



# An Efficient Human Computer Interaction through Hand Gesture Using Deep Convolutional Neural Network

Md. Milon Islam<sup>1</sup> · Md. Repon Islam<sup>1</sup> · Md. Saiful Islam<sup>1</sup>

Received: 3 June 2020 / Accepted: 8 June 2020 / Published online: 23 June 2020  
© Springer Nature Singapore Pte Ltd 2020

## Abstract

This paper focuses on the achievement of effective human–computer interaction using only webcam by continuous locating or tracking and recognizing the hand region. We detected the region of interest (ROI) in the captured image range and classify hand gestures for specific tasks. Firstly, background subtraction is used based on the main frame captured by webcam, and some preprocessing are done, and then YCrCb skin segmentation is used on RGB subtracted image. The ROI is detected using Haar cascade classifier for hand palm detection. Next, kernelized correlation filters tracking algorithm is used to avoid noise or background influences for tracking the ROI, and the median-flow tracking algorithm is used for depth tracking. The ROI is converted to a binary channel (black and white), resized to  $54 \times 54$ . Then gesture recognition is done using a 2D convolutional neural network (CNN) by entering the preprocessed ROI on the architecture. Two predictions are made based on skin segmented frame and image dilated frame, and gesture is recognized from the maximum value of those two predictions. The tracking and recognition process is continued until the ROI is presented on the frames. Finally, after validation, the proposed system has successfully obtained a recognition rate of 98.44%, which is usable for the practical and real-time application.

**Keywords** Human–computer interaction · Hand gesture detection · Hand tracking · Convolutional neural network

## Introduction

In computer science and language technology, gesture recognition [1] is getting emphasized. Hand, face, and other bodily motion are engaged to originate different types of gestures. Gestures are helpful to control electronic devices without physically touching them. Using cameras and computer vision algorithms [2, 3], interpretation of sign language [4]

is easily made and different signs have given different meaning to perform electronic devices functionalities.

Detection, tracking, and recognition are essential parts to achieve success in gesture recognition work. Following this, hand gesture recognition needs these three parts as well. In addition, two types of hand gestures are important which are static hand gesture and dynamic hand gesture. Static hand gestures can be created using static hand sign or fixed hand sign [5] and the dynamic hand gestures [6] can be created by recognizing the movement of the hand with sign, e.g., grabbing or swiping [7].

Recently, Microsoft introduced its depth camera and named as Kinect [8]. By the influence of Kinect depth camera, many methods that are based on depth information have evolved with directive knowledge. For example, Memo et al. [9] and Keskin et al. [10] proposed two frameworks in which they used some effective machine learning algorithms such as the random forest [11] to train up the architecture for capturing the skeletal structure of the hand. But comparing with a typical web camera, such depth cameras are expensive and the environment can affect to obtain a better result, which is limitations. In recent years, several human device interaction frameworks are developed based on sensor technology

---

This article is part of the topical collection “Advances in Computational Approaches for Artificial Intelligence, Image Processing, IoT and Cloud Applications” guest edited by Bhanu Prakash K N and M. Shivakumar.

---

✉ Md. Milon Islam  
milonislam@cse.kuet.ac.bd

Md. Repon Islam  
cse.repon@gmail.com

Md. Saiful Islam  
cse.saiful2119@gmail.com

<sup>1</sup> Department of Computer Science and Engineering, Khulna University of Engineering & Technology, Khulna 9203, Bangladesh

[12–15], computer vision [16–19], deep learning [20], smartphone [21, 22] and Internet of Things (IoT) [23–25] for different purposes.

The pipeline of the proposed system is as follows. Some preprocessing techniques are applied to detect the hand such as background subtraction, skin color segmentation, and noise processing. Cascade classifier is used to find the hand from skin segmented frame. Initially, background subtraction is done with the captured frame to remove an unnecessary region; then dilate effect or skin segmentation is applied to prepare frame to apply hand cascade classifier that obtains the desired hand region, our ROI in this work. The tracking algorithm is initialized after detecting the palm using a Haar cascade palm detector from the frame that entering on the lens of the webcam. Then the ROI is resized and passed into the CNN network to recognize the gesture. The tracking is then continuing with the ROI without detecting again the hand until the ROI exist in the frames and the CNN network continues to recognize the ROI. In the recognition phase, five different classes of the hand are recognized by 2D CNN with “ReLU” activation function and “softmax” activation function in the last layer and “adam” optimizer to compile the architecture by setting loss equal to categorical “cross\_entropy.” Distinguishing between ROI and background skin color is a difficult task in a dynamic environment. Other problems are raised due to the changes of intensity of light in different environments and variations of human skin color. Because of these, the background selection of the running system is a very sensitive task by avoiding those disturbances.

The remaining parts of the paper are organized as follows. “Materials and Methods” section describes the hand detection and tracking methods and designs the CNN network architecture for the recognition of hand gestures. The experimental results are shown in “Experimental Results Analysis”

section. Finally, the conclusions and future research with potential direction are discussed in “Conclusion” section.

## Materials and Methods

For the better result of recognition performances, some essential preprocessings have done on the initial images, background subtraction [26], noise processing, skin segmentation [27] using the YCrCb skin [28] range (0,133,77) to (235,173,127). The ROI is detected using the cascade classifier [29]. Then hand tracking is done using KCF and median-flow algorithm. Finally, the processed images resized to  $54 \times 54$ , converted to binary, and fed into the CNN network for gesture recognition. The overall processes of hand gesture recognition are shown in Fig. 1.

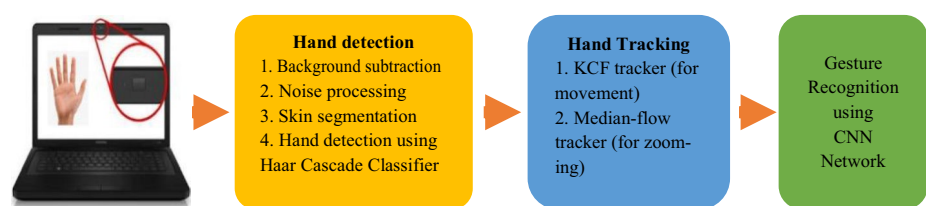
### Hand Detection Method

The initial step of hand gesture recognition is hand detection that is performed on webcam captured RGB images. Some critical preprocessing techniques are used to meet this purpose. Background subtraction, noise processing, skin color segmentation, and Haar cascade classifier are used for preprocessing the images. The hand detection process of the proposed system is demonstrated in Fig. 2 which is described as follows.

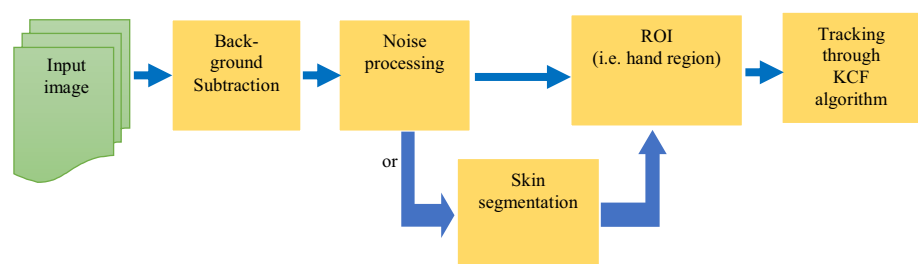
### Background Subtraction

Background subtraction is the most basic technique for computer vision preprocessing. Firstly, background subtraction is applied to webcam captured image based on the previous fifth frame and then again subtracted from the initial captured basic frame. Gaussian blur function [30], the

**Fig. 1** Overview of the overall processes of the hand gesture recognition system



**Fig. 2** The hand detection process of the proposed system



thresholding technique, is associated with performing absolute difference between two image frames. The followings are the steps of background subtraction.

- Step 1:** Gaussian Blur function to reduce noise
- Step 2:** Absolute difference between two image frames
- Step 3:** Convert color from RGB to gray scale
- Step 4:** Image thresholding technique for converting grayscale image to binary image
- Step 5:** Morphology function
- Step 6:** Image dilate effect

### Noise Reduction

To get a decent result from CNN architecture, obviously, the CNN network requires a noise free and predictable 2D image frame. To reduce or preprocess the noises, Gaussian blur, median blur, and dilation effect are used on a different part of this work. Figure 3 shows the frames altogether.

### Skin Segmentation

Then skin color segmentation is associated with that frame using YCrCb color channel. Besides, a binary frame is generated by converting the RGB image to grayscale image and applied image dilation effect. Finally, Haar cascade is used to detect hand from the preprocessed image frame.

### Haar Cascade Classifier

Haar cascade classifiers are effective feature-based object detection method proposed by [7]. Haar cascade classifiers are work with face detection, hand detection, and other object detection. For training the classifier, a number of positive images (images of hands) and negative images (images without hands) are entered into the algorithm to train the classifier. Then classifier is got ready to extract features from it.

Figure 3 demonstrates all the frames of this work. Here, Fig. 3a and b is the initial image that is used to background subtraction and one of Fig. 3a is used for very basic background image and another is for repeated subtraction. Figure 3c contains first hand to detect the ROI; then, further processes are executed through this image frame. Figure 3d shows the background subtracted frame, actually subtraction performs two times: first, based on Fig. 3b frame and then based on the initial frame. Figure 3e and f is binary images generated by thresholding and skin color segmented images, respectively. After that, KCF is used to track the hand region (ROI), which describes in section below.

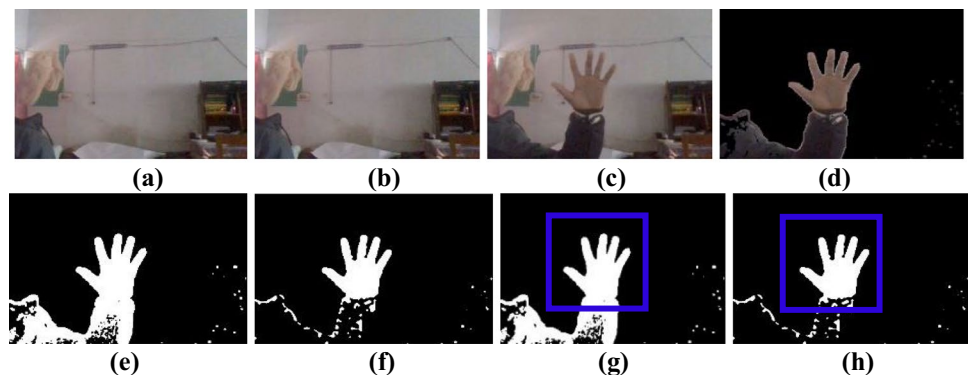
### Hand Tracking Method

After detecting the hand region as ROI, the second phase is hand tracking to determine the movement of ROI. Tracking means locating an object in successive frames of a video. Different ideas such as dense optical flow, sparse optical flow, Kalman filtering, meanshift and camshift, single object trackers, and multiple object track finding algorithms are exist there. Here, in this proposed system, we considered a single object tracker. We used KCF and median-flow single object tracker API which is provided by OpenCV as built in functionalities. KCF is used for avoiding the interception of moving skin color objects. Median-flow is used for zooming purposes. These two single object trackers are described below in brief.

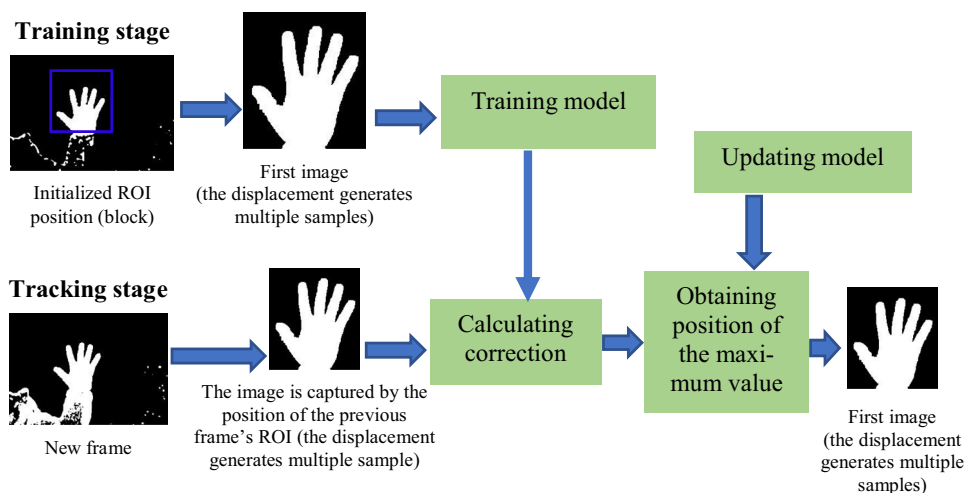
### KCF Algorithm

The KCF algorithm [31] can be described into two stages shown in Fig. 4. The training stage is the first stage which is indicated in Fig. 4 (top). In the training stage, the initial frame of the ROI (which is detected by the Haar cascade hand palm detection classifier from the background subtracted frame or further skin segmented frame) is used for the positive sample to train up the algorithm for tracking the object. Firstly, multiple training samples (negative samples) are generated using this initial ROI frame. Then,

**Fig. 3** a Indicates very basic initial background image, b indicates initial background that is used for repeated subtraction, c indicates the image with first-hand region, d indicates background subtracted image, e binary image obtaining from d, f skin segmented image, g detected hand region on binary image frame, h detected hand region on skin segmented image frame



**Fig. 4** Flowchart of the training stage of KCF tracking (top) and tracking stage of KCF tracking (bottom)



each positive and negative sample is fed to train; then based on the Gaussian probability density function (PDF) model, when a sample is closer to the positive sample, it obtains higher PDF value otherwise not.

The tracking phase is the second stage of the KCF algorithm as indicated in Fig. 4 (bottom). Whenever a hand comes in the captured frame, the system captures the image of the selected ROI of the previous frame; by this time, multiple samples are produced for displacement. The samples and the new frame are prepared for the KCF trained model which is indicated in Fig. 4 (top) and using those samples correction is calculated. Then, the ROI is modified based on the position of the maximum value. The targeted image will be repeatedly captured when it finds a new ROI position, then trained, and updated the model as indicated in Fig. 4 (top). When the hand is out of the camera range, the system stops its execution.

**Median-Flow Algorithm**

This tracker is used to zoom-in and zoom-out any document, photograph, and PowerPoint slide. Using this tracker, we measured the movement of ROI in a forward or backward direction [32]. If the movement is in a forward direction, then zoom-in operation will be performed otherwise zoom-out. Internally, this tracker tracks the object in both forward and backward directions in time and calculates the inconsistencies between these two trajectories. We realized that this tracker works best when the motion is predictable and small.

**CNN Architecture for Gesture Recognition**

This study proposed a convolutional neural network containing three convolutional and max pooling layers. After a

tensor is passed through the convolutional layers, it is flattened into a vector and passed through the dense layers [33]. The overview of the CNN architecture is shown in Table 1 and the overall architecture is shown in Fig. 5, which contains 3 convolutional layers, 3 max-pooling layers, 2 fully connected layers, and the final output layer connected to 5 classes to recognize gestures of 5 categories. Each step of the CNN architecture is as follows.

**Convolutional Layer**

Convolution layer is the basic building block in CNN [34, 35]. In the proposed system, we used three convolution layers with different convolution kernel 32-32-64 sequentially and kernel size is 3×3 for all. The first layer is given 54×54 sized input image; then, after going through the pooling layer, the second convolution layer gets 26×26 sized image and the last layer gets 12×12 sized input image. Also, ReLU activation function is performed in every convolution layers.

**Table 1** Overview of the CNN architecture

Layer	Type	Kernel size	Stride	Kernel	Input size
1	Conv2D	3×3	1	32	54×54×1
2	Pool	2×2	2	–	52×52×32
3	Conv2D	3×3	1	32	26×26×32
4	Pool	2×2	2	–	24×24×32
5	Conv2D	3×3	1	64	12×12×32
6	Pool	2×2	2	–	10×10×64
7	Full Connected	–	–	1600	5×5×64
8	Full Connected	–	–	64	1600
9	Output	–	–	5	64

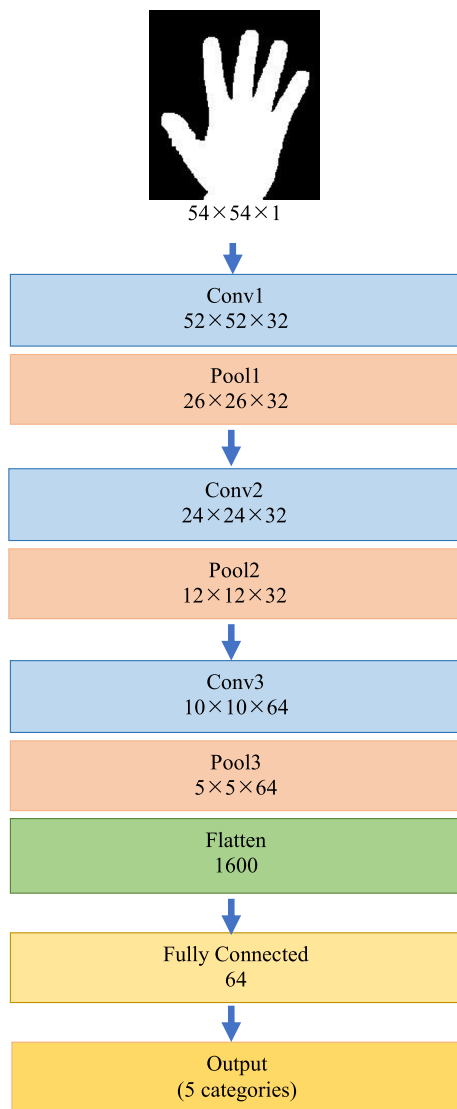


Fig. 5 Architecture of the 2D CNN network

### Pooling Layer

After every convolution layer, the pooling layer is merged to downsampling. Consequently, the obtained feature map sample by the convolution layer becomes one-fourth of the original sample. Max pooling method is used for pooling with  $2 \times 2$  kernel. This method takes maximum element among the  $2 \times 2$  sized mapped area elements. Around three times are used in the architecture, results in a fully connected layer get the input of  $5 \times 5$  sized sample.

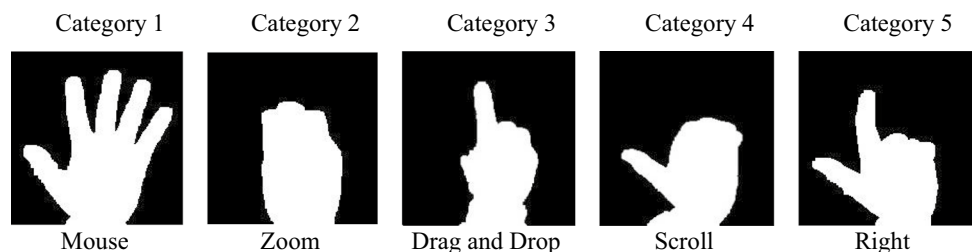
### Fully Connected Layer

The output of the last pooling layer gets into a fully connected layer. In this architecture, two fully connected layers are used to classify five common hand gestures. Around 64 neurons are set as input parameters. To overcome the overfitting problem, we add a dropout (0.5) method before fully connected layer and “softmax” activation function is used. Finally, after network testing is finished, all features are combined.

### Tuning Parameters

To initialize the network, weight parameters are randomly set in every layer of the network. The network batch size is 16, and “categorical\_crossentropy” is used as the loss function. Adam optimizer is used as an accuracy metric. After passing the result using the “softmax” function, the “categorical\_crossentropy” loss function is used to measure the error that occurs between the real label value and the result of the prediction. The index of the maximum value of predicted output probability array of the five categories is used for the gesture selection from the five classes which is shown in Fig. 6.

Fig. 6 Five hand gestures with the corresponding gesture label

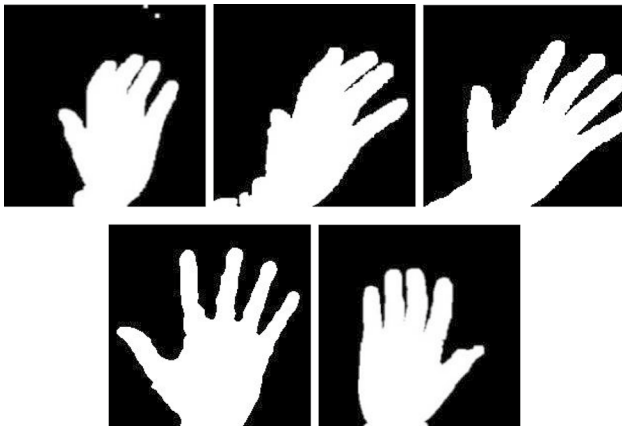


## Experimental Results Analysis

This section analyzed the results of our proposed system. We fixed the input image size to a ratio of 1:1, resized it to  $54 \times 54$ , and converted the images to the binary channel (black and white) to fix the neurons quantity in the fully connected layer. In order to recognize the gesture, we entered the converted images into the CNN architecture. We subtracted most of the background information, detected the skin by filtering YCrCb values, and applied some noise removal filters (e.g., blurring) to remove the unnecessary information so that the complexity of the system can be reduced and can be trained the network efficiently and faster.

### Prepared Data

A number of images are collected for different gestures, and a total of 3,000 images are created after preprocessing (background subtraction, skin filtering, and using other noise removal filters), 600 images for each gesture ( $5 \times 600 = 3000$



**Fig. 7** Examples of images after preprocessing at different angles

images). For training, we used 2,500 images and 500 images for each gesture ( $5 \times 500 = 2500$  images). These images were collected at various angles and different backgrounds, then preprocessed, resized to  $54 \times 54$  (a ratio of 1:1), and converted to binary to fed into the network. And finally, 500 images were used to validate the CNN architecture. Figure 7 shows some example images after preprocessing.

### Performances of the System

The settings of the network parameter for the CNN architecture are shown in Table 2. We subtract the background information and segmented the skin using YCrCb range from (0, 133, 77) to (235,173,127). After resizing the images and converting them into binary, we fed it into the network.

Several performance measures are used to evaluate a system like accuracy [36], error rate [37], precision [38], sensitivity [39], specificity [40], f1-score [41, 42], MCC [43], AUC [44] etc. In this system, we just considered accuracy and loss as evaluation metrics. The recognition results of the training set and validation set are shown in Table 3. The system successfully achieved the training set accuracy of 93.25%, loss of 0.19, and validation set accuracy of 98.44%, loss of 0.04 for gesture recognition. Figures 8 and Fig. 9 show the accuracy and loss curve of the CNN architecture, respectively, in which the accuracy of validation is higher than the accuracy of the training set and the validation loss is lower than the training loss.

**Table 3** Recognition results of the CNN architecture

Performance measures	Training set result	Validation set result
Accuracy	93.25%	98.44%
Loss	0.19	0.04

**Table 2** Network parameters for the CNN architecture

Tuning parameters	Value
Probability of dropout	0.5
Model optimizer	Adam
Size of batches	16
No. of epochs	26
Steps per epoch	2000
No. of images in the training set	$500 \times 5 = 2500$ images
No. of images in the validation set	500 images
Preprocessing of images	Background subtraction, skin segmentation using YCrCb range from (0,133,77) to (235,173,127), binary conversion using dilation

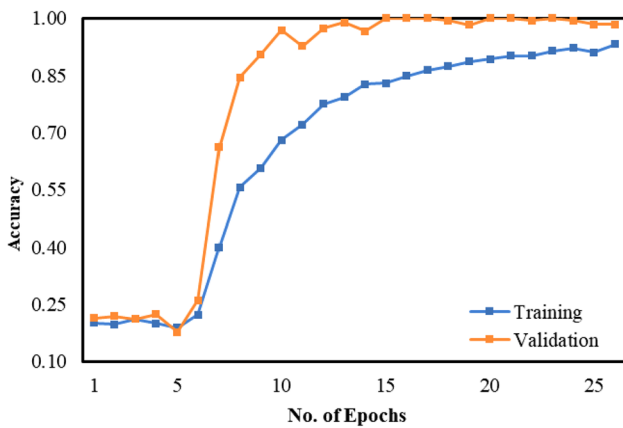


Fig. 8 Accuracy of the proposed system both for training and validation

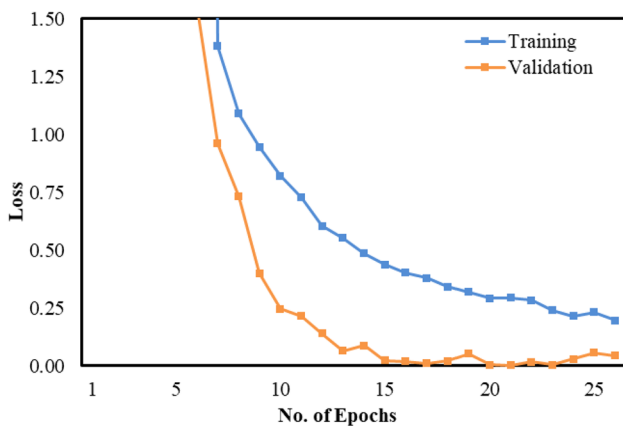
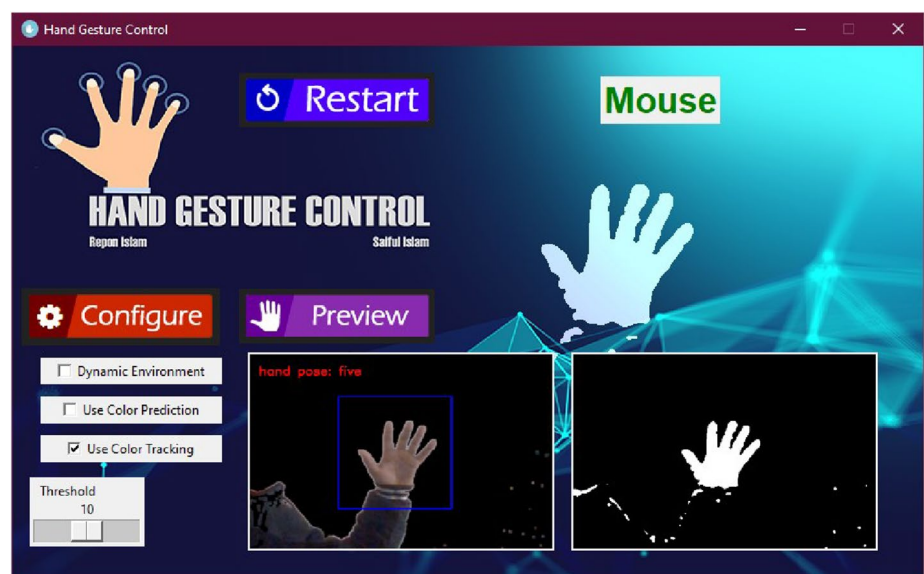


Fig. 9 Loss of the proposed system both for training and validation

Fig. 10 User interface of the proposed human–computer interaction system including all features



We developed an interface using this CNN architecture for human–computer interaction by performing some mouse and keyboard operations (e.g., mouse movements, clicking, scrolling, drag, and dropping, left key press, right key press, etc.). The user interface including all features is depicted in Fig. 10.

### Conclusions

The achievement of effective human–computer interaction using hand gesture recognition is the main focus of our study. Hand detection, tracking, and gesture recognition are the three main components of our proposed system. For the better result of recognition performance, some essential preprocessing, background subtraction, noise processing, skin segmentation using the YCrCb skin detection range have been done on the initial images. The ROI is detected using the Haar cascade classifier. Then hand tracking is done using KCF and median-flow algorithm. Finally, the processed images resized to  $54 \times 54$ , converted to binary images (black and white), and fed into the 2D convolutional neural network for gesture recognition from the five categories. Our proposed system achieved a higher performance result of recognition with validation accuracy of 98.44%

For detecting hand, background subtraction, skin segmentation, and noise processing are applied. But sometimes due to environmental complexity or in a dynamic environment, the developed system behaves unexpected, because in some conditions and cases skin segmentation finds some other objects that matches with human skin color. As a result, hand tracking is humped and the recognition step cannot

occur. Hence, it can be a future research scope to mitigate the errors due to the limitations of the environment, lighting condition, or color.

**Acknowledgment** The authors would like to thank the Department of Computer Science and Engineering (CSE), Khulna University of Engineering & Technology (KUET) to facilitate the work.

**Funding** No funding sources.

## Compliance with Ethical Standards

**Conflict of Interest** On behalf of all authors, the corresponding author states that there is no conflict of interest.

## References

- G Coleman, R Ward. Gesture recognition performance, applications and features. 4th ed. New York : Nova Science Publishers, Inc; 2018.
- A Voulodimos, N Doulamis, A Doulamis, E Protopapadakis. Deep learning for computer vision: a brief review. *Comput Intell Neurosci*. 2018;2018:1–13. <https://doi.org/10.1155/2018/7068349>.
- Rezwanul Haque M, Milon Islam M, Saeed Alam K, Iqbal H. A computer vision based lane detection approach. *Int J Image, Graph Signal Process*. 2019;11:27–34. <https://doi.org/10.5815/ijgsp.2019.03.04>.
- Dabre K, Dholay S (2014) Machine learning model for sign language interpretation using webcam images. In: 2014 International conference on circuits, systems, communication and information technology applications, CSCITA 2014
- Hasan H, Abdul-Kareem S. Static hand gesture recognition using neural networks. *Artif. Intell Rev*. 2014;41:147–81. <https://doi.org/10.1007/s10462-011-9303-1>.
- Molchanov P, Yang X, Gupta S, Kim K, Tyree S, Kautz J (2016) online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural Networks. In: proceedings of the IEEE computer society conference on computer vision and pattern Recognition
- Haria A, Subramanian A, Asokkumar N, Poddar S, Nayak JS. Hand gesture recognition for human computer interaction. *Procedia Comp Sci*. 2017;115:367–74.
- Zhang Z. Microsoft kinect sensor and its effect. *IEEE Multimed*. 2012;19(2):4–10. <https://doi.org/10.1109/MMUL.2012.24>.
- Memo A, Minto L, Zanuttigh P (2015) Exploiting silhouette descriptors and synthetic data for hand gesture recognition. In: Italian chapter conference 2015—smart tools and apps in computer graphics, STAG 2015
- Keskin C, Kiraç F, Kara YE, Akarun L (2011) Real time hand pose estimation using depth sensors. In: proceedings of the IEEE international conference on computer vision
- Rigatti SJ. Random Forest J Insur Med. 2017. <https://doi.org/10.17849/insm-47-01-31-39.1>.
- Rahman MA, Sadi MS, Islam MM, Saha P (2019) Design and Development of Navigation Guide for Visually Impaired People. In: 2019 IEEE International conference on biomedical engineering, computer and information technology for health (BECITHCON). IEEE, pp 89–92
- Khanom M, Sadi MS, Islam MM (2019) A comparative study of walking assistance Tools developed for the visually impaired people. 1st Int conf adv sci eng robot technol 2019, ICASERT 2019 2019:1–5. <https://doi.org/10.1109/ICASERT.2019.8934566>
- Rahman MM, Islam M, Ahmmed S. “BlindShoe”: an electronic guidance system for the visually impaired people. *J Telecommun Electron Comp Eng (JTEC)*. 2019;11:49–54.
- Habib A, Islam MM, Kabir MN, Mredul MB, Hasan M. Staircase detection to guide visually impaired people: a hybrid approach. *Rev d’Intell Artif*. 2019;33:327–34. <https://doi.org/10.18280/ria.330501>.
- Islam MM, Sadi MS, Zamli KZ, Ahmed MM. Developing walking assistants for visually impaired people: a review. *IEEE Sens J*. 2019;19:2814–28. <https://doi.org/10.1109/JSEN.2018.2890423>.
- Alam N, Islam M, Habib A, Mredul MB. Staircase detection systems for the visually impaired people : a review. *Int J Comp Sci Inf Secur (IJCSIS)*. 2018;16:13–18.
- Kamal MM, Bayazid AI, Sadi MS, Islam MM, Hasan N (2017) Towards developing walking assistants for the visually impaired people. In: 2017 IEEE region 10 humanitarian technology Conference (R10-HTC). IEEE, pp 238–241
- Islam MM, Sadi MS, Islam MM, Hasan MK (2018) A New Method for Road Surface Detection. In: 2018 4th International conference on electrical engineering and information & communication technology (iCEEICT). IEEE, pp 624–629
- Islam MM, Sadi MS (2018) Path Hole Detection to Assist the Visually Impaired People in Navigation. In: 2018 4th International conference on electrical engineering and information & communication technology (iCEEICT). IEEE, pp 268–273
- Islam MM, Neom NH, Imtiaz MS, Nooruddin S, Islam MR, Islam MR. A review on fall detection systems using data from smart-phone sensors. *Ing des Syst d’Inf*. 2019;24:569–76. <https://doi.org/10.18280/isi.240602>.
- Islam MM, Hasan MK, Billah MM, Uddin MM (2017) Development of smartphone-based student attendance system. In: 2017 IEEE region 10 humanitarian technology conference (R10-HTC). IEEE, pp 230–233
- Nooruddin S, Milon Islam M, Sharna FA. An IoT based device-type invariant fall detection system. *Internet Things*. 2020;9:100130. <https://doi.org/10.1016/j.iot.2019.100130>.
- Rahaman A, Islam M, Islam M, Sadi M, Nooruddin S. Developing IoT based smart health monitoring systems: a review. *Rev d’Int Artif*. 2019;33:435–40. <https://doi.org/10.18280/ria.330605>.
- Islam MM, Rahaman A, Islam MR. Development of smart healthcare monitoring system in IoT environment. *SN Comput Sci*. 2020;1:185. <https://doi.org/10.1007/s42979-020-00195-y>.
- Brutzer S, Höferlin B, Heidemann G (2011) Evaluation of background subtraction techniques for video surveillance. In: proceedings of the IEEE computer society conference on computer vision and pattern recognition
- Shaik KB, Ganesan P, Kalist V, Sathish B, Jenitha JMM. Comparative study of skin color detection and segmentation in HSV and YCbCr color space. *Procedia Comp Sci*. 2015;57:41–8. <https://doi.org/10.1016/j.procs.2015.07.362>.
- Tsagaris A, Manitsaris S. Colour space comparison for skin detection in finger gesture recognition. *Int J Adv Eng Technol*. 2013;6(4):1431.
- Soo S. Object detection using Haar-cascade Classifier. *Inst Comp Sci Univ Tartu*. 2014;2(3):1–12.
- Fisher R, Perkins S, Walker A, Wolfart E (2003) Spatial Filters—Gaussian Smoothing. *Image Process. Learn. Resour*.
- Tang M, Yu B, Zhang F, Wang J (2018) High-speed tracking with Multi-kernel correlation Filters. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition
- Grigorev A, Derevitskii I, Bochenina K. Analysis of special transport behavior using computer vision analysis of video from



- traffic cameras. *Commun Comput Inform Sci.* 2018;858:289–301. [https://doi.org/10.1007/978-3-030-02843-5\\_23](https://doi.org/10.1007/978-3-030-02843-5_23).
33. Cheng C, Parhi KK. Fast 2D convolution algorithms for convolutional neural networks. *IEEE Trans Circuits Syst I Regul Pap.* 2020. <https://doi.org/10.1109/TCSI.2020.2964748>.
  34. Lecun Y, Bengio Y, Hinton G. Deep learning. *Nature.* 2015;521:436–44.
  35. Haque S, Sadi MS, Rafi MEH, Islam MM, Hasan MK (2020) Real-Time Crowd Detection to Prevent Stampede. pp 665–678
  36. Muhammad LJ, Islam MM, Sharif US, Ayon SI. Predictive data mining models for novel coronavirus (COVID-19) Infected Patients Recovery. *SN Comput Sci.* 2020;1:216. <https://doi.org/10.1007/s42979-020-00216-w>.
  37. Hasan MK, Islam MM, Hashem MMA (2016) Mathematical model development to detect breast cancer using multigene genetic programming. In: 2016 5th International conference on informatics, electronics and vision (ICIEV). IEEE, pp 574–579
  38. Das S, Sadi MS, Ahsanul Haque M, Islam MM (2019) A machine learning approach to protect electronic devices from damage using the concept of outlier. In: 2019 1st international conference on advances in science, engineering and robotics technology (ICASERT). IEEE, pp 1–6
  39. Haque MR, Islam MM, Iqbal H, Reza MS, Hasan MK (2018) Performance evaluation of random forests and artificial neural networks for the classification of Liver Disorder. In: 2018 International conference on computer, communication, chemical, material and electronic engineering (IC4ME2). IEEE, pp 1–5
  40. Islam MM, Iqbal H, Haque MR, Hasan MK (2017) Prediction of breast cancer using support vector machine and K-nearest neighbors. In: 2017 IEEE Region 10 Humanitarian technology conference (R10-HTC). IEEE, pp 226–229
  41. Islam Ayon S, Milon Islam M. Diabetes prediction: a deep learning approach. *Int J Inf Eng Electron Bus.* 2019;11:21–7. <https://doi.org/10.5815/ijieeb.2019.02.03>.
  42. Milon Islam M, Kabir MN, Sadi MS, Morsalin MI, Haque A, Wang J. A novel approach towards tamper detection of digital holy quran generation. *Lect Notes Electr Eng.* 2020;632:297–308.
  43. Ayon SI, Islam MM, Hossain MR. Coronary artery heart disease prediction: a comparative study of computational intelligence techniques. *IETE J Res.* 2020. <https://doi.org/10.1080/03772063.2020.1713916>.
  44. Hasan M, Islam MM, Zarif MII, Hashem MMA. Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. *Internet Things.* 2019;7:100059. <https://doi.org/10.1016/j.iot.2019.100059>.

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