Chapter

Cognitive Visual Tracking of Hand Gestures in Real-Time RGB Videos

Richa Golash and Yogendra Kumar Jain

Abstract

Real-time visual hand tracking is quite different from commonly tracked objects in RGB videos. Because the hand is a biological object and hence suffers from both physical and behavioral variations during its movement. Furthermore, the hand acquires a very small area in the image frame, and due to its erratic pattern of movement, the quality of images in the video is affected considerably, if recorded from a simple RGB camera. In this chapter, we propose a hybrid framework to track the hand movement in RGB video sequences. The framework integrates the unique features of the Faster Region-based Convolutional Neural Network (Faster R-CNN) built on Residual Network and Scale-Invariant Feature Transform (SIFT) algorithm. This combination is enriched with the discriminative learning power of deep neural networks and the fast detection capability of hand-crafted features SIFT. Thus, our method online adapts the variations occurring in real-time hand movement and exhibits high efficiency in cognitive recognition of hand trajectory. The empirical results shown in the chapter demonstrate that the approach can withstand the intrinsic as well as extrinsic challenges associated with visual tracking of hand gestures in RGB videos.

Keywords: hand tracking, faster R-CNN, hand detection, feature extraction, scale invariant, artificial neural network

1. Introduction

Hand Gestures play very significant roles in our day-to-day communication, and often they convey more than words. As technology and information are growing rapidly in every sector of our life, interaction with machines has become an unavoidable part of life. Thus, a deep urge for natural interaction with machines is growing all around [1, 2]. One of the biggest accomplishments in the domain of Hand Gesture Recognition (HGR) is Sign language recognition (SLR) where machines interpret the static hand posture of a human standing in front of a camera [3]. Recently, implementation of HGR-based automotive interface in BMW cars is very much appreciated. Here, five gestures are used for contactless control of music volume and incoming calls while driving [4]. Project Soli is the ongoing project of Google's Advanced Technology; in this project a miniature radar is developed that understands the real-time motion of the human hand at various scales [5].

Hand gestures are very versatile as they comprise static as well as dynamic characteristics, physical as well as behavioral characteristics, for example, movement in any direction, fingers can bend to many angles. Hand skeleton has a complex structure with a very high freedom factor, and thus its two-dimensional

RGB data sequence has unpredictable variations. Visual recognition of dynamic hand gestures is complex because the complete process requires the determination of hand posture along with a cognitive estimation of the trajectory of motion of that posture [3, 6–9]. Due to these intricacies to date, vision-based HGR applications mainly dominate with static hand gesture recognition.

2. Challenges in online tracking hand motion

In context with computer vision and pattern recognition, a human hand is described as a biological target with a complex structure. Uneven surface, broken contours, and erratic pattern of movement are some of the natural characteristics that complicate DHGR [10]. Thus, in comparison to the other commonly tracked moving object, a hand is a non-rigid subtle object and covers a very small area in the image frame. The scientific challenges accompanied in the online tracking of the hand region in an unconstrained environment in RGB images captured using a simple camera are categorized as follows: [3, 4, 6–11].

i. Intrinsic Challenges: Intrinsic challenges are related to a target that is "Hand" physical and behavioral nature. The features such as

   Hand Appearance: The number of joints in the hand skeleton, the appearance of the same hand posture has a large variation, known as shape deformation. Different postures have a wide difference in occupancy area in an image frame, and some postures only cover 10% of the image frame, which is a very small target size in computer vision. In a real-time unconstrained environment, the two-dimensional (2-D) posture shows large variation during movement.

   Manner of Movement: There is a large diversity among human beings in performing the gesture of the same meaning, in terms of speed and path of movement. The moving pattern of the hand is erratic, irregular, and produces blur in the image sequence. Furthermore, the two-dimensional data sequence of a moving hand is greatly affected by background conditions, thus tracking and interpretation of dynamic hand gestures are a challenging task in the HGR domain. The unpredictable variation in target trajectory makes the detection and classification process complex in pattern recognition.

ii. Extrinsic Challenges: These challenges mainly arise due to the environment in which the hand movement is captured. Some of the major factors that deeply impact the real-time visual tracking of the dynamic hand gestures are as follows:

   Background: In the real-time HGR applications, backgrounds are unconstrained, we cannot use fixed background models to differentiate between the foreground and the background. Thus, the core challenge in the design of a real-time hand tracking system is the estimation of discriminative features between background and target hand posture.

   Illumination: Illumination conditions in real-time applications are uneven and also unstable. Thus, 2-D (two-dimensional) projection of the 3-D (three-dimensional) hand movement produces loss of information in RGB images. This loss is the major reason for errors in the visual tracking of hand movement.

Presence of other skin color objects in the surroundings: The presence of objects with similar RGB values such as the face, neck, arm, etc., is the serious cause for track loss in the RGB-based visual tracking techniques.

3. Components of DHGR

There are four main components in cognitive recognition of dynamic hand gestures [3, 10–12].

i. Data Acquisition.

ii. Interest Region Detection.

iii. Tracking of Interest Region.

iv. Classification of Trajectory.

In Dynamic Hand Gesture Recognition (DHGR), acquisition of signals plays a very important role in deciding the technique to recognize and deduce the hand pattern into meaningful information. Contact-based sensors and contactless sensors are two main types of sensors to acquire hand movement signals. Contact-based sensors are those sensors that are attached to the body parts of a user example. Data gloves are hand gloves, accelerometers are attached the arm region, and egocentric sensors are put on the head to record hand movement. Wearable sensor devices are equipped with inertial, magnetic sensors, mechanical, ultrasonic, or barometric [7]. Andrea Bandini et al. [13], in their survey, presented many advantages of egocentric vision-based techniques as they can acquire hand signals very closely. Although the contact-based techniques require fewer computations, but wearing these devices gives uneasiness to the subject. Due to the electrical and magnetic emission of signals, it is likely to produce hazardous effects on the human body.

Contactless sensors or vision-based sensor technology is becoming encouraging technology to develop natural human-machine interfaces [1–4, 14]. These devices consist of visual sensors, with a single or a group of cameras situated at a distance from the user to record the hand movement. In vision-based methods, the acquired data is image type, a user does not have to wear any devices, and he can move his hand naturally in an unconstrained pattern. The important assets of vision-based techniques are large flexibility for users, low hardware requirements, and no health issues. These methods have the potential to develop any natural interface for remote human-machine interaction, this can ease the living of physically challenged or elderly people with impaired mobility [2, 9, 15].

In vision-based methods, the information is two-dimensional, three-dimensional, or multiview images. Two-dimensional images are RGB images with only intensity information about the object, captured using simple cameras and. Three-dimensional images are captured from advanced sensor cameras such as Kinect, Leap Motion, Time of flight, etc.; these cameras collect RGB along with depth information of the object in the scene. The third and the most popular choice in HGR is multiview images; here two or more cameras are placed at different angles to capture the hand movement from many views [3, 6, 8].

Wang J. et al. [16] used two calibrated cameras to record hand gestures under stable lighting conditions. They initially segmented the hand region using YCbCr color space and then applied SIFT algorithm for feature extraction. After then, they tracked using Kalman Filter. But due to similarity with other objects, the author imposes position constraints to avoid track loss.

Poon G. et al. [17] also supported multiple camera setups that can observe the hand region from diversified angles to minimize the errors due to self-occlusion. They proposed three camera setups to recognize bimanual gestures in HGR. Similarly, Bautista A.G. et al. [18] used three cameras in their system to avoid complex background and illumination. Marin G. et al. [19] suggested combining Kinect data with Leap motion camera data to exploit the complementary characteristics of both the cameras. Kainz O. et al. [20] combined leap motion sensor signals and surface electromyography signals to propose a hand tracking scheme.

Andreas Aristidou discussed that high complexity in hand structure and movement make the animation of a hand model a challenge. They preferred a marker-based optical motion capture system to acquire the orientation of the hand [21]. With the same opinion, Lizy Abraham et al. [22] placed infrared LEDs on the hand to improve the consistency of accuracy in tracking. According to the study conducted by Mais Yasen et al. [9], surface electromyography (sEMG) as wearable sensors and Artificial Neural Network (ANN) as classifiers are the most preferable choices in hand gesture recognition.

The important factor in HGR is that information obtained using a monocular camera is not sufficient to extract the moving hand region. The loss of information in RGB images is maximum due to unpredictable background, self-occlusion, illumination variation, and erratic pattern of the hand movement [8, 10, 14].

The second component in the design of DHGR is description of the region of interest or "target modeling." In this section, features that are repetitive, unique, and invariant to general variations, e.g., illumination, rotation, translation of the hand region are collected. These features model the target of tracking and are responsible for detecting and localizing the target in all frames of a video. This step is very significant because it helps to detect the target in an unconstrained environment [10, 12].

Li X. et al. [12] presented a very detailed study of the building blocks of visual object tracking and the associated challenges. They stated that effective modeling of the appearance of the target is the core issue for the success of a visual tracker. Practically, effective modeling is greatly affected by many factors such as target speed, illumination conditions, state of occlusion, complexity in shape, and camera stability, etc. Skin color features are the most straightforward characteristic of the hand used in the HGR domain to identify the hand region in the scene. Huang H. et al. simply detected skin color for contour extraction and then classified them using VGGNet [23]. M. H. Yao et al. [24] extracted 500 particles using the CAMShift algorithm for tracking the moving hand region. In this case, the real-time performance of the HGR system decreases when a similar color object (face or arm region) interferes. As the number of particles increases the complexity of the system increases. The HGR technique proposed by Khaled H. et al. [25] emphasized the use of both shape and skin color features for hand area detection because of background conditions, shadows, visual overlapping of the objects. They stated that noise added due to camera movement is one of the major problems in real-time hand tracking. Liu P. et al. [26] proposed a single-shot multibox detector ConvNet architecture that is like Faster R-CNN to detect hand gestures in a complex environment. Bao P. et al. [27] expressed that since the size of hand posture is very small, therefore misleading behavior or the overfitting problem becomes prominent in regular CNN.

In the method discussed in [10], we have shown that though the local representation of the hand is a comparatively more robust approach to detect the hand region, but they often suffer from background disturbance in a real-time tracking. In general, hand-crafted features result in large computations and loss of trajectory visual while tracking in real-time hand movement is very common. Henceforth, it is difficult for hand-crafted features to perfectly describe all variations in target as well as background [10, 12]. According to Shin J. et al. [28], the trackers that visually trace the object, based on appearance and position, must have a high tolerance for appearance and position. Tran D. et al. [29] initially detected the palm region from depth data collected by Kinect V2 skeletal tracker followed by morphological processing. They determined hand contour using a border tracing algorithm on binary image converted using a fixed threshold. After detecting fingertip by K-cosine algorithm, hand posture is classified using 3DCNN.

Matching of hand gesture trajectory is another important phase in the cognitive recognition of DHGR. The main constrain in generating similarity index in HGR is the speed of hand motion and the path of movement. Both these factors are highly dependent on the user's mood and surrounding conditions at the instant of movement. Similarity matching based on distance metrics generally fails to track efficiently as hand gestures of the same meaning do not follow the same path always.

Dan Zhao et al. [30] used a hand shape fisher vector to find the movement of the finger and then classified it by linear SVM. Plouffe et al. [31] proposed Dynamic Time Wrapping (DTW) to match the similarity between target and trained gesture. In [32], a two-level speed normalization procedure is proposed using DTW and Euclidean distance-based techniques. In this method, for each test gesture, 10 best-trained gestures are selected using the DTW algorithm. Out of these 10 gestures, the most accurate gesture is selected by calculating Euclidean distance. Pablo B. et al. [33] suggested a combination of the Hidden Markov Model (HMM) and DTW, in the prediction stage.

4. Proposed methodology

The proposed system is designed by using a web camera; it is a simple RGB camera. The use of the RGB camera is limited in the field of hand gesture tracking because of various difficulties as discussed above (**Figure 1**).

The proposed system is divided into three modules:

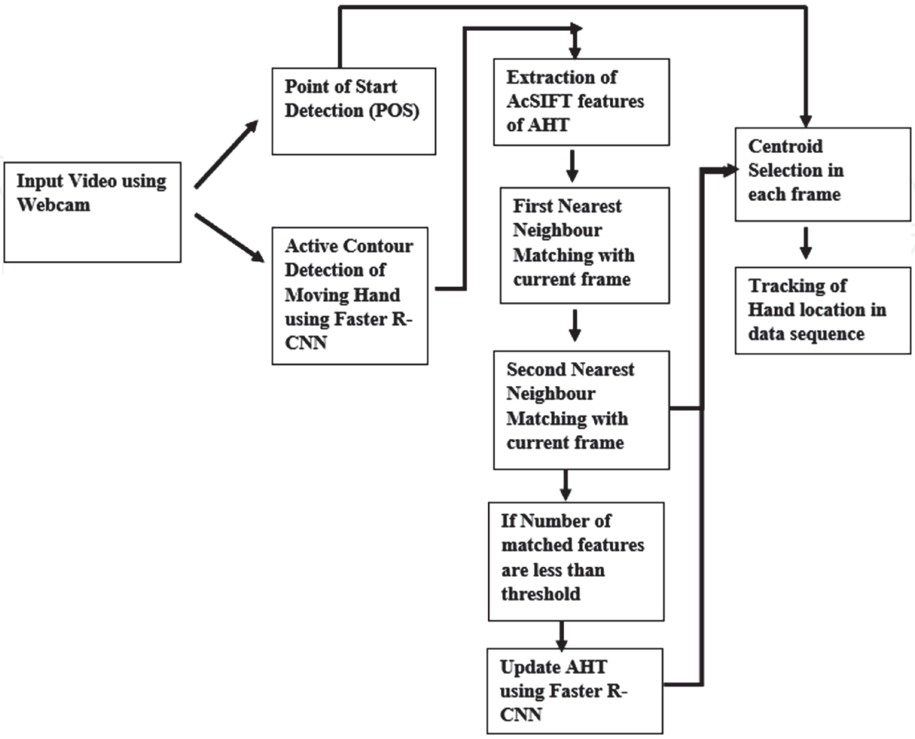

Input Video using Webcam
Point of Start Detection (POS)
Active Contour Detection of Moving Hand using Faster R-CNN
Extraction of AcSIFT features of AHT
First Nearest Neighbour Matching with current frame
Second Nearest Neighbour Matching with current frame
If Number of matched features are less than threshold
Update AHT using Faster R-CNN
Centroid Selection in each frame
Tracking of Hand location in data sequence

Figure 1.
Architecture of the proposed system.

4.1 Module I

This module is also known as the "hand detection module." Here the posture of the hand, which is used by the user in real-time hand movement events, is detected. When the user moves his hand in front of the web camera attached to any machine acquires a video of 5–6 seconds at a rate of 15 frames per second. This video comprises a raw data sequence of length 100–150 frames; it is saved in a temporary folder, resizing all the frames to size [240, 240]. In this module, detection of an online Active Hand Template (AHT) is made using Faster Region-based Convolutional Neural Network (Faster R-CNN).

4.1.1 Faster R-CNN

We have proposed the design of an online hand detection scheme (AHT) using Faster Region-based Convolutional Neural Network (Faster R-CNN) [34], on Residual Network (ResNet101) [35], a deep neural architecture. Three major issues that are encountered in online tracking of hand motion captured using simple cameras are as follows:

1. A hand is a versatile object in comparison with other objects. The area occupied in the image frame has a high variation that depends on the posture selected.

2. It is not fixed that the subject starts the motion from the first frame or the fixed position in the frame.

3. Anthropometric and scale variation in the hand are very prominently seen during hand movement in RGB images.

Thus, the essential requisite of any technique is to cope with the abovementioned factors. In the proposed method, these issues are solved by using Faster-RCNN, a Deep Neural Network (DNN) architecture. Deep learning algorithms (DLAs) are models for a machine to learn and execute any task as human beings perform. Deep networks directly learn features from raw data by exploiting local information of the target, with no manual extraction or elimination of background. Convolutional Neural Network (ConvNet) is a powerful tool in the computer vision field that mainly deals with images.

Ren S. et al. [34] modified fast RCNN to Faster Region-based Convolutional Neural Network (Faster R-CNN). They added a region proposal network (RPN) (a separate CNN network) that simultaneously estimates objectness score and regresses the boundaries of the object using the anchor box concept.

The architecture of the proposed Faster R-CNN developed on ResNet 101 is shown in **Figure 2**. Region Proposal Network (RPN) is an independent small-sized ConvNet, designed to directly generate region proposals from an image of any size without using a fixed edge box algorithm. The process of RPN is shown in **Figure 3**; here region proposals are generated from the activation feature map of the last shared convolutional layer between the RPN network and Fast-RCNN. It is implemented with an $m \times m$ convolutional layer followed by two siblings: box regression layer and box classification layer each of size 1×1. At each sliding grid, multiple regions are proposed depending upon the number of anchor boxes (Q). Each predicted region is classified by a score and four tuples (x, y, L, B) where (x, y) are the coordinates of the top left corner of the bounding box, L and B are the length and breadth of the box. If $M \times N$ is the size of the feature map

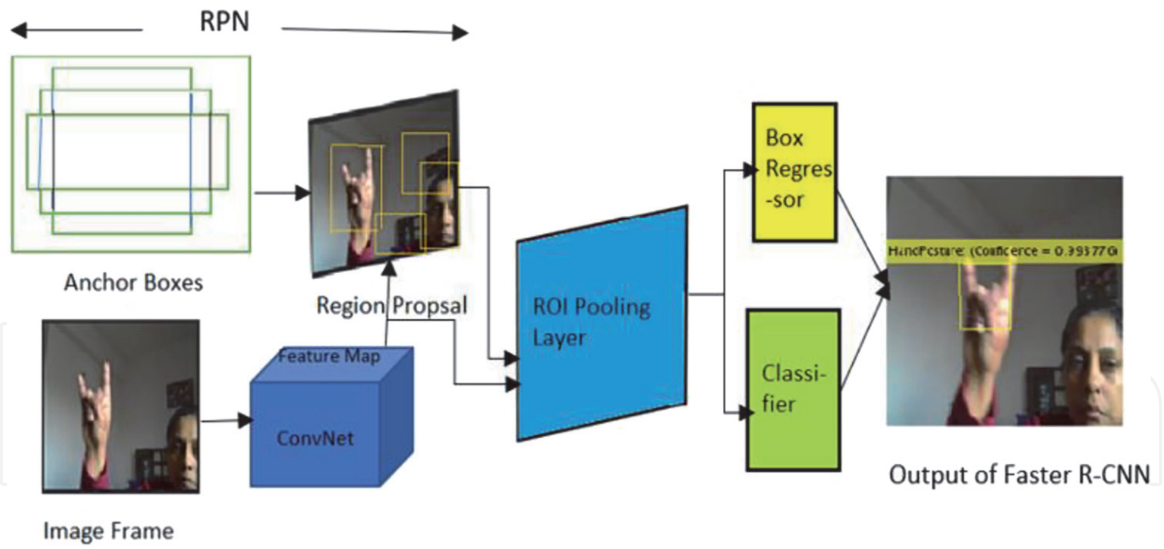


Figure 2.
The architecture of the proposed faster R-CNN.

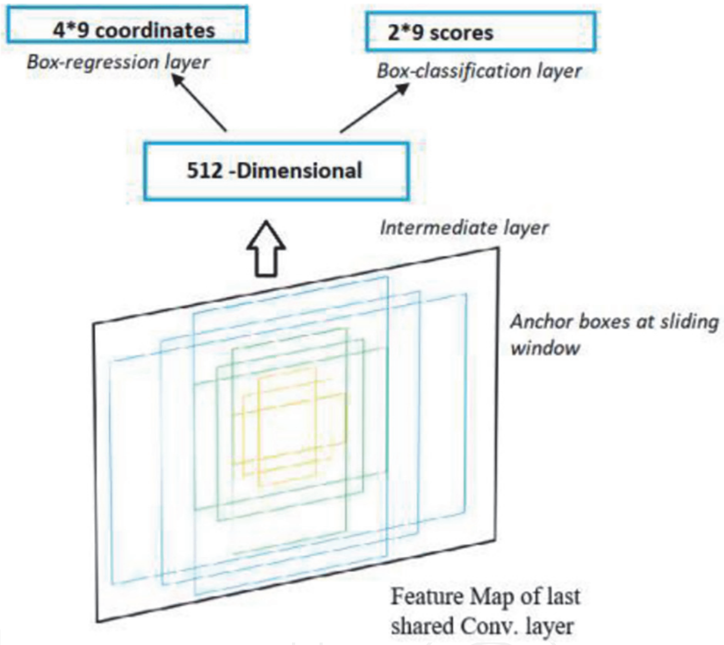


Figure 3.
Process in RPN.

and number of anchors, the technique is Q, then total anchors created will be $M \times N \times Q$.

Anchor boxes are bounding boxes with predefined height and width to capture the scale and aspect ratio of the target object. There are pyramids of anchors. The anchor-based method is translation invariant and detects objects of multiple scales and aspect ratios. For every tiled anchor box, the RPN predicts the probability of object, background, and intersection over union (IoU) values. The advantage of using the anchor boxes in a sliding window-based detector is to detect, encode, and classify the object in the region in a single process [34].

The design of the proposed Faster-RCNN technique is accomplished on Residual network (Resnet) resnet 101. Resnet architecture network was proposed in 2015 by Kaiming He et al. [35], to ease the learning process in a deeper network. They exhibited that a resnet architecture eight times deeper than VGG16 still has less complexity in training on ImageNet dataset. The proposed use of resnet 101 in the design of Faster R-CNN solves the complex problems in object classification by using a large number of hidden layers without increasing the training error. Furthermore, the network does not have a vanishing and exploding gradient problem because of the "skip connection" approach.

4.2 Module II

This module handles the feature extraction process of the AHT that helps in the continuous localization of the moving hand region. Our method processes a hybrid framework that combines Scale Invariant Feature Transform (SIFT) and Faster-RCNN. A framework with hybrid characteristics is selected because in real-time movement, the geometrical shape of any posture changes many times, and thus it is difficult to detect the moving hand region with only hand-crafted features i.e., SIFT. Whenever the posture is changed above the threshold (number of matched features < = 3), then AHT is determined using Faster R-CNN, and the previous AHT is updated with new AHT. During this process, a bounding box is also constructed around the centroid of the hand movement to determine the current two-dimensional area covered by the hand region.

4.2.1 Scale invariant feature transform (SIFT)

In motion modeling, we have used SIFT algorithm designed by David Lowe [36], for local feature extraction of AHT. As compared with global features such as color, contour, texture, local features have high distinctiveness, better detection accuracy toward local image distortions, viewpoint change, and partial occlusion. Therefore, SIFT detects the object in the cluttered background without performing any segmentation or preprocessing algorithms [36, 37]. The combination of SIFT and Faster-RCNN is helpful in real-time fast-tracking of the non-rigid subtle object hand.

SIFT algorithm comprises of feature detector as well as a feature descriptor. In general, features are high-contrast areas example point, edge, or small image patch, in an image. These features are extracted such that they are detectable even in noise, scale variation, and during the change in illumination. Each SIFT feature is defined by four parameters: $f_i = \{p_i, \sigma_i, \varphi_i\}$, where $p_i = (x_i, y_i)$ is the 2D position of SIFT keypoint, σ_i is the scale, φ_i is gradient orientation within the region. Each key point i d is described by 128-dimensional descriptor d [36].

In our approach, we find the SIFT features of the AHT template obtained in module-I, since it contains only the target hand posture and is small as compared with the image frame [240, 240]. Therefore, this approach saves time in matching unnecessary features and pruning them further [20, 21].

Let there be m key features in AHT frame, given as $S_{AHT} = \{f_i\}^m$, where f_i is the feature vector at i^{th} location. Let $S_{cur} = \left\{f_j\right\}^k$ are k numbers of SIFT features in the current frame, where f_j is the SIFT feature at j^{th} location. We use the best-bin-first search method that identifies the nearest neighbors of AHT features with current frame features. The process of SIFT target recognition and localization in the subsequent frames of a video is accomplished in three steps:

Initially, we find the first nearest neighbors (FNN) of all the SIFT features in AHT with SIFT features in the current frame. The First Nearest Neighbors (FNNs) are defined as the pairs of key points in two different frames with a minimum sum of squared differences for the given descriptor vector

$$distanceFNND(a_{AHT}, b_{cur}) = \sqrt{\sum_{i=1}^{128}(a_i - b_i)} \tag{1}$$

where a_{AHT} and b_{cur} are descriptor vectors of features in AHT and current frame, respectively.

In the second step, matching is improved by performing Lowe's Second Nearest Neighbor (SNN) test using Eq. (2).

$$\frac{distance(a_{AHT}, b_{cur}\ )}{distance\ (a_{AHT}, c_{cur})} > 0.8 \tag{2}$$

SNN test is done by calculating the ratio between the FNND of a_{AHT} feature with two nearest neighbors b_{cur} and c_{cur} in the current frame.

Further to find the geometrically consistent points, we apply the geometric verification test (Eq. (3)) on the key points obtained after SNN.

$$\begin{bmatrix} x^* \\ y^* \end{bmatrix} = vR(\alpha)\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \end{bmatrix} \tag{3}$$

Here v is isotropic scaling, α is rotation parameter, (T_x, T_y) are translation vectors for the i^{th} SIFT keypoint located at a distance (x, y) .

4.3 Module III

This module deals with the cognitive recognition of the trajectory. Here the cognitive recognition means vision-based intellectual development of machine for the interpretation of hand movement. Because hand movements do not have a fixed pattern, by nature movement patterns are erratic. Due to this characteristic, till now static hand gesture recognition is more preferred than dynamic hand gesture recognition. We have determined the centroids of hand location in the tracked frames. To derive the meaning of hand movement, we have used the modified back-propagation Artificial Neural Network (m-BP-ANN) Match of test trajectory to train database. This cognitive stage is very significant for DHG because the way we collect and transform the centroid of hand movement C_{HM} of every frame in a particular data sequence, helps to classify the hand gesture. In the proposed system we have kept this stage simple but efficient because complex algorithms increase the error rate and time of interpretation.

We have made use of the concept of the quadrant system of the Cartesian plane to transform the image frame into a 2-D plane. The two-dimensional Cartesian system divides the plane of the frame into four equal regions called Quadrants. Each quadrant is bound by two half-axes, with the center in the middle of a frame. The translation of the image frame axis to a Cartesian axis is done using Eqs. (4) and (5):

$$x_c = (C_{HMx} - I_x)/n_x \tag{4}$$

$$y_c = (C_{HMy} - I_y)/n_y \tag{5}$$

Here I_x, I_y are the dimensions of the image frame [240, 240] and n_x, n_y [12, 12] are normalization factors for the X and Y-axis. To convert the hand trajectory into meaningful command, we have applied Modified Back-Propagation of Artificial Neural Network (mBP-ANN) using start and end location of the hand gesture.

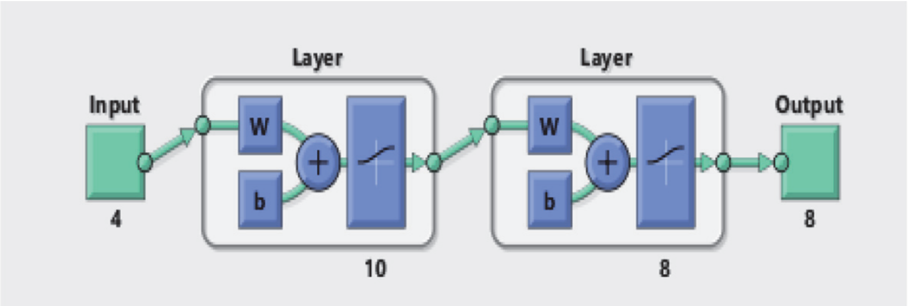


Figure 4.
Architecture of the proposed ANN model.

Back-propagation (BP) is a supervised training procedure in feed-forward neural networks. It works on minimizing the cost function of the network using the delta rule or gradient descent method. The value of the weights with which we obtain the minimum cost function is the solution for the given learning problem. The error function $'E_f''$ is defined as the mean square sum of the difference between the actual output value of the network (a_j) and the desired target value (t_j) for the jth neuron. E_f calculated for N_L number of output neurons in$''{}'L''$ a number of layers are given as Eq. (6):

$$E_f = 1/2 \sum_{p=1}^{P} \sum_{j=1}^{N_L} \left(t_j - a_j\right)^2 \tag{6}$$

The minimization of the error function is carried out using gradient descent or delta rule. It determines the amount of weight update based on gradient direction along with step size. It is given by Eq. (7):

$$\frac{\partial C(t+1)}{\partial \delta_{ij}(t)} = \frac{\partial C(t+1)}{\partial w_{ij}(t+1)} \text{x} \frac{\partial w_{ij}(t+1)}{\partial \delta_{ij}(t)} \tag{7}$$

In the traditional BP, the optimization of the multidimensional cost function is difficult because step size is fixed, since the performance parameters are highly dependent on the learning rate δ. Hence, to overcome the problems of fixed step size and slow learning, we use adaptive learning and momentum term to modify BP. The updated weight value at any node is given by Eq. (8):

$$\Delta w_{ij}(t) = \eta \delta_j a_i + m \Delta w_{ij}(t-1) \tag{8}$$

The term momentum $(0 < m < 1)$ updates the value of weight using the previous value of it. Adaptive learning rates help to learn the characteristic of the cost function. If the effort function is decreasing, then the learning rate will increase, and vice versa [38].

In the proposed prototype, we have developed eight vision-based commands to operate and machine remotely by showing hand gestures. The proposed model of ANN has three layers, input layer, hidden layer, and output layer as shown in **Figure 4**. The input layer has 4 neurons, the hidden layer has 10 neurons, and the outer layer consists of 8 neurons.

5. Experimental analysis

In this research work, we have taken three hand postures (as shown in **Table 1**) to demonstrate the vision-based tracking efficiency of our proposed concept. It is

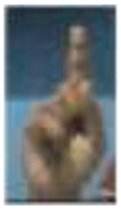
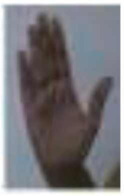
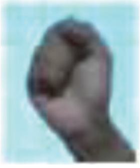
		
Posture I	Posture II	Posture III

Table1.
Types of postures used in the proposed system.

the unique feature of this work as most of the techniques demonstrate tracking of hand movements performed by a single posture [32]. For consolidated evaluation, we have taken approximately 100 data sequences captured in different environments as shown in **Figure 5**. Our database is a collection of publicly available dataset [32] and self-prepared data sequence. In [32], hand movements are mainly performed by a single hand posture (Posture III as shown in **Table 1**) and in a constrained laboratory environment.

In self-prepared dataset, we have collected hand movements performed by six participants of three different age groups: two kids (age 10–16 years), two adults (age 20–40 years), and two seniors (age 65 years). In this, the hand movement is carried out using three different postures (as illustrated in **Table 1**), in linear as well as circular pattern. In self-collected dataset, 15 frames per second are taken through the web camera, and gesture length varies from 120 to 160 frames.

The evaluation of the proposed online adaptive hand tracking methodology is carried out on four test parameters. The methodology is also compared with the

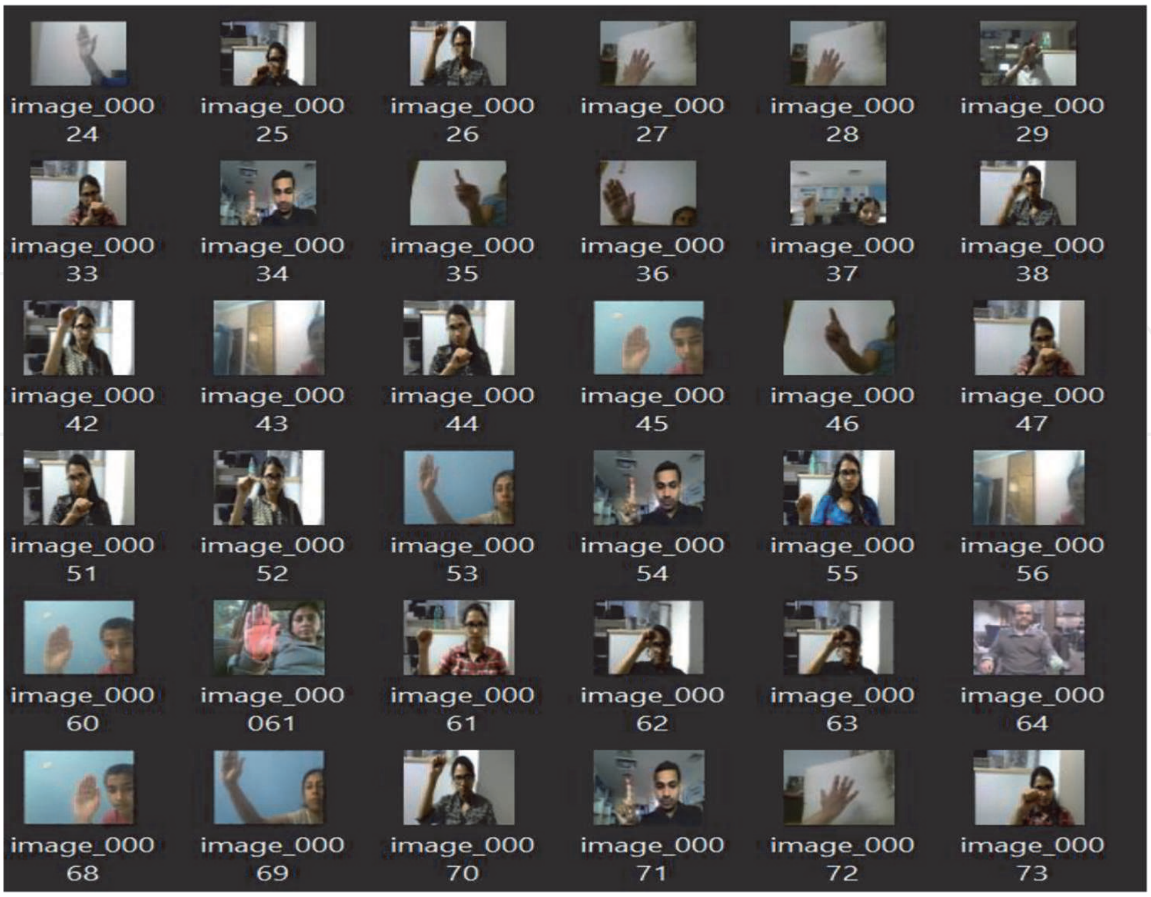


Figure 5.
Dataset for training faster R-CNN.

contemporary techniques that are based on RGB images or webcam images. The four test parameters are as follows:

1. Accuracy in hand detection in real-time complex images, i.e., video is captured in unconstrained background and covers natural variations occurring in geometrical contour of the postures.

2. Parametric evaluation of the proposed Faster R-CNN on resnet101 architecture on training and validation data.

3. The efficiency of a hybrid tracking system in complex environment.

4. Effectiveness of cognitive recognition of hand trajectory as machine command.

5.1 Accuracy in hand detection

Figure 6 demonstrates the outcome of the hand recognition stage of different data sequences captured (using three hand postures demonstrated in **Table 1**) in different backgrounds under d

ifferent illumination conditions. To test the accuracy of the hand detection scheme in recognizing the hand region, we have considered nearly all possible combinations: only the hand is visible in the camera view, subject face along with arm region is in the camera view, illumination conditions are unstable, background has same color as the hand region, etc. Thus, the hand detection results in **Figure 6** illustrate the following distinguishing key features of our proposed system:

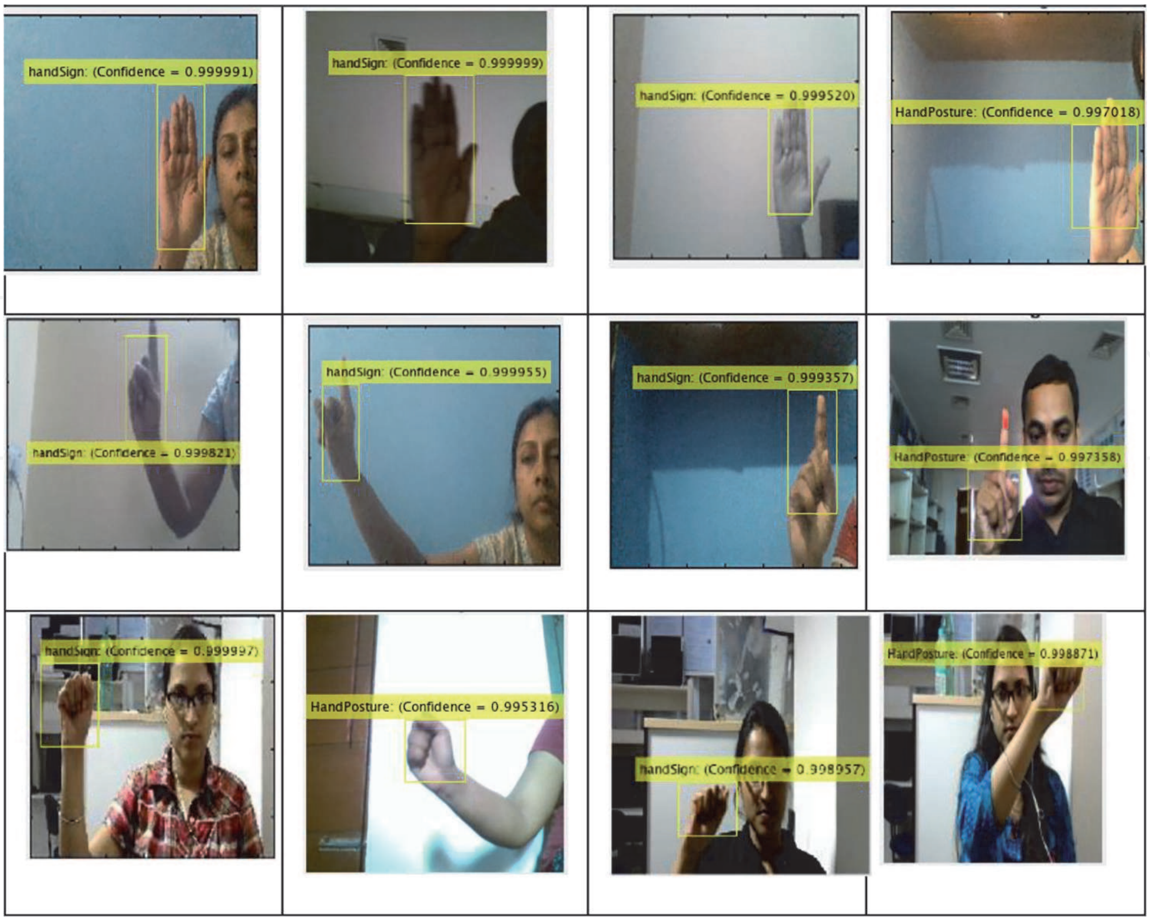

handSign: (Confidence = 0.999991)
handSign: (Confidence = 0.999999)
handSign: (Confidence = 0.999520)
HandPosture: (Confidence = 0.997018)
handSign: (Confidence = 0.999821)
handSign: (Confidence = 0.999955)
handSign: (Confidence = 0.999357)
HandPosture: (Confidence = 0.997358)
handSign: (Confidence = 0.999997)
HandPosture: (Confidence = 0.995316)
handSign: (Confidence = 0.998957)
HandPosture: (Confidence = 0.998871)

Figure 6.
Various outcomes of module-I (simple background, complex background, the subject is also visible in camera range).

i. Large diversity is present in hand shape and sizes also, the same posture differs in geometrical shapes and area of coverage in the image frame. Our technique does not require any foreground and background modeling. It detects the hand region by automatic learning, the discriminative deep features of the hand postures.

ii. The subject's state of mind at the instance of hand movement is not alike. Thus, it is not necessary that the hand is completely visible from the first frame. Our technique is not affected by the location from which the user starts their motion, it is also unaffected by the face region or other body parts of the user present in the data sequence.

5.2 Parametric evaluation of module i

The proposed hand detection module, developed on Faster R-CNN architecture, has been evaluated on the following parameters:

i. Accuracy: It is the parameters by which the network is evaluated and selected. It gives the count of accurate predictions.

ii. Loss: The loss curve is the most useful diagnostic curve that accounts for variation in the predicted and actual value. Loss information helps to learn the optimization behavior of the model parameters.

iii. Model Behavior: It talks about the learning behavior of the model. Learning pattern helps to diagnose the character of the train or validation dataset concerning the problem domain.

iv. Root Mean Square Error (RMSE): It calculates the standard deviation between the actual value and the predicted value. The RMSE is applied in regression analysis and classification of the predicted bounding box with the ground truth bounding box. It is calculated during the training process for both train data and validation data.

Table 2 illustrates detailed performance outcomes of the proposed Faster R-CNN based on the abovementioned parameters. The observations are taken at intervals of 50, 100, 150, 200, 220 iterations. The outcomes illustrate following points of our proposed architecture on Faster R-CNN constructed on resnet101:

1. As the number of iterations increases, the accuracy of train data increases, and it reaches the maximum value at the 200th iteration.

2. Validation data achieve the maximum accuracy at 220th iterations.

3. There is a linear decrement in the RMSE and the loss values of both the train and validation dataset. This linear decrement reflects the stable learning behavior of the proposed model.

4. It is observed that at the 200th iteration, RMSE and loss of train data reached at its minimum value of 0.14 and 0.154, respectively. Similarly, in the case of the validation dataset, the value of RMSE and loss reached their minimal at the 200th iteration.

No. of iteration	Train data accuracy	Validation data accuracy	Train data RMSE	Validation data RMSE	Train data loss	Validation data loss
1	30.24	78.26	0.23	0.22	2.9331	2.2522
50	97.07	98.80	0.19	0.19	0.9396	0.6902
100	98.32	98.87	0.15	0.19	0.4791	0.6049
150	99.03	98.85	0.16	0.17	0.2671	0.5956
200	99.07	97.86	0.14	0.17	0.1544	0.55544
220	98.92	98.76	0.17	0.17	0.2052	0.6262

Based on above outcomes, the characteristic features of the proposed trained resnet101 are:
Accuracy: 98.76%
Loss: 0.17
The behavior of the Network: Well fit.

Table 2.
Outcomes in the training process of the proposed faster R-CNN model.

5.3 Efficiency of hybrid tracking system

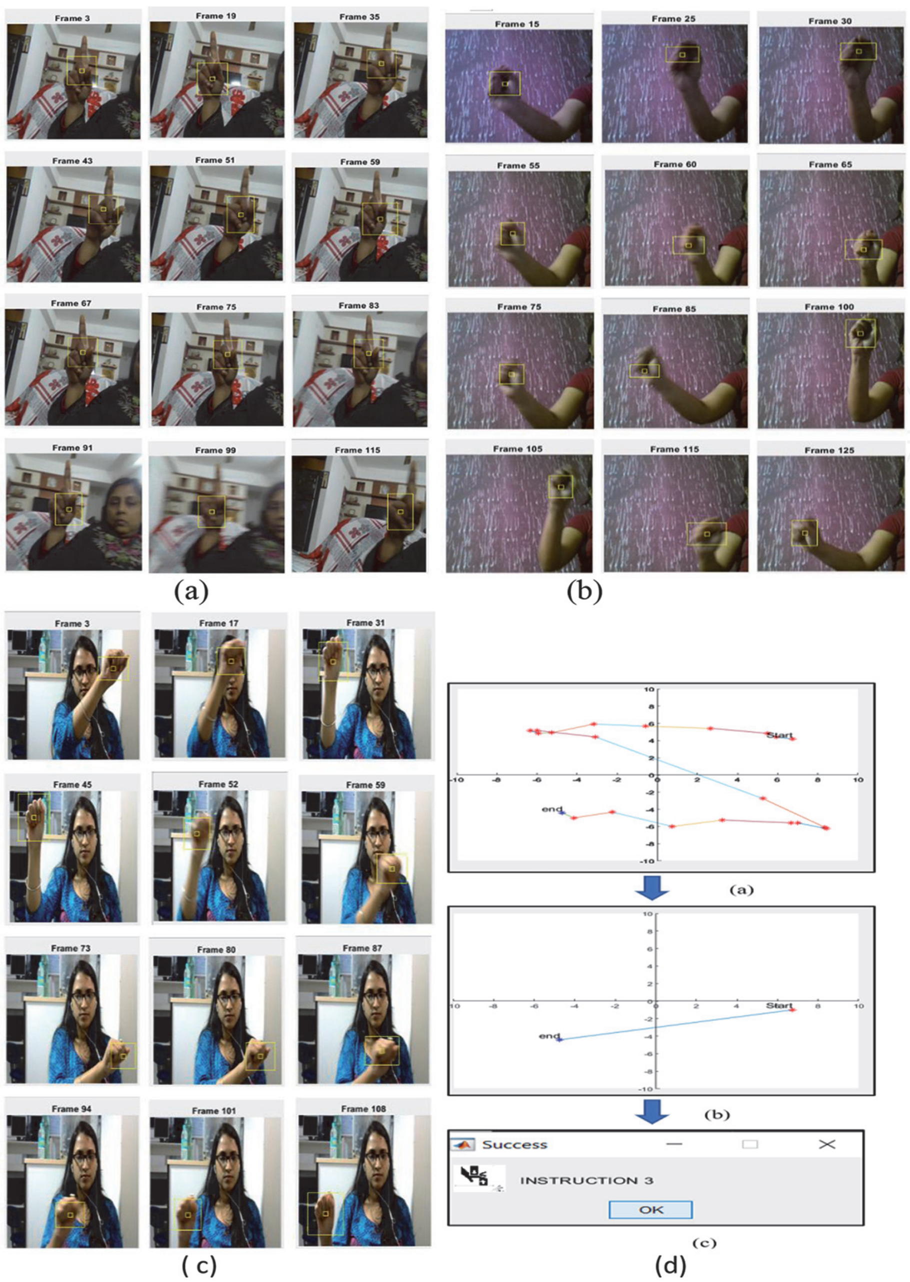
In this section, we have evaluated the tracking efficiency of our proposed hybrid method. The data sequences captured are of variable length ranging from 100 to 150 frames. **Figure 7** shows results of tracking in different data sequences, approximately 10–12 frames of each data sequence are shown here to highlight the tracking efficiency of module II. Each frame is illustrated by its frame number, a yellow box enclosing the hand region and a yellow dot inside the yellow box represent the instant position of the centroid of the hand region. **Figure 7**(a) shows the tracking of P-I posture in a cluttered background. This data sequence is captured in a background that has many similar colored objects as that of the hand. Our proposed system discriminates and localizes the hand region efficiently due to the robust deep feature learning capability of our hybrid tracking system. It is also noticeable that the hand is properly identified even when the hand region was blurred due to sudden erratic movement by the subject as shown in frame 99 of the data sequence.

Figure 7(b) displays the tracking results of the P-III hand posture in improper illumination conditions. It can be noticed that in **Figure 7**(a) and (b), the FoS are frame 3 and frame 15, respectively. This data sequence is mainly affected by the color reflection of the background wall, and thus, it is visible that the edges of the P-III posture are nearly mixed with the background in some frames.

Figure 7(c) demonstrates the tracking results of a data sequence [32] in which a teenage girl is moving her hand (posture P-III) in front of her face. It is noticeable that the hand region and face region nearly overlap in frame 17. The fast change in the hand position in the frames indicates that the subject is moving her hand in a speedy manner. The change in the distance between the two positions of the hand frame 45 to frame 59 along with the change from a clear image of the hand region to the blurred image of the hand image proves the fast movement of the hand. During the movement, the subject is also changing the orientation of the hand posture as can be seen from the frames 59, 73, 80, 87.

5.4 Efficiency of hybrid tracking system

Cognitive efficiency means the development of the semantic between the trajectory of the dynamic hand gestures and machine command. Since, hand gestures do not follow a fixed line of movement to convey the same meaning. Therefore, syntax formation to match train data and test data is a challenge. Hence, the main



INSTRUCTION 3

Figure 7.
Tracking outcomes of different data sequence are shown in (a), (b), (c), and (d) shows the cognitive recognition of hand movement in (c).

limitation in DHGR is the development of a process that can convert the trajectory of hand movement to machine command. Our proposed method handles this difficult challenge in a schematic manner.

In our proposed technique, we have developed eight vision-based commands "INSTRUCTION 1–8" (abbreviated as INT-1 to INT-8). For the vision-based instruction, we have drafted a process to convert trajectory of the hand movement obtained in module-II to a machine command by using Cartesian plane system as illustrated in **Figure 8**.

Figure 7(d) illustrates the process in developing cognitive ability to recognize hand movement by the machine. This process consists of three steps: (i) trajectory plot of the hand movement, (ii) position of start and end point in Cartesian plane, and (iii) conversion to machine command. **Figure** 7(d) demonstrates the results of the cognitive recognition of a data sequence shown in **Figure** 7(c) [32]; here an adult girl moves her hand from right to left and the machine recognizes this movement as command 7.

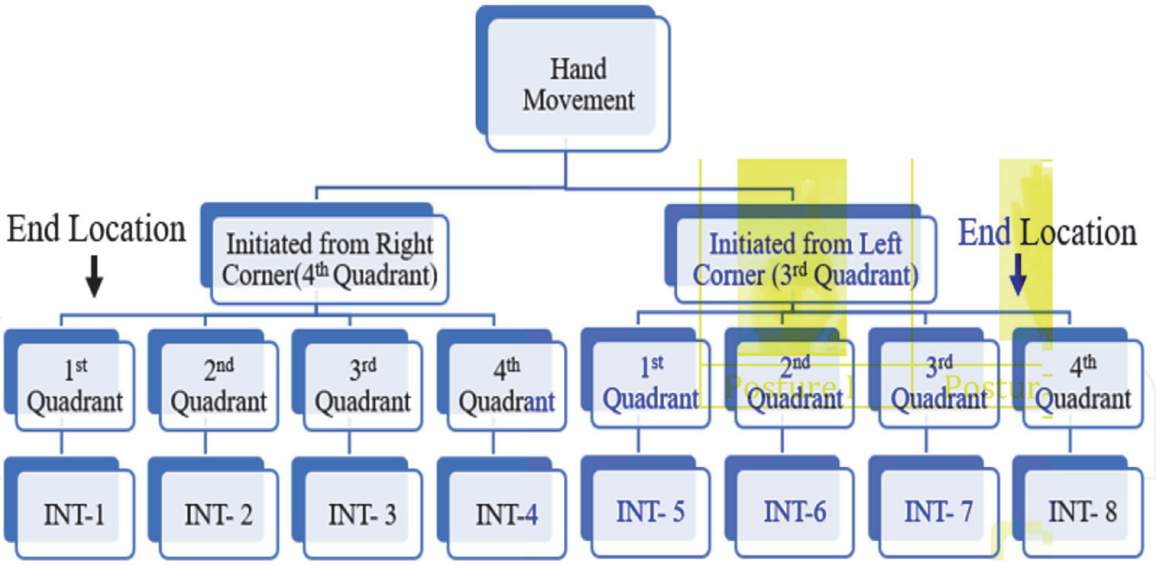


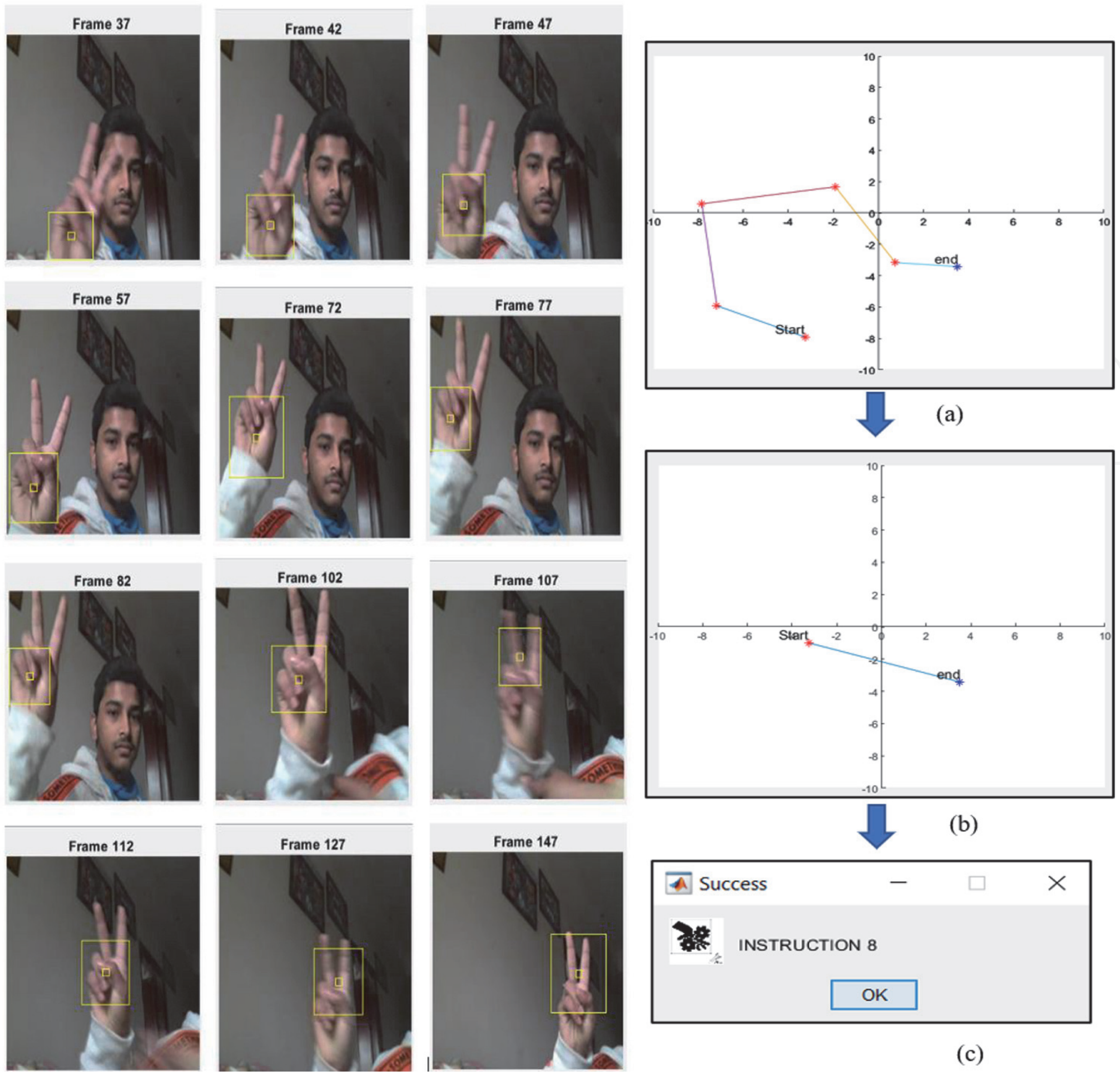
Figure 8.
Conversion of trajectory of hand movement to machine command.

Figure 9(a) shows tracking results of P-III posture performed by a teenage boy. In this data sequence, we can notice that scale change of the hand region is very prominent (as the size of the hand region is continuously changing from frame to frame). The posture area is big in frame 37, and it gradually decreases till frame 147. This indicates the distance between the subject's hand and the camera, it is minimum in frame 37 and maximum at frame 147. **Figure 9**(b) displays the result of cognitive recognition of the trajectory in the three steps in trajectory to command interpretation of left initiated data sequences The movement starts from the bottom left, moves in a zigzag manner, and finally reaches close to the initial starting place. The PoS and end location of this sequence both are in the third and fourth quadrant respectively; thus, "INSTRUCTION 8" is generated through this hand movement.

5.5 Comparison with contemporary techniques

In this section, we compare our process and results with two different approaches used recently in the field of DHGR. In the first approach [32] technique utilizes true RGB images. This approach mainly involves hand-crafted features for hand detection and tracking. The research work conducted by Singha J. et al. [32] focused on only fist posture tracking in a fixed background, they have achieved 92.23% efficiency when no skin color object is present in the surroundings. One of the prominent limitations in their approach is that they have applied sequence of algorithms for precise detection of hand region. This method is complex and unsuitable for real-time implementation of DHGR.

In the approach proposed by Tran DS et al. [29] for fingertip tracking, depth coordinates of fingertip provided by the inbuilt software of the advanced sensor-based camera are directly used. According to the researchers, RGB camera images are largely affected by illumination variation, and thus, to avoid background and illumination complexities in DHGR, they utilized RGB-D data sequences captured through the Microsoft Kinect V2 camera. It is a skeletal tracker camera that provides the position of 25 joints of the human skeleton including fingertips. This method is designed for tracking only seven hand movements comprised of 30–45 frames in three fixed backgrounds; besides, subjects are also trained to perform correct hand movement. In this research work, each frame is allotted an individual 3DCNN for classification. Thus, the experiments can perform fingertip tracking only for short



INSTRUCTION 8

Figure 9.
Tracking results of P-III posture performed by a teenage boy. (b) Cognitive recognition of hand movement.

gesture length. The training time of the 3D CNN is 1 hr. 35 minute with a six-core processor of 16GB RAM, which indicates the complex architecture of the technique. The accuracy of the trained 3D CNN model is 92.6% on validation data. **Table 3** illustrates and compares different technical aspect of the above two mentioned approaches with our proposed method:

6. Conclusion

This research work presents solutions to many crucial and unresolved challenges in vision-based tracking of hand movement captured using a simple camera. The methodology has the potential to provide a complete solution from hand detection to tracking and finally for cognitive recognition of trajectory to machine command for contactless Human-Machine interaction via dynamic hand gestures. Since the proposed design is implemented around a single RGB webcam, thus the system is economical and user-friendly. The accuracy achieved in the online and adaptive hand detection scheme with Faster R-CNN is 98.76%. The proposed hybrid tracking scheme exhibits high efficiency to adapt scale variation, illumination variation, and background conditions. It also exhibits high accuracy when camera is in motion during the movement. The overall accuracy achieved by our proposed system in complex conditions is 95.83%.

Parameters	Research work-I (2018) [36]	Research work-II (2020) [29]	Proposed research work
Camera/ Image type	Simple webcam/RGB	Microsoft Kinect Sensor version 2/depth,	Webcam/RGB
Preprocessing	Face segmentation using ViolaJones and the background subtraction using skin filtering	Noise Removal using median filtering and morphological processing. Conversion to binary image	Not Required
Initial stage-Hand detection	Three frames differencing on colored and grayscale images.	Hand Contour is extracted using Moore -Neighbor algorithm. Fingertip extraction using K-cosine algorithm.	Designed Faster-RCNN constructed on ResNet101. Used region-based network (RPN) for defining hand region.
Feature Extraction	Eigen features of the detected hand region. Remove unwanted features using compact criteria	Position of Fingertip calculated through inbuilt software of the camera.	SIFT feature extraction of AHT
Tracking methods	KLT features followed 44 features matching by compact criteria	For each frame, a 3D CNN is allotted.	Combination of Faster RCNN with SIFT algorithm.
Classification	Results of ANN, SVM, kNN classifiers are fused to get the final classified value	Ensemble learning to generate a final probability for classification	Using ANN with Cartesian quadrant system.
Background to conduct experiments	Fixed laboratory environment without any skin color object	Three fixed backgrounds	Any real-time background.
Accuracy of Methodology	92.23%	92.60%	95.83%
Limitations	KLT features get reduced in subsequent frames.	(i) Preprocessing is required (ii) For each frame separate 3D CNN is required this makes the system slow. (iii) fixed gesture length of 20 frames.	Initially trained for five gestures and can be extended for many more postures

Table 3.
Comparative analysis of two recent methods with the proposed methodology based on different parameters.

The comparative analysis demonstrates that our system gives users the freedom to select posture and to start the hand movement from any point in the image frame. Also, we do not impose any strict conditions in terms of geometrical shape of any posture. The hybrid framework and cognitive recognition features of our proposed method give a robust solution to classify any hand trajectory in a simple manner. This feature has not been discussed in any existing technique working with RGB images till date. The cumulative command interpretation efficiency of our system in real-time environment is 96.2%. The various results justify the “online” hand detection and “adaptive” tracking feature of the proposed technique. In the future, the method can be further extended to track multiple hand movements.

Acknowledgements

No funding is received

Author details

Richa Golash* and Yogendra Kumar Jain
Samrat Ashok Technological Institute, Vidisha, Madhya Pradesh, India

*Address all correspondence to: golash.richa@gmail.com

References

[1] Wachs JP, Kölsch M, Stern H, Edan Y. Vision-based hand-gesture applications. Communications of the ACM. 2011;**54**(2):60-71. DOI: 10.1145/1897816.1897838

[2] Golash R, Jain YK. Economical and user-friendly Design of Vision-Based Natural-User Interface via dynamic hand gestures. International Journal of Advanced Research in Engineering and Technology. 2020; **11**(6)

[3] Rautaray SS, Agrawal A. Vision based hand gesture recognition for human computer interaction: A survey. Artificial Intelligence Review. 2015; **43**(1):1-54. DOI: 10.1007/s10462-012-9356-9

[4] Mcintosh J. How it Works: BMW's Gesture Control. Available from: https://www.driving.ca/auto-news/news/how-it-works-bmw-gesture-control [Accessed: March 23, 2021]

[5] Gu C, Lien J. A two-tone radar sensor for concurrent detection of absolute distance and relative movement for gesture sensing. IEEE Sensors Letters. 2017;**1**(3):1-4. DOI: 10.1109/LSENS.2017.2696520

[6] Oudah M, Al-Naji A, Chahl J. Hand gesture recognition based on computer vision: A review of techniques. Journal of Imaging. 2020;**6**(8):73. DOI: 10.3390/jimaging6080073

[7] Li Y, Huang J, Tian F, Wang HA, Dai GZ. Gesture interaction in virtual reality. Virtual Reality & Intelligent Hardware. 2019;**1**(1):84-112

[8] Chakraborty BK, Sarma D, Bhuyan MK, MacDorman KF. Review of constraints on vision-based gesture recognition for human–computer interaction. IET Computer Vision. 2018; **12**(1):3-15

[9] Yasen M, Jusoh S. A systematic review on hand gesture recognition techniques, challenges and applications. PeerJ Computer Science. 2019; **16**(5):e218

[10] Golash R, Jain YK. Trajectory-based cognitive recognition of dynamic hand gestures from webcam videos. International Journal of Engineering Research and Technology. 2020;**13**(6): 1432-1440

[11] Yang H, Shao L, Zheng F, Wang L, Song Z. Recent advances and trends in visual tracking: A review. Neurocomputing. 2011;**74**(18):3823-3831

[12] Li X, Hu W, Shen C, Zhang Z, Dick A, Hengel AV. A survey of appearance models in visual object tracking. ACM Transactions on Intelligent Systems and Technology (TIST). 2013;**4**(4):1-48

[13] Bandini A, Zariffa J. Analysis of the hands in egocentric vision: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2020

[14] Golash R, Jain YK. Robust tracking of moving hand in coloured video acquired through simple camera. International Journal of Computer Applications in Technology. 2021;**65**(3): 261-269

[15] Bandara HM, Priyanayana KS, Jayasekara AG, Chandima DP, Gopura RA. An intelligent gesture classification model for domestic wheelchair navigation with gesture variance compensation. Applied Bionics and Biomechanics. 2020;**30**:2020

[16] Wang J, Payandeh S. Hand motion and posture recognition in a network of calibrated cameras. Advances in Multimedia. 2017;**2017**:25. Article ID 216207. DOI: 10.1155/2017/2162078

[17] Poon G, Kwan KC, Pang WM. Real-time multi-view bimanual gesture recognition. In: 2018 IEEE 3rd International Conference on Signal and Image Processing (ICSIP). IEEE; 2018. pp. 19-23

[18] Cruz Bautista AG, González-Barbosa JJ, Hurtado-Ramos JB, Ornelas-Rodriguez FJ, González-Barbosa EA. Hand features extractor using hand contour–a case study. Automatika. 2020;**61**(1):99-108

[19] Marin G, Dominio F, Zanuttigh P. Hand gesture recognition with jointly calibrated leap motion and depth sensor. Multimedia Tools and Applications. 2016;**75**(22):14991-15015

[20] Kainz O, Jakab F. Approach to hand tracking and gesture recognition based on depth-sensing cameras and EMG monitoring. Acta Informatica Pragensia. 2014;**3**(1):104-112

[21] Aristidou A. Hand tracking with physiological constraints. The Visual Computer. 2018;**34**(2):213-228

[22] Abraham L, Urru A, Normani N, Wilk MP, Walsh M, O'Flynn B. Hand tracking and gesture recognition using lensless smart sensors. Sensors. 2018; **18**(9):2834

[23] Huang H, Chong Y, Nie C, Pan S. Hand gesture recognition with skin detection and deep learning method. Journal of Physics: Conference Series. 2019;**1213**(2):022001

[24] Yao MH, Gu QL, Wang XB, He WX, Shen Q. A novel hand gesture tracking algorithm fusing Camshift and particle filter. In: 2015 International Conference on Artificial Intelligence and Industrial Engineering. Atlantis: Atlantis Press; 2015

[25] Khaled H, Sayed SG, Saad ES, Ali H. Hand gesture recognition using modified 1$ and background subtraction algorithms. Mathematical Problems in Engineering. 2015;**20**:2015

[26] Liu P, Li X, Cui H, Li S, Yuan Y. Hand gesture recognition based on single-shot multibox detector deep learning. Mobile Information Systems. 2019;**30**:2019

[27] Bao P, Maqueda AI, del-Blanco CR, García N. Tiny hand gesture recognition without localization via a deep convolutional network. IEEE Transactions on Consumer Electronics. 2017;**63**(3):251-257

[28] Shin J, Kim H, Kim D, Paik J. Fast and robust object tracking using tracking failure detection in kernelized correlation filter. Applied Sciences. 2020;**10**(2):713

[29] Tran DS, Ho NH, Yang HJ, Baek ET, Kim SH, Lee G. Real-time hand gesture spotting and recognition using RGB-D camera and 3D convolutional neural network. Applied Sciences. 2020;**10**(2): 722

[30] Zhao D, Liu Y, Li G. Skeleton-based dynamic hand gesture recognition using 3d depth data. Electronic Imaging. 2018; **2018**(18):461-461

[31] Plouffe G, Cretu AM. Static and dynamic hand gesture recognition in depth data using dynamic time warping. IEEE Transactions on Instrumentation and Measurement. 2015;**65**(2):305-316

[32] Singha J, Roy A, Laskar RH. Dynamic hand gesture recognition using vision-based approach for human–computer interaction. Neural Computing and Applications. 2018; **29**(4):1129-1141

[33] Barros P, Maciel-Junior NT, Fernandes BJ, Bezerra BL, Fernandes SM. A dynamic gesture recognition and prediction system using the convexity approach. Computer

Vision and Image Understanding. 2017; **1**(155):139-149

[34] Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2016;**39**(6):1137-1149

[35] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE; 2016. pp. 770-778

[36] Lowe DG. Distinctive image features from scale-invariant keypoints. International Journal of Computer vision. 2004;**60**(2):91-110

[37] Lindeberg T. Scale invariant feature transform. Scholarpedia. 2012;**7**(5): 10491

[38] Rojas R. Fast learning algorithms, in neural networks. Springer. 1996;**1996**: 183-225