Smart Electronic Switching (ON/OFF) System Based on Real-time Detection of Hand Location in the Video Frames

Duaa A. Taban¹, Ali Al-Zuky², Shaymaa H. Kafi³, Anwar H. Al-Saleh⁴, Haidar J. Mohamad⁵

¹Department of physics, College Of Education for Pure Science (Ibn Al-Haitham), University of Baghdad
²,³,⁵ Department of physics, College of Science, Mustansiriyah University
⁴ Department of Computer Science, College of Science, Mustansiriyah University
¹duaa_taban@yahoo.com, ²prof.alialzuky@uomustansiriyah.edu.iq,
³shaymaa.h.kafy@uomustansiriyah.edu.iq, ⁴anwar.h.mi@uomustansiriyah.edu.iq, ⁵haidar.mohamad@uomustansiriyah.edu.iq

Abstract
Human looking forward to living in a modern and comfortable environment like smart houses. In this study, an effective user-friendly smart home prototype designed with low cost. The prototype contains eight of Light Emitting Diode (LED) considered as home appliances and controlled in real-time using eight suggested hand cases. The hand cases have different position regarded to head and shoulder levels. The hand position is detected using a new suggested algorithm programmed in Matlab software. Viola-Jones method used to detect hand in a complex background (hand with a different background) by training computer using positive (hand) and negative (non-hand) image datasets. To make computer training faster and accurate, a new idea depends on a skin detection used before computer training to determine the location and size of all positive images automatically. The LEDs in prototype switched ON/OFF using the suggested hand cases in a fast time. Where the response time of LEDs to hand cases was 0.43 second.

Keywords: Viola-Jones method, Arduino, hand detection, smart home.

1. Introduction
Nowadays, lifestyle is changing dramatically as a result of increasing of modern technology. Where the idea of a simple home is developed into a smart home [1]. The smart home can be described as the reaction according to the surrounding environment orders and give the feedback accordingly [2]. Smart home control is one of the important technology keys in home automation. The smart home can be controlled using intuitive and intelligent interface methodology like hand gestures since the interaction between user and home devices achieved in a simple way [3]. Adopting hand gesture as interfaces in a vision-based smart home require to detect hand (determining hand location and its size in a still image or video sequences), where one or more camera should be used to capture the hand images. Hand gestures can be detected using computer vision techniques basing on extracting some feature such as hand shape, skin color, and hand models. Hand gesture detection is a challenging task as the hand has different and variable gestures (signs). Also, the hand has more than 25 degrees of freedom, considering
fingers, wrist and elbow joints. These challenges make detecting the hand is very difficult in cluttered backgrounds and inappropriate illumination [4, 5]. However, controlling a smart home using hand gestures has some advantages since the user can interact with home appliances from a distance. Moreover, the devices are far to have physical contact.

Many researchers attempt to detect hand in input images using different approaches like Van-Toi Nguyen et. al [6] by presenting a method to detect hand based on the Viola-Jones algorithm. Their technique lacks a big database as well as it avoids false detection of background as hand. Ruchi. M. Gurav et. al [7] detected hand gesture in real-time using Adaptive Boosting (AdaBoosting) algorithm based on the Viola-Jones method with Haar-like feature. Their experiment developed on (OpenCV) library, which is a library for computer vision application with open source. Aashni Haria et. al [8] designed a system for hand gesture recognition basing on detecting palm and fist gestures using Haar cascade classifier. The detection hand gestures translated into specific actions for lunching application in the computer. Their cascade classifier trained using a small number of positive and negative images. The detector trained using Open Computer Vision Library OpenCV. Abhishek B. et. al [9] developed a hand gesture recognition system that performs specific actions like switching pages, scrolling up or down in a page. They detect hand using Haar Cascade classifier. The hand gestures recognized using three Dimensional Convolutional Neural Network. The efficiency of their model is around 80% since the model requires time to get trained with the dataset.

In this study, a human hand used to control a smart home prototype system in real-time video using a built-in computer webcam. The designed prototype contained eight LEDs that served as home appliances. These LEDs controlled by means ON/OFF using different hand cases. The human hand cases differed by location of hand within the human head and shoulder level. The human hand is detected using the Viola-Jones method, where their method developed to generate XML-file of hand detection that contains hand features. A new algorithm introduced to determine the location of the detected hand within head and shoulder level using the XML-file of hand detection. There are two cases in the image, one and two hands case to obtain eight different hand case. Each case used to control one LED (ON/OFF) of the designed prototype.

2. Hand Detection Using Viola-Jones Method

Human hand detection has a wide range of application like as smart home control. Different methods and algorithms employed to determine hand features to detect hand in a still image or real-time video. The human hand can be recognized using a set of features by training an image dataset using the Viola-Jones robust algorithm for object detection [10]. The proposed algorithm based on training a detector to detect a human face in real-time video using two image dataset: positive (image with a human face) and negative (an image without a face). The algorithm based on machine learning combined of Haar-like features computation using the integral image and boosted classifier using the (cascaded) AdaBoost algorithm. Each Haar-like feature \( f(x) \) consists of connected “black” and “white” rectangles for each rectangle there is a set of image pixels. The value of \( f(x) \) is the difference between the sums of the pixel values \( I(x) \) in the black and white rectangles i.e. [7] as in equation (1). The computation time minimizes by converting each input image into an integral image. This is the first step in the Viola-Jones algorithm of the Haar feature of the image dataset. Image integral is fast and efficiently in computing the sum of values in the rectangle set of pixels.

\[
f(x) = \sum_{x=black} I(x) - \sum_{x=white} I(x)
\]  

(1)

Viola-Jones used in their system variant of AdaBoost (Adaptive Boosting which is a machine learning algorithm [11]) to select small number of features and also train classifier system to detect object (hand) in image. A cascade of boosted classifiers is composed by several
integrated (nested) stages, each one containing a boosted classifier. The cascade works as a single classifier that integrates the classifiers of every layer [5]. At first layer a classifier is trained to detect almost all objects of interest (hand) and triggers evaluation of classifier of second layer which adjusted to achieve detection at very high rate. A positive result from the second layer triggers a third layer, and so on [12].

3. Experimental work

In this work, a low-cost user-friendly prototype system designed as a smart home. The prototype designed using eight Light Emitting Diodes (LEDs) considered as home appliances. The LEDs are controlled in real-time using eight suggested hand cases. Each hand case used to control ON/OFF of single LED (same hand case turned LED ON and turned it OFF). A suggested algorithm used to detect and interpret input hand cases then the input case is translated into the command to control the LEDs. Therefore, the work in this study is consist of two parts. The first, hardware part which is the part of designing the smart home prototype. The second, software part where input hand cases are detected, interpreted and translated into the command to control ON/OFF of the LEDs in the suggested prototype.

3.1 Hardware System Architecture

Smart homes are intelligent buildings. The objective is to make human life more comfortable. This part of the study is to illustrate designing an electronic prototype as a smart home. The prototype system designed using the following tools showed in Fig (1). Eight LEDs (turns ON/OFF when it receives commands), Arduino Uno card (provide a voltage of (5 volts) to LEDs), jumper wire (used to connect LEDs into Arduino), and breadboard (used to connect LEDs and jumper wire on it). The LEDs in prototype considered as home devices (TV, refrigerator, lights, motors, etc.). The Arduino and the computer connected using USB-cable illustrated in Fig. (1). The connection between Arduino and computer allows MATLAB to access Arduino, resulting in LEDs controlling.

![Figure (1) The suggested smart home prototype system.](image-url)
The smart home prototype tools connected as follow: The eight LEDs settled on breadboard then LEDs connected to Arduino using jumper wires by connecting the negative end of each LED with ground GND of Arduino and connect the positive end of each LED with digital output pins of the Arduino. The introduced prototype connection described as a schematic sketch in Fig. (2).

![Schematic of the introduced prototype system.](image)

3.2 Software Part

In this section, eight suggested hand cases are detected in real time-video using a new suggest algorithm. The hand cases introduced using one and two human hands, and the hand cases different by differing hand location in the image plane. The suggest algorithm used to detect the difference in the position of the human hand in the input image for one or two hands case. Then, the data of the hand case is converted into the command to control ON/OFF LEDs in the introduced prototype. The hand in the input image is detected based on the Viola-Jones algorithm as the first step of the suggested algorithm.

- **Viola-Jones Algorithm to Detect Hand**

  The human hand is detected based on the Viola-Jones method in the complex background by training computer using Matlab system statement `trainCascadeObjectDetector`. To train the computer, two image datasets are provided: negative and positive dataset. Negative dataset (include images with any objects except human hand) consists of 12000 images downloaded from the Internet. Whereas, Positive dataset contains 2900 different images of palm hand posture. These images captured with a different light condition and a different distance from the camera. The dataset of positive image frames collected using a different size, color, tilt, and orientation of hands. Figure (3) shows an example of negative and positive images used in the training.
Before computer training using cascade training, the data of location and size of each 2900 positive image are required. In Matlab, Computer Vision System Toolbox includes Image Labeler application is used to provide location and size data for a positive image. By using this application the target (hand), in all 2900 positive image, is manually required to enclosed by bounding box. Each bounding box of each positive image has the format \([x, y, \text{width}, \text{height}]\) to specify an object size (width and height) and the location where \((x, y)\) is coordinate of the upper left corner of rectangle enclosed hand. The disadvantage of using Image Labeler is a time-consuming process because of the manual determination of the human hand shape in 2900 positive image. In this study, a new idea introduced to provide the same details of the Image Labeler application. For all positive dataset, the location and size determined automatically depending on skin detection method. Figure (4) shows skin detection of positive image and the enclosed rectangle of detected skin of the positive image.

For cascade training three data are required: the location and size from skin detection data, positive, and negative images dataset. These data are used to train computer basing on Viola-Jones method where the training was done using cascade of 24 stages of classifier. Each stage of cascade trained using AdaBoost training algorithm. Haar like feature used to classify images whether it contain hand or not at each stage of classifier. After training process classification model of hand is obtained as an XML-file named with specific name (in this study named as (hand.xml)). This file is saved in computer to use it later for hand detection in complex background regardless of hand gesture type and size and. Figure (5) shows block diagram of training process.
Hand Detection in Different Locations of Head and Shoulders Level

In this section, the human hand detected in different locations in the image using the suggested algorithm. The environment of study has dimensions of 7m length, 3.7m width, and 3m height. The environment lighted using two adjacent LED strip light placed at one side of the wall. Figure (6a) illustrates the scheme of the environment study, where the red circle indicate the hand cases location. The distance between hand case and webcam was 1m with appropriate lighting condition for detection of hand as illustrated in Fig. (6b). The environment lightening at this distance was 70 Lux which measured in a Luxmeter device. These illuminations obtained using two LED strip and portable LED bulb settled above webcam.

![Figure (6a) Environment scheme, (b) Experiment Setting.](image)

In this work, different eight hand cases are suggested by differing hand location within head and shoulders level that shown in the table (1). The hand location is detected using the newly introduced algorithm. The algorithm steps illustrated in the block diagram in Fig. (7). For real-time detection, an image \( I \) extracted then the image plane divided into nine regions named from R1 to R9, as shown in Fig (8). The image is divided into nine regions to determine hand location within the shoulder and head level. Moreover, determine hand location if it is at the right or left body side using the following relationships:

\[
\begin{align*}
  w &= \frac{r}{3} \\
  h &= \frac{c}{3}
\end{align*}
\]

where \( r \) and \( c \) represent length and width of \( I \), \( w \) and \( h \) are the dimension of each reigon.

The XML-file (hand.xml) is scanned over all divided regions in the image to detect hand in each reigon. The detected hand is enclosed by bounding box with the title at the upper left corner of the box. The hand location determined according to the extracted region. If the hand detected in region R2 or R8, then it is within the head level. Otherwise, regions R3 and R9 refer to location of hand within shoulder level. The hand location at right or left human body side determined by regions R2, R3, R8, and R9. The left hand recognized by region R2 and R3 and indicated as hand 2 (enclosed by the bound box of the the title “hand 2”) as in Fig. (8). While regions R8 and R9 are referred to the right hand and marked as hand 1 in Fig. (8). The regions R1, R4, R5, R6, and R7 are ignored in this work, because the locations of hand in these regions are not used. However, it is easy to study these regions in other cases to extend the project.

Each hand cases is described by a specific decimal code number \( C_N \), as illustrated in the table (1). For one raised hand (within head or shoulder level), the suggested \( C_N \) equal to 4, 8, 1, and 2 for the regions R2, R3, R8, and R9, respectively. Therefore, \( C_N=1 \) and \( 2 \) for hand in right bodyside (right hand) and \( C_N=4 \) and \( 8 \) for the left hand in the left body side. For two raised hands cases, (with head or shoulder level), the \( C_N \) became 5, 6, 9, and 10 according to hand case appearance obtained by adding the code numbers \( C_N \) of each single detected hand case. Namely, the hand case of output index \( id=5 \), the right and left hands are within the head level. The single-hand detected has code number \( C_N=4 \), and \( C_N=1 \), respectively. Thus, the code number for this case (of \( id=5 \)) is obtained by adding these two numbers \( C_N=4+1=5 \). The rest of the hand cases has the same procedure.
The eight hand cases used to switch ON/OFF of eight LEDs based on output $i_d$, as shown in table (1). Where the $i_d$ is used to arrange the hand cases ascendingly and used as a command sent to Arduino to control LEDs in the smart home prototype. For instance, in the table (1), LED number 7 (LED 7) switched ON/OFF using hand case of $i_d=7$. When the hand case of $i_d=7$ displayed infront of webcam the algorithm is detect and determined hand location then an output index $i_d=7$ is assigned. The assigned $i_d=7$ is sent to Arduino as the command to control LED 7. For LED 7 with status OFF, it turned ON when the same hand case of $i_d=7$ used again i.e. Arduino receive $i_d=7$. When LED 7 with statue ON, it turned OFF when the same hand case of $i_d=7$ used again i.e. Arduino receive $i_d=7$.

Table (1) input hand cases and the output.

| Input hand case | Code number ($C_N$) | Output index ($i_d$) | LED No. | Input hand case | Code number ($C_N$) | Output index ($i_d$) | LED No. |
|-----------------|---------------------|----------------------|---------|-----------------|---------------------|----------------------|---------|
| Right hand      | 1                   | 1                    | LED 1   | Right hand      | 5                   | 5                    | LED 5   |
| Left hand       |                     |                      |         |                 |                     |                      |         |
|                 | 2                   | 2                    | LED 2   |                 | 10                  | 6                    | LED 6   |
|                 | 4                   | 3                    | LED 3   |                 | 9                   | 7                    | LED 7   |
|                 | 8                   | 4                    | LED 4   |                 | 6                   | 8                    | LED 8   |
4. Results and Discussions

In this study, a smart home prototype system introduced where the human hands used to switch controlled (turned ON/OFF) a set of eight LEDs in real-time video. The right and left hand are employed to have eight cases used to control eight LEDs in the prototype system. The hand location detected in the image plane by training computer basing on the Viola-Jones algorithm. The new idea based on a skin detection method introduced to make computer training more precise and faster.

To control a certain LED, a specific hand case shown in front of a computer webcam. The hand case interpreted using the suggested algorithm steps (showed in Fig. (7)) where the hand/s detected based on the Viola-Jones algorithm. Then the hand location within the head or shoulder level is assigned, and a specific index \( id \) is determined. After that, an index of hand case is sent to Arduino as the command to control (ON/OFF) a LED. The delay time between showing a hand case in front of a webcam and LED responding is 0.43 second, whereas the algorithm interpreted and assign an index in time of 0.31 second. The delay time is determined based on the Matlab software version and the computer processor. The work of this study implemented using Matlab R2018a runs on a laptop Lenovo 520 with Windows 10, 8th Gen Intel® Core™ i7 processor and 16 GB of RAM.

The results of the algorithm responding (software part) to three hand cases of indices \( (id=1, id=5, \) and \( id=8) \) demonstrated in Fig. (9). Where the image plane is divided into nine regions to determine hand location. While the results of controlling three LEDs (LED1, LED5, and LED8) using three hand cases of three indices \( ids (id=1, id=5, \) and \( id=8) \) respectively shown in Fig. (10). The results of controlling the rest of the LEDs not shown because it is beyond paper limits and its control in the same procedure of the LEDs in Fig. (10).

![Figure (9) The introduced algorithm response to three hand cases.](image)

(a) Controlling of LED 1 using hand case of

\[ \text{hand case of } id=1 \]

\[ \text{hand case of } id=5 \]

\[ \text{hand case of } id=8 \]
From Fig. (10), it can be noticed that the left and right hand presented in the image of the input hand case flipped at the output images from the algorithm (as in Fig. (9)). These images are flipped by Matlab software to be convenient for the user to use the system.

5. Conclusions
The introduced system, low cost and user friendly prototype system to turn ON/OFF of LEDs, has many application in the remote control environments. The controlling of LEDs using eight suggested human hand cases implemented inaccurate performance. Where the eight studied cases detected efficiently at 100% accuracy using a new robust algorithm. The response time of detect hand case using the introduced algorithm could reduce lower than 0.31 second using different upgraded PC processor. Delay time between capturing hand case using the webcam and receiving commands in LEDs to control switch ON/OFF is about 0.43 sec. The lighting condition has a vital effect on the detection of hand cases. Therefore, special consideration has taken to avoid false hand detection. The system may be converted to a small device and manufactured as a compact device within smart TV, smart home, factory, or smart cars.
6. References

[1] A. Howedi, A. Jwaid, Design and implementation prototype of a smart house system at low cost and multi-functional, in: 2016 Future Technologies Conference (FTC), IEEE, 2016, pp. 876-884.

[2] J. See, S.-W. Lee, An integrated vision-based architecture for home security system, IEEE Transactions on Consumer Electronics, 53 (2007).

[3] D.-L. Dinh, T.-S. Kim, Smart home appliance control via hand gesture recognition using a depth camera, in: Smart Energy Control Systems for Sustainable Buildings, Springer, 2017, pp. 159-172.

[4] M.M. Jin, Real Time Hand detection and gesture recognition system, (2011).

[5] H. Francke, J. Ruiz-del-Solar, R. Verschae, Real-time hand gesture detection and recognition using boosted classifiers and active learning, in: Pacific-Rim Symposium on Image and Video Technology, Springer, 2007, pp. 533-547.

[6] V.-t. Nguyen, T.-l. Le, R.M. Thi-thanh-hai Tran, V. Courboulay, A method for hand detection based on Internal Haar-like features and Cascaded AdaBoost Classifier, in: The International Conference on Communications and Electronics (ICCE), 2012, pp. 608-613.

[7] R.M. Gurav, P.K. Kadbe, Vision based hand gesture recognition with haar classifier and AdaBoost algorithm, Int J Latest Trends Eng Technol (IJLTET), 5 (2015) 155-160.

[8] A. Haria, A. Subramanian, N. Asokkumar, S. Poddar, J.S. Nayak, Hand gesture recognition for human computer interaction, Procedia Computer Science, 115 (2017) 367-374.

[9] B. Abhishek, K. Krishi, M. Meghana, M. Daaniyaal, H. Anupama, Hand gesture recognition using machine learning algorithms, Computer Science and Information Technologies, 1 (2020) 116-120.

[10] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, IEEE, 2001, pp. I-I.

[11] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, Journal of computer and system sciences, 55 (1997) 119-139.

[12] P. Viola, M.J. Jones, D. Snow, Detecting pedestrians using patterns of motion and appearance, International Journal of Computer Vision, 63 (2005) 153-161.