architecture illustrating the sequential stages of the hand sign recognition framework

After preprocessing, the images are passed on to feature extraction stage, where key characteristics like edges, shapes, and contours are identified. These features are then used by a machine learning classifier to distinguish between different hand signs. The model is trained and validated to ensure good performance and to avoid overfitting. Finally, the trained model predicts the correct gesture label for each new image. This process provides an effective way to recognize static hand signs in practical applications.

Methodology

Dataset Collection

The dataset used in this study consists of Indian Sign Language (ISL) hand gesture images created for alphabet recognition. There are about 26,000 images in total, with 1,000 images for each letter from A to Z. Each image in the dataset is 250×250 pixels and includes a range of backgrounds, lighting conditions, and hand orientations, making the collection both diverse and realistic[13]. The images were captured using a 720p laptop camera to mimic real-world settings where lighting and backgrounds can vary. All hand gestures were carefully demonstrated and referenced from the official ISLRTC website to ensure accuracy and authenticity. Additionally, the dataset includes augmented variations in angle, lighting, and background to improve robustness and help models generalize across different conditions. This well-structured and thoughtfully created dataset provides a strong foundation for developing and evaluating preprocessing, feature extraction, and classification techniques for Indian Sign Language recognition systems.

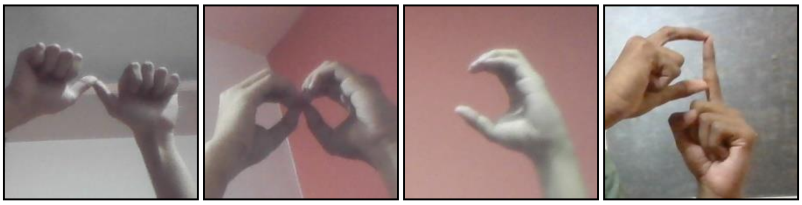

Figure 2. Sample Images from Dataset

Preprocessing

The preprocessing stage is crucial for preparing Indian Sign Language (ISL) hand gesture images before moving on to feature extraction and model training. The process starts with cropped gesture images, neatly organized into folders for each letter from A to Z. Because these images can look quite different due to changes in lighting, busy backgrounds, and variations in skin tone, a combination of enhancement techniques is used to improve image quality and consistently isolate the hand. Each image is resized to 128×128 pixels and converted to grayscale to reduce dimensionality while preserving structural information. Hand segmentation is performed in the YCrCb color space, which separates luminance from chrominance and provides robust skin detection under diverse lighting conditions—an approach also validated in prior work using HSV and YCbCr segmentation [8]. Morphological operations refine the mask by removing noise and closing small gaps. If the skin detection isn't reliable, Otsu's thresholding method is used as a backup to separate the hand from the background based on differences in brightness. The resulting mask is used to clearly isolate the hand area. To make important features stand out, the image is enhanced using Contrast Limited Adaptive Histogram Equalization (CLAHE), and any remaining noise is smoothed out with adaptive Gaussian filtering. The processed image is then normalized to a [0,1] range and converted to three channels so it can be used with models that expect RGB images. The final output is a set of gesture images and corresponding binary masks, all neatly organized by letter. This preprocessing pipeline ensures images have consistent contrast, minimal background distractions, and clear hand outlines, making it much easier to extract features and classify gestures accurately in the next steps.

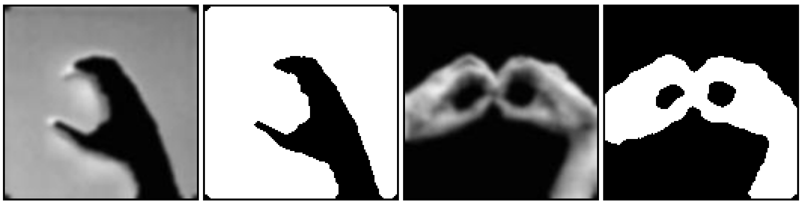

Figure 3. Grayscale and masked images after preprocessing

Feature Extraction

Feature extraction is a key step in the Indian Sign Language (ISL) hand gesture recognition system. In this stage, each preprocessed image is turned into a set of numbers that the model can use to recognize gestures. These features capture important details about the hand's shape, structure, and texture, making it easier for the system to tell different gestures apart.. In this work, three complementary feature categories were used: Hu Moments, Contour-Based Shape Features, and Histogram of Oriented Gradients (HOG). These features were concatenated to form a comprehensive feature vector for each image.

Hu Moments

The Hu moments are derived from the geometric image moments and represent invariant shape descriptors that remain unchanged under rotation, translation, and scaling. A total of seven Hu moments were extracted for each grayscale image. These moments effectively capture the overall hand shape and spatial distribution of intensity patterns within the gesture.

Normalized Central Moment

$$\eta pq = \mu pq/\mu_{00}^{\gamma}, \quad \gamma = \frac{p+q}{2} + 1 \qquad 1)$$

Central Moment

$$\mu pq = \sum y \sum (x - \bar{x})p(y - \bar{y})\, qI(x,y) \qquad 2)$$

Centroid

$$\bar{x} = M10 / M00 , \bar{y} = M01/M00 \qquad 3)$$

Where

μpq: central moment of order p+q
ηpq: normalized central moment
p,q: moment orders (non-negative integers)

Contour-Based Shape Features

Identifying the largest contour corresponding to the hand region, a set of twelve geometric and structural shape features was extracted. These include the contour area, perimeter, bounding box width and height, aspect ratio, extent (ratio of contour area to bounding box area), solidity (ratio of contour area to convex hull area), number of corners (via polygonal approximation), convexity defects count, eccentricity (from an ellipse fit), normalized area, and bounding box area

Figure 4. Sample images of contour based shape features

Contour Area

$$A=Area(C) \qquad 4)$$

Where

A: total area enclosed by the contour

C: hand contour (set of boundary points)

Contour Perimeter

$$P=\mathrm{ArcLength}(C,\mathrm{true}) \quad 5)$$

Where

P: contour perimeter or boundary length

C: contour points

true: indicates a closed contour

Aspect Ratio

$$AR=h/w \quad 6)$$

Where

AR: ratio of width to height

w: width of the bounding rectangle

h: height of the bounding rectangle

Extent

$$\mathrm{Extent}= A / w\times h \quad 7)$$

Where

Extent: proportion of the bounding box filled by the hand

A: contour area

w×h: area of the bounding box

Solidity

$$\mathrm{Solidity}= A / \mathrm{Ahull} \quad 8)$$

Where

Solidity: measure of shape compactness

A: contour area

Ahull: area of the convex hull surrounding the contour

Equivalent Diameter

$$\mathrm{Deq} = \sqrt{4A/\pi} \quad 10)$$

Where

Deq: diameter of a circle with the same area as the hand

A: contour area

π: constant Pi (3.14159)

Eccentricity

$$\mathrm{Eccentricity} = \sqrt{1-(b/a)2} \quad 11)$$

Where

Eccentricity: elongation measure of the fitted ellipse

a: semi-major axis length

b: semi-minor axis length

Circularity (Roundness)

$$\mathrm{Circularity} = 4\pi A/P^2 \quad 12)$$

Where

Circularity: measure of how close the contour is to a perfect circle

A: contour area

P: contour perimeter

π: constant Pi (3.14159)

Histogram of Oriented Gradients (HOG) Features

To capture the fine-grained edge and texture patterns, Histogram of Oriented Gradients (HOG) features were computed from each grayscale image. HOG encodes local intensity gradients by dividing the image into cells and accumulating gradient direction histograms.

Figure 5. Sample images of HOG features

The HOG descriptor is generated by first computing the image gradients in the horizontal and vertical directions using simple derivative filters:

$$Gx = I(x+1,y) - I(x-1,y), \qquad Gy = I(x,y+1) - I(x,y-1) \tag{13}$$

where
I(x,y) is the pixel intensity at location (x,y),
Gx and Gy represent the horizontal and vertical gradient components.

The gradient magnitude and orientation at each pixel are then calculated as:

$$M(x,y) = \sqrt{\{G_x^2 + G_y^2\}} \qquad \theta(x,y) = \tan^{-1}\left(\frac{Gx}{Gy}\right) \tag{14}$$

where
M(x,y) denotes the strength of the gradient,
θ(x,y) is the orientation (angle) of the gradient.
For each cell, a histogram of gradient orientations is constructed:

$$H(b) = \Sigma_{\{(x,y)\in \text{cell}\}} M(x,y) \tag{15}$$

where
H(b) is the histogram value for orientation bin b for given cell
Σ :Sum of gradient magnitudes of all pixels whose gradient orientation falls into bin b
M(x, y): Gradient magnitude at pixel (x,y)
(x,y) ∈ Cell All pixel locations (x,y) that lie inside the current HOG cell.
HOG Feature Vector Normalization

$$v\_norm = v \,/\, (\|v\|_2 + \varepsilon) \tag{16}$$

Where
v is the unnormalized HOG feature vector,
||v||2 is its L2-norm,
ϵ is a small constant to avoid division by zero.

Combined Feature Vector
The final feature vector for each image is formed by concatenating all three feature groups — Hu Moments (7), Contour-Based Shape Features (12), and HOG Features (≈1760) — resulting in a feature vector of approximately 1779 dimensions. This hybrid feature representation captures both global shape characteristics and local structural details, ensuring high discriminative power across gesture classes. The extracted feature vectors were standardized and provided as input to the classifiers

Table 1. Combined Feature Vector count

Feature Group	Feature Name / Description	Count	Purpose / Interpretation
Hu Moments	7 invariant statistical moments derived from image moments (log-scaled). Invariant to rotation, scale, and translation.	7	Capture global shape and spatial distribution of the hand region.
Contour-based Shape Features	Contour area, perimeter, bounding box width and height, aspect ratio, extent (area/bounding box area), solidity (area/convex hull area), number of contour corners, number of convexity defects, eccentricity (from fitted ellipse), normalized area, and bounding box area.	12	Quantify geometric and structural properties of the hand, finger spread, and palm shape.
HOG (Histogram of Oriented Gradients)	Gradient-based local feature descriptor capturing edge and contour orientation patterns.	N (depends on cell and block size)	Encodes fine-grained local texture and edge orientation information useful for distinguishing hand postures.

Total Features per Image	Hu (7) + Shape (12) + HOG (≈ 1764 in current setup)	≈ 1783	Combined descriptor used for classification with the SVM (Polynomial kernel).

Classification
To assess the performance of the proposed feature extraction approach, several machine learning classifiers were applied to the same feature dataset generated from the preprocessed ISL hand gesture images. Three versions of the Support Vector Machine (SVM) algorithm were used — **Linear SVM**, **Polynomial SVM**, and **RBF SVM** — allowing the evaluation of different kernel functions on a uniform feature set. This ensured that variations in accuracy were purely due to classifier behavior and not differences in the input data. Along with SVM, the **K-Nearest Neighbors (KNN)** classifier was also employed with **K = 3** and **K = 5** to study how neighborhood size influences recognition performance. Using multiple classifiers provided a comprehensive understanding of how effectively the extracted features represent hand gestures and how well they support accurate classification across different learning models.

SVM (RBF kernel)

The RBF SVM employs a Gaussian radial basis function to model highly nonlinear decision boundaries, making it well-suited for gesture datasets with large variability in shape and pose. It generally offers the best balance of accuracy and generalization across diverse feature sets.

SVM (Linear Kernel)

The Linear SVM uses a straight hyperplane to separate gesture classes and is effective when the feature space is linearly separable. It provides fast training and good performance for high-dimensional feature vectors such as HOG and LBP.

SVM (Polynomial Kernel)

The Polynomial SVM maps input features into a higher-degree space, enabling the classifier to capture more complex relationships between gesture patterns. It is useful when class boundaries are nonlinear but still require controlled model complexity.

K-Nearest Neighbors (KNN)

The K-Nearest Neighbors classifier operates by identifying the k closest feature vectors to a test sample in the feature space and assigning the class based on majority voting. KNN is a non-parametric, distance-based method that performs well for gesture recognition when using discriminative features such as HOG, Hu Moments, and contour-based descriptors. In this study, K values of 3 and 5 were evaluated to analyze the effect of neighborhood size on classification accuracy and decision stability.

Results And Discussions
In this study, two supervised machine learning classifiers—Support Vector Machine (SVM) and K-Nearest Neighbors (KNN)—were employed for hand sign recognition. The SVM classifier was implemented with different kernel functions, including linear, polynomial, and radial basis function (RBF), to evaluate their ability to handle linear and non-linear feature separations. The RBF kernel demonstrated superior generalization due to its capacity to map input features into a higher-dimensional space, effectively separating complex gesture patterns. The KNN classifier was tested with multiple values of K to determine the optimal neighborhood size for accurate classification. KNN, being a distance-based model, classified samples by comparing feature similarities within the dataset. The comparative analysis of these classifiers helped identify the most effective approach for recognizing Indian Sign Language (ISL) gestures from extracted shape and texture-based features.

Table 2. Results of SVM Classifier

Metric	SVM (RBF kernel)	SVM (Linear kernel)	SVM (Poly kernel)
Total Samples	52,000 (26 classes × 2,000 images each)	52,000 (26 classes × 2,000 images each)	52,000 (26 classes × 2,000 images each)
Train-Test Split	41,600 train / 10,400 test	41,600 train / 10,400 test	41,600 train / 10,400 test
Feature Vector Length	1,783 features per image	1,783 features per image	1,783 features per image

Accuracy	98.62%	98.48%	98.49%
Average Precision	0.9863	0.9848	0.9851
Average Recall	0.9863	0.9848	0.9849
Average f1 score	0.9863	0.9848	0.9850

The performance evaluation of the classifiers is summarized in Table 2 and Table 3, which present the results of SVM with different kernels and KNN with varying values of K. As shown in Table 2, the SVM classifier using the RBF kernel achieved the highest performance, with an accuracy of 98.62%, along with the highest average precision, recall, and F1-score (all 0.9863). The polynomial and linear kernels also performed well but with slightly reduced accuracy values of 98.49% and 98.48%, respectively. These results indicate that the hand gesture dataset is non-linearly separable, and the RBF kernel provides a more effective mapping to higher-dimensional feature space, enabling better discrimination between gesture classes.

Table 3. Results of KNN Classifier

Metric	KNN (K=3)	KNN (K=5)
Total Samples	52,000 (26 classes × 2,000 images each)	52,000 (26 classes × 2,000 images each)
Train-Test Split	41,600 train / 10,400 test	41,600 train / 10,400 test
Feature Vector Length	1,783 features per image	1,783 features per image
Accuracy	97.78%	97.50%
Average Precision	0.98	0.97
Average Recall	0.97	0.98
Average f1 score	0.98	0.98

The results of the KNN classifier, presented in **Table 3**, show that **K = 3** yielded the best performance with an accuracy of **97.78%**, when the value of K was increased to 5, the accuracy dropped slightly to **97.50%**, with corresponding decreases in precision, recall, and F1-score. Although KNN performed competitively but its accuracy remained lower than that of SVM-RBF classifier. This indicates that KNN is sensitive to the choice of neighborhood size, and larger values of K may smooth class boundaries too much in the high-dimensional feature space used in this study.

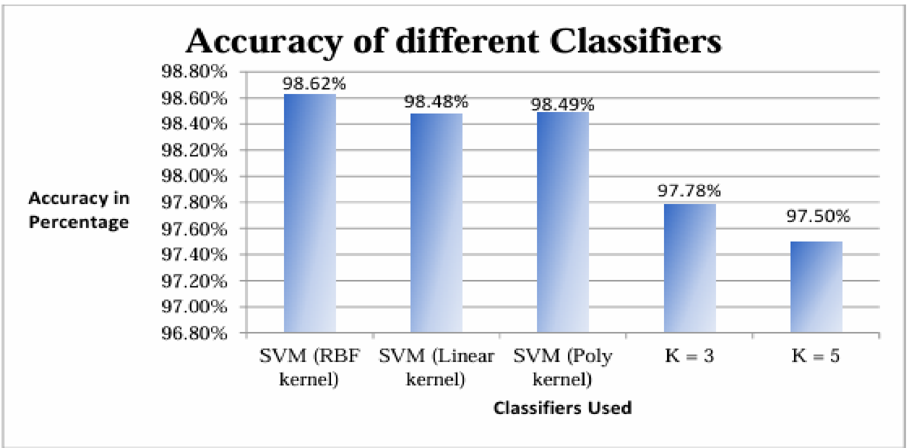

Figure 6. Comparative Graph of different Classifiers.

Overall, based on the comparative analysis in **Table 2 and Table 3**, the results clearly show that SVM with RBF kernel consistently outperforms other SVM kernels and KNN variants, offering the best balance between accuracy and generalization. KNN remains a competitive yet parameter-sensitive classifier, while SVM-RBF demonstrates superior robustness for hand sign recognition using the extracted feature set.

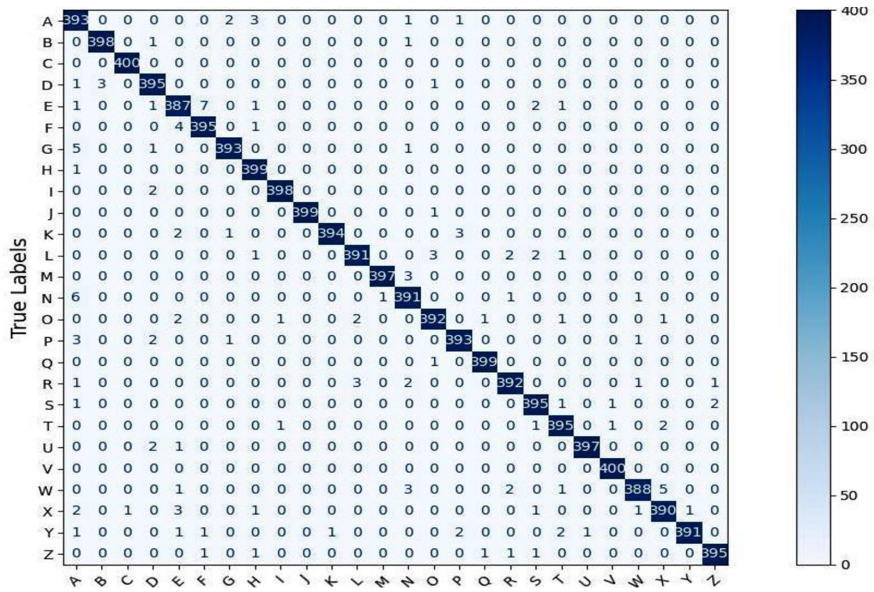

Figure 7. Confusion Matrix of SVM (RBF) classifier

To further validate the classification performance of the SVM model with the RBF kernel, a confusion matrix was generated for all 26 gesture classes. The confusion matrix provides a detailed view of class-wise predictions and helps identify specific classes where misclassifications may occur. The matrix shows that the SVM-RBF achieved **near-perfect classification**, with the diagonal elements consistently holding the highest values, indicating correct predictions for almost every class. Only a very small number of samples were misclassified, and these errors were scattered across a few visually similar hand signs. This aligns with the high accuracy, precision, recall, and F1-score reported in **Table 2**, confirming that the RBF kernel is highly effective in distinguishing subtle variations in hand shape and orientation. The confusion matrix therefore reinforces the robustness and reliability of the SVM-RBF model, demonstrating its suitability for real-world hand sign recognition applications where class-level accuracy is critical.

The Comparative work of researchers on ISL is shown in **Table 4**, but most of the studies are related to **Marathi Sign Language**.

Table 4. Comparison of Results of researchers on ISL and our proposed system

Name of the Researcher	Dataset	Technique Used	Features Extracted	Accuracy
Amitkumar Shinde. et al. [1]	A dataset Marathi of 43 sign language words.	Web camera, Segmentation, HSV color-based hand extraction, and Database Comparison	Center of Gravity (Centroid), Average Height of the sign, and Euclidean Distance	85% (Web-cam/43 signs),
Shalvee Meshram. et. al. [3]	36 hand gestures with 400 images each (total 14,400 images)	Deep Learning using 2D Convolutional Neural Networks (CNN) and Transfer Learning (utilizing the VGG16 pretrained model).	Automated Feature Extraction: The CNN layers automatically extract edges, colors, corners, and complex shapes. Spatial Features: Hand region/Region of Interest (ROI) and blob centroid coordinates for tracking.	99.08% (Proposed system using 2D CNN + VGG16)

Ashwini M. Deshpande. et. al. [4]	25 Marathi alphabets, with 25 samples each from 5 participants.	Video-Based Convolutional Neural Network (CNN) using 5 convolutional layers, pooling layers, and fully connected layers,Implemented using OpenCV and Python,	Automated Feature Construction: The system automatically constructs features from gesture images; the CNN learns the gestures itself, removing the need for manual/handcrafted feature extraction,.	99.28% (Testing accuracy for 25 Marathi sign language alphabets)
G.R.S. Murthy. et. al. [7]	10 hand gestures, including directional and finger-count gestures	Vision-based technique using a web camera. Utilizes a supervised feed-forward neural network with a back propagation algorithm for classification.	Edge counting to identify the thumb and the start of the hand portion. Weighted average (WA) of the relevant finger area and extreme coordinates (bounding box) to find hand direction. Uses a 3D Euclidean Space of binary values.	89% overall correct results on a typical test set
Proposed System	26 English alphabets with 1000 samples of each. Images have varied backgrounds, lighting conditions, and hand orientations	Hand segmentation is performed in the **YCrCb** color space for robust skin detection, Binary mask generation is used to isolate the hand region, CLAHE is applied to enhance contrast, SVM with RBF classifier was used.	The hybrid feature set of Hu Moments, Contour-based shape features, Histogram of Oriented Gradient (HOG)	SVM with RBF kernel achieved overall accuracy of 98.62%

Conclusion

This study presented an extensive evaluation of Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) classifiers for static hand sign recognition using a 26-class Indian Sign Language (ISL) dataset comprising 52,000 preprocessed images. The hybrid feature set—combining Histogram of Oriented Gradients (HOG), Hu Moments, and Contour-based shape descriptors—proved highly discriminative and effective for traditional machine-learning models.

Among all classifiers, the SVM with RBF kernel achieved the highest performance with an accuracy of **98.62%**, supported by strong diagonal dominance in the 26×26 confusion matrix and balanced per-class precision, recall, and F1-scores. Its ability to model complex, non-linear decision boundaries enabled superior handling of variations in hand orientation, finger articulation, background patterns, and intra-class shape differences. The Linear SVM and Polynomial SVM (degree 3) also performed competitively, achieving **98.48%** and **98.49%** accuracy, respectively. These results indicate that the extracted 1783-dimensional feature vector provides a feature space that is mostly linearly separable, with only a few gesture classes benefiting substantially from non-linear mapping.

The KNN classifier also gave good results in this study. When **K = 3**, it achieved an accuracy of **97.78%**, but increasing the value to **K = 5** caused a small drop in accuracy to **97.50%**, showing that KNN is sensitive to the choice of K, especially with high-dimensional features. Although KNN performed well, the **SVM with RBF** kernel achieved the highest accuracy of **98.68%**, making it more reliable for this task. Compared to KNN, SVM-RBF handled variations in the data better and provided more consistent performance.

Per-class performance analysis showed that most gestures achieved **94%–100%** F1-scores, with gestures such as **M, U, V, O, B, C, and Z** consistently surpassing **99%.** Misclassifications were minimal and primarily occurred among visually similar gestures—such as **E–F**, **S–W**, and **W–X**—highlighting inherent structural ambiguities rather than model-specific weaknesses.

Overall, the results conclusively demonstrate that **SVM with RBF kernel is the most robust and reliable classifier** for static ISL hand sign recognition, providing superior generalization and minimal misclassification. KNN serves as a strong baseline model with competitive accuracy, particularly at lower values of *K*, but exhibits greater variability relative to SVM. The study also validates that combining HOG, Hu Moments, and contour features creates a powerful and efficient feature representation for high-accuracy gesture recognition.

References

[1] Amit Kumar Shinde, Ramesh Kagalkar, “Advanced Marathi sign Language Recognition using computer vision”. International Journal of Computer Applications (0975-8887) Vol. 118-No. 13 May 2015.

[2] Brandon Gracia, Sigberto Alarcon Viesca., Real-time American Sign Language Recognition with Convolutional Neural Network.

[3] Shalvee Meshram, Roshan Singh, Prashant Pal, Shashank Kumar Singh, “Convolution Neural Network based Hand Gesture Recognition System” 3rd international conference on advances in electrical, computing , communication and sustainable technologies (ICAECT) – IEEE 978-1-6654-9400-7 – 2023

[4] Ashwini M. Deshpande and Snehal R. Kalbhor, “Video based Marathi sign language recognition and Text conversion using Convolutional Neural Network,” Emerging Trends in Electrical, Communications and Information Technologies proceeding of ICECIT – 2018.

[5] Jose L. Flores C.,E. Gladys Cutipa A.,Lauro Enciso R, “Application of Convolutional Neural Networks for Static Hand Gestures Recognition Under Different Invariant Features” in IEEE 978-1-5090-6363-5/17 2017.

[6] Gummadi, V. P. K. (2021). Streaming in Mule 4: High-volume processing. Journal of Information Systems Engineering and Management, 6(4), 1–9.

[7] G.R.S. Murthy, R.S. Jadon “Hand Gesture Recognition using Neural Networks” in IEEE 978-1-4244-4791-6/10 2010.

[8] Sourav Bhowmick, Sushant Kumar and Anurag Kumar. (2015) IEEE 2nd International conference on Recent Trends in Information Systems (ReTIS)

[9] Mengmeng Han, Jiajun Chen, Ling Li, Yuchun Chang “Visual Hand Gesture Recognition using Convolutional Neural Network” (2016) IEEE 978-1-5090-2239.

[10] Md. Mohiminul Islam, Sarah Siddiqua and Jawata Afnan “Real Time Hand Gesture Recognition Using Different Algorithms Based on American Sign Language” 2017 IEEE 978-1-5090-6004-7/17

[11] A. Sharmila Konwar, B. Sagarika Borah, C. Dr.T.Tuithung “An American Sign Language Detection System using HSV Color Model and Edge Detection,” 2014 IEEE International conference on Communication and Signal Processing

[12] Felix Zhan “Hand Gesture Recognition with Convolution Neural Networks”, IEEE 2019.

[13] Link to the Indian Sign Language Dataset.

[14] https://www.kaggle.com/datasets/atharvadumbre/indian-sign-language-islrtc-referred/data