ACCSCIENCE PUBLISHING

International Journal of Systematic Innovation

ARTICLE

Variability in gesticulation patterns: A robust framework for recognizing self co-articulated dynamic gestures

Shweta Sharda[1]*, Ritu Vyas[1], and Joyeeta Singha[2]

[1]Department of Electronics & Communication, Jaipur Engineering College and Research Center, Jaipur, Rajasthan, India

[2]Department of Electronics & Communication Engineering, The LNM Institute of Information Technology, Jaipur, Rajasthan, India

***Corresponding author:**
Shweta Sharda
(shweta.saboo.y18pg@lnmiit.ac.in)

Citation: Sharda, S., Vyas, R. & Singha, J. (2026). Variability in gesticulation patterns: A robust framework for recognizing self co-articulated dynamic gestures. *Int J Systematic Innovation*. 10(1):19-34.
https://doi.org/10.6977/IJoSI.202602_10(1).0003

Received: February 3, 2025

Revised: October 27, 2025

Accepted: January 5, 2026

Published online: February 13, 2026

Abstract

Dynamic hand gesture recognition has become an important research area in human–computer interaction, virtual reality, sign language interpretation, and intelligent surveillance systems. With the increasing demand for natural and contactless communication interfaces, gesture-based systems are gaining significant attention due to their intuitive and user-friendly nature. However, one of the major challenges in dynamic gesture recognition is inter-user variability, where differences in speed, style, and articulation patterns among users reduce the overall robustness and accuracy of recognition systems. Another critical issue is self co-articulation, which occurs when gestures overlap or influence each other during continuous motion, making feature extraction more complex. This study presents a dynamic hand gesture recognition system that addresses inter-user variability in gesticulation patterns. In our proposed system, a new set of features was employed, which divides the gesture into two halves, and feature extraction was performed after the removal of self-co-articulation. The efficiency of the proposed system was validated on a new set of gestures recorded in the LNM Institute of Information Technology Dynamic Hand Gesture Dataset-4, which consists of videos recorded according to different patterns. The performance of the proposed system was calculated with different features combined with individual as well as combinations of classifiers, such as support vector machine, k-nearest neighbor, naïve Bayes, adaptive neuro-fuzzy inference system, and discriminant analysis classifiers. The recognition accuracy of the naïve Bayes classifier was 93.13%, which is the best among all the classifiers. Recognition accuracy improved by about 10% with an increase in the number of features.

Keywords: Hand gesture recognition; Pattern variation; Self-co-articulation; Trajectory features

1. Introduction

A promising area in pattern-variation research is the systems developed for recognizing hand gestures. Various hand gesture systems have been designed and proposed in the literature, which can make human–computer interaction easier and simpler. Moreover, challenges arise when there is variation in gestures according to the user, which produces a hindrance in the flexibility of various practical applications. Additionally, the presence of self-co-articulation raises the problem of misclassifying gestures, which leads to incorrect recognition. If the challenges associated with pattern variation and self-co-articulation can be addressed by the hand gesture recognition systems, it will be easier to use the hand gesture systems in applications such as sign language communication, video-based surveillance, and the use of a mouse.

Static hand gestures are easy to recognize and use; however, the major challenge lies in dynamic hand gestures, as they are recorded by moving the hand in space according to the user's comfort level. In literature, users typically follow the reference pattern of the gestures provided to them. Li *et al.* (2018) used the Mobile Gesture Database, which consists of A–F letters and 1–6 numerals. Bhuyan *et al.* (2014) and Pun *et al.* (2011) developed a hand gesture recognition system using only 10 numerals from 0–9, which were recorded with a fixed set of patterns. Tang *et al.* (2018) used lowercase letters a–z for designing the hand gesture recognition system using dynamic time warping. Beh *et al.* (2014) considered numerals ranging from 1–9, a few letters like A, B, C, D, X, Y, and Z, along with five direction-based symbols. Singha *et al.* (2016) included the gestures from A–Z and 0–9 with pattern variation using a colored marker. However, previous studies have not included the bare hand gesture pattern variation. In this study, some of the natural variations of the hand gestures have been recorded and used for detection, tracking, and recognition. Figure 1 shows the gestures with the self-co-articulated strokes that need to be detected and removed to obtain the original gesture. Specifically, stroke 3 in "A," strokes 3 and 5 in "E," stroke 3 in "F," stroke 2 in "T," and stroke 2 in "X" correspond to self-co-articulated strokes (**Figure 1**). Self-co-articulation refers to strokes that occur either at the beginning of a gesture or between strokes during the formation of the gesture shape.

The main contributions of this paper are as follows:

(i) The design of a new dataset named the LNM Institute of Information Technology (LNMIIT) Dynamic Hand Gesture Dataset-4. This dataset mainly consists of numbers and letters recorded by different users. As very few datasets consisting of videos with pattern variation are publicly available, we developed our own dataset.
(ii) The detection and removal of self-co-articulation strokes occurring at the start of the gesture and those occurring at the time of gesture completion.
(iii) The proposal of a new feature that divides the stroke into two parts horizontally, such as the ratio of the number of points in the first half, the ratio of stroke length, the distance between the starting and ending points of each half, and the curliness of both halves.
(iv) Recognition accuracy was calculated for each of the gestures using different classifiers, including support vector machine (SVM), artificial neural network (ANN), k-nearest neighbor (k-NN), naïve Bayes, adaptive neuro-fuzzy inference system (ANFIS), and discriminant analysis (DA).
(v) A combination of different sets of features with classifiers was performed and analyzed to reduce the misclassification of the gestures.

The paper is organized as follows: Section 2 reviews related work. Section 3 provides an explanation of the detailed architecture of the proposed system, and describes the dataset used and the effect of pattern variation in gestures. Section 4 presents the recognition results, analyzed using different classifiers. Finally, the paper is concluded in Section 5.

2. Related works

In this section, we review various models used for tracking and summarize the features extracted from dynamic gesture trajectories for accurate recognition. Various traditional methods have been used to track gesture trajectories under certain conditions. Comaniciu *et al.* (2003) used a color histogram to develop a hand tracker model in which hand detection is determined by calculating a color histogram, which is used as a mean shift and locates the hand in video frames along with tracking. The CamShift algorithm is able to track any feature distribution representing the target in a successful way (Bradski & Kaehler, 2008). There are many techniques where the CamShift was combined with various other tracking methods, which led to improved tracking efficiency. For example, the CamShift algorithm was combined with a Kalman filter (Huang & Hong, 2011; Wang & Li, 2010). The possible positions of a target were predicted by the Kalman filter, and CamShift was subsequently used to search and match the target in the predicted areas. Kolsch & Turk (2004) introduced an algorithm using a Kanade–Lucas–Tomasi-based tracker. However, the Kanade–Lucas–Tomasi tracker did not yield good results when the hand shape changed during gesticulation.

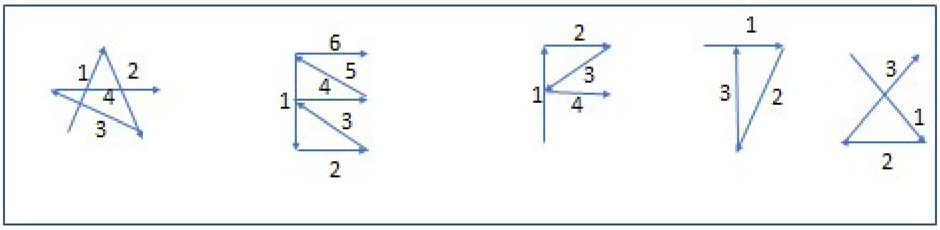


Figure 1. Gestures showing self-co-articulated strokes

Accurate tracking of the trajectory is a prerequisite for effective feature extraction, which in turn improves recognition accuracy. For developing a gesture recognition system, a single feature was used in several studies. Elmezain *et al.* (2008) tried to recognize both isolated and continuous gestures with the help of an orientation feature. With the help of this feature, gesture motion direction can be calculated using trajectory points. The quantization process was performed using code words from 1 to 18 on the angle of orientation. Orientation feature was also used by Kao & Fahn (2011) to design a real-time hand gesture recognition system. Gestures were classified using a Hidden Markov Model after the quantization process. Location, orientation, and velocity features were among the most used features by researchers (Elmezain *et al.*, 2009; Xu *et al.*, 2015; Yoon *et al.*, 2001). Xu *et al.* (2015) proposed a novel hand gesture recognition system for robotic applications using features such as orientation, with a chain code of 1–8 code words, location, and velocity. Yoon *et al.* (2001) used the combination of features such as location, orientation, and velocity.

Singla *et al.* (2019) used the normalized sequence of captured three-dimensional space coordinates as input, and a sequence of features was computed along the trajectory. Gesture direction, curvature, aspect, curliness, slope, and lineness are some of the features that have been calculated and used for the development of feature space and recognition. Misra and Laskar (2019) presented novel spatio-temporal trajectory features that provide output as structural values of the gestures. These features included the area of two halves, local geometrical area ratios, and curve-area features. The gesture was divided into two halves equally, and the area of each half provided the output as the area of the two halves. For the calculation of local geometrical area ratios, the ratio between enclosed areas in each case of stroke length was measured. Variation in the patterns of bare-hand dynamic gestures has not been adequately addressed in the existing literature, as users are typically required to record videos according to predefined gesture patterns. In this study, we developed a hand gesture recognition system that is capable of recognizing gestures with varying patterns.

Anish *et al.* (2021) proposed a gesture formation model for the removal of self-co-articulation gestures. Features like Euclidean distance, velocity, and the minimum-maximum-polarity algorithm were used as global and local measures. The experiments were done on the NITS hand gesture dataset, which showed an improvement in accuracy of about 40%. Although self-co-articulation strokes were identified and removed, pattern variations were not considered in the work. The effects of variable illumination were considered for real-time recognition of twenty-six American Sign Language signs. The model employed scale-invariant feature transform features to represent gesture patterns with inherent symmetry, with the ANN model showing a recognition accuracy of 97.03%. Cheng *et al.* (2023) used graph convolutional networks along with the path signature theory for extracting features from various skeleton joints. Since the model relied on differential features for recognition, variations in gesture patterns were not taken into account.

3. Methodology

The proposed system is mainly divided into seven phases: data acquisition, detection, tracking, trajectory smoothening, self-co-articulated stroke detection due to pattern variation, feature extraction, and the recognition of the gesture being provided as input.

Figure 2 shows the detailed flowchart of the proposed hand gesture recognition system. The detailed description of each phase is given below in subsections.

3.1. Data acquisition

Figure 3 shows a sample of the LNMIIT Dynamic Hand Gesture Dataset-4, which has videos recorded using pattern variation. Various numerals and letters have been considered for pattern variation, such as 5, A, B, D, E, F, and P. Self-co-articulation represents the unwanted strokes when recording the gestures. These unwanted movements can change the shape of the character; hence, they have to be recognized. It has been observed that the gestures that show the pattern variation consist of an extra stroke at the start of the gesture trajectory. This extra stroke, if

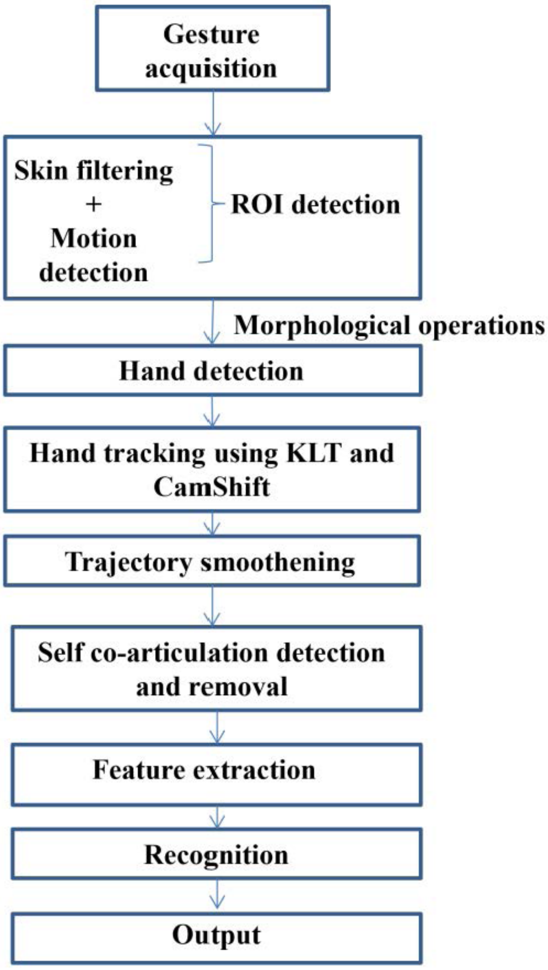


Figure 2. Flowchart of the proposed system
Abbreviations: KLT: Kanade–Lucas–Tomasi; ROI: Region of interest.

removed, will lead to an ordinary gesture that can be easily recognized by the proposed system. The self-co-articulated stroke present will extract the changed features, such as the number of points, stroke length, and angle, leading to a misclassification of the gesture and hence, an incorrect recognition. However, there is a requirement to remove the other self-co-articulated strokes also.

We developed the LNMIIT Dynamic Hand Gesture Dataset-4 with the help of various users who were using different ways of gesturing alphanumeric characters. Acquisition of gestures was performed in an indoor environment using a Logitech (Switzerland) C922 Pro Stream Webcam (640 × 360 pixels) with an aspect ratio of 16:9 and 30 frames per second. Figure 4 shows two patterns for different alphanumeric characters. Pattern 1 shows the gestures without any extra stroke, while pattern 2 represents gestures with self-co-articulated strokes. In numeral 5, pattern 2 consists of stroke number 2, which represents the self-co-articulated stroke. In the letter "A," stroke 2 and stroke 4 represent the two different self-co-articulated strokes. Only some of the alphanumeric characters have been shown in the figure. Both types of self-co-articulated strokes must be detected and removed to obtain the proper gesture, which can then be appropriately recognized.

Gestures from the LNMIIT Dynamic Hand Gesture Dataset-3 have also been used for the proposed system. This dataset consists of numbers from 0–9, letters from A–Z, and lowercase letters from a–z.

The LNMIIT Dynamic Hand Gesture Dataset-4 consists of alphanumeric characters showing pattern variation. Details of various datasets are given in Table 1. The constraints used for recording gestures in the LNMIIT Dynamic Hand Gesture Dataset-4 are:

(i) At the start of recording the gesture, the hand must be placed in the correct position.
(ii) The palm should exhibit clearer motion than the forearm to illustrate the gesture distinctly.
(iii) The movement of the hand must be smooth and continuous.
(iv) Lighting must be adequate at the time of recording.
(v) The hand must be kept in a static gesture position for a few seconds before the completion of the gesture.
(vi) The users were asked to record a video in their natural way of writing the characters.

3.2. Detection

Detection plays the most important role in the proper recognition of the gesture. After separating the region of interest with the help of a skin detection process, morphological operations were applied to remove non-hand skin regions and refine the detected hand mask. Skin filtering was followed by motion detection, giving only the hand region as the output. A bounding box was created around the hand region so that its coordinates can be used for obtaining the trajectory points. A combination of HSV and YCbCr color models was used for this purpose (Saboo *et al*., 2021). Figure 5 shows the various steps involved in the process of hand detection.

3.3. Tracking and trajectory smoothing

The motion of the detected hand must be tracked to obtain the trajectory of the recorded gesture. All the frames of the gesture video were taken into consideration, and the center of the hand region of each frame was marked to form the hand gesture trajectory. As the plotted trajectory contained some irregular strokes due to the movement of the hand at the start of the gesture, such as the diversion

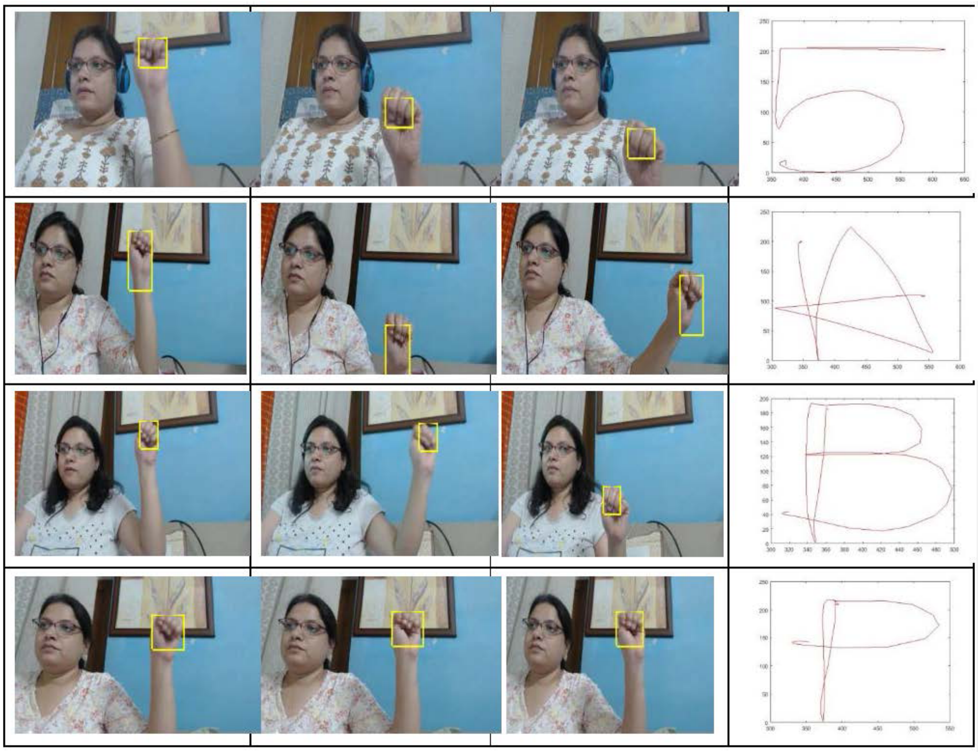


Figure 3. Video frames of some of the gestures from the LNM Institute of Information Technology Dynamic Hand Gesture Dataset-4

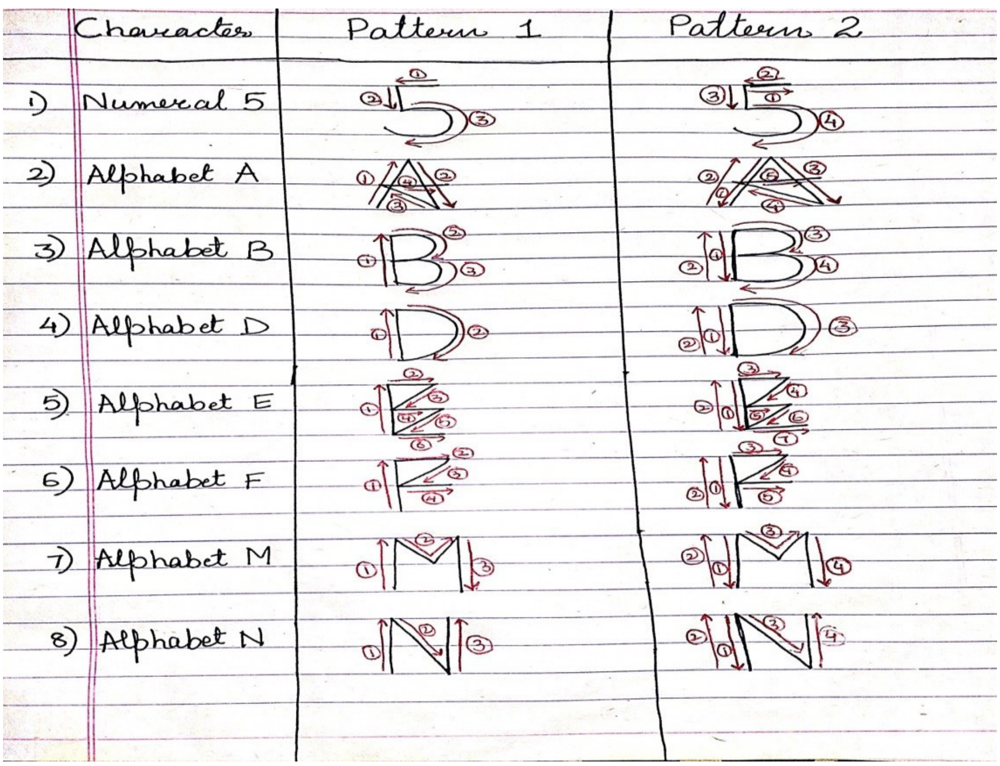


Figure 4. A few samples of gesture pattern variations

Table 1. Description of the hand gesture dataset

Characteristics	LNMIIT Dynamic Hand Gesture Dataset-1	LNMIIT Dynamic Hand Gesture Dataset-2	LNMIIT Dynamic Hand Gesture Dataset-3	LNMIIT Dynamic Hand Gesture Dataset-4
No. of videos	500	900	1,860	720
Details	0–9, A, C, S	0–9 numbers, a–z excluding f, i, j, k, t, and x	0–9 numbers, A–Z, and a–z	0–9 numbers, A–Z
Variations	Hand shape variation, illumination variation, and the presence of multiple persons	No variation	No variation with self-co-articulation	Pattern variation with self-co-articulation
Acquisition device	ArcSoft Web Companion	Logitech C922 Pro Stream Webcam	Logitech C922 Pro Stream Webcam	Logitech C922 Pro Stream Webcam
Resolution	640 × 480 pixels	640 × 360 pixels	640 × 360 pixels	640 × 360 pixels
No. of users	3–4	5	10	12
Frame rate (frames per second)	30	30	30	30

Abbreviation: LNMIIT: LNM Institute of Information Technology.

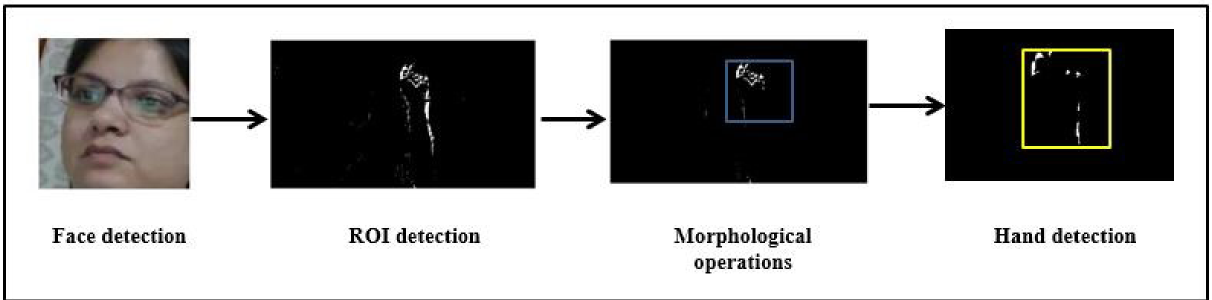


Figure 5. Steps of hand detection
Abbreviation: ROI: Region of interest.

of the bounding box, trajectory smoothening becomes a necessary process. A smoothed trajectory was obtained by averaging the previous, current, and next points and replacing all three of them with the calculated point (Saboo *et al*., 2022). Figure 6 shows the frame-to-frame tracking, indicating the original trajectory of the gesture. Removal of the starting noise and averaging of the gesture points produced a smoothed gesture trajectory as the output (Equations 1–3).

$$x(t), y(t) = \frac{x(i-1)+x(i)+x(i+1)}{3}\ ;\ \frac{y(i-1)+y(i)+y(i+1)}{3} \quad (1)$$

$$x(s), y(s) = \frac{x(1)+\cdots x(5)}{5}; \frac{y(1)+\cdots y(5)}{5} \quad (2)$$

$$x, y = \big(x(s){:}\,x(t)\big)\ ;\ \big(y(s){:}\,y(t)\big) \quad (3)$$

where $x(s)$ and $y(s)$ were used to remove starting noise; $x(t)$ and $y(t)$ represent the averaged points, and x and y represent the current array of trajectory points. Smoothed trajectory helps in the proper recognition of the gesture and hence is an important process in the steps of a hand gesture recognition system.

3.4. Self-co-articulation detection and removal

A gesture comprises different types of self-co-articulated strokes. In this work, we focused on removing two such strokes: one arising due to pattern variation and another occurring during the formation of the gesture itself. These self-co-articulated strokes can introduce unwanted feature variations and lead to incorrect gesture recognition. Thus, proper detection and removal of self-co-articulated strokes become an important phase in the proposed hand gesture recognition system. As illustrated in Figure 3, the self-co-articulation caused by pattern variation typically occurs at the beginning of the gesture. To address this, experiments were conducted to analyze the number of points in the majority of gestures, which was then used to determine an appropriate threshold for point consideration.

Table 2 consists of the various values of the number of points that help in deciding the threshold value required for extracting the points to be removed. Hence, we considered starting from 15 points and checking the

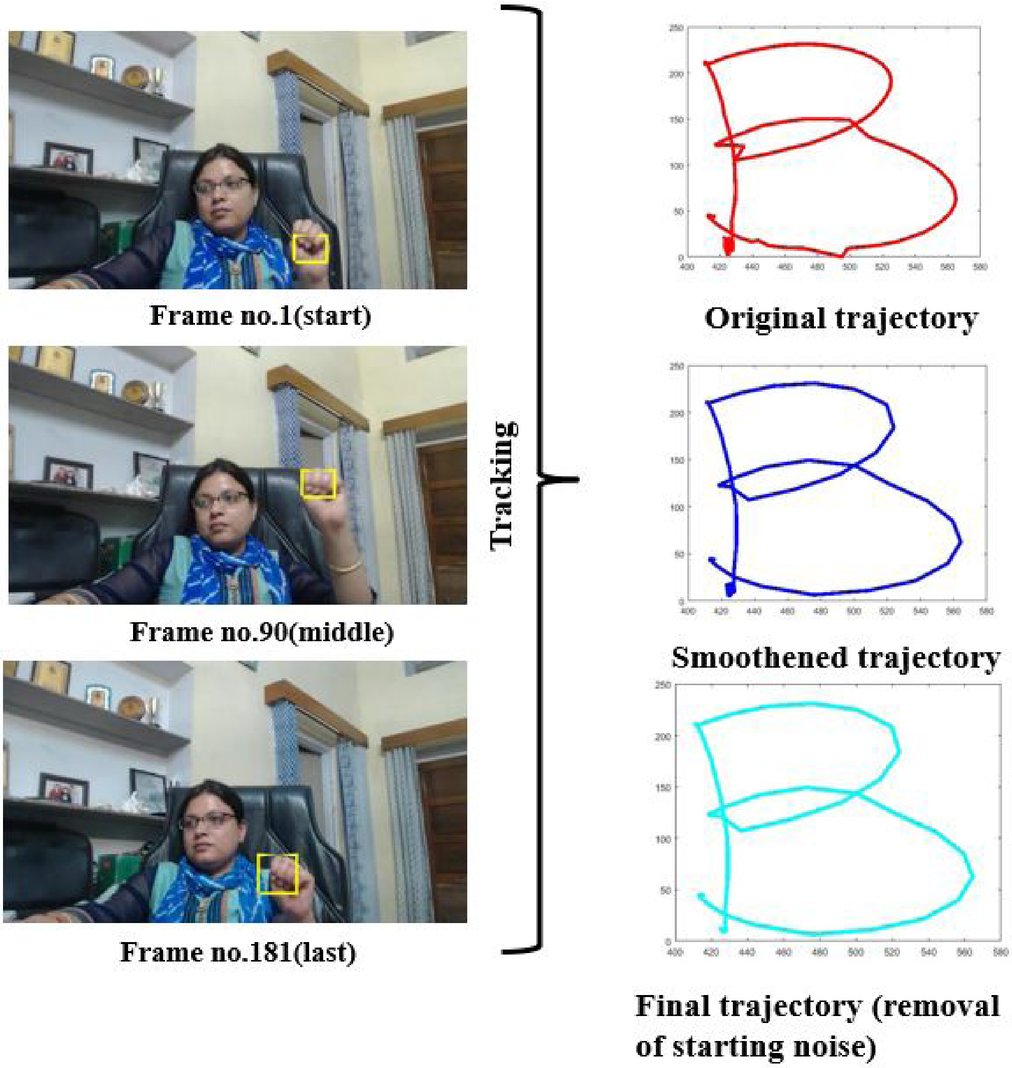


Figure 6. Gesture tracking showing the original and smoothed trajectory

difference between each point and the next point.

A point is considered the start of the gesture when the difference changes sign compared to previous values. All points occurring before this sign-change point are regarded as part of self-co-articulated strokes and should be removed.

Another type of self-co-articulation is the stroke occurring due to the formation of the gesture. It can be seen from Figure 7 that all the gestures have a co-articulation stroke in the negative x or y direction. The methodology for removing self-co-articulated strokes began by calculating the differences between consecutive trajectory points. Points with negative difference values were identified and separated (Equations 4 and 5). Figure 7 shows the flowchart of the detection and removal of self-co-articulated strokes.

$$diff = y(t+1) - y(t) \tag{4}$$

$$Y = Y(i) \ if \ diff > 0 \tag{5}$$

where $y(t)$ represents the current point, and all those points should be removed for which the difference is negative.

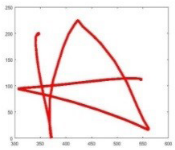
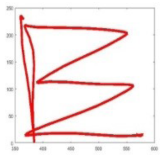
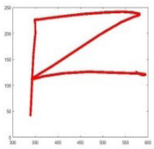
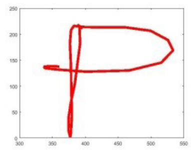
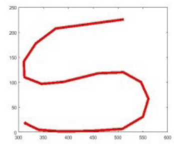
In continuous hand gesture movements, self-co-articulation strokes encounter problems during movement from one gesture to another. It is observed that the duration of a gesture is longer than the duration of the co-articulated strokes.

3.5. Feature extraction

Extraction of the features from the trajectory points is required for matching the trajectory with the gesture.

These features are provided as the input to various

Table 2. The number of points for starting the self-co-articulation stroke

Gesture					
No. of self-co-articulation strokes present	Two (one at the start of the gesture and another after three strokes)	Three (one at the start of the gesture and two after three strokes)	One (After two strokes)	One (At the start of the stroke)	No self-co-articulation stroke
No. of points for starting self-co-articulation	12	15	17	14	16

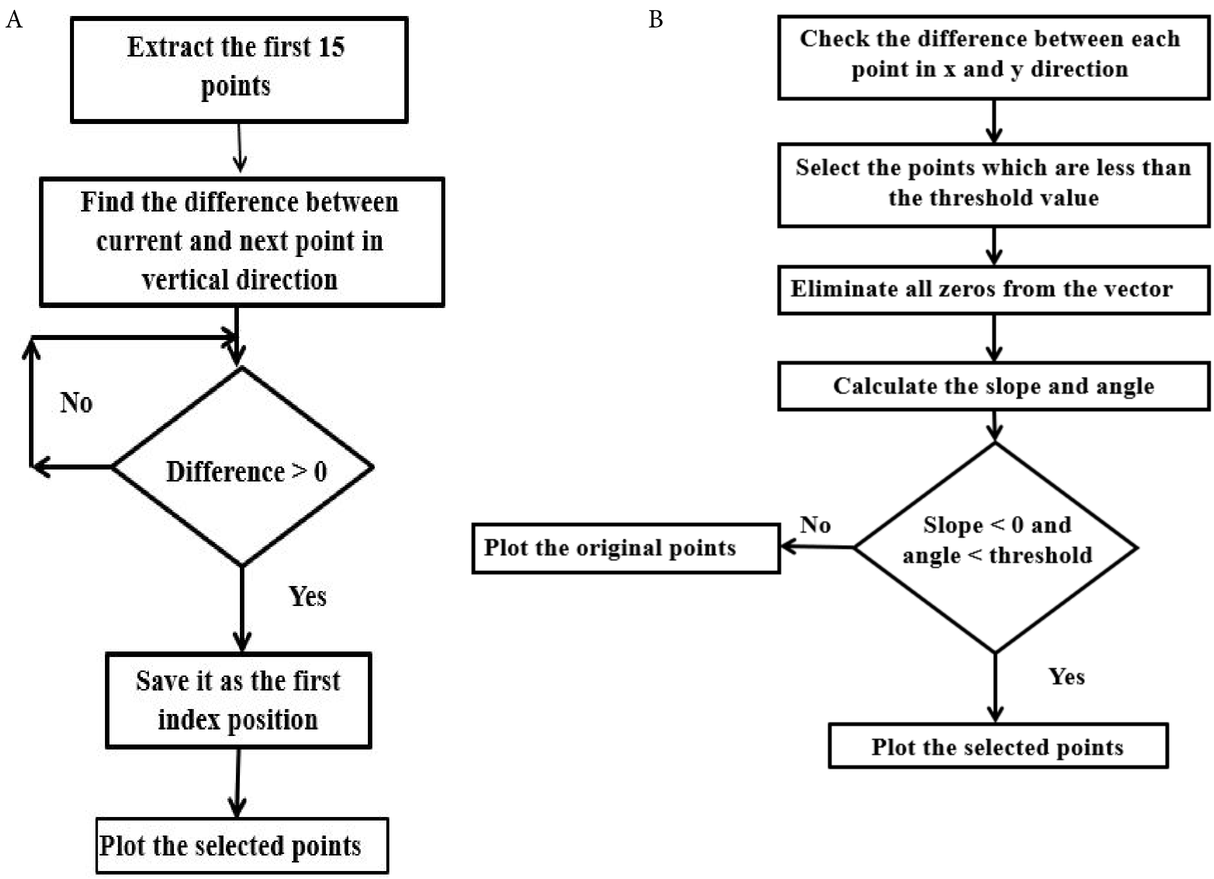


Figure 7. Flowcharts for the detection and removal of self-co-articulation strokes. (A) Starting of the self-co-articulated strokes. (B) Self-co-articulated strokes at the time of formation of the gesture shape.

classifiers, and hence they should be appropriately determined. In general, features that exhibited consistency across instances of the same gesture were considered for analysis. A total of 12 features were considered in this study, among which some were taken from the literature, and some were proposed by dividing the trajectory into two halves horizontally. The features, such as the distance feature (Rubine, 1991b), the location feature (Xu *et al*., 2015), and the close figure test feature (Misra *et al*., 2018), were taken from the literature. A subset of features was selected from the available feature set provided by Rubine (1991a). Table 3 provides the list of proposed and existing features.

3.5.1. The ratio of the number of points in the first and second half, and the ratio of the midpoint of the first and second half

After removing the self-co-articulated strokes, the resulting gesture shape matches the original gesture. The gesture trajectory was divided into two halves in the horizontal direction, and the number of points was calculated for the two halves. The ratio of these numbers was considered as one of the features. For each half of the trajectory, the midpoint was calculated, and the ratio of both values was taken as a new feature, which matches the pattern variation gestures.

3.5.2. Stroke length of half/Distance between first and last point of half

This feature was calculated by first identifying the complete distance traversed by the trajectory and dividing it by the distance between the first and last points, as shown in Equation 6:

$$\frac{\sum_{i=1}^{n}\sqrt{[x(i+1)-x(i)]^2+[y(i+1)-y(i)]^2}}{\sqrt{\left(x_{p-1}-x_o\right)^2+\left(y_{p-1}-y_o\right)^2}} \tag{6}$$

where, x_{p-1} represents the last point and x_o represents the first point of the trajectory. A feature reflects more than one entry in the taxonomy; for example, the entropy feature was considered a measure of density.

3.5.3. Perimeter efficiency of halves

To find the value of this feature, all the two-dimensional coordinates were used to create an alpha shape, and the perimeter was calculated. Area of the alpha shape when multiplied by pi and divided by the perimeter gives the perimeter efficiency as Equation 7:

$$PE = 2 * \frac{\sqrt{\pi * A}}{P} \tag{7}$$

where A and P represent the area and perimeter, respectively. The selected value of the alpha radius gives a scalar quantity specifying the radius of the alpha disk or sphere used to recover the alpha shape. Figure 8 represents different gestures with some of the features taken into consideration.

3.5.4. Ratio of convex hull area of the first and second half

A convex hull can be defined as the shape of the smallest convex set that contains it. To calculate the convex hull, the point with the minimum x coordinate value or the leftmost point was taken as the starting point, and the points were wrapped up in a counterclockwise direction.

3.5.5. Curliness of halves

The curliness feature states the deviation of a gesture from a straight line in the vicinity of a curl in two dimensions, as shown in Equation 8:

$$\text{Curliness, } C(t) = \frac{P}{\max(\Delta x, \Delta y)} - 2 \tag{8}$$

3.5.6. Minimum bounding rectangle

This feature finds the average of the rectangles bounding the various trajectory points. The area of the minimum bounding rectangle is calculated by finding the ratio of the difference between each trajectory and the minimum value by the difference between the maximum value of the trajectory points and the minimum value of the trajectory points, as shown in Equations 9 and 10:

Table 3. List of total features used in the proposed system

Existing features	Proposed features
• Length feature (Rubine, 1991a) • Angle feature (Rubine, 1991a) • Distance feature (Rubine, 1991b) • Close figure test (Misra *et al*., 2018) • Location feature (Xu *et al*., 2015)	• Ratio of the number of points in the 1st and 2nd half • Stroke length of half/Distance between first and last point of half • Perimeter efficiency of halves • Curliness of halves • Ratio of the midpoint of the 1st half and the 2nd half • Ratio of the convex hull area of the 1st and 2nd half • Minimum bounding rectangle

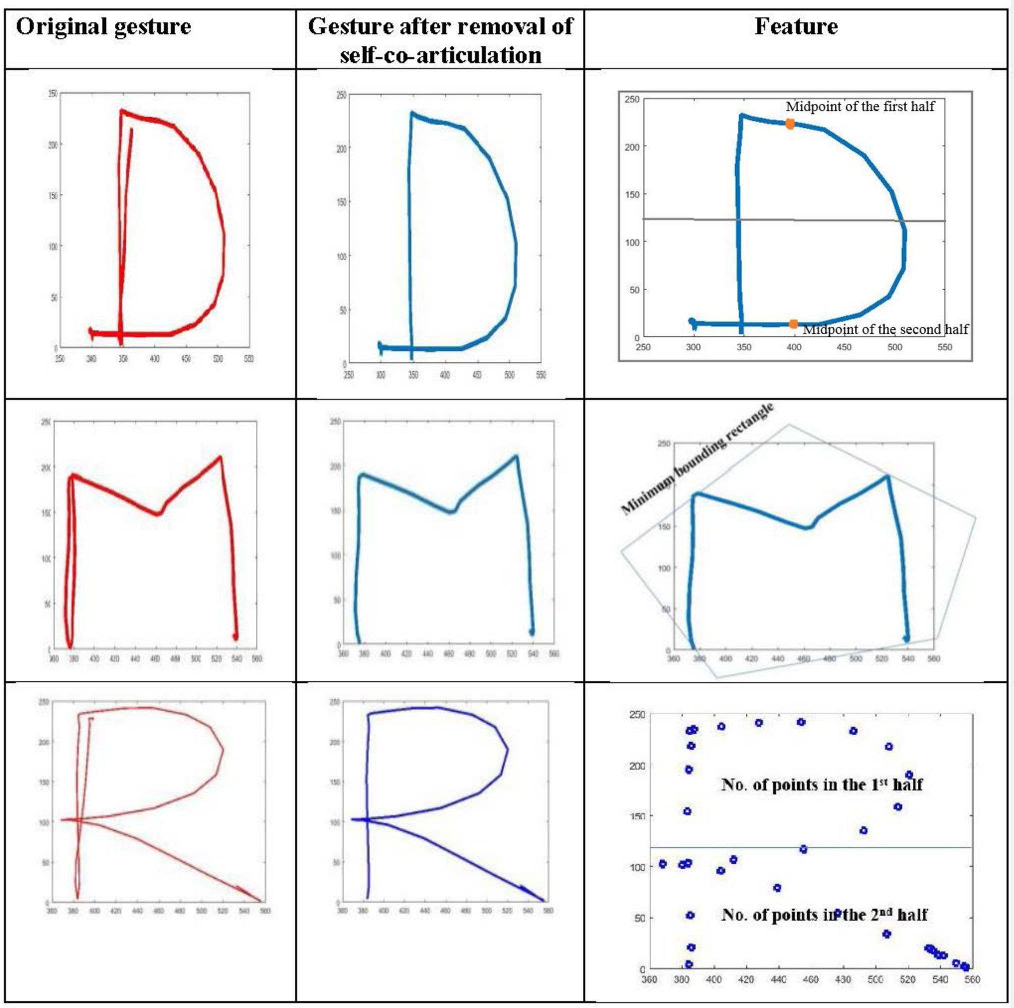


Figure 8. Gestures representing some of the proposed features

$$x_{max} = \max_{t=1} n\,(X_t)\ ; \quad x_{min} = \min_{t=1} n\,(X_t)$$

$$y_{max} = \max_{t=1} n\,(Y_t); \quad y_{min} = \min_{t=1} n\,(Y_t)$$

$$x_t = \frac{X_t - x_{min}}{x_{max} - x_{min}}\ ; \quad y_t = \frac{Y_t - y_{min}}{y_{max} - y_{min}} \tag{9}$$

$$mbr = x_t * y_t \tag{10}$$

3.6. Recognition

Support vector machine can be used to classify for both supervised and unsupervised learning. SVM uses various types of kernels like linear, Gaussian, and polynomial. It projects the data into a higher-dimensional space, enabling the separation of linearly non-separable data and reducing classification errors (Hsu & Lin, 2002). k-NN is a classifier that solves both classification and regression problems by using supervised machine learning methods. This algorithm assumes that similar things are available in proximity and thus captures the idea of similarity.

Naïve Bayes classifications assume that, given the class, the features are conditionally independent, even though the class itself may depend on multiple features (Singha *et al.*, 2016). An ANN is a gesture recognition approach used by various researchers. ANN comprises an input, hidden, and output layer with neurons used according to the available dataset (Bamwenda & Özerdem, 2019).

Adaptive neuro-fuzzy inference system (Subasi, 2007) uses a hybrid learning algorithm and applies a combination of the least-squares method and the back-propagation gradient descent method for training FIS membership function parameters to emulate a given training dataset. The quadratic DA (Tharwat, 2016) model is a discriminant model of classification used for predicting the response of the test data. This type of classifier can be linear or quadratic in nature.

4. Results and discussion

Tracking accuracy was calculated for various classifiers individually as well as with different combinations of features. The LNMIIT Dynamic Hand Gesture Dataset-3 and the LNMIIT Dynamic Hand Gesture Dataset-4 were used, which consist of 20 samples each of numbers and letters; hence, a total of 720 gestures were used in this experiment. Out of 720 gestures, the ratio of training and testing sets was separated as 70% and 30%, respectively. Different kernel functions, such as linear, Gaussian, and polynomial, were used in SVM. The polynomial kernel provided the best accuracy of 85.88% for the complete dataset (Table 4). Tests were conducted for numbers and letters separately, and then both were considered together.

We also evaluated the performance of the ANN classifier, and the results are shown in Table 4. Training, testing, and validation accuracies were also calculated along with the overall classifier accuracy. The highest accuracy for 20 hidden neurons and network structure 22–36–20 was obtained, which was 100%, 94.56%, and 90.21% for numbers, letters, and alphanumeric characters, respectively. The number of hidden neurons and layers was changed, and different permutations and combinations were tested to obtain reliable accuracy.

For different values of k, the k-NN classifier was also used to calculate the recognition accuracy. The highest accuracy was calculated for k = 1. Recognition accuracy for

Table 4. Recognition accuracy of the classifiers

Classifiers	Accuracy measures		
	Numerals	Letters	Alphanumeric characters
Support vector machine (SVM): polynomial kernel			
SVM + existing features	65.52%	63.31%	60.24%
SVM + 15 features	74.22%	72.23%	69.87%
SVM + 20 features	88.33%	86.36%	85.88%
Artificial neural network (ANN): network structure 22–36–20			
ANN + existing features	74.45%	70.23%	65.52%
ANN + 15 features	82.24%	81.12%	78.86%
ANN + 20 features	100%	94.56%	90.21%
k-nearest neighbor (k-NN): k = 1			
k-NN + existing features	74.25%	72%	68.8%
k-NN + 15 features	82.2%	75.05%	70.01%
k-NN + 20 features	95%	91.67%	88.89%
Naïve Bayes (NB): Data = kernel			
NB + existing features	80.1%	79.5%	75.44%
NB + 15 features	87.74%	83.34%	81.14%
NB + 20 features	96.43%	95.94%	93.13%
Adaptive neuro-fuzzy inference system (ANFIS): Linear			
ANFIS + existing features	67.52%	62.33%	60.01%
ANFIS + 15 features	75.29%	72.55%	70.14%
ANFIS + 20 features	96.50%	92.23%	89.96%
Discriminant analysis (DA): Linear			
DA + existing features	80.21%	79.96%	78.08%
DA + 15 features	86.65%	82.22%	81.14%
DA + 20 features	95%	95.51%	90.28%

numbers, letters, and a combination of both was obtained as 95%, 91.67%, and 88.89%, respectively (Table 4). The value of k is generally taken as an odd number. Naïve Bayes classifier, when used with data distribution as "kernel," yielded the best accuracy of 96.43% for the number dataset. Using the existing features, the classifier achieved a maximum accuracy of 80.1% for numerals. Increasing the number of features to 15 improved the accuracy by approximately 7%. Finally, expanding the feature set to 20 yielded a satisfactory recognition accuracy of 96.43%. When performed with an alphanumeric dataset, the accuracy obtained was 93.13% (Table 4).

The ANFIS classifier was used to solve the problems regarding fuzzy classification. Table 4 reports the accuracy of 89.96% when the function used was linear. This type of classifier helps in recognizing similar types of gestures like "o" and "0," "2," and "Z." Another type of classifier used for checking the effectiveness of the proposed system is the quadratic type of linear DA, which helps in classifying the gestures that are generated based on Gaussian distributions. As the number of features increased, the recognition accuracy increased to 95%, which is better than the classification conducted with 15 features (Table 4). Analysis of the individual classifiers shows that naïve Bayes achieved the highest accuracy, while SVM yielded the lowest (Figure 9).

The reduced accuracy is attributed to confusing gestures such as O, 0, 2, and Z. Excluding these misclassified cases improved recognition accuracy by approximately 3–4%. Table 5 shows the results with and without misclassified cases. This misclassification arises because these characters share similar shapes and patterns, as shown in Figure 10. The proposed system is designed for individuals who have become impaired due to accidents and are unfamiliar with

Table 5. Results of classifiers with and without misclassification

Classifiers	With misclassified gestures	Without misclassified gestures
SVM	85.88%	91.26%
ANN	90.21%	96.62%
k-NN	88.89%	94.44%
Naïve Bayes	93.13%	96.36%
ANFIS	89.96%	92.22%
Quadratic DA	90.28%	92.37%

Abbreviations: ANFIS: Adaptive neuro-fuzzy inference system; ANN: Artificial neural network; DA: Discriminant analysis; k-NN: k-nearest neighbor; SVM: Support vector machine.

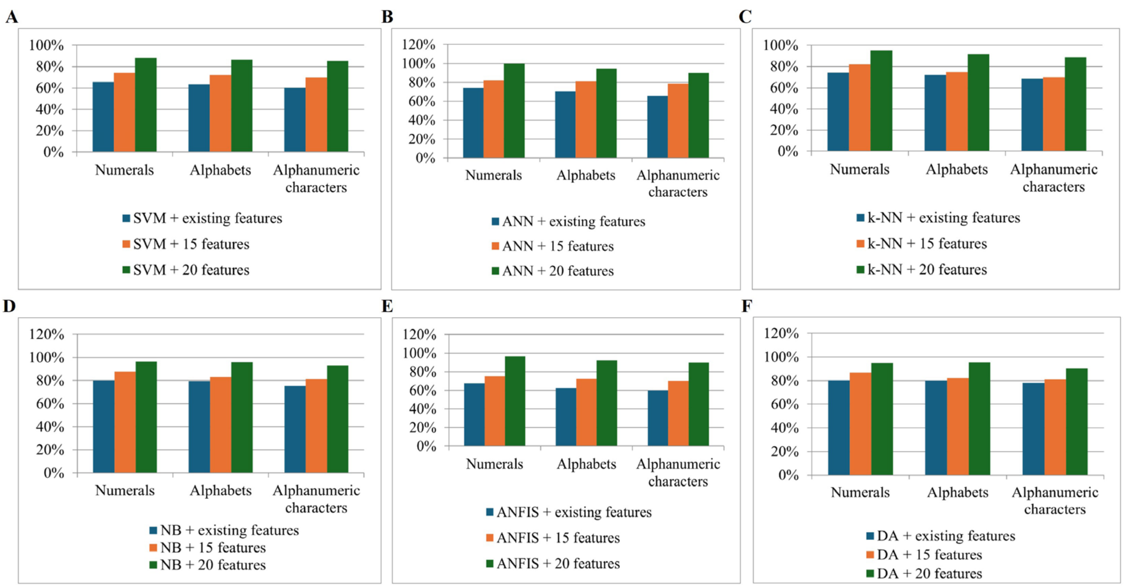


Figure 9. Performance in terms of accuracy using different feature combinations. (A) Support vector machine (SVM), (B) artificial neural network (ANN), (C) k-nearest neighbor (k-NN), (D) naïve Bayes (NB), (E) adaptive neuro-fuzzy inference system (ANFIS), and (F) quadratic discriminant analysis (DA).

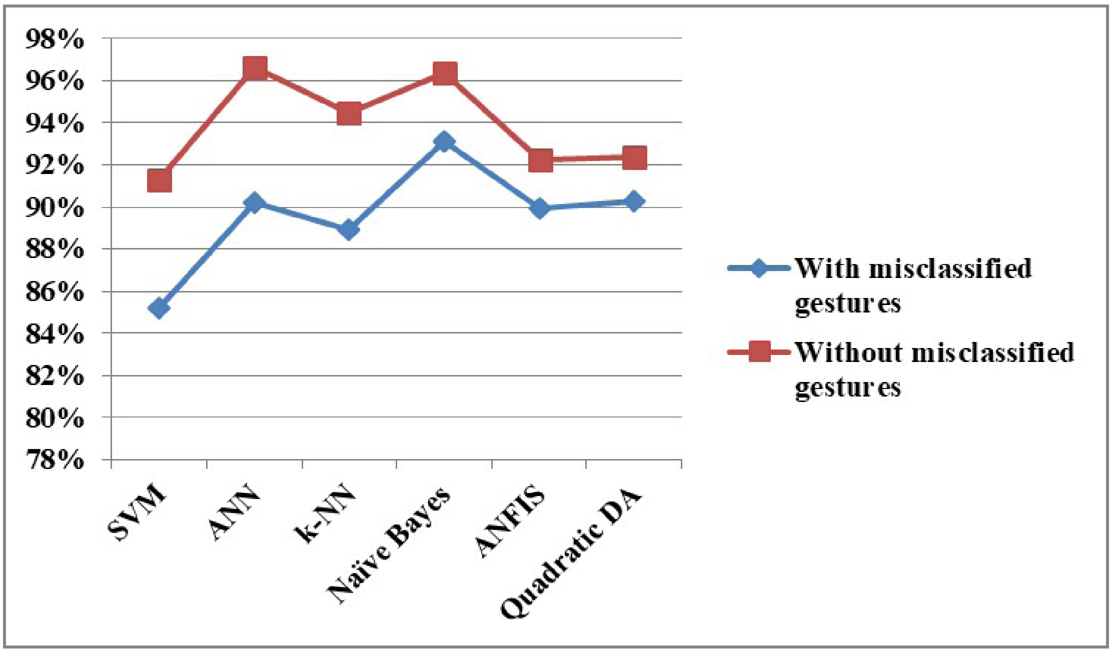


Figure 10. Output graph showing the change in efficiency with respect to misclassified gestures
Abbreviations: ANFIS: Adaptive neuro-fuzzy inference system; ANN: Artificial neural network; DA: Discriminant analysis; k-NN: k-nearest neighbor; SVM: Support vector machine.

sign language. Figure 11 shows one of the applications implemented using the proposed system.

To validate the results of the proposed system, a comparison with existing literature was conducted, and the results are shown in Table 6. Accuracies calculated in these papers using different machine learning algorithms, such as SVM, ANN, k-NN, and naïve Bayes, were compared with the proposed system. It can be observed that the proposed system achieved the highest mean accuracy for all the individual classifiers.

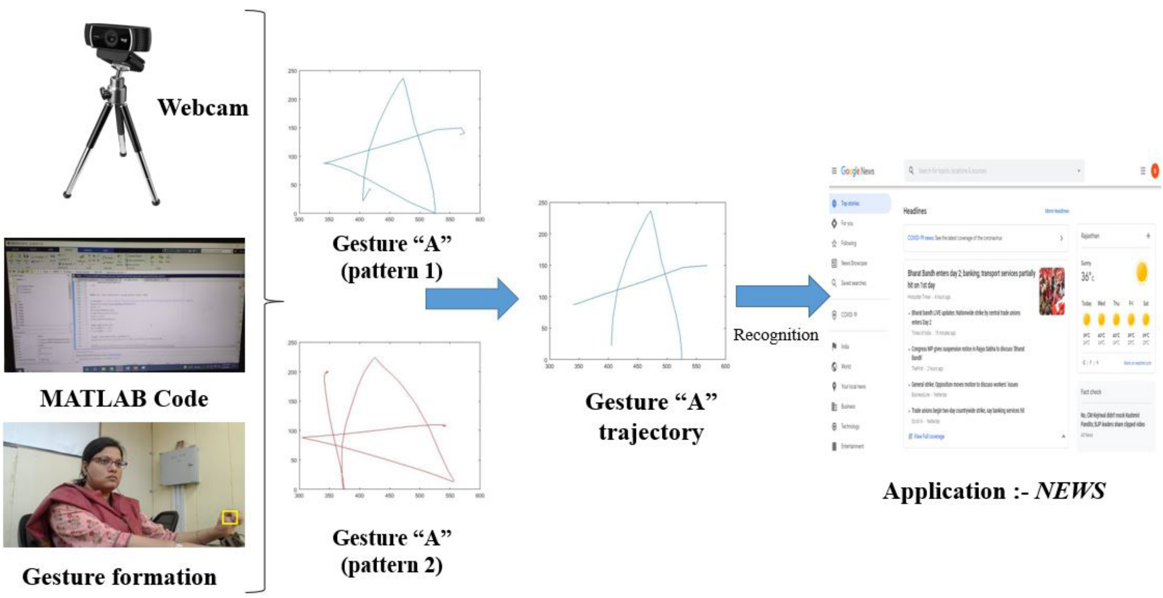


Figure 11. Block diagram for the application interface

Table 6. Comparison of existing literature with the proposed system

Method	SVM accuracy (%)	ANN accuracy (%)	k-NN accuracy (%)	Naïve Bayes accuracy (%)
Bhuyan et al. (2014)	60.31	62.77	59.64	55.98
Singha et al. (2016)	85.34	88.69	82.21	79.65
Proposed	85.88	90.21	88.89	93.13

Abbreviations: ANN: Artificial neural network; k-NN: k-nearest neighbor; SVM: Support vector machine.

5. Conclusion

Few studies in the literature have focused on gestures exhibiting pattern variations. Publicly available hand gesture datasets contain only a limited number of dynamic gestures, which are typically recorded following a predefined pattern that serves as the reference. To calculate the recognition accuracy of the various hand gestures, a dataset named LNMIIT Dynamic Hand Gesture Dataset-4 was developed, which contains all the alphanumeric characters with different variations in the pattern of the gestures. Experiments were conducted on 720 samples recorded with more than 10 people. Hand detection, hand tracking, and feature extraction were supported with the detection and removal of self-co-articulated strokes, which improved the recognition accuracy of the proposed hand gesture recognition system. An algorithm was developed to remove the starting self-co-articulated strokes, followed by the removal of other self-co-articulated strokes that occurred during the formation of some of the gestures. Several new features, such as the ratio of the points contained in the two halves of the gesture trajectory and the curliness feature, were introduced. Different features and their combination with classifiers, including SVM, k-NN, ANN, naïve Bayes, ANFIS, and DA classifiers, were explored in this study. The naïve Bayes classifier achieved the best accuracy of 93.13%, which is an improvement of 3% compared to ANN and quadratic classifiers. The proposed system was tested on different gestures of numerals and letters with pattern variation, gestures without pattern variation, and the total set of gestures. It can be observed that reduced accuracy was due to misclassification of gestures, such as 2, Z, O, 0, 5, and S. Future work should include the development of a new set of features to improve the discrimination of the gestures from one another. Additionally, recognition of gestures to form words may be included in future research work.

Acknowledgments

We would like to express our sincere gratitude to all those who contributed to the successful completion of this research project. Their support, guidance, and expertise were invaluable in shaping the outcome of this study.

Funding

None.

Conflict of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization: Shweta Sharda
Formal analysis: Joyeeta Singha
Investigation: Shweta Sharda
Methodology: Shweta Sharda
Visualization: Shweta Sharda
Writing–original draft: Shweta Sharda
Writing–review & editing: Joyeeta Singha, Ritu Vyas

Ethics approval and consent to participate

Ethics approval was not required for this study. All participants provided informed consent before participating in this study.

Consent for publication

All participants provided consent for the publication of the study's findings and any accompanying images.

Availability of data

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Anish Monsley, K., Yadav, K. S., Misra, S., Khan, T., Bhuyan, M. K., & Laskar, R. H. (2021). Segregation of meaningful strokes, a pre-requisite for self co-articulation removal in isolated dynamic gestures. *IET Image Processing,* 15(5), 1166-1178.

https://doi.org/10.1049/ipr2.12095

Bamwenda, J., & Özerdem, M. S. (2019). Recognition of static hand gesture with using ANN and SVM. *Dicle University Journal of Engineering*, 10, 561-568.

https://doi.org/10.24012/dumf.569357

Beh, J., Han, D., & Ko, H. (2014). Rule-based trajectory segmentation for modeling hand motion trajectory. *Pattern Recognition*, 47(4), 1586-1601.

Bhuyan, M. K., Ajay Kumar, D., MacDorman, K. F., & Iwahori, Y. (2014). A novel set of features for continuous hand gesture recognition. *Journal on Multimodal User Interfaces*, 8(4), 333-343.

https://doi.org/10.1007/s12193-014-0165-0

Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library*. Sevastopol, CA: O'Reilly Media, Inc..

Cheng, J., Shi, D., Li, C., *et al.* (2023). Skeleton-based gesture recognition with learnable paths and signature features. *IEEE Transactions on Multimedia*, 26, 3951-3961.

https://doi.org/10.1109/TMM.2023.3318242

Comaniciu, D., Ramesh, V., & Meer, P. (2003). Kernel-based object tracking. *IEEE Transactions on pattern analysis and machine intelligence*, 25(5), 564-577.

https://doi.org/10.1109/TPAMI.2003.1195991

Elmezain, M., Al-Hamadi, A., Appenrodt, J., & Michaelis, B. (2008). A hidden markov model-based continuous gesture recognition system for hand motion trajectory. In: Proceedings of the 2008 19th international conference on pattern recognition; December 8-11, 2008, Tampa, FL, USA. pp. 1-4.

https://doi.org/10.1109/ICPR.2008.4761080

Elmezain, M., Al-Hamadi, A., & Michaelis, B. (2009). Hand gesture recognition based on combined features extraction. *International Journal of Electrical and Computer Engineering*, 3(12), 2389-2394.

Hsu, C. W., & Lin, C. J. (2002). A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, 13(2), 415-425.

https://doi.org/10.1109/72.991427

Huang, S., & Hong, J. (2011). Moving object tracking system based on camshift and Kalman filter. In: Proceedings of the 2011 International conference on consumer electronics, communications and networks (CECNet); April 16-18, 2011, Xianning, China. pp. 1423-1426.

https://doi.org/10.1109/CECNET.2011.5769081

Kao, C. Y., & Fahn, C. S. (2011). A human-machine interaction technique: hand gesture recognition based on hidden Markov models with trajectory of hand motion. *Procedia Engineering*, 15, 3739-3743.

https://doi.org/10.1016/j.proeng.2011.08.700

Kolsch, M., & Turk, M. (2004). Fast 2d hand tracking with flocks of features and multi-cue integration. In: Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop; June 27-July 2, 2004; Washington, DC, USA. pp. 158-158.

https://doi.org/10.1109/CVPR.2004.345

Li, C., Xie, C., Zhang, B., Chen, C., & Han, J. (2018). Deep Fisher discriminant learning for mobile hand gesture recognition. *Pattern Recognition*, 77, 276-288.

https://doi.org/10.1016/j.patcog.2017.12.023

Misra, S., Singha, J., & Laskar, R. H. (2018). Vision-based hand gesture recognition of alphabets, numbers, arithmetic operators and ASCII characters in order to develop a virtual text-entry interface system. *Neural Computing and Applications*, 29(8), 117-135.

https://doi.org/10.1007/s00521-017-2838-6

Misra, S., & Laskar, R. H. (2019). Development of a hierarchical dynamic keyboard character recognition system using trajectory features and scale-invariant holistic modeling of characters. *Journal of Ambient Intelligence and Humanized Computing*, 10(12), 4901-4923.

https://doi.org/10.1007/s12652-019-01189-2

Pun, C. M., Zhu, H. M., & Feng, W. (2011). Real-time hand gesture recognition using motion tracking. *International Journal of Computational Intelligence Systems*, 4(2), 277-286.

https://doi.org/10.2991/ijcis.2011.4.2.15

Rubine, D. (1991a). Specifying gestures by example. *ACM SIGGRAPH computer graphics*, 25(4), 329-337.

https://doi.org/10.1145/127719.122753

Rubine, D. (1991b). The automatic recognition of gestures [PhD thesis]. Carnegie Mellon University.

Saboo, S., & Singha, J. (2021). Vision based two-level hand tracking system for dynamic hand gestures in indoor environment. *Multimedia Tools and Applications*, 80(13), 20579-20598.

https://doi.org/10.1007/s11042-021-10669-7

Saboo, S., Singha, J., & Laskar, R. H. (2022). Dynamic hand gesture recognition using combination of two-level tracker and trajectory-guided features. *Multimedia Systems*, 28(1), 183-194.

https://doi.org/10.1007/s00530-021-00811-8

Singha, J., Misra, S., & Laskar, R. H. (2016). Effect of variation in gesticulation pattern in dynamic hand gesture recognition system. *Neurocomputing*, 208, 269-280.

https://doi.org/10.1016/j.neucom.2016.05.049

Singla, A., Roy, P. P., & Dogra, D. P. (2019). Visual rendering of shapes on 2D display devices guided by hand gestures. *Displays*, 57, 18-33.

https://doi.org/10.1016/j.displa.2019.03.001

Subasi, A. (2007). Application of adaptive neuro-fuzzy inference system for epileptic seizure detection using wavelet feature extraction. *Computers in biology and medicine*, 37(2), 227-244.

https://doi.org/10.1016/j.compbiomed.2005.12.003

Tang, J., Cheng, H., Zhao, Y., & Guo, H. (2018). Structured dynamic time warping for continuous hand trajectory gesture recognition. *Pattern Recognition*, 80, 21-31.

https://doi.org/10.1016/j.patcog.2018.02.011

Tharwat, A. (2016). Linear vs. quadratic discriminant analysis classifier: a tutorial. *International Journal of Applied Pattern Recognition*, 3(2), 145-180.

https://doi.org/10.1504/IJAPR.2016.079050

Wang, X., & Li, X. (2010, December). The study of MovingTarget tracking based on Kalman-CamShift in the video. In: Proceedings of the 2nd International Conference on Information Science and Engineeringl; October 26-28, 2012; Chongqing, China. pp. 1-4.

https://doi.org/10.1109/ICISE.2010.5690826

Xu, D., Wu, X., Chen, Y. L., & Xu, Y. (2015). Online dynamic gesture recognition for human robot interaction. *Journal of Intelligent & Robotic Systems*, 77(3), 583-596.

https://doi.org/10.1007/s10846-014-0039-4

Yoon, H. S., Soh, J., Bae, Y. J., & Yang, H. S. (2001). Hand gesture recognition using combined features of location, angle and velocity. *Pattern recognition*, 34(7), 1491-1501.

https://doi.org/10.1016/S0031-3203(00)00096-0