International Journal of Technical Research and Applications e-ISSN: 2320-8163,
www.ijtra.com Volume 4, Issue 3 (May-June, 2016), PP. 89-92

CONTROLLING HOME APPLIANCES USING EYE BLINK WITH ADVANCE SECURITY

Parin Patel[1], Hitesh Patel[2], Jignesh Patoliya[3]
[1]M.Tech(Embedded System and VLSI Design), [2,3] Faculty of Electronics & Communication
C.S.P.I.T, CHARUSAT University Changa,Gujarat,India
[1]patelparin58@gmail.com
[2]hiteshpatel.ec@charusat.ac.in
[3]jigneshpatoliya@charusat.ac.in

Abstract— **A vision based human computer interface is presented in this research paper. The interface detects eye blinks and interprets them as control commands. The employed image processing methods include Haar-like features: for automatic eye detection and face detection. The main goal of this paper is to design a real time interactive system that can assist the person to control appliances viz, Light, Fan, T.V, A.C etc. Using user friendly GUI which also provides password security.**

Index terms- **Human computer interface, Face detection, Eye blink detection**

I. INTRODUCTION

Paralyzed person cannot move any body part except what is above the neck. The main Aim of this project is to design a real time interactive system that can assist the paralyzed to control appliances. System is intended to assist the paralyzed and physically challenged. System is wake up by when patient see in camera and do particular eye pattern (like... three time blinking eye), we can set it as a system start code. Once the system is started, menu is shown to the display by system, in which focus is rotating between every menu option. Patient can enter in particular menu by eye signal, when focus is on menu that he/she wants to select. Same way he/she can make selection in sub menu. All question or selection on screen is controlled by eye signal. This way patient can control home appliances (like, light, fan, TV etc. - ON/OFF or ring the bell to call someone) without need of anyone[2][3]

II. EYE-BLINK DETECTION SYSTEM

The eye-blink detection algorithm consists of two major Steps:

(1) Face detection, (2) Eye-Detection.

Face detection using Haar Feature-based Cascade Classifiers and same for eye detection using Haar Feature- based Cascade Classifiers [2]

A. Face Detection

The face detection algorithm proposed by Viola and Jones is used as the basis of our design. The face detection algorithm looks for specific Haar features of a human face. When one of these features is found, the algorithm allows the face candidate to pass to the next stage of detection. A face candidate is a rectangular section of the original image called a sub-window. Generally these sub-windows have a fixed size (typically 24×24 pixels). This sub-window is often scaled in order to obtain a variety of different size faces. The algorithm scans the entire image with this window and denotes each respective section a face candidate. The algorithm uses an integral image in order to process Haar features of a face candidate in constant time. It uses a cascade of stages which is used to eliminate non-face candidates quickly. Each stage consists of many different Haar features. Each feature is classified by a Haar feature classifier. The Haar feature classifiers generate an output which can then be provided to the stage comparator. The stage comparator sums the outputs of the Haar feature classifiers and compares this value with a stage threshold to determine if the stage should be passed. If all stages are passed the face candidate is concluded to be a face [2]

B. Eye Detection

Eye Detection use Haar-like features are computed by convolving the image with templates of different size and orientation. These feature prototypes can be grouped into 3 categories: edge, line and center-surround masks. Each template is composed of two or three black and white rectangles [1]

The feature value for the given mask is calculated as the weighted sum of the intensity of the pixels covered by black rectangle and the sum of pixel intensities covered by the whole mask. Not all calculated features are necessary to correctly detect Eye in the image. An effective classifier can be built using only a part of the features with smallest error rates. In order to find these features, boosting algorithm was used, namely Gentle Adaptive Boosting (Gentle AdaBoost). The boosting process is repeated several times to build a cascade of classifiers. Each stage of the cascade is a weighted sum of weak classifiers, where the complexity of the stages increase with the number of the stage [2]

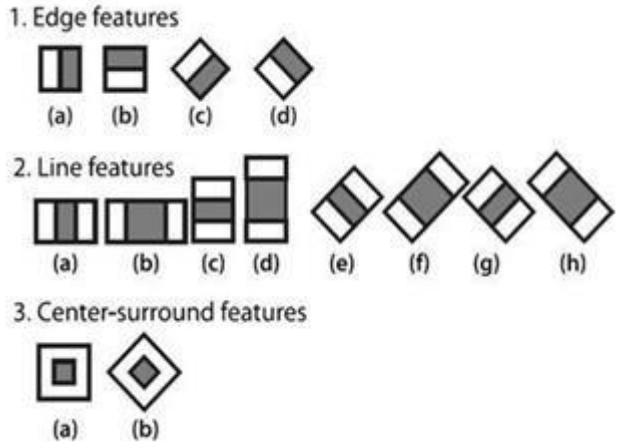

Fig 1. Detection

C. Eye Blink Detection

For recognizing Eye Blink, I have used flag mechanism. This Eye Blink Detection gives accurate results.

III. DEPENDANCY ON THE SYSTEM

The developed system for eye-blink detection and monitoring was tested using RapberryPi2 Development Board at 900 MHz processor on the sequences from the USB Logitech 2MP web cam [7]. Testing of the system took place in a room daylight from multiple window. The person was sitting in front of the monitor, on which the instructions for blinking were displayed. Each person was asked to blink 20 times (20 short blinks, alternately). The USB camera was fixed at the top of the monitor, about 65 cm away from the person's face. Two kinds of errors were identified: false detection (the system detected an eye-blink when it was not present) and missed blinks (a present eye-blink that was not detected by the system).

Eye Blink detection depends on illumination condition and camera feature.

I notice that poor illumination get false detection and range of detection also depend on camera's megapixel.

Results are presented in Tables 1 and 2. Note that the achieved results are comparable to the performance rates reported for active vision-based eye-blink detection systems

Light Condition	Interval between open eye and close eye(seconds)	Blinks	True Detection	False Detection
Full	1	20	19	1
Medium	1	20	18	2
Poor	1	20	2	18

Table 1 Eye-blink detection system (Lighting Condition effects)

Interval between open eye and close eye(seconds)	Sensitivity (Blink-detection)
Less seconds(short Eye Blink Duration)	increase
Maximum seconds(Long Eye Blink Duration)	Decrease

Table 2 Eye-blink detection system (sensitivity of Detection)

IV. EYE-BLINK INTERACTION

The algorithm for automatic detection of voluntary eye-blinks was employed in the development of a user interface. The applications were written using C++ Qt (Quasar Technologies) GUI Development tool. And OpenCV library. The system is built from off-the-shelf components: a camera and Raspberrypi2 Development Board. For best performance of the system, the distance between the camera and the user's head should be not greater than 100cm [1]

V. FUNCTIONALITY

A. The proposed interface has the following functionality

- Fully paralyzed person cannot move any body part except what is above the neck [3]. System is woken up by when patient see in camera and do particular eye pattern (like. Three time blinking eye), we can set it as a system start code as shown in figure 2 and figure 3.
- Once the system is started, menu is shown to the display by system, in which focus is rotating between every menu option. Patient can enter in particular menu by eye signal, when focus is on menu that he/she wants to select. Same way he/she can make selection in sub menu. All question or selection on screen is controlled by eye signal. Refer figure 3,figure 4,figure 5 and figure 6 for this.
- This way patient can control home appliances (like, light, fan, TV etc. - ON/OFF) without need of anyone.
- User can also changes the password using system setting menu in this Product as shown in figure 7
- Product has Active and Inactivate mode in this product.
- If Active mode enable, Appliances and system setting menu control using eye blink.
- If Inactive mode enable, Appliances and system setting Menu can't control using Eye Blink.

- Eye signal detection done by OpenCV and displaying menu or GUI done by Qt application [4] [8]. Appliances are handled by Raspberrypi2 GPIO through application [5] [6] [7].

B. Main Menu of Project contain following buttons

- Start: It includes home appliances and each appliances has on/off button.
- Close All: It is used to close all running Home appliances.
- System: It has some features like change password of system.
- Active: Appliances and system setting menu control using eye blink.
- Inactive: Appliances and system setting menu can't control using eye blink.

Fig 2. Security

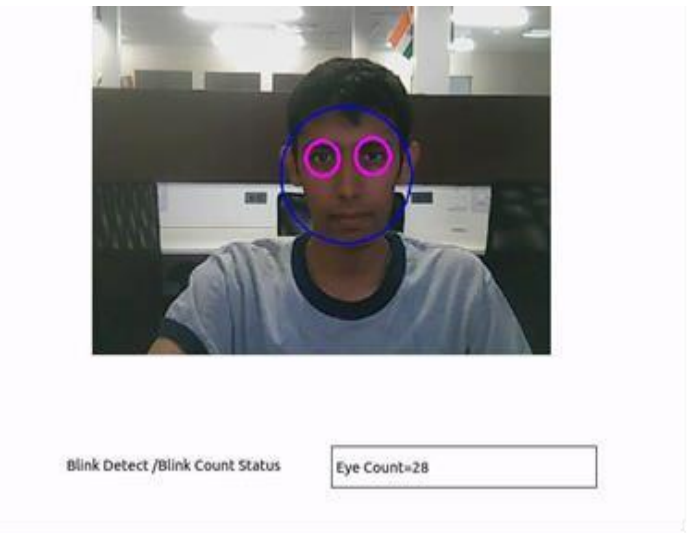

Fig 3. Face and Eye detect

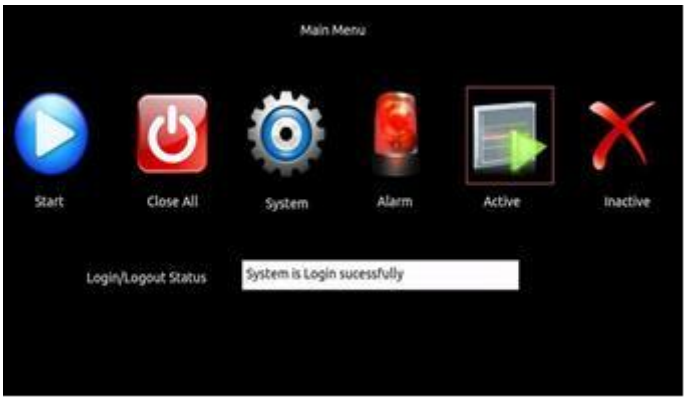

Fig 4. Main Menu

C. Start Button Includes Following Home Appliances

- Light
- FAN
- T.V
- A.C
- LAMP
- Refrigerator
- Back button is used for return to Main menu.

Each above Appliance includes on/off button and Back button, when it's clicked by user, on/off button and Back button is displayed.

- On/off button is used to start and stop Appliance.
- Back button is used for return to previous menu.

Fig 5. Home Appliance

Fig 6. Home Appliances

D. System Setting Button includes following buttons

- Change Password: for Change Password.
- Back: return to previous menu

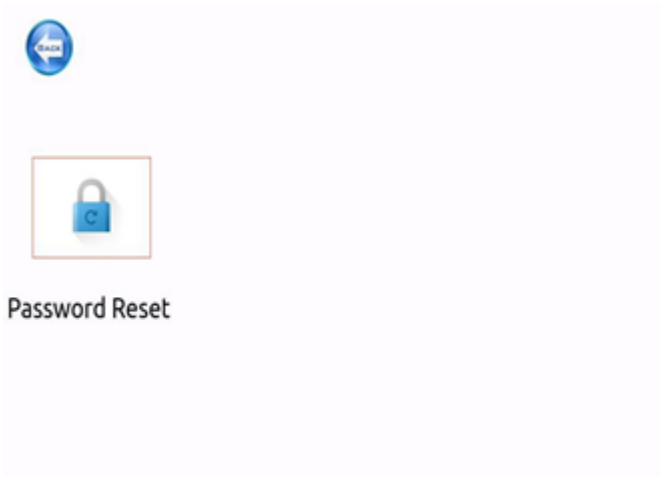

Fig 7. System Setting Menu

VI. CONCLUSION

This project is fully depend on lighting condition and also this application is very helpful for paralyzed person which are not able to control appliances. In Future, It can also be used for variety of other functions.

REFERENCS

[1] Aleksandra Królak, Paweł Strumiłło:Eye blink detection system for human–computer interaction. Universal Action in the Information Society,2011.
[2] Viola,Jones:RobustReal-timeObjectDetection,IJCV2001
[3] Atish Udayashankar, Amit R. Kowshik ,S. Chandramouli ,H. S. Prashanth Assistance for the Paralyzed Using Eye Blink Detection Dept. of Telecomm. Eng., PES Inst. of Technol., Bangalore, India.
[4] https://en.wikipedia.org/wiki/OpenCV
[5] http://docs.opencv.org/2.4.9/
[6] http://www.qt.io/support/
[7] https://www.raspberrypi.org/
[8] https://en.wikipedia.org/wiki/Qt%28sof