An ANN-based gesture recognition algorithm for smart-home applications

Phat Nguyen Huu[1*], Quang Tran Minh[2], and Hoang Lai The[1]
[1]School of Electronics and Telecommunications, Hanoi University of Science and Technology
Hanoi, Vietnam
[2]Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology, VNU-HCM
Ho Chi Minh City, Vietnam
[e-mail: phat.nguyenhuu @hust.edu.vn ; quangtran@hcmut.edu.vn; laithehoangbk@gmail.com]
*Corresponding author: Phat Nguyen Huu

Received October 17, 2019; revised December 27, 2019; accepted March 5, 2020; published May 31, 2020

Abstract

The goal of this paper is to analyze and build an algorithm to recognize hand gestures applying to smart home applications. The proposed algorithm uses image processing techniques combing with artificial neural network (ANN) approaches to help users interact with computers by common gestures. We use five types of gestures, namely those for Stop, Forward, Backward, Turn Left, and Turn Right. Users will control devices through a camera connected to computers. The algorithm will analyze gestures and take actions to perform appropriate action according to users requests via their gestures. The results show that the average accuracy of proposal algorithm is 92.6 percent for images and more than 91 percent for video, which both satisfy performance requirements for real-world application, specifically for smart home services. The processing time is approximately 0.098 second with 10 frames/sec datasets. However, accuracy rate still depends on the number of training images (video) and their resolution.

Keywords: 3-Dimensional Convolutional Network, Human-Computer Interaction, Smart-home, Machine Learning, IoT Applications.

1. Introduction

Today, computers are effective tools used in both work and human life by developing technologies. Therefore, human-computer interaction (HCI) systems have become more diversified. Humans interact with computers through mouse and keyboard. There are many interaction methods for example voice and motion recognition that make visualization for users. As a result, HCI systems have become one of the most interesting technologies.

We are able to use voice to interact with computer by analyzing audio signals from user. The method has many practical applications especially for smart homes. However, signal processing is still challenging since it requires high performance recording equipment. For the method of using motion sensors, their cost is too expensive.

Using of gestures in HCI systems is an effective method for people to communicate each other by hand. The gesture is combinations of different parts to convey information as shown in **Table 1**. Hand gesture is a type of important language to convey necessary signal or information because of the flexibility of hands, various forms, and postures. It brings a great deal to communicate each other. Therefore, hand gesture is the most suitable for HCI systems [1], and recognition and interaction hand gestures are the greatest research.

Table 1. The different parts of body for communicating [1]

Body parts	The number of differences
Head and Fig.s	10
Foot	8
Object and Fig.s	15
Hand and objects	20
Hand and Fig.s	24
Hand and head	28
Body	30
Others	35
Fig.s	40
Multiple hands	50
Objects	54
Hand	83

Hand gestures consist of static and dynamic ones. Depending on different applications, they can be categorized into many groups, namely conversational, controlling, and communication gestures. Sign Language (SL) is a case of conversational gesture. It uses the shapes and positions of hands to help people to communicate each other instead of speech. Controlling gestures are used in HCI systems to control Smart TV devices, Xbox game, and industrial robots. They are often used in human-to-human communication. Besides, people use hand gestures to communicate information. Using gestures in human-computer interaction is an easy method. Hand applications in HCI systems are interesting. Static and dynamic gestures are a posture of hand that is a specific combination of its position and direction at a defined time [2]. "STOP" and "GOODBYE" are the static and dynamic gestures. However, the systems remain many disadvantages since human gestures are varied in different contexts and their recognition cannot be successfully applied in all cases.

Based on the results, we propose a novel gesture recognition algorithm to apply for smart-home applications. Main contributions of paper is to propose model that recognizes

hand gestures using skin color and body posture combining with an artificial neural network (ANN) model to make action and developing a set of gestures for specific actions.

Gesture recognition and computer interface system are key factors in controlling digital devices in the future. This is an advanced technology in smart home applications. Currently, many companies and research offices are actively working on these systems, and they are important in automotive industry. High-tech models allow controlling screens without touching the devices. Peripheral devices such as the leap motion controller have been applied for gesture recognition. These technologies use cameras and infrared sensors to track activity of hands and fingers and then perform commands and make decisions for machine [1]÷ [4].

There are many researches for gesture recognition [1][3][4][5][6][8][9]. Tracking [1] is the method of monitoring hands after detection. In the paper, the authors detected and combined with tracking to improve computation speed. However, it depends on detection methods. The authors in [5] proposed a solution for recognizing static and dynamic gestures using Haar-like features and motion history images (MHI) to analyze area of interest (ROI). ROI is left or right of a face. Its size is 1.5 times of the face area. As a result, the exact recognition is 95.07% for static gestures and 95.66% for dynamic gestures. Although the accuracy of the method is quite high, user is dependent on characteristics of sample data. The authors in [6] used the convexity technique described as a technique of extracting all points surrounding a hand to find minimum set of points. After extracting feature vectors, the author uses hidden Markov model (HMM). The accurate of the method is 93.98%. Although the accuracy rate of method is high, it is only effective for gestures separated from background. Machine learning algorithms are applied for classifying gestures [8].

We divide machine learning approaches into four types, namely supervised learning, unattended learning, semi-supervised learning, reinforcement learning. We find that Artificial Neural Network (ANN) is a suitable approach to hand gesture recognition. The authors in [9] built a large dataset of 25 common gestures recorded from webcams (148092 videos). Their proposal is to build a three-dimensional convolutional network (3D-CNN) model to train the recognition model. They applied the technique and gained an accuracy rate up to 96.6%. Therefore, identification methods using CNN is often used for recognition methods.

However, there are several challenges for gesture recognition as follows:

- *Developing training sample:*
  Identification using machine learning requires suitable sample datasets. It takes long time for collecting data to create the standard samples.
- *Processing time:*
  We need to process large amounts of data. To recognize hand gestures, the system requires real-time identification and responses to their changes.
- *The accuracy of method:*
  For regular cameras (webcams), accuracy is influenced by other conditions such as light, background images, hand movement speed since we have to give several assumptions for applications.

Based on the results, we propose a novel gesture recognition system combining with ANN to make decisions. One of main points is the recognition of gestures and body postures.

The paper includes five parts and organized as follows. Section I presents an overview of human hand gestures and related work. Section II presents the proposed algorithm. Section III will evaluate the proposed model and analyze the results. In the final section, we give conclusions and future research directions

2. Proposal system

2.1 Overview of the proposed system

The proposed system is built for application in smart-home models. The goal of this system is to build simple and common gesture data. The proposed gestures consist of six gestures, namely Start, Stop, Forward, Backward, Turn Left, and Turn Right. Starting gesture is represented by action of fist hand and raising two fingers. Stop gesture is represented by action of spreading out and grasping hand. Movement forward, backward, left, and right gestures are represented by movement of hands toward up, down, left, and right directions. They are illustrated in **Fig. 1**.

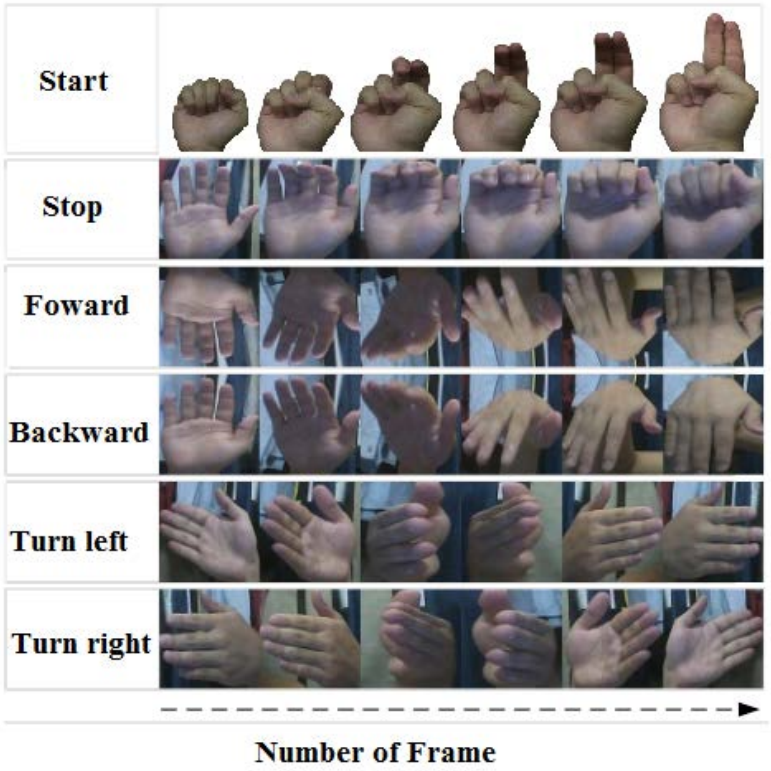


Fig. 1. Illustration of hand gestures.

Dynamic gesture is a series of hand movement that changes over time. Therefore, it is necessary to determine the starting and ending time of the gesture before performing identification. In the paper, we propose a method for recognizing hand gestures based on 2D images as shown in **Fig. 2**.

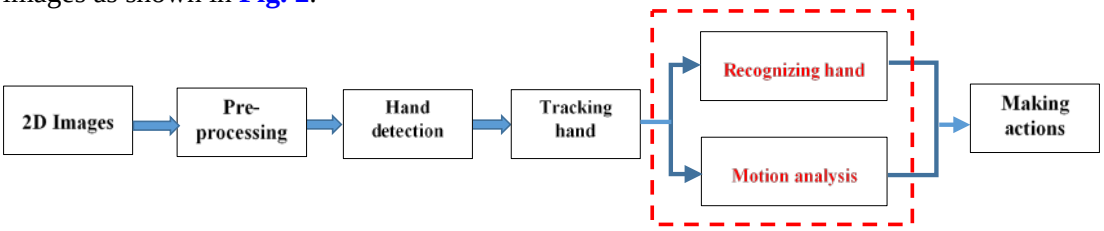


Fig. 2. Diagram of the proposed system on recognizing hand gestures based on 2D images.

The proposed system consists of three main steps, namely hand detection, hand tracking and gesture recognition. The first step is to detect hands using extracting features combining machine learning techniques to identify hands and divide hand areas. Since hand shape is diverse, the detection requires to use complex methods such as ANN with large training samples [8]. Gestures are started with opening hands where fingers are stretched out to the palm of the hand. Therefore, the first step will detect opening hands instead of different shapes in all frames.

In the tracking step, we use the technique to predict the position of a hand in the next frames to build movement trajectory. The final step is gesture recognition. It analyzes image data from detection and then performs identification that applies an extraction feature technique combining with an ANN model to identify the starting and ending positions of a gesture. Based on the results, we make decision about gestures. The diagram consists of five blocks as follows:

- Pre-processing block helps to improve the quality, light balance, and noise reduction of images.
- Hand detection block helps to detect partitions of a hand to performs following steps. When the hand areas are detected, we will apply tracking techniques in the next step.
- Tracking hand block is used to build the movement trajectory of gestures.
- Recognizing hand block helps to detect starting and ending gestures of hand.
- Motion analysis block analyzes trajectory of gestures based on information in the tracking step. Since dynamic gesture recognition is not able to peform on hand shapes, it needs to combine with motion, direction, and trajectory.

The advantages of proposed method are not dependent on the context and using simple backgrounds comparing with other methods [5][6]. The details of the method will be presented in the following section.

2.2 Process of implementation

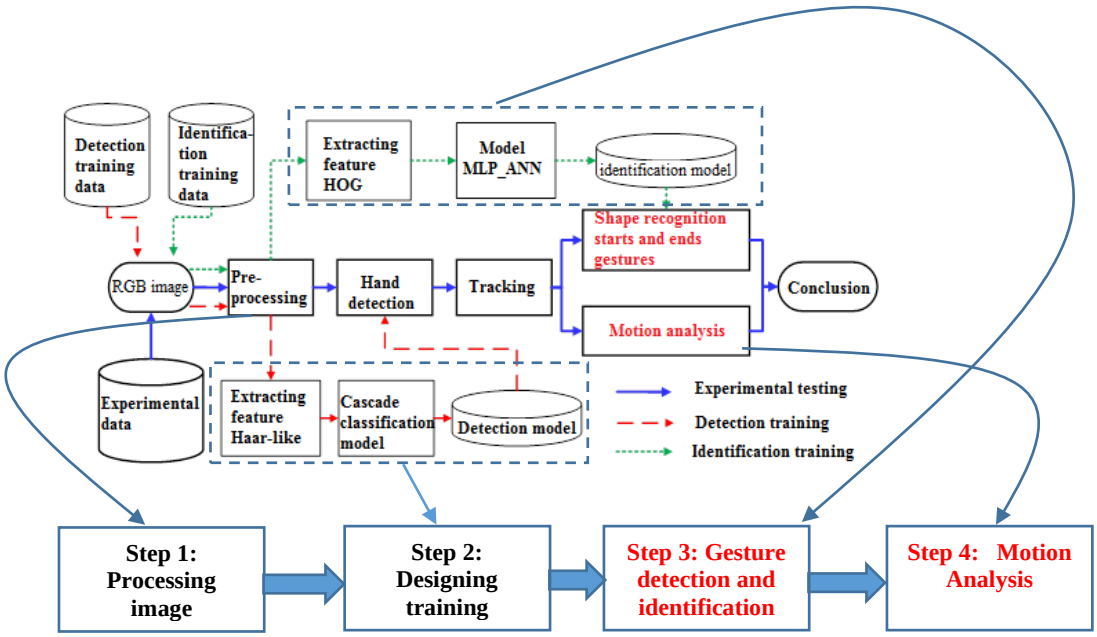


Fig. 3. Details of the proposed system implementation scheme

This section describes the details of the steps to perform the gesture recognition algorithm. The steps are described in **Fig. 3**.

Step 1: Processing image

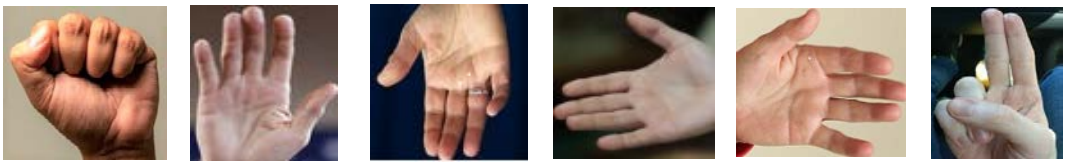
In the preprocessing step, we use a histogram balance and a low pass filter to eliminate noise. Histogram of an image is a graph describing distribution of gray values of pixels. Based on histogram, we can see to change the bright and dark of images. For input image data, we can balance the dark images. For filtering noise, we use to smooth function using a filter matrix (kernel) as in Eq. (1)

$$M = \frac{1}{(b+2)^2}\begin{bmatrix} 1 & b & 1 \\ b & b^2 & b \\ 1 & b & 1 \end{bmatrix}, \qquad (1)$$

where $b = 1$. Image will be improved the quality and noise is eliminated after preprocessing. It is applied for gesture in the detecting and the training phases.

Step 2: Designing training sample

Several training samples have been used such as MSRGesture3D, Cambridge-Gesture [6], 20bn-jester [9]. Each of them is built for a specific application based on types of gestures. During the processing implementation, we have built a sample set and combined selection of gesture samples (20bn-jester) for the proposed method. We divide them into two types, namely the detection and identification samples.



(a) (b) (c) (d) (e) (f)

Fig. 4. Detecting gesture (a) starting position of START, (b) starting position of BACKWARD, (c) starting position of FORWARD, (d) starting position of TURN LEFT, (e) starting position of turn right, (f) ending position of START [9].

As analysis above, we first have to detect the starting and ending gestures. Therefore, positions of detection is opening hand and stretching fingers straight with palm (first gesture and two fingers) as shown in **Fig. 4**. The sample images are captured and collected under a variety of lighting and background conditions. Combining with images as shown in **Fig. 4** [9], we have built the sample set with 5635 images containing starting positions of hands as shown in **Fig. 4**. To design the recognition sample, we perform similar to training data set. **Table 2** shows the starting and ending shapes of each gesture.

Table 2. Classification gestures

Type of gesture	Illustrations	
	Starting	Finishing
START		
STOP		
GO FORWARD		
GO BACKWARD		
TURN LEFT		
TURN RIGHT		

In **Table 2**, we recognize that starting of START is similar to ending of STOP, starting of STOP is similar to starting of BACKWARD, and ending of FORWARD. Besides, starting of FORWARD is similar to ending of BACKWARD. Starting of the TURN LEFT is similar to ending of TURN RIGHT and starting of TURN RIGHT is similar to ending of TURN LEFT. Therefore, it is only necessary to identify the six types of postures as shown in **Fig. 5**.

Fig. 5 illustrates directions of hand that need to be identified. They are labeled from 0 to 5. Positions 1, 2, 3, 4, and 5 are collected from detected dataset. Labeled 0 is built by ourself with 640 images in different context and light. Besides, we need to add images that are not belong to **Fig. 5**. They are identified by a machine learning model.

Fig. 5. Illustrations of hand positions.

Step 3: Gesture detection and identification training model

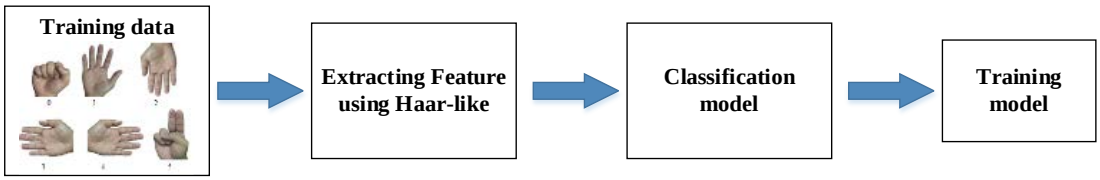


Fig. 6. Gesture training model based on hand detection algorithm [5][7].

As described in the previous section, hand detection will be applied for opening gestures as shown in **Fig. 6**. The purpose of this step is to train hand data to carry out detection and partition of them. Training model is shown in **Fig. 6** [5].

Preparing a training sample: The sample images are cut and preprocessed to eliminate noise before training. Besides, we also add background images for training processing. Total of 6 handsets are identified for four positions for each sample. There are 5635 postures and 2000 backgrounds as shown in **Fig. 7**.

Fig. 7. Several background for detection training (Source Internet).

Conducting the training dataset: We conduct a training dataset with six types of gestures. The training process includes two steps, namely extracting Haar-like feature and inputting data into a machine learning model to train using OpenCV program as follows.

Firstly, we create a characteristic vector with OpenCV library by command:
opencv_createsamples.exe -info location.txt -vec positive.vec -w 32 -h 32,
where “info” is the file that contains name and location of the cutting images,
“vec” is the file that contains output vectors, and *w* and *h* are width and height of images.

Secondly, we use OpenCV library to train data after creating the vector file by command:
opencv_traincascade.exe -data ..\\ out -vec pos.vec -bg negative.txt -numPos 1110 -numNeg 2000 -w 32 -h 32,
where *data* is the training output file, *vec* is the input characteristic vector, *bg* is the name and location of background image (negative image), and *numPos* and *numNeg* are number of positive and negative images, *w* and *h* are width and height of images. Training time depends on speed of computer and resolution of positive and negative images. After completing training, we will generate the "cascade.xml" file that contains trained data. **Fig. 8** shows several results where hand area is defined by red square.

Fig. 8. Several results of hand identification using Haar-like [5][9].

Gesture regonization: At this stage, we start to recognize the starting and the ending of a gesture as shown in **Fig. 4** as follows.

Preparing the training sample: This step preprocesses sample images where we choose 64×64 image and assign labels for them as shown in **Fig. 5**. Therefore, the training dataset consists of six types of gestures and are assigned from 0 to 5.

Training implementation: The goal of this step is to find parameters of multi-layer perceptron artificial neural network (MLP_ANN) model to identify labels for input data. The process consists of two steps, namely extracting HOG features [10] and building MLP_ANN as follows.

Firstly, we perform to extract features with training images based on the histogram of oriented gradients (HOG). We use several functions *HOGDescriptor* (win_size, block_size, block_stride, cell_size) to extract their features, where win_size is size of input image, block_size is size of window, block_stride is window jumping, cell_size is size of cell, and *n_bin* is number of characteristic vectors. After extracting characteristics, we get a 1764-dimensional vector. Extracting features from each gesture, obtained results are characteristic vectors as digital data.

Secondly, we use MPL network with reverse propagation algorithm to carry out training data. We carry out network training through weighting of hidden and output layers.

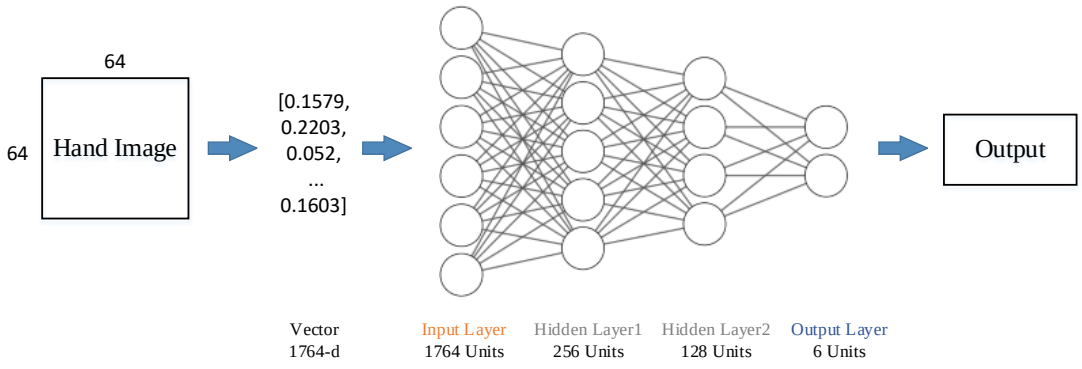


Fig. 9. Diagram of gesture recognition using MLP_ANN

We perform with different network architectures such as changing number of layers, number of units, selecting appropriate learning rate and several neural network optimization techniques. We also conduct training and evaluation on testing data. Based on comparing between network size and accuracy, we select MLP network that has 1764 input layer units corresponding to HOG characteristic vector value and 256 hidden layer units, and 6 output layer units. After training the network, output is a file with network configuration parameters that is used to build data. The results are shown in subsection 3.2.

Step 4: Motion Analysis

The gesture trajectory is designed from coordinates of hand obtained by each frame in the tracking step. It is motion direction of hands from starting to ending gestures.The postilions are analyzed by motion vector as in Eq. (2).

$$\varphi_n = \arctan\left(\frac{(y_n - y_{n-1})/h}{(x_n - x_{n-1})/w}\right), \qquad (2)$$

where, $\boldsymbol{\varphi_n}$ is direction value of the motion vector (radian); $\mathbf{x_n}$ and $\mathbf{y_n}$ are coordinates of current frame; $\mathbf{x_{n-1}}$ and $\mathbf{y_{n-1}}$ are coordinates of previous frame; and **w** and **h** are width and height of the image.

Table 3 shows hand coordinate and direction of the vector based on motion trajectory. From the data, it has been found that magnitude of motion vectors from 5th to 23rd frames are about zero. Therefore, it can be concluded that orbital motion is almost linear and is movement direction to left side.

Table 3. Motion direction gesture into left side.

Image	x	Y	$\boldsymbol{\varphi_n}$ (radian)
5	167	247	0, 45
6	169	246	0,58
7	172	246	0
8	176	247	-0,32
9	180	248	-0,32
10	188	247	-0,16
11	198	245	-0,26
12	208	242	-0,38
13	222	239	-0,27
14	238	235	-0,32
15	255	233	-0,15
16	308	237	-0,10
17	309	240	0,02
18	313	235	0,13
19	304	232	0,41
20	303	232	0,12
21	305	235	0,10
22	314	244	0,27
23	320	249	0,37

Based on the results, we conclude that the type of gestures depends on magnitude of the motion vector. Due to limitations of 2D camera, it is difficult to detect all hand gestures in the paper. Therefore, it is necessary to rely on starting and ending positions combining with the motion to decide exactly.

3. Results and Analysis

The platform is Ubuntu on Laptop with **CPU** (i5, 1.6GHz, 4GB RAM). We capture images from 2D cameras with low resolution. We use data based on [6][8]. After implementing program with training and identifyin samples, we obtain several results as follows.

3.1. Hand detection

We evaluate position hands in section 2. The evaluated data includes 2000 images for each gesture from [9]. Images are captured from different contexts and lighting and each of them only contains one hand. **Tables 4 and 5** present the results of four types of gestures with 352 × 200 and 704 × 400 input images.

Table 4. The results are detected with 352x200 input image.

Type of posture of the gesture in Fig. 9	Number of images	Detecting correctly	Not detecting	Detecting wrong	Time of detecting gesture (second/frame)	Accuracy
0	1000	973	9	18	0.037	97.3%
1	1000	947	22	31	0.042	94.7%
2	1000	975	13	12	0.037	97.5%
3	1000	987	8	5	0.041	98.7%
4	1000	980	11	9	0.036	98.0%
5	1000	932	25	43	0.042	93.2%

Table 5. Test results are detected with 704x400 input image.

Type of posture of the gesture in Fig. 9	Number of images	Detecting correctly	Not detecting	Detecting wrong	Time of detecting gesture (second/frame)	Accuracy
0	1000	981	7	12	0.059	98.1%
1	1000	952	31	17	0.073	95.2%
2	1000	975	11	14	0.063	97.5%
3	1000	983	9	8	0.071	98.3%
4	1000	972	12	16	0.068	97.2%
5	1000	931	24	45	0.075	93.1%

The accuracy of hand detection is greater than 91% and proceesing time is 0.039s with image size 352x200; 0.068s with image size 704 × 400. The accuaracy rate depends on the number of training images and their resolution. **Fig. 9** shows several cases that can not detect hand.

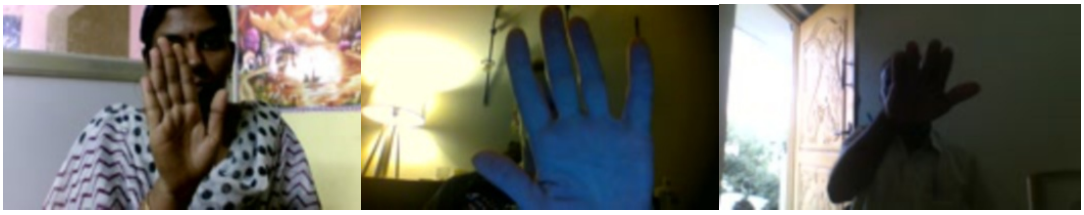


Fig. 9. Several results do not detect hand posture.

Fig. 10 shows the results of hand detection where images are captured on different backgrounds. Hand area is identified by blue rectangle. The average accuracy of the detection is greater than 91.5% and the detection time is around 0.039s (24 frames/sec). The accuracy rate of the detection depends on the number of training images and their resolution.

Table 6 gives results comparing with several related works. Most hand detectors using object detection algorithms with CNN architectures achieve high accuracy. However, they use highly configurable hardware. In this paper, we propose to remove background and using Haar-like features and an appropriate machine learning model. As a result, the accuracy of the proposed method is 91.5% with lower hardware while the highest accuracy is 97%.

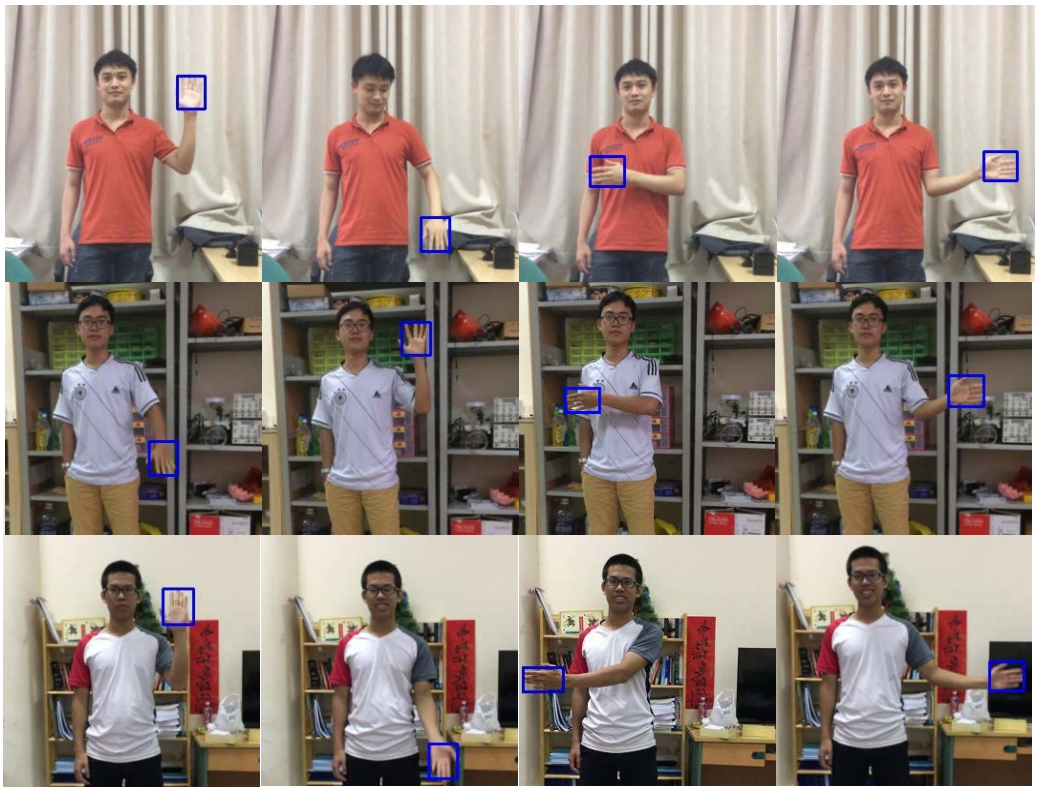


Fig. 10. Several results are detected by right hand.

Table 6. Comparision with other hand gesture detection systems.

Reference method	Vocabulary set	Platform	Accuracy (%)	Processing time for frame detection (second)	Hardware
M.Kounavis [12]	Combination edge detection, finger template description	N/A	82.82	From 10 to 15	**CPU** (i5, 2.3GHz, 16GB RAM)
K. P. Feng , F. Yuan [13]	HOG characters and support vector machines	CPU	91.3	N/A	N/A
Tofighi et al. [14]	Feature extraction and using k-NN classifier and SVM classifiers	N/A	93÷96	N/A	N/A
V. Dibia [8]	Single shot detector method with CNN	GPU	97	0.5	**CPU** (i7, 2.5GHz, 16GB RAM)
Proposal method	**Haar-like characters and cascade classifier**	**CPU**	**91.5**	**0.039**	**CPU** (i5, 1.6GHz, 4GB RAM)

3.2. Recognizing gestures of static image

We identify with input data containing starting and ending positions of gestures as shown in **Fig. 5**. The experimental data are collected by [9] with 1500 images for 1, 2, 3, 4, and 5 labels. We build 500 images for 0 label. Conducting identification with five labels, the results are shown in **Table 7** and **Fig. 11**.

Table 7. Results of recognition for starting and ending of gestures

Type of starting posture of the gesture	Number of images	Classifying correctly	Classifying wrong	Inference time (second/frame)	Accuracy	F1-Score
0	500	488	12	0.056	97,60%	0.98
1	1500	1473	27	0.062	98,20%	0.99
2	1500	1468	32	0.061	97,86%	0.96
3	1500	1479	21	0.059	98,60%	0.97
4	1500	1475	25	0.061	98,33%	0.98
5	1500	1472	28	0.060	98.13%	0.97

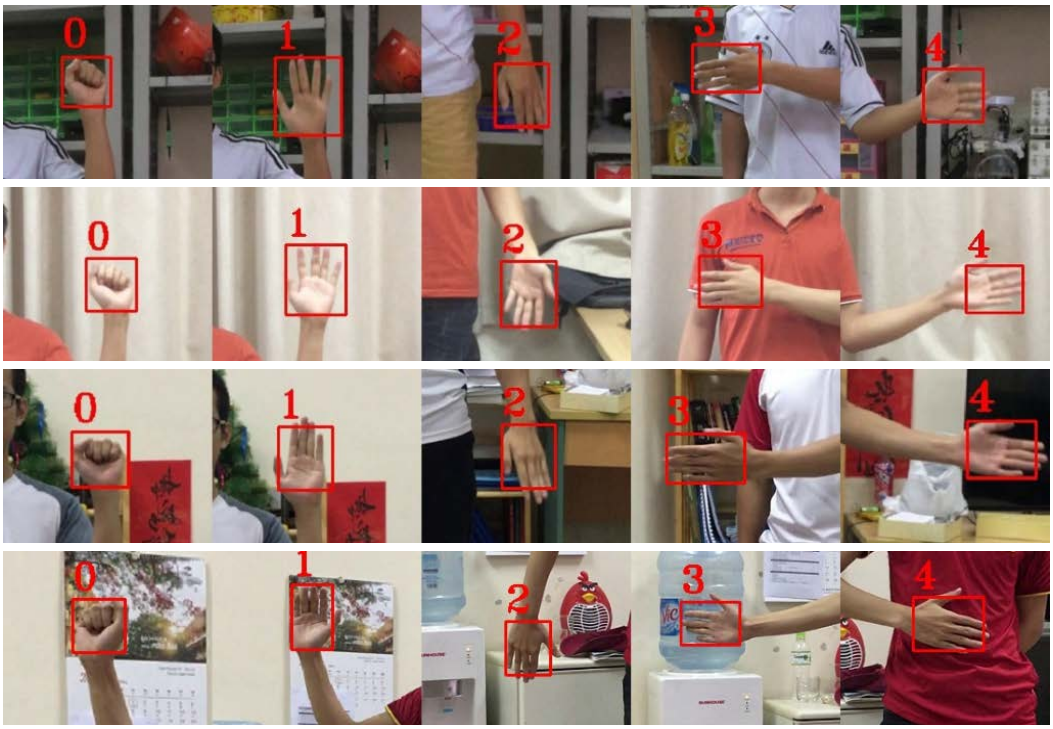


Fig. 11. Results for several correct hand postures.

3.3. Recognizing gesture of video

Experimental samples are videos that are attributed in different context and lighting conditions. Applying results in two steps of detection and combining to beginning and ending gestures, we have discussed about types of gestures. Experimenting with 40 self-video for each gesture, identification results are presented in **Table 8**.

Experiments show that gesture identification achieves average results up to 91.5% of accuracy. The method is not dependent on regional recognition (ROI) [5], less affected by background image and is comparable with [6]. The algorithm can run on low hardware configuration without GPU comparing to [9] but still achieves real-time results that is suitable for HCI systems.

Table 8. Results for recognizing gesture based on video

Type of the gesture	Number of videos	Recognizing correctly	Not recognizing	Time of detecting gesture (second/frame)	Accuracy
Starting	40	38	2	0.18	95.0%
Stoping	40	40	0	0.16	100,0%
Going forward	40	36	4	0.21	90,0%
Going backward	40	35	5	0.19	87,5%
Turning left	40	38	2	0.20	95,0%
Turning right	40	34	6	0.20	85,0%

In this paper, we have built an algorithm to recognize hand gestures and identified five basic gesture types. Although experimental data are limited, it can be improved by collecting additional gestures for re-training to increase the accuracy in steps.

4. Conclusion

We solved the problem of gesture recognition in the paper. The algorithm is designed to apply for motion to control robot movement. The main contributions of the paper are as follows:

- Proposing the sample set for moving robot. It is not only natural but also practical in order to command moving robots.
- Proposing a novel gesture recognition approach based on space and time characteristics of gestures. Neural networks are used in gesture recognition combining with motion trajectory analysis to provide accurate decisions.
- In the future, we will develop algorithm by:
- Collecting more gesture samples to increase the accuracy of gesture detection models.
- Processing of input data is deep image and combining with other methods such as CNN, neural network models.
- Replacing hand detection using Haar feature-based cascade classifiers with specific deep learning object detection methods, namely SSD [8][23] or YOLO [24][25].
- Upgrading powerful system hardware to perform real-time deep learning tasks.
- Dynamic gesture recognition can be analyzed by direction of video recognition, sequence of real time videos. This problem is performed by Recurrent Neural Network (RNN).
- Deploying RNN with high hardware devices to solve image sequence identification problems will be performed on our system in the future.
- Combining motion recognition and robot control when determining distance to obstacle based on the results of our papers [16] ÷ [21] to command control-moving robot.

Acknowledgment

This research is carried out in the framework of the project funded by the Ministry of Education and Training (MOET), Vietnam with the title "*Research and development of gesture recognition system, action application of artificial intelligence in smart homes*" under the grant number B2020-BKA-06. The authors would like to thank the MOET for their financial support.

References

[1] H. Hasan and S. Abdul-Kareem, "RETRACTED ARTICLE: Human–computer interaction using vision-based hand gesture recognition systems: a survey," *Neural Comput & Applic*, 25, 251–261, 2014. Article (CrossRef Link)

[2] M. V. Lamar, "Hand gesture recognition using T-Comb net a neural network model dedicated to temporal information processing," *Ph.D. thesis, Nagoya Institute of Technology, Japan*, 2002. Article (CrossRef Link)

[3] S. S. Rautaray and A. Agrawal, “Vision based hand gesture recognition for human computer interaction: a survey,” *Artif Intell Rev.*, 43, 1–54, 2015. Article (CrossRef Link)
[4] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Kauai, HI, USA*, I-I, 2001. Article (CrossRef Link)
[5] C. C. Hsieh and D. H. Liou, “Novel Haar features for real-time hand gesture recognition using SVM,” *J Real-Time Image Proc*, 10, 357–370, 2015. Article (CrossRef Link)
[6] P. Barros, N. T. M. Junior, B. J. T. Fernandes, B. L. D. Bezerra , and S. M. M. Fernandes, “A dynamic gesture recognition and prediction system using the convexity approach,” *Computer Vision and Image Understanding*, 155, 139-149, 2017. Article (CrossRef Link)
[7] S. Bilal, R. Akmeliawati, M. J. E. Salami, A. A. Shafie, and E. M. Bouhabba, "A hybrid method using haar-like and skin-color algorithm for hand posture detection, recognition and tracking," in *Proc. of 2010 IEEE International Conference on Mechatronics and Automation, Xi'an*, 934-939, 2010. Article (CrossRef Link)
[8] V. Dibia, “How to Build a Real-time Hand-Detector using Neural Networks (SSD) on Tensorflow,” *GitHub repository*, 2017. retrieved from official website: Article (CrossRef Link)
[9] T. B. Neurons, “Gesture recognition using end-to-end learning from a large video database,” *Medium Corporation*, 2017. Article (CrossRef Link)
[10] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. of 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA*, 886-893, 2005. Article (CrossRef Link)
[11] C. Huang and W. Huang, “Sign language recognition using model-based tracking and a 3D Hopfield neural network,” *Machine Vision and Applications*, 10, 292–307, 1998. Article (CrossRef Link)
[12] M. Kounavis, "Fingertip detection without the use of depth data, color information, or large training data sets," in *Proc. of 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Banff, AB*, 2396-2401, 2017. Article (CrossRef Link)
[13] K. Feng and F. Yuan, "Static hand gesture recognition based on HOG characters and support vector machines," in *Proc. of 2013 2nd International Symposium on Instrumentation and Measurement, Sensor Network and Automation (IMSNA), Toronto*, 936-938, 2013. Article (CrossRef Link)
[14] G. Tofighi, A. Venetsanopoulos, K. Raahemifar, S. Beheshti, and H. Mohammadi, “Hand posture recognition using K-NN and Support Vector Machine classifiers evaluated on our proposed HandReader dataset.” in *Proc. of 18th International Conference on Digital Signal Processing (DSP)*, 1-5, 2013. Article (CrossRef Link)
[15] E. Stergiopoulou and N. Papamarkos, “Hand gesture recognition using a neural network shape fitting technique,” *Engineering Applications of Artificial Intelligence*, 22 (8), 1141-1158, 2009. Article (CrossRef Link)
[16] P. N. Huu and H. L. The, “Low-Complexity Image Processing Algorithm for Estimating Distance and Actual Size of Subject,” in *Proc. of 7th International Conference on Communications and Electronics (ICCE 2018), Hue, Vietnam*, 463-468, 2018. Article (CrossRef Link)
[17] Y. Freund and R. E. Schapire, “A Short Introduction to Boosting,” *Journal of Japanese Society for Artificial Intelligence*, 14(5), 771-780, 1999. Article (CrossRef Link)
[18] P. N. Huu, V. T. Quang, and T. Miyoshi, "Multi-hop Reed-Solomon encoding scheme for image transmission on wireless sensor networks," in *Proc. of 2012 Fourth International Conference on Communications and Electronics (ICCE), Hue, Vietnam*, 74-79, 2012. Article (CrossRef Link)
[19] P. N. Huu, V. T. Quang, and T. Miyoshi, “Low-Complexity and Energy-Efficient Algorithms on Image Compression for Wireless Sensor Networks,” *IEICE Transactions on Communications*, E93-B (12), 3438-3447, 2010. Article (CrossRef Link)
[20] V. T. Quang, P. N. Huu and T. Miyoshi, "Adaptive transmission range assignment algorithm for in-routing image compression on wireless sensor networks," in *Proc. of International Conference on Communications and Electronics 2010, Nha Trang, Vietnam*, 18-23, 2010. Article (CrossRef Link)

[21] S. Bilal, R. Akmeliawati, and M. Salami, “A hybrid method using haar-like and skin-color algorithm for hand posture detection, recognition and tracking,” in *Proc. of IEEE Intl Conf. on Mechatronics and Automation*, 4-7, 2010. Article (CrossRef Link)
[22] Z. Zhao, P. Zheng, S. Xu, and X. Wu, "Object detection with deep learning: a review," *IEEE Transactions on Neural Networks and Learning Systems*, 30(11), 3212-3232, 2019. Article (CrossRef Link)
[23] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multiBox detector," *Computer Vision and Pattern Recognition (ECCV)*, 21-37, 2016. Article (CrossRef Link)
[24] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proc. of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV*, 779-788, 2016. Article (CrossRef Link)
[25] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: inverted residuals and linear bottlenecks," in *Proc. of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT*, 4510-4520, 2018. Article (CrossRef Link)

Phat Nguyen Huu received his B.E. (2003), M.S. (2005) degrees in Electronics and Telecommunications at Hanoi University of Sceience and Technology (HUST), Vietnam, and Ph.D. degree (2012) in Computer Science at Shibaura Institute of Technology, Japan. Currently, he lecturer at School of Electronics and Telecommunications, HUST Vietnam. His research interests include digital image and video processing, wireless networks, ad hoc and sensor network, and intelligent traffic system (ITS) and internet of things (IoT). He received the best conference paper award in SoftCOM (2011), best student grant award in APNOMS (2011), hisayoshi yanai honorary award by Shibaura Institute of Technology, Japan in 2012.

Quang Tran Minh (quangtran@hcmut.edu.vn) is an associate professor at Faculty of Computer Science and Engineering, Hochiminh City University of Technology, Vietnam and a visiting researcher at Shibaura Institute of Technology, Tokyo, Japan. He has been a researcher at Network Design Department, KDDI Research Inc., Japan (2014-2015) and a researcher at Principles of Informatics Research Division, National Institute of Informatics (NII), Japan (2012-2014). His research interests include mobile and ubiquitous computing, IoT, network design and traffic analysis, disaster recovery systems, data mining, and ITS systems. Prof. Quang received his Ph.D. in Functional Control Systems from Shibaura Institute of Technology. He is a member of IEEE, ACM.

Hoang Lai The has received his B.E degree in Electronic and Telecommunications at Hanoi University of Science and Technology(HUST), Vietnam in 2018. Currently, he is working in R&D department at Lumi Viet Nam.,Jsc. His main duty is developing smart products which relates to digital image, video processing, machine learning and internet of things (IoT).