Eye Tracking Based Mobile Application

Rawan Ali Taban, Dr. Muayad Sadik Croock and Ayad Esho Korial

Abstract—**The development in employing the human face actions in the interaction with different types of devices is widely considered in the scientific researching fields. One of the most important parts of face actions is the eyes. This paper proposes a mobile application for human eye tracking using the front camera. The presented application can track the movement of eyes in order to select specific buttons. These buttons can be used for performing different activities. Moreover, eye blinking can be detected and used as an additional action. The motivation of the presented system over previous research works is the using of the front camera without the need for another supporting instrument. This means that the proposed system does not need any calibration with additional sensors, in which the accuracy and efficiency are guaranteed. The obtained results proved that the achieved system accuracy was 66% indoor and 78% outdoor.**

Index Terms— **eye tracking, blink detection, HAAR cascade classifier, android application.**

I. Introduction

Due to the advent of mobile devices and the continuous development of different technologies, it becomes necessary to expand and improve the human-computer interaction techniques to serve elderly and physically disabled persons. In addition, they facilitate their interaction with the novel technologies because, in some cases, it is difficult to use the traditional existing input devices such as keyboard, mouse or touch screen [1][2].

The eye is one of the important sensors of human body to be used as an input unit for many smart devices. Eye tracking represents a faster non-invasive method to estimate the user reactions; it provides seamless method to interact with modern graphical user interface [3].

Most of eye tracking techniques use expensive hardware such as, the Electrooculography (EOG) method which uses the differences in voltage between the cornea and retina of eye by using small electrodes that positioned around the eye of the user. In addition, the search coil method is another eye tracking technique which uses induced voltage with coil including in contact lenses attached to user’s eyes. These methods and others cause psychological burden to the user and can be uncomfortable as a consequence to the direct sensors or the cameras attached to user’s face, eye or head which are required in these methods [2].

Recent techniques employed by optical pupil tracking which are applicable to any device with sufficient processing power and a camera that is near or far to the user's eye or attached on the head [4]. Gaze tracking and blink detection fields achieve the attention of numerous researchers especially in the recent years and are still under development particularly in the field of Human-Computer Interaction (HCI).

In [5], the authors proposed an EOG based eye tracking system for HCI. The system was employed to control the TV, eye controlling game and eye sight level check by using the four detected directions of eye. The results showed that the system works precisely by more than 90% only in the first 30-90 min because of the sweating of skin attached to it.

The authors of [1], proposed an android application that process a video received from a head mounted camera to detect the eye movement directions in addition to blink detection, and then send the results via a network link to the clients to be used in controlling purpose. The results proved that the processing was slow and it could be improved by replacing the camera driver rather than the accuracy was about 90.88%. In spite of that, the weakness point in this system was the hardware complexity related to two devices one for processing and another for controlling.

In [6], authors used the Convolutional Neural Network (CNN) to train an eye tracking system from large scale dataset consisting of data from over 1450 people. The obtained results proved that the system was capable of robustly detecting eye gaze with low error rate achieved when tested on mobile phones and tablets. In [7], a gaze tracking based game for android platform was developed to help people with motoric impairment to interact with mobile device. The system included single eye blink feature detection. Low accuracy found by testing especially under low light intensity with some limitations such as the face position related to the camera must be constant and without any vertical or horizontal rotations.

In [8], the authors designed a system that replaced the mouse with gaze based controlling over windows operating system. The system used the smooth pursuit eye movement based interaction method with online recalibration algorithm to improve accuracy. The confirmed results proved that the user could successfully use the system for example in web browsing with some limitations.

In this paper, an eye gaze tracking and blink detection based mobile application is proposed. It is based on tracking the movement of user's pupils, sensed by the build-in front camera of an android mobile device in real time video. The proposed application does not need an additional sensor or an external device to be calibrated, which leads to reduce the hardware complexity at the same accuracy. Furthermore, the proposed algorithm and related subroutines have been performed using the underlying device without the need to delegating the processing to an external server.

II. THE PROPOSED SYSTEM

As aforementioned, an android mobile device is used as a hosting device in this application to be run in real time using Android Studio (2.3.3) IDE and OpenCV (2.4.9) library. For face and eye detection and tracking, viola jones algorithm [9] is used, which is a machine learning based algorithm. This algorithm is used to train a classifier from many positive and negative images, which is used to detect a specific object in these images. This is done by applying it on the image and scans it several times at different scales to detect different sizes of the object.

The adopted classifier in proposed application is Haar-like features based cascade because of the short time needed to compute extensive set of features and the efficient results of detection related to using this classifier [10]. Fig. 1 shows the proposed algorithm expressed as a flowchart. This algorithm performs different processing operations that are generally divided into four stages as follow:

A. Face detection stage:

It is the first stage of the proposed algorithm, which is based on Haar cascade classifier. The current frame enters to this stage for human face detection and a bounding rectangle insertion as shown in Fig. 2a.

B. ROI calculation stage:

In this stage the Region of Interest (ROI) which is the eyes-region is approximately calculated related to the dimensions of the detected face rectangle. The region of eyes is assumed to occupy 1/3 of the height of the face and ¾ of the width of the face. Thus, the ROI can be positioned by calculating the coordinates of the upper left corner of the eye region (x, y), the width and the height using (1), (2) and (3) respectively [11].

$$EyeArea_{(x,y)} = (FR.x + \frac{FR.w}{8}, FR.y + \frac{FR.h}{4.5}) \quad (1)$$

$$EyeArea_{width} = \frac{3}{4} FR.w \quad (2)$$

$$EyeArea_{height} = \frac{FR.h}{3} \quad (3)$$

The resulted region shown in Fig. 2b is splitted into two regions in form of rectangle to each region by following (4) to (6)[11].

$$RightEye_{(x,y)} = (FR.x + \frac{FR.w}{2}, FR.y + \frac{FR.h}{4.5}) \quad (4)$$

$$RightEye_{width} = \frac{7}{16} FR.w \quad (5)$$

$$LeftEye_{(x,y)} = (FR.x + \frac{FR.w}{16}, FR.y + \frac{FR.h}{4.5}) \quad (6)$$

The width of the left eye is computed using (5) while the heights of both eyes separately are computed using (3).

Where, FR is the face rectangle upper left corner point. Moreover, w and h are the width and the height of the rectangle respectively. The resulted image after splitting is shown in Fig. 2c.

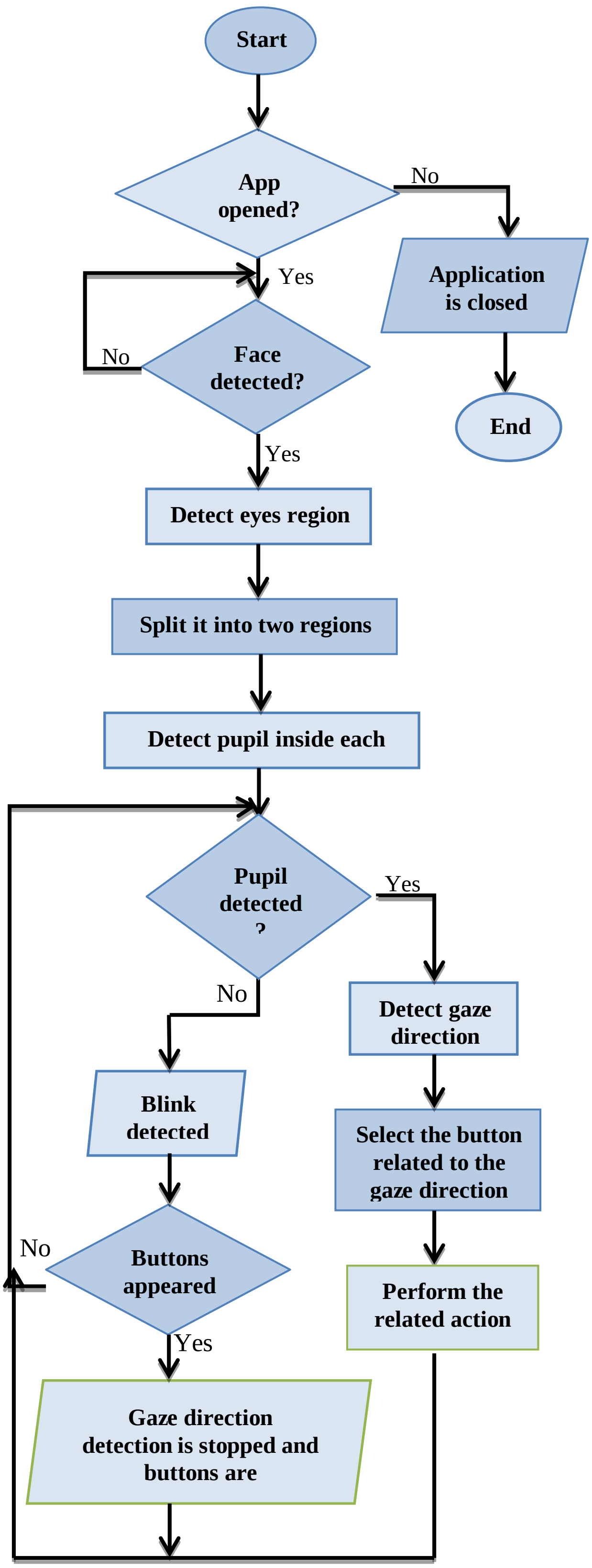

Fig. 1: Flowchart of the overall application processing stages.

C. Pupil detection stage:

The ROI has been reduced by detecting only the eye region without the eyebrow and other additional skin regions by applying another cascade classifier on the regions that obtained in the previous stage. After that, by searching for the darkest point inside the reduced region, the pupil's point is obtained (which is the darkest point in this region). Then, it is surrounded by a circle to distinguish and track it as shown in Fig. 2d.

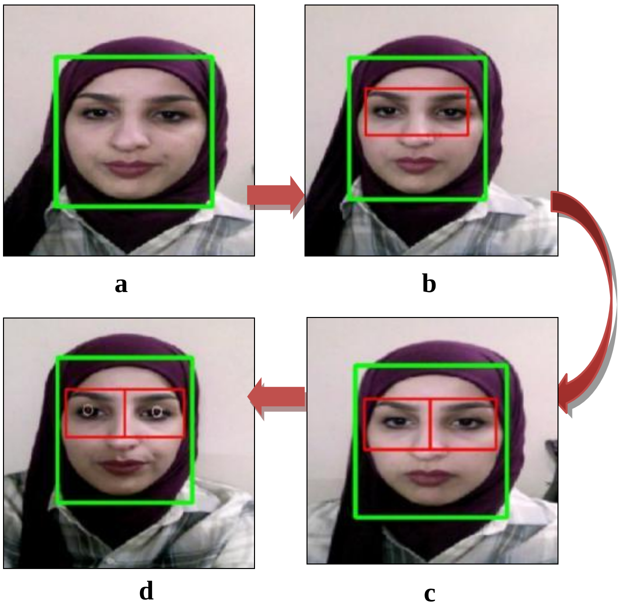

Fig. 2: The first three stages of processing: (a) is the first stage, (b-c) for the second stage and (d) is the third stage.

D. Gaze direction and blink detection stage:

It is the final stage in the application processing, in which the regions of both eyes obtained from the second stage are divided into five rows and five columns for each region as shown in Fig. 3. This is to use them as thresholds, which leads to perform the following two actions:

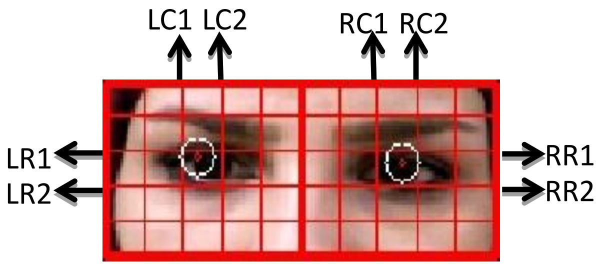

Fig. 3: The division of both eyes regions into 5x5 grid.

1) Gaze direction detection:

In this stage, the direction of eye gaze is detected. Ultimately, the application responds by a specific action which is button pressing on its interface depending on the detected direction. The proposed algorithm adopts a single eye which is the right one. By applying the taken thresholds, the intended direction is evaluated using gaze direction detection method that can be illustrated in Fig. 4 as a pseudo code.

```
/*detect the eye gaze direction depending on the right
eye*/
Let: RP = right eye pupil
     LP = left eye pupil
     RPC = right pupil circle's center
     GD = gaze direction
/*in addition to the rows and columns names mentioned
in figure (3) */
If RP&& LP are detected then
  If RPC >= RC2 then
```

```
GD = "right direction"
  elseif RPC <= RC1 then
  GD = "left direction"
elseif RPC <= RR1 then
  GD = "up direction"
elseif RPC >= RR2 then
  GD = "down direction"
endif
endif
endif
endif
  endif
```

Fig. 4: pseudo code of gaze direction detection method.

2) Blink detection:

This action is detected under the condition of the non-existence of the pupil of both eyes. The application can consider this action under some conditions which are:

- The user's gaze must be at the forward position first before blinking (explained in the next section), to avoid the overlap with the gaze directions action.
- The user's eye must be continually closed for 2sec.
- The blink must be hardly enough to guarantee the eyelashes disappeared. This is to avoid overlapping with the condition of detecting the pupil depending on the darkest point which is the eyelashes in the case of closed eye.

III. The Detected Actions and the GUI Design:

The user interface panel of the proposed application consists of four buttons that represent the four eye gaze directions (right, left, up and down) as shown in Fig. 5. The user can select a specific button by gazing to the button's position direction. Gazing again to the same selected button is employed to remove the selection of that button (see Figures 6-9). If the user looks forward and then blinks hardly for two seconds as shown in Fig. 10, the buttons are disappeared and the gaze direction detection is stopped. The same operation can be used to return the buttons shown in Fig. 5.

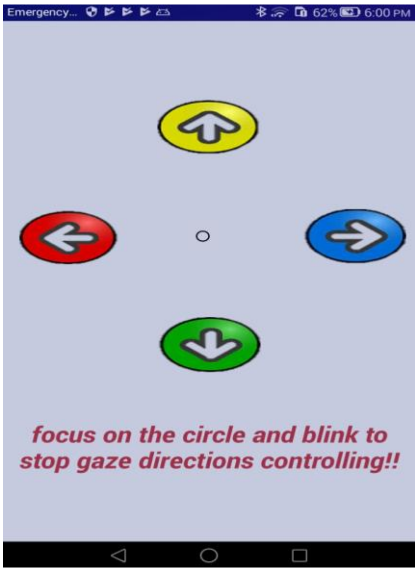

Fig. 5: The application user interface (without any action or response).

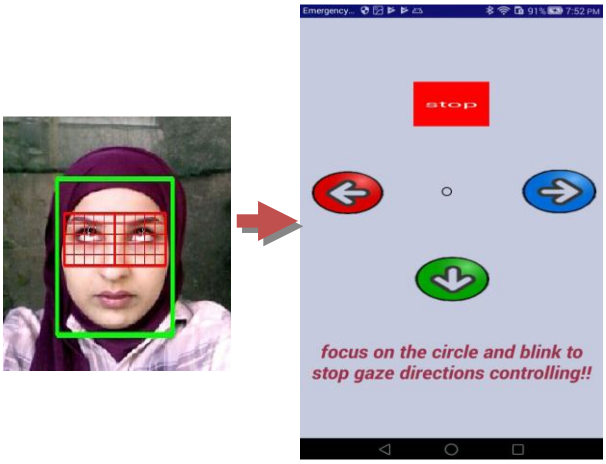

Fig. 8: Up gaze direction (action and response).

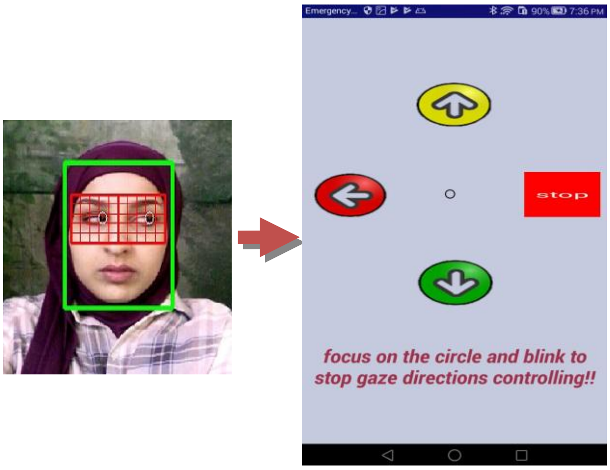

Fig. 6: Right gaze direction (action and response).

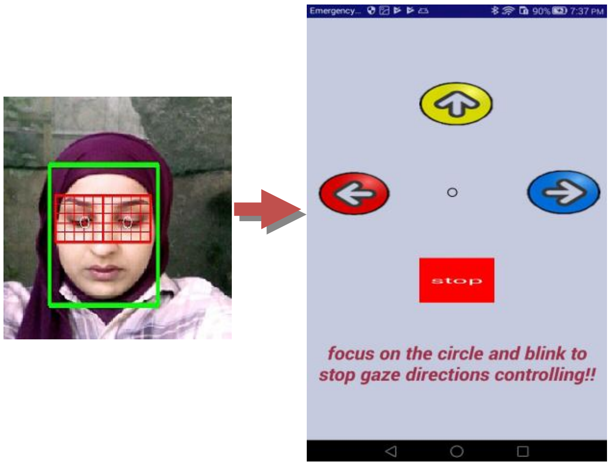

Fig. 9: Down gaze direction (action and response).

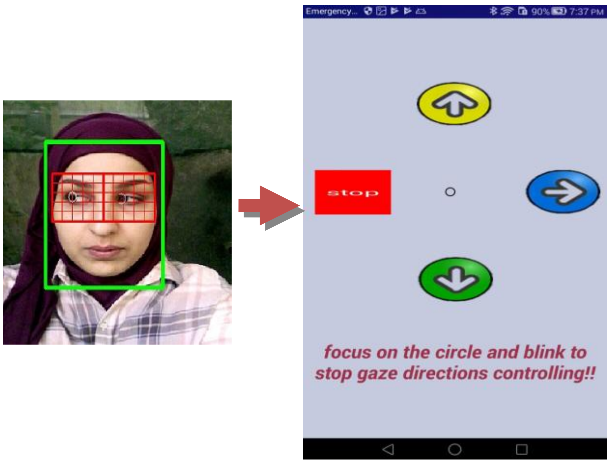

Fig. 7: Left gaze direction (action and response).

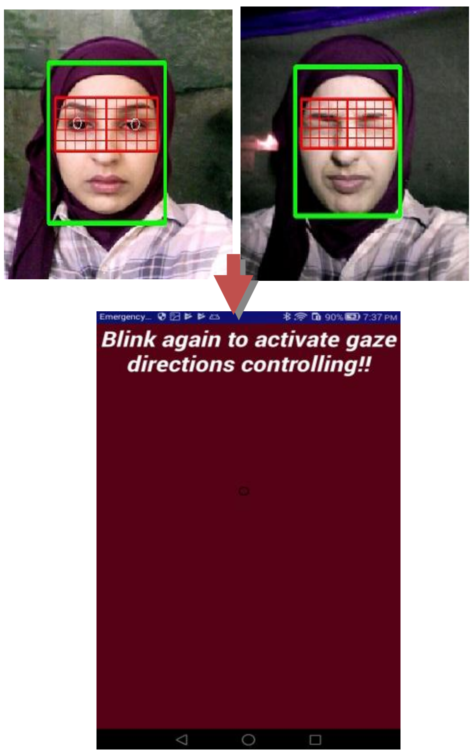

Fig. 10: the blink action (action and response).

IV. Testing and Evaluation:

The proposed system was tested and evaluated on Huawei T1-701u tablet with Quad core 1.2 GHz, 1.0GB RAM, 2MP camera and 600 x1024 resolution which running Android 4.4.2 version. This non-powerful device was chosen to prove the ability of track the eye even under low device specification with satisfied results related to these non-powerful specifications.

From the experiments, it is found that the performance of proposed application depends on the processing power of the device and the resolution of its camera in addition to the intensity of the used light in the surrounding area, such as indoor or outdoor usage. Table I shows the efficiency ratio of performing the underlying actions over different environment circumstances with limited distance between the user and the device (50cm indoor and 54cm outdoor).

The intensity of light must not be very high because this can make the size of pupil small causing difficult pupil detection[12]. Furthermore, the increased intensity of light can make the pupil with more brightness which causes incorrectly detecting. This is because the pupil does not remain dark as needed. In addition, the low light intensity causes low detection accuracy.

Table I: The obtained results.

Action	**State**	**Accuracy**	
		Indoor	**Outdoor**
Right gaze direction	The right button is selected	90%	70%
Left gaze direction	The left button is selected	90%	60%
Up gaze direction	The up button is selected	70%	50%
Down gaze direction	The down button is selected	70%	60%
Blinking	Stop gaze directions detection	70%	90%
The whole system accuracy:		78%	66%

The low detection accuracy ratio of the up and down directions in comparison with the right and left directions is appeared because of the horizontal line of the eye is wider than the vertical line. As a consequence, the horizontally movement range of the eye from left to right is faster to detect than the vertical movement range.

V. Conclusion

This paper presented a proposed eye tracking mobile application working under the operating system of android. The proposed application provided the ability of selecting different buttons on the designed user interface panel. These buttons represented the eye direction: up, down, right and left. In addition, the blinking eye action was considered to activate and deactivate the panel. The proposed algorithm consisted of four processing stages to perform the classification using HAAR-cascades method, face detection, eye region detection, pupil detection, and gaze detection. The introduced application was working efficiently even with low technical conditions of the smart devices. This was to prove the claiming of low cost application. Moreover, the proposed application adopted the built-in front camera of the smart device without the need of additional sensors and instruments. The obtained results proved that the system is fast, efficient and precise especially in distinguishing blink from gaze directions in contrary to similar systems. The accuracy of the overall system was 66% indoor and 76% outdoor which is efficient rate related to the used device's specifications. This rate can be improved by using another powerful device with high camera resolution.

References

[1] D. Garcia and I. Sintos, "EyeDROID: Android eye tracking system", University of Copenhagen, 2014.
[2] K. Arai and R. Mardiyanto, "Eyes based eletric wheel chair control system", International Journal of Advanced Computer Science and Applications, vol. 2, no. 12, pp. 92-105, 2011.
[3] I. Karkanis, "Exploring Eye Tracking Techniques On Smartphones", Appsala University, July 2015.
[4] H. R. Chennammaand X. Yuan, "A survey on eye-gaze tracking techniques", Indian Journal of Computer Science and Engineering, vol. 4, no. 5, pp. 388-393, Nov. 2013.
[5] L. Y. Deng, C. Hsu, T. Lin, J. Tuan and S. Chang, "EOG-based Human–Computer Interface system development", Expert Systems with Applications, vol. 37, no. 4, pp. 3337-3343, 2010.
[6] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik and A. Torralba, "Eye tracking for everyone", arXiv preprint arXiv:1606.05814, June 2016.
[7] H. Yaya and M. James, " Mata: An Android Eye-Tracking Based User Interface Control Application" Journal of Game, Game Art, and Gamification, vol. 1, no. 1, pp. 35-40, 2017.
[8] S. Schenk, M. Dreiser, G. Rigoll and M. Dorr, "GazeEverywhere: Enabling Gaze-only User Interaction on an Unmodified Desktop PC in Everyday Scenarios", Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, ACM, pp. 3034-3044, 2017.
[9] V. Paul and J. Michael, "Rapid object detection using a boosted cascade of simple features", in Computer Vision and Pattern Recognition, IEEE Computer Society Conference on, vol. 1, pp. 511-518, 2001.
[10] A. Kasinski and A. Schmidt, "The architecture and performance of the face and eyes detection system based on the Haar cascade classifiers", Pattern Analysis and Applications, vol. 13, no. 2, pp. 197-211, 2010.
[11] B. Javidi, Image recognition and classification: algorithms, systems, and applications. CRC press, 2002.
[12] Resolution of an eye, 28 Feb. 2018 at 11:30 PM, https://www.wikilectures.eu/w/Resolution_of_an_eye