



Dynamic hand gesture recognition using combination of two-level tracker and trajectory-guided features

Shweta Saboo^{1,2} · Joyeeta Singha^{1,2} · Rabul Hussain Laskar^{1,3}

Received: 8 December 2020 / Accepted: 17 May 2021 / Published online: 14 June 2021
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2021

Abstract

Hand gesture recognition system helps in development of interface system for entering text in human computer interaction. In this paper, we have presented a hand gesture recognition system designed for dataset consisting of numerals and alphabets in lower case. The proposed system detects the hand with the help of skin color and motion information. Hand tracking is done with the help of two-level tracking system using modified Kanade–Lucas–Tomasi (KLT) tracking algorithm. The existing KLT was not able to track the gesture trajectory once the skin detected becomes less in area resulting in decreased number of points. In this paper, traditional KLT has been modified with a new additional feature to overcome this difficulty. In feature extraction process, a feature matrix consisting of 30 features have been created. Among these 30 features, few features like density-1, density-2, and perimeter efficiency have been introduced and are used for calculating efficiency along with some existing features. Inclusion of new features helps in improving the performance and accuracy of the system. Recognition is done using six classifiers including SVM (Support Vector machine), Decision Tree, Naïve Bayes, k -NN (K nearest neighbor), ANN (Artificial neural Network) and ELM (Extreme learning Machine). The experimental results prove that 89.67% of accuracy is achieved for the recognition of dataset containing both numerals and alphabets. Our proposed system is also compared with two existing literatures and it has been observed that better accuracy is exhibited by the proposed system.

Keywords Hand gestures · Recognition system · Classifiers · Extreme learning machine · Lower-case alphabets · Feature extraction

1 Introduction

Hand gestures are a way to reinforce information by people to express their feelings and thoughts along with information conveying also. Hand gesture helps to express ideas of

people as an option to speech and gives emphasis to various points. Hand gesture recognition systems find wide applications in fields of human computer interaction. Static gestures are used for persons with disabilities. Dynamic gestures also exhibit features which can be helpful for persons with disabilities especially if the hearing and speech disability is after occurrence of some accident. A person can use his hand for making understand the gestures to other person or gestures can also be used for interfacing devices. The process involves detection of an image which separates the relevant data from the image background based on color along with motion. Detection is followed by tracking which corresponds to track the movement of hand in each frame of the segmented hand region according to the movement of hand. Dynamic recognition system faces many challenges like the abstraction of invariant features, movement transition between gestures, automatic segmentation of features, matching techniques, recognition of mixed gestures and gestures occurring in complex backgrounds. In this paper we have tried to recognize numerals along with lower-case

Communicated by Y. Kong.

✉ Shweta Saboo
shweta.saboo.y18pg@lnmiit.ac.in

Joyeeta Singha
joyeeta.singha@lnmiit.ac.in

Rabul Hussain Laskar
rabul18@yahoo.com

¹ Department of Electronics and Communication Engineering, Jaipur, India

² The LNM Institute of Information Technology, Jaipur 302031, India

³ National Institute of Technology, Silchar, Assam 788010, India

alphabets which will be further helpful in formation of words.

A combination of YCbCr and HSV color model was proposed to separate skin-colored pixels from the background. Malima et al. [1] used the ratio of red and green color to determine the skin-colored regions for robotic application. A solution to reduce this problem is to use the background subtraction technique [2]. The first frame of the video was considered as the background for the entire processing of the system. Guo et al. [3] proposed a hand tracking system using a combination of skin filtering and pixel based hierarchical feature AdaBoosting technique, along with background cancelation. Koh et al., used skin and developed a color model which helped in tracking of hand gestures [4]. Elmezain et al. [5] suggested a system to recognize continuous and isolated gestures. Feature extraction stage uses orientation feature which provides motion direction between various trajectory points as output. Kao and Fahn [6] developed a real time hand gesture recognition system using orientation feature in feature extraction stage. Bhuyan et al. [7], recognized hand gestures with the help of four static features along with two dynamic features successfully. Rubine developed 13 features [8] some of them are: cosine and sine of the initial angle with respect to axes, the length of bounding box diagonal, the angle of the bounding box, the distance between first and last point of the trajectory, the cosine and sine of the angle between the first and last point of the trajectory and the total gesture length. Chan et al. [9] used a combination of HMM and RNN which provided better performance compared to the performance of the individual classifiers such as HMM or RNN. Wang et al. proposed a system to recognize gestures using combination of AdaBoost and rotation forest [10]. Some of the 3-D techniques employing combination of multi-view method with deep learning techniques have been used in literature to develop hand gesture recognition systems and similar other applications [26–28].

The contribution of paper is as follows:

- First, a new dataset named “LNMIIT Dynamic Hand Gesture-2” has been created with dynamic hand gestures consisting of numerals and lower-case alphabets are recorded.
- The hand has been detected using a combination of color and motion information. Color information includes skin filtering using conversion to YCbCr color coding and motion information includes 3-frame differencing. A check condition of face and hand area has been introduced to carry out the detection process efficiently.
- A two-level tracking algorithm has been designed using a combination of feature based and color-based tracking system. First level is the Modified KLT algorithm where additional information such as increasing the

number of detected points is used and second level tracking uses Camshift to enhance the performance.

- Feature extraction stage gives a set of 30 features including few new features along with available features of the presented hand gestures without pattern variation and self-co-articulation.
- Gestures are classified using features extracted and providing these features to various individual classifiers like Artificial Neural Network (ANN), Naïve Bayes, Support Vector machine (SVM), ELM and decision tree.

The paper is divided into different sections. Section 2 consists of detailed work done in recent past by various researchers. Section 3 contains detailed characteristics of datasets considered followed by the flowchart of the processes used in detection according to the changed parameters checking the ratio of area and hand. This section also consists of proposed tracking system reducing the problems in existing KLT algorithm and Camshift. Section 4 consists of details of the features extracted from the hand gestures Sect. 5 contains experimental analysis and analysis using various recognition classifiers. Finally, conclusion is presented in last section.

2 Proposed system

Figure 1 shows the proposed block diagram of the system. The details of each step are provided in following subsections.

2.1 Dataset creation by video acquisition

Dynamic videos have been recorded using Logitech C922 Pro Stream Webcam with 360 pixels having an aspect ratio of 16:9 and 30 frames per second. Gestures have been recorded with five users and 900 gestures were recorded. The description of the database is summarized in Table 1. The constraints used for recording gesture in LNMIIT Dynamic Hand Gesture Dataset-2 are:

- The palm should be moving majorly as compared to complete hand and should clearly illustrate the gesture.
- The movement of hand should be smooth and continuous.
- Lightening should be adequate at the time of recording.
- The hand should be kept in a static gesture position for few seconds before the completion of gesture.

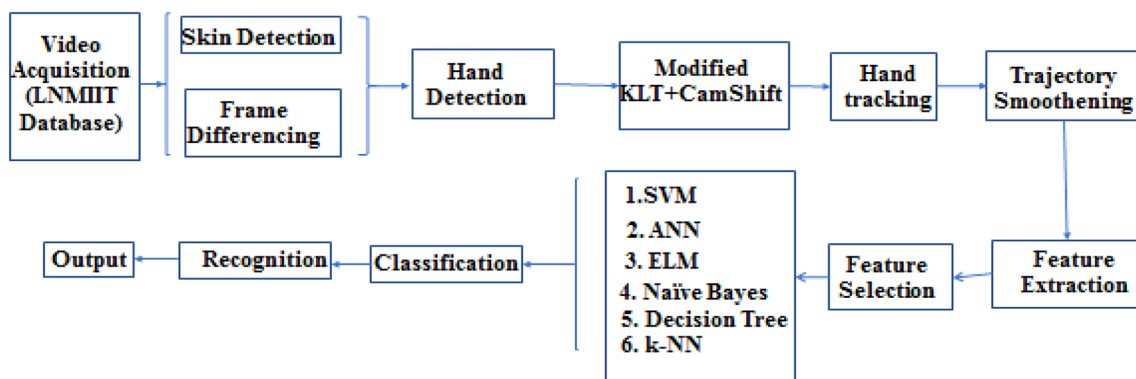


Fig. 1 Proposed model block diagram

Table 1 Dataset details

Dataset details	
Total Gestures	900 gestures
Details	300 gestures (30 gestures each of 0–9 numerals) 600 gestures (30 gestures each of a-z excluding f, i, j, k, t and x)
Number of Users	5
Acquisition device	Logitech C922 Pro Stream Webcam
Resolution	360 pixels with 16:9 aspect ratio
Training gestures	300
Testing gestures	600

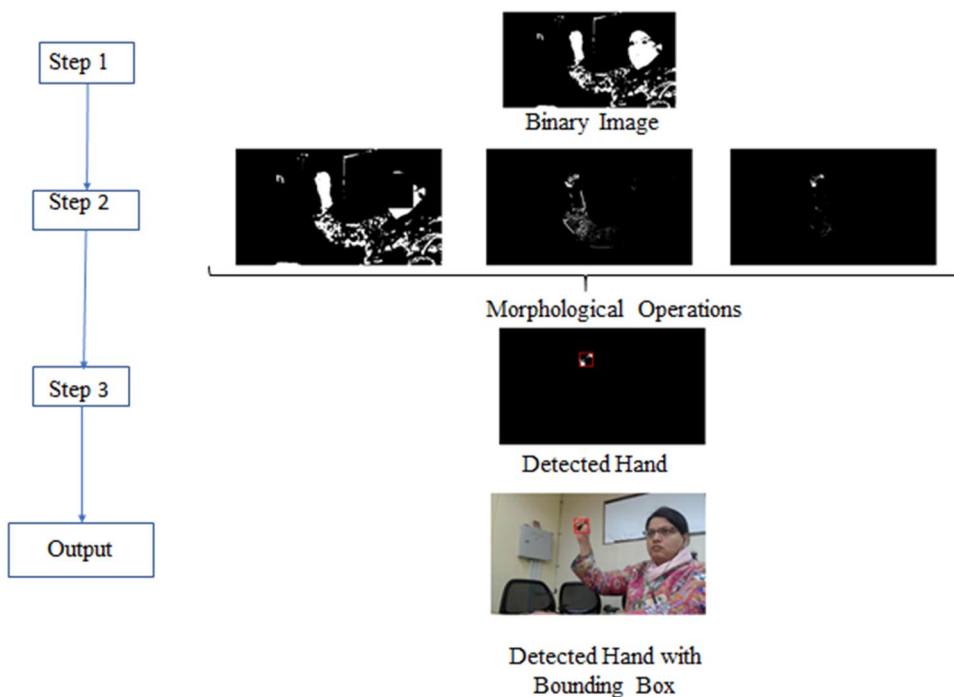
The first important step in the hand gesture recognition is removal and detection of hand from background. Figure 2 shows the step by step process of operations being carried out by the detection process. Following are the three steps to obtain the desired hand:

- Face detection followed by skin filtering
- Three-frame differencing for colored frames
- Three-frame differencing for grayscale frames

Firstly the 3rd frame of the video is provided as an input. Face is detected using Viola-Jones algorithm [11] and then this information is used to remove face. Then skin filtering is done, which results in generating skin color objects in

2.2 Hand detection

Fig. 2 Detection by skin filtering and three-frame differencing



the frames according to the values of Y, Cb and Cr. 3-frame differencing is done between the first three frames, firstly between the first and third frame and secondly between third and fifth frame of both grayscale and binary images. Morphological operation (OR) is being carried out of the binary images obtained after the differencing is being carried out (Fig. 3).

2.3 Tracking

The next step is to track the hand after successful hand detection. With dynamic existence, many videos have been taken into account. Existing tracking algorithms do not yield adequate results and therefore there is a need to develop an algorithm which overcomes with the difficulties of existing tracking routines like KLT, Camshift etc.

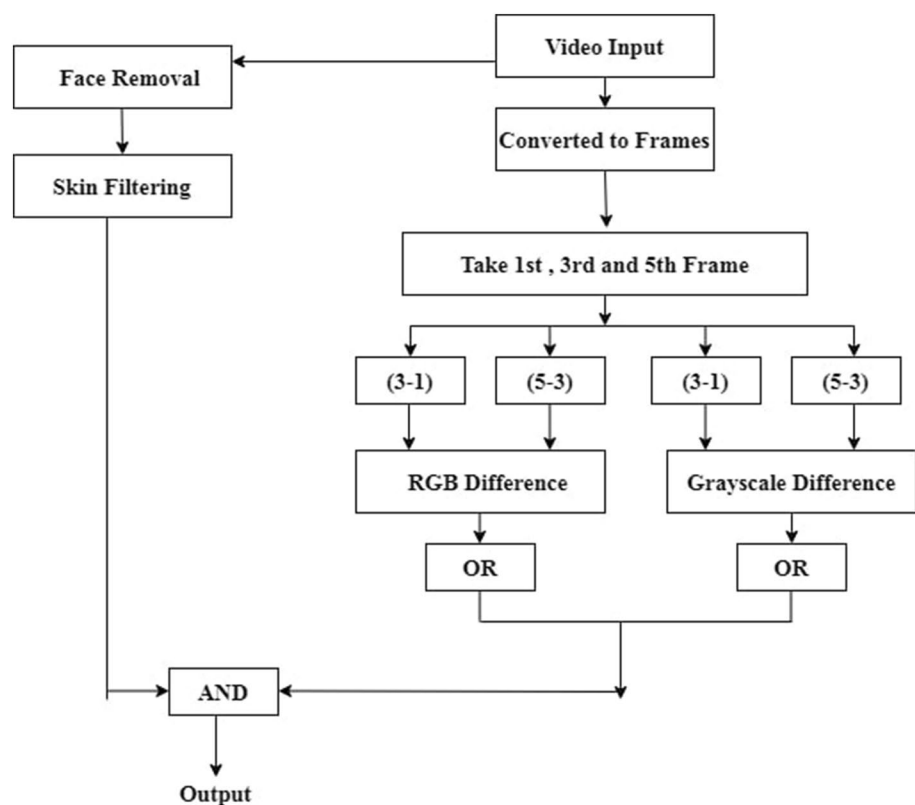
The traditional KLT [12] has been used by various researchers and it has been observed that it makes use of spatial intensity information for searching the position yielding best match. The KLT algorithm effectively detects the gesture until the condition occurs when within the detected area there are at least two visible points. It is necessary to develop an algorithm that takes care of it when the measurable points are reduced by more than 2 in number.

The Camshift algorithm [13] uses color histogram of moving target as target mode and is thus known as target tracking algorithm. Camshift algorithm when used as mean

shift algorithm has advantages like simplicity and fast speed to process and converge. The window size of this algorithm is constant so that the object location cannot be exactly right even if the size of object changes and hence tracking loses the path sometimes. In the proposed system detected hand initializes the tracking window thus making the system more efficient and automatic. Detection should be proper to set the tracking window properly as initial tracker will decide complete trajectory (Fig. 4).

After the initialization of tracking region, proper selection of features is essential. Existing KLT uses eigen features to track the object. As the video moves further, detected Eigen points starts decreasing and a time comes when tracking is lost due to loss of all points. Moreover, change in hand shape also results in loss of eigen features, whereas Camshift tracker also loses its path when there are some skin-colored objects causing occlusion. To reduce these challenges, a modified system is defined using the re-detection of hand according to the newly defined area. So, when the points reduce in number, detection is performed again to eliminate the issue of tracking at reduced points as soon as the number of observable points decreases. Until detection, area is doubled so that the resulting area is greater to reach the visible points and then skin filtering and 2-frame differencing is done. Logical AND process is performed between the binary and RGB differenced frames. Then the double area bounding box is inserted into the current frame and new eigen features

Fig. 3 Flowchart of detection process



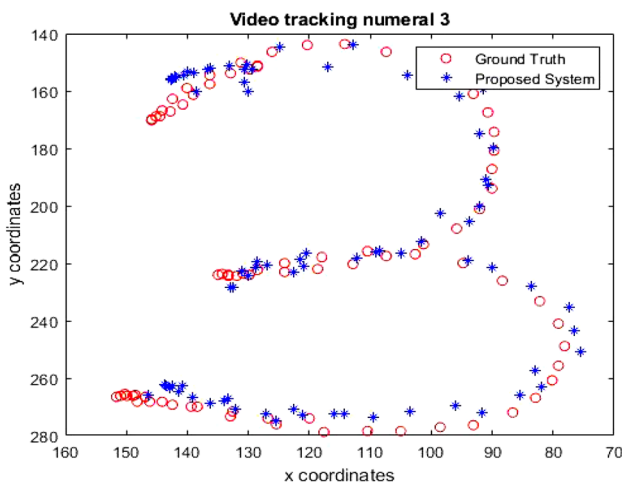


Fig. 4 Trajectory showing tracking of numeral “3”

are identified and points are inserted afterwards. Tracker is run again, and visible points are counted and then the complete process of KLT is again performed. To form the gesture trajectory, centroids of the detected region is marked and then the path can be monitored.

2.3.1 Overall tracking algorithm

1. Read the video frame along with the detected region
2. Detect the Eigen Features
3. Initialize Tracker
4. No. of detected points (Count) = 0
5. LOOP: If Count > 10 then
 - i. Change the detected area according to the hand position
 - ii. Insert Points
 - iii. Checking the visible points lying inside the detected area Count = Count + 1
 - Else
 - i. Double the detected area
 - ii. Skin Filtering
 - iii. 2 Frame Differencing
 - iv. A = Difference of 2 RGB frames
 - v. B = Difference of 2 Binary Frames
 - vi. C = AND(A,B)
 - vii. Insert a rectangle around the area
 - viii. Detect new features
 - ix. Insert points and initialize tracker
 - x. Count Visible Points
6. If Count > 10 then go to LOOP

The proposed system is tested on all the input videos which are taken into consideration and trajectory is formed and trajectory points are stored in matrix form for feature extraction.

2.4 Trajectory smoothening

For the tracked region to display gesture we need to calculate the centroid of each bounding box which is being used to enclose the hand being detected and tracked. The centroid points are being stored of each frame which is obtained by dividing the recorded video into frame. To obtain gesture trajectory all the centroid points of the bounding boxes of consecutive frames are joined together. Due to the uneven movements of hand the gesture trajectory is uneven and hence we need to smoothen the gesture trajectory. Small amount of noise is present in starting of each gesture due to the starting movement of hand as the detection is done with the help of motion produced by the hand [14]. This noise is reduced by averaging starting 4–5 points of the gesture obtained. Rest of the trajectory is smoothened by substituting each centroid point with the mean value of current centroid point, previous centroid point and next centroid point given as

$$(X_c, Y_c) = ((x_{i-1} + x_i + x_{i+1})/3), ((y_{i-1} + y_i + y_{i+1})/3) \tag{1}$$

Smoothened trajectory is finally obtained $\{(x_{c1}, y_{c1}) \dots \dots \dots\}$.

Figure 5 shows the output of various trajectories formed of numerals as well as alphabets. The difference between the trajectory before and after smoothening can be observed. At the time of recording of the various sequences, a movement of hand is required as the detection process is done with the help of motion as well as frame differencing.

2.5 Feature extraction

A combination of twenty-three existing features and seven new features were used to develop a new feature set in our proposed system. Each feature is defined using a formal mathematical description or the formalized algorithm is used to calculate the feature. Description of all the features is given in the following subsections:

2.5.1 Existing features

- Start point and end point location: After finding the trajectory points by smoothening, start and end point location is obtained by dividing the area into four parts. It is being checked that the start location of the trajectory lies in which of the four quadrant and end of the




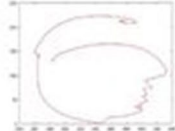
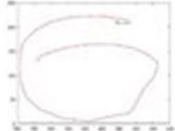




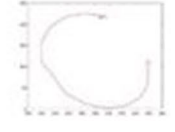



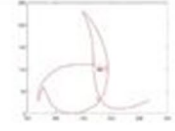
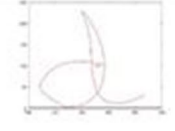



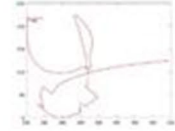
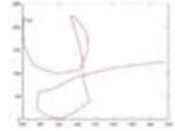
Starting frame	Middle frame	Last frame	Trajectory before Smoothing	Trajectory After Smoothing
				
				
				
				

Fig. 5 Tracking of numerals and lower-case alphabets

hand location lies in which quadrant and hence first two features specifying starting and ending point are derived [15].

- **Ellipse features:** Next feature is finding the chain code by calculating the angle of ellipse orientation [15]. Given measurements are used to fit best ellipse. Equation of ellipse which needs to be used can be given in its mathematical form as:

$$\text{Ellipse} = a * x^2 + b * x * y + c * y^2 + d * x + e * y + f = 0 \quad (2)$$

- **Bounding box parameters:** Bounding Box is used to define and describe target location. Bounding Box can be defined as rectangular box determined by x - y coordinates in the leftmost upper corner and the origin is at the rightmost lower corner. Dimensions of Bounding box change according to the coordinates of the estimated trajectory (Fig. 6).

As bounding box is one of the important results of the object detection, parameters of the bounding box can be taken as important features for recognition purposes [23]. Bounding Box area can be calculated with the help of width and height of the bounding box. Diagonal length of bounding box can be calculated as

$$D = \sqrt{a^2 + b^2} \quad (3)$$

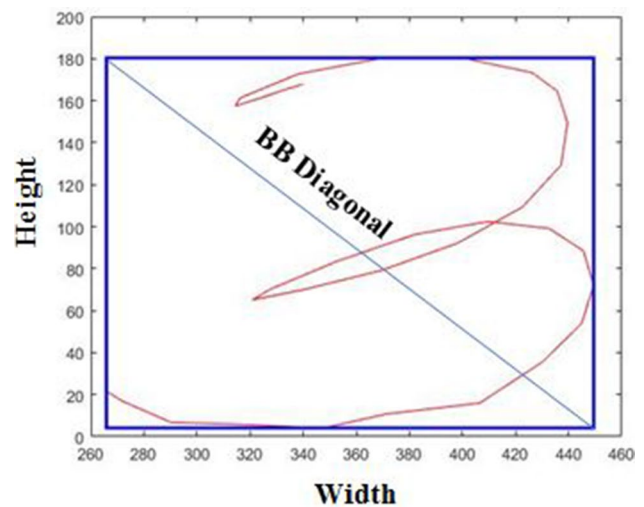


Fig. 6 Bounding box parameters

- **Distance feature:** Along with the no. of points, distance feature is also calculated [24]. Distance feature is calculated as the Euclidean distance between the current point and the next closest point and the distance is being added up and the feature is being calculated as

$$f = \sqrt{(x_{p-1} - x_o)^2 + (y_{p-1} - y_o)^2} \tag{4}$$

- Angle features: For recognizing the gesture properly, total angle traversed should be calculated as it will be similar in same gestures while different in different gestures [23].
- Statistical features: Gestures can be classified using statistical features like variance and mode. Standard deviation is calculated for matrices which gives Y as a row vector having standard deviation of each column. Mode is also one of the important statistical features for vector X and Y which computes M as the sample mode which is the most frequently occurring values in X and Y [8].
- Close figure test: Close figure test helps in distinguishing between gesture having nearby end point and far end points [18]. Two dimensional features are obtained as output using this test as:

$$CFT_x = \frac{x_{end} - x_{start}}{\text{Trajectory Length}}; CFT_y = \frac{y_{end} - y_{start}}{\text{Trajectory Length}} \tag{5}$$

- Convex hull features: Convex hull can be defined as the shape of the smallest convex set which consist it. To calculate convex hull, point with minimum x coordinate value or the leftmost point is taken as the starting point and points are being wrapped up in counterclockwise direction [24]. Ratio of bounding box area and convex hull area gives output as the next feature.
- Location feature: Location feature extracts the feature which measures the distance between the center of gravity and the points in a gesture trajectory [24]. Center of gravity is given as,

$$x_c = \left(\frac{1}{N}\right) \sum x_i; y_c = \left(\frac{1}{N}\right) \sum y_i \tag{6}$$

$$L_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} \tag{7}$$

- Length ratio: Length ratio uses the cumulative distance property by adding the distance between two successive points to the previous distance and the ratio between the distances traversed and the cumulative distance calculated by taking the square root of the respective cumulative distances [24].

2.5.2 New features

A set of new features have been proposed to improve the accuracy of the recognition of gestures when combined with existing gestures. Features proposed have been used keeping in mind the movement of hand while recording the gesture videos.

- Density-1: Density 1 is being calculated by first finding out the complete distance traversed by the trajectory and dividing it by distance between first and last point.

$$\frac{\sum_{i=1}^n \sqrt{[x(i+1) - x(i)]^2 + [y(i+1) - y(i)]^2}}{\sqrt{(x_{p-1} - x_o)^2 + (y_{p-1} - y_o)^2}} \tag{8}$$

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 is considered to be a measure of density.

- Density-2: This feature takes bounding box dimensions in consideration for calculating the area of bounding box and dividing the length of trajectory by the area of the bounding box.

$$\frac{\text{Area of the Bounding box}}{\sum_{i=1}^n \sqrt{[x(i+1) - x(i)]^2 + [y(i+1) - y(i)]^2}} \tag{9}$$

- Minimum bounding rectangle: This feature finds the average of the rectangles bounding the various trajectory points. Area of the minimum bounding rectangle is calculated by finding the ratio of difference between each trajectory and minimum value by the difference between maximum value of trajectory points and minimum value of trajectory points (Fig. 7).

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

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

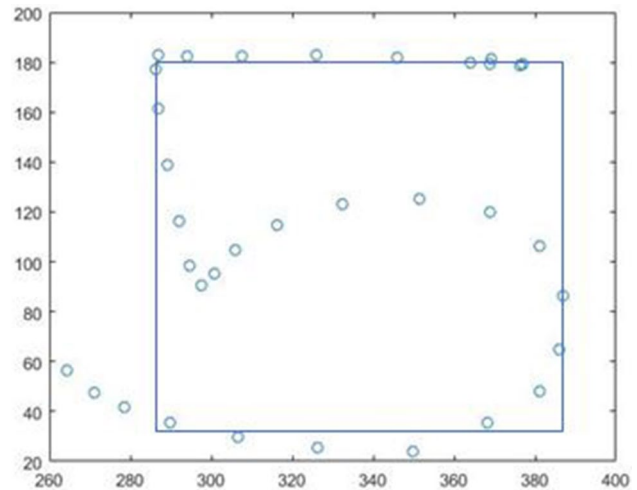


Fig. 7 Feature showing minimum bounding rectangle (for number 5)

$$x_t = \frac{X_t - x_{\min}}{x_{\max} - x_{\min}}; y_t = \frac{Y_t - y_{\min}}{y_{\max} - y_{\min}} \quad (10)$$

$$\text{mbr} = x_t * y_t \quad (11)$$

- Perimeter efficiency: To find the value of this feature all the 2D coordinates are used to create alpha shape and their perimeter is being calculated. Area of the alpha shape when multiplied by pi and divided by the perimeter gives the perimeter efficiency as:

$$\text{PE} = 2 * \frac{\sqrt{\pi * A}}{P}, \quad (12)$$

where A and P represent the area and perimeter, respectively. The selected value of alpha radius gives a scalar quantity specifying the radius of the alpha disk or sphere used to recover the alpha shape.

- Angle between points along trajectory: It is a feature which calculates the dot product of both the available set of points along the trajectory with the norm values. Then inverse cosine resulting indegrees gives the value of this feature (Fig. 8).
- Pairwise distribution between two set of trajectory points: This feature finds out the pairwise distance between two sets of trajectory points which returns values containing the Euclidean distances between each set of points. Among various options available when finding pairwise distance, one minus the sample linear correlation between trajectory points is used and all the values are treated as sequences of values. The squared Euclidean distance between the point $p = (p_1, p_2, \dots, p_n)$ and the point $q = (q_1, q_2, \dots, q_n)$ is the sum of the squares of the differences between the components: $\text{Dist}^2(p, q) = \sum_i (p_i - q_i)^2$. The Euclidean distance is then the square root of $\text{Dist}^2(p, q)$.

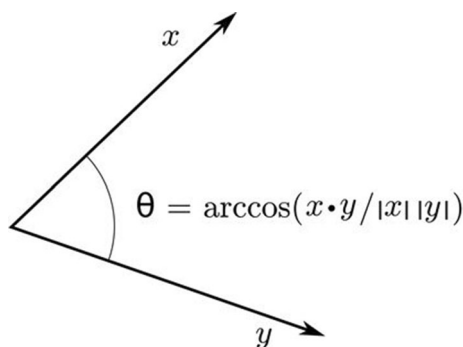


Fig. 8 Angle between points along trajectory

3 Recognition

3.1 SVM

Support vector machine (SVM) can be used to classify for both supervised as well as unsupervised learning. SVM with supervised learning type of classifier can be used in multiple pattern recognition problems to classify and identify gestures being produced. Multiclass SVM can be used as a classifier for gesture trajectory guided recognition. SVM helps to separate the linearly non separable data as the data was projected to high dimensional data so that the error is being reduced [17].

3.2 k-NN algorithm

K nearest neighbor (KNN) is a classifier which solves both classification and regression problems using supervised machine learning methods. Training and testing has been done for different values of K and to select correct and appropriate value of K , KNN algorithm needs to be run several times and the value of K which reduces the number of errors while making predictions accurately is being selected [19]. Value of K is selected to be odd if the numerals of classes are odd to avoid the situation of draw of votes.

3.3 Naïve Bayes and decision tree

Naïve Bayes classifications have the fact that given class participation there is some multivariate appropriation in the perceptions; however, the indicator or highlights creating the perception are autonomous [18]. This type of classifier is used when within each class the predictors available are independent of each other. Decision trees, also known as classification or regression trees are used to predict responses to data. Tree starts from the root and then is divided into leaf nodes down the tree. The decision starts from the beginning node and moves down to leaf node for response prediction. Finally, the response is stored in the leaf node. Output of classification tree can be obtained in the form of true or false while regression tree gives output in the form of numeric responses [21].

3.4 ANN

Artificial neural network is one of the gesture recognition approaches being used by various researchers. ANN comprises of input, hidden and output layer with neurons used according to the available dataset. It can be defined as fully connected multi-layer neural networks. Each node of one layer relates to all nodes available in next layer. The

weighted sum of each node’s inputs is being calculated and output is provided by passing these through some non-linear activation function [20]. Dataset is being trained and tested with changing the number of hidden neurons.

3.5 ELM

Extreme machine learning based classification is the newly developed algorithm which trains a neural network having a single hidden layer. Its structure consists of hidden nodes in a single layer and the weights of inputs and hidden nodes are assigned randomly and are kept constant during training, testing and prediction phases [22]. At the time of classification ELM type along with the number of hidden neurons needs to be specified. In hidden neurons, piecewise continuous functions like sigmoid, hardlimit etc. can be used.

3.6 Results

Feature extraction stage provides feature set as output. This feature matrix is given as input to the various recognition classifiers. The performance of different classifiers is provided in Table 2. Table 2 shows the results obtained using SVM classifier along with different kernel functions. It can be observed that SVM provides highest accuracy with polynomial function as kernel with an accuracy of 97% in case of numerals, 92.5% in alphabets and 89.67% for the dataset containing both numerals and alphabets. In case of k-nearest neighbor algorithm the accuracy has been calculated for various odd values of *K* like 1, 3, 5, 7 and 9. The train and test accuracies of the dataset have been calculated using *k*-NN classifier for different *k* values. The highest accuracy of 96% has been achieved for *k* having

Table 2 Results of accuracy of different classifiers

		Numerals	Alphabets	Numerals + Alphabets			
SVM	Linear SVM	94%	74%	59%			
	RBF SVM	94%	87%	81.33%			
	Gaussian SVM	94%	87%	81.67%			
	Polynomial SVM	97%	92.50%	89.67%			
K-NN	K=1	96%	94%	91.33%			
	K=3	95%	92%	87%			
	K=5	96%	91%	86.67%			
	K=7	91%	91%	84.67%			
	K=9	89%	89%	81.67%			
ANN	No. of Hidden neurons	96%	90%	74%			
	10	97%	90.50%	80.30%			
	15	94%	91%	80.70%			
	20	96%	95.50%	82.70%			
	25	98%	95.50%	86.70%			
	30	95.2%	94.6%	82.3%			
Naive Bayes		96.67%	94.17%	88.89%			
DecisionTree		92%	85.50%	85.67%			
		Numerals		Alphabets		Numerals+Alphabets	
No. of Hidden neurons (Activation Function)		Training Accuracy	Testing Accuracy	Training Accuracy	Testing Accuracy	Training Accuracy	Testing Accuracy
ELM	10 (sig)	73.85%	75.32%	88.49%	84.57%	88.46%	85.59%
	10 (hardlim)	87.45%	87.91%	88.17%	84.60%	88.05%	85.52%
	20 (sig)	82.56%	83.74%	86.91%	83.90%	87.80%	85.80%
	20 (hardlim)	83.43%	83.70%	87.47%	84.48%	88.65%	86.15%
	30 (sig)	76.83%	75.08%	86.79%	86.33%	87.98%	85.89%
	25 (hardlim)	83.33%	87.54%	84.01%	84.75%	86.04%	83.69%

value 5. In case of dataset with alphabets highest accuracy achieved is for value of k as 1 as 94%. For $k=5$, accuracy value can be calculated as 86.67% for dataset having both alphabets and numerals.

Next, ANN classifier is used for recognition of the gestures of the dataset. The train and test accuracies were calculated for different hidden neuron units of ANN. The highest accuracy was observed for the network structure having 25 hidden neurons. Value of accuracies comes out to be 98% for numerals, 95.5% for alphabets and 86.7% for combination dataset of alphabets and numerals. Accuracy obtained in the case of ANN is better as compared to SVM and k -NN classifiers.

The results for Naïve Bayes classifier are shown in Table 2. It can be observed that best results were obtained for the numeral dataset as 96.67% for the normal kernel case. However, for dataset comprising of both numerals and alphabets accuracy is improved to a value of 88.89% which is better as compared to ANN and k -NN classifiers. The next classifier used for recognition of gestures is decision tree classifier. The training and testing accuracies obtained in this case is less as compared to other classifiers. Highest accuracy among the three datasets used is of dataset comprising numerals and its value comes out to be 92%. In this classifier it is also observed that combination dataset exhibits better accuracy as compared to dataset with alphabets.

Another classifier which is employed to compare the test gestures with predicted gestures is ELM. This extreme machine learning technique calculates the training and testing accuracy with the help of hidden neurons along with different activation functions. ELM classifier offers very fast and advanced machine learning technique and gives the best output for hardlimit activation function operating for 20 hidden neurons and gives 88.65% training accuracy and 86.15% testing accuracy for dataset consisting of both numerals and alphabets. From the analysis of results of individual classifiers and the representation shown in Fig. 9, performance of the classifiers can be arranged as SVM > Naïve Bayes > ANN > KNN > ELM > Decision Tree.

Singha et al. [25] proposed a system which was applied on the dataset recorded with colored marker consisting of Numerals and alphabets. Another system was proposed by Misra et al. [16] with a dataset with ASCII characters also included in it. Various classifiers like SVM, ANN, k -NN and naïve bayes have been used in these literatures to calculate the recognition accuracy. Algorithms used by them were applied on the “LNMIIT Dynamic Hand Gesture Dataset-2” and it has been observed that accuracy calculated using the proposed algorithm was better as compared to the algorithms used in the above stated papers. Figure 10 shows the graphical representation of this comparison process.

4 Conclusion

In this paper, a hand recognition system is developed which can be utilized for various applications of human computer interaction. A new dataset named “LNMIIT Dynamic Hand Gesture Dataset-2” have been created with dynamic gestures containing numerals and alphabets in lowercase. The users were required to gesticulate according to the required gestures. In the proposed system, detection process is initialized with skin detection and combined with three-frame differencing. A checking condition of hand and face area is utilized to decide upon the number of frames used in motion information. Hand tracking is done by modified KLT which tracks Eigen points again in increased area if tracking is lost. New features like density-1, angle between trajectory points and perimeter efficiency have been introduced and when used with few existing features which results in better accuracy. It has been observed that numerals and lower-case alphabets exhibit better recognition accuracy. The performance of the system was evaluated for different individual classifiers such as SVM, k -NN, Naïve Bayes, ANN, Decision Tree and ELM. An accuracy of 98% has been achieved for numerals, 95.5% for alphabets in lower case and 89.67% for mixture of both the numerals and lower-case alphabets. Few gestures like e, f, i, j, k, t and x have not been considered due to self-co-articulation which will be taken care in future. The issues related to complex and dynamic backgrounds needs to be considered also to make system robust and accurate.

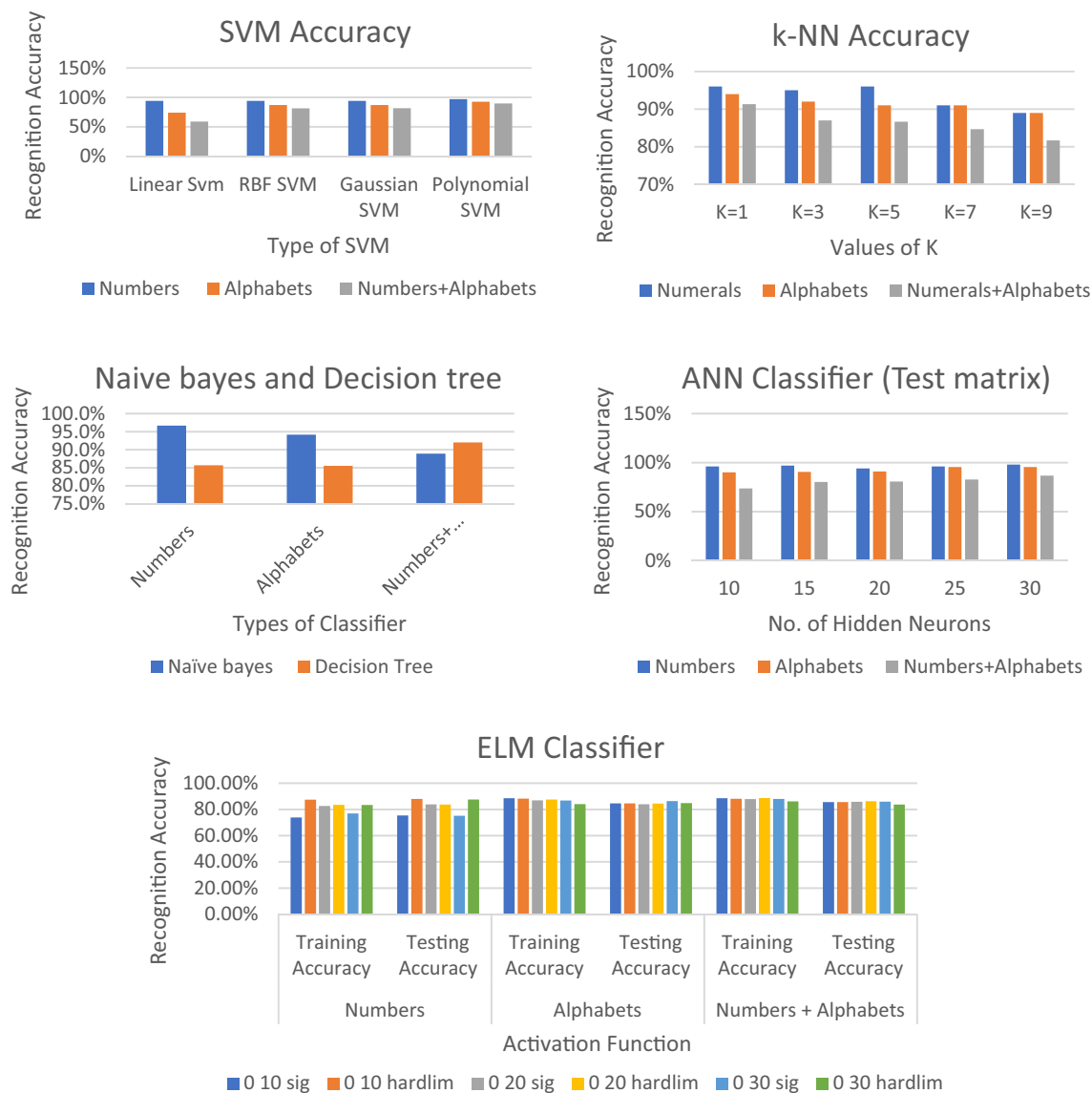


Fig. 9 Graphs showing accuracy of various classifiers

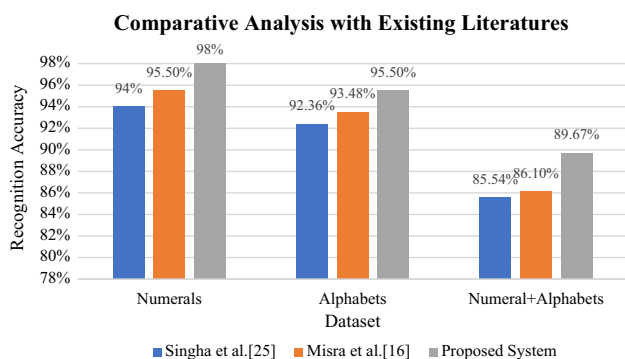


Fig. 10 Graphical representation of comparative analysis

Acknowledgements This work is supported by DST (Govt. of India) under the SEED Division [SP/YO/407/2018].

References

1. Malima, A., Ozgur, E., Cetin, M.: A fast algorithm for vision-based hand gesture recognition for robot control. In: Proceedings of 14th IEEE Conf. on signal processing and communications applications, antalya, pp. 1–4. (2006)
2. Rehg, JM., Kanad, T.: Digiteyes: vision-based hand tracking for human-computer interaction. In: Proceedings of 1994 IEEE workshop on motion of non-rigid and articulated objects, pp. 16–22. IEEE (1994)

3. Guo, J.M., Liu, Y.F., Chang, C.H.: Improved hand tracking system. *IEEE Trans. Circuits Syst. Video Technol.* **22**(5), 693–701 (2012)
4. Koh, E., Won, J., Bae, C.: On-premise skin color modeling method for vision-based hand tracking. In: 2009 IEEE 13th international symposium on consumer electronics, pp. 908–909. IEEE (2009)
5. Elmezain, M., Ayoub, A.-H., Jorg, A., Bernd, M.: A hidden markov model-based continuous gesture recognition system for hand motion trajectory. In: 2008 19th international conference on pattern recognition, pp. 1–4. IEEE (2008)
6. Kao, C.Y., Fahn, C.S.: A human-machine interaction technique: hand gesture recognition based on hidden Markov models with trajectory of hand motion. *Procedia Eng.* **15**, 3739–3743 (2011)
7. Bhuyan, M.K., Bora, P.K., Ghosh, D.: Trajectory guided recognition of hand gestures having only global motions. *Int. J. Comput. Sci.* **2**(9), 753–764 (2008)
8. Rubine, D.: Specifying gestures by example. *Computer graphics (SIGGRAPH '91 Proceedings)*, 25(4):329–337 (1991)
9. Ng, C.W., Ranganath, S.: Real-time gesture recognition system and application. *Image Vis. Comput.* **20**, 993–1007 (2002)
10. Wang, G.W., Zhang, C., Zhuang, J.: An application of classifier combination methods in hand gesture recognition. *Math. Probl. Eng.* **2012**, 1–17 (2012)
11. Viola, P., Jones, M.J.: Robust real-time face detection. *Int. J. Comput. Vis.* **57**(2), 137–154 (2004)
12. Lee, H.-K., Choi, K.-W., Kong, D., Won, J.: Improved Kanade-Lucas-Tomasi tracker for images with scale changes. In: Proceedings of the IEEE international conference on consumer electronics, pp. 33–34. Berlin, Germany (2013)
13. Yu, Y., Bi, S., Mo, Y., Qiu, W.: Real-time gesture recognition system based on Camshift algorithm and Haar-like feature. In: 2016 IEEE international conference on cyber technology in automation, control, and intelligent systems (CYBER), pp. 337–342. IEEE (2016)
14. Singha, J., Laskar, R.H.: Self co-articulation detection and trajectory guided recognition for dynamic hand gestures. *IET Comput. Vis.* **10**(2), 143–152 (2016)
15. Bhuyan, M.K., Kumar, D.A., MacDorman, K.F., Iwahori, Y.: A novel set of features for continuous hand gesture recognition. *J. Multimod. User Interfaces* **8**(4), 333–343 (2014)
16. Misra, S., Singha, J., Laskar, R.H.: Vision-based hand gesture recognition of alphabets, numerals, arithmetic operators and ASCII characters in order to develop a virtual text-entry interface system. *Neural Comput. Appl.* **29**(8), 117–135 (2018)
17. Wang, Z., Xue, X.: Multi-class support vector machine. In: Ma, Y., Guo, G. (eds.) *Support vector machines applications*, pp. 23–48. Springer International Publishing, New York (2014)
18. McCue, R.: A comparison of the accuracy of support vector machine and Naive Bayes algorithms. In: *Spam classification*. University of California, Santa Cruz (2009)
19. Liu, Y., Wang, X., Yan, Ke.: Hand gesture recognition based on concentric circular scan lines and weighted K-nearest neighbor algorithm. *Multimed. Tools Appl.* **77**(1), 209–223 (2018)
20. Singha, J., Roy, A., Laskar, R.H.: Dynamic hand gesture recognition using vision-based approach for human-computer interaction. *Neural Comput. Appl.* **29**(4), 1129–1141 (2018)
21. Saha, S., Ganguly, B., Konar, A.: Gesture recognition from two-person interactions using ensemble decision tree. In: *Progress in intelligent computing techniques: theory practice, and applications*, pp. 287–293. Springer, Singapore (2018)
22. Lu, D., Yuanlong, Y., Huaping, L.: Gesture recognition using data glove: An extreme learning machine method. In: 2016 IEEE international conference on robotics and biomimetics (ROBIO), pp. 1349–1354. IEEE (2016)
23. Paulson, B., Rajan, P., Davalos, P., Gutierrez-Osuna, R., Hammond, T.: What!?! no rubine features?: using geometric-based features to produce normalized confidence values for sketch recognition. In: *HCC workshop: sketch tools for diagramming*, pp. 57–63 (2008)
24. Blagojevic, R., Chang, S.H.-H., Plimmer, B.: The power of automatic feature selection: rubine on steroids. In: *Proceedings of the seventh sketch-based interfaces and modeling symposium. SBIM 10*. Eurographics association, Aire-la-Ville, Switzerland, pp. 79–86 (2010)
25. Singha, J., Laskar, R.H.: Hand gesture recognition using two-level speed normalization, feature selection and classifier fusion. *Multimed. Syst.* **23**(4), 499–514 (2017)
26. Yan, C., Gong, B., Wei, Y., Gao, Y.: Deep multi-view enhancement hashing for image retrieval. *IEEE Trans. Patt. Anal. Mach. Intell.* (2020). <https://doi.org/10.1109/TPAMI.2020.2975798>
27. Yan, C., Shao, B., Zhao, H., Ning, R., Zhang, Y., Feng, Xu.: 3D room layout estimation from a single RGB image. *IEEE Trans. Multimed.* **22**(11), 3014–3024 (2020)
28. Yan, C., Li, Z., Zhang, Y., Liu, Y., Ji, X., Zhang, Y.: Depth image denoising using nuclear norm and learning graph model. *ACM Trans. Multimed. Comput. Commun. Appl.* **16**(4), 1–17 (2020)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.