

HCI based on eye movements for unlocking mobile devices

Isaías Arrieta-Arellano, Francisco López-Orozco, J. Israel Hernández-Hernández and Juan-Gabriel Ruiz-Ruiz

Published: 30 November 2021

Abstract

This paper addresses the implementation of an eye detection mobile application for unlocking mobile devices using its built camera as an eye tracker. The implementation allows a real-time face and eye detection of a mobile phone user. The idea behind our development was the creation of a preventive model for the benefit and safety of mobile users, including disabled users (e.g., users with arthritis). The product is a visual application that detects the pupils and can simulate the mouse cursor on the screen to unlock the device. Although some screen lock methods have been already developed, our project aims to complement the security in mobile devices. Moreover, with a new mechanism to unlock mobile devices it would offer a more robust alternative in terms of safety of such devices. Evaluation results suggest our implementation is a promising alternative for security in these devices.

Keywords:

Human-centered computing; Eye tracking; Eye movements; Mobile devices; Gaze detection.

1 Introduction

Nowadays, we can find vast functionality in mobile devices, like simple phone calls, text messages, calculators, chat, social networks, etc. In the last decade, they were incorporated in daily work and today it has become an important element for our use [7]. The mobile device, also called a smartphone, changed the phone into a type of small computer, having in its system, file's protection models by the time the user is storing information in it. However, storing personal data on the mobile device could be at risk of being stolen if the protection measures are not well implemented. Personal and external data are exposed daily by the mobile device. However, we need no count with the security to keep safe such data, that's something users should always take in consideration [10]. Unfortunately, users must face more and more vulnerabilities

regarding the security of their data, so it is important to have alternatives that allow them to maintain their security.

Security is increasing with the flow of time; it becomes necessary to improve the lock method's rigor to keep safe the information against possible data attacks. Screen blockers on mobile devices are part of this lock's methods. Some measures that work with security are using the gaze to generate visual patterns as a protection model to contribute to new security systems for mobile devices. Biometric studies have generated a big impact on security. This unique form detects human features implemented on devices. For example, a fingerprint detection as a locking model, however, studies about gaze detection had created a new model that could establish the gaze as a new locking model.

This work proposes an implementation of an application to unlock a mobile device with the gaze.

2 Previous work

Several research projects have proposed these kinds of authentication schemes. Eye movement biometrics for human identification were initially investigated by Kasprowski and Ober [14]. They performed a personalized identification based on characteristic eye movement, specifically using eye tracking equipment. Their experiment tested the possibility of identifying people with that unique biological aspect. Bednarik [16] presented a case study investigating the potential of eye movement information for biometric purposes. The results indicated that the eye movement and the gaze characteristic contain the discriminatory information between the individuals. Later, Komogorsteve suggested [3,11,12] a human identification by eye movement, including the types of stimulating effects of the eye and movement patterns in the precision in biometric verification. The results verified that certain removable fixation features can specify individual distinctions. For example, authors in [6] show a new authentication perspective based on gaze movement using the front camera. Here, users are not required to do a complex task, just follow a target as a password. They describe an assigned pattern to follow up with gaze. Regarding a biometric analysis, the study carried out by Song et al. [5] explains the impact of biometrics before the creation of a visual identifier for mobile devices. Another example is presented by Christina, et. al. [4]. They discuss the use of the gaze as a support for authentication and private protection. In addition, tracking the user's gaze behavior can help to understand their actions for the detection of insecure behaviors. Several attempts to build mechanisms to support or improve security have been proposed by observing and analyzing the behavior of the user's gaze when performing security tasks.

Arrieta-Arellano, Isaías., López-Orozco, Francisco., Hernández-Hernández, Jesús-Israel.
DMCU, Universidad Autónoma de Ciudad Juárez
Ciudad Juárez, México.
al154022@alumnos.uacj.mx, francisco.orozco@uacj.mx,
israel.hernandez@uacj.mx

Ruiz-Ruiz, Juan-Gabriel.
Universidad de la Sierra Juárez (UNSIJ)
Ixtlán de Juárez, Oaxaca.
jugaruiz@unsij.edu.mx

Summarizing, in [6], the authors describe an authentication process based on eye movement, following a specific objective; authors in [5], present a possibility of identifying users by means of eye movement, like the biometric verification process that is handled with fingerprint detectors that are seen in mobile devices. Finally, in [4] the relationship between private protection and the analysis of the behavior in the user's gaze is observed to detect security behaviors or provide authentication support.

Our work proposes to cover most of these analyzes and to take advantage of visual movement as an ocular signature to unlock/lock mobile devices with its built camera as in [2,9].

3 Methodology

For the current development and its simplicity, the cascade methodology was followed with next stages: requirements, design, implementation, testing and maintenance.

Requirements' phase helped to determine the software features for its development without technical details. Then the internal structure of the software and the relationships between the entities were described. The software **design** document describes the general structure of the system and its specifications [13]. In the **implementation** phase, the requirements were coded in a programming language making use of the designed structure. During the **testing** phase, a verification that all the system components were working correctly and that all the requirements requested at the end of the previous phase are being met. Finally, in the **maintenance** phase, it was sought to verify that it is working correctly in the environment where it will be used [13].

The determination of requirements started from the following points: it must show the eye detection through the front camera of the mobile device, an eye calibration method should be included, a limited space will be shown where the signature will be made, the signature pattern must be shown within the space allowed and the device must notify if the ocular signature for unlocking was correct or incorrect.

Figure 1 shows an example of the eye signature. The scanpath is represented by a blue line.



Figure 1. Example of the eye signature with a locking/unlocking pattern in blue.

Figure 2 shows the locking/unlocking process. First, an initial idle state is shown on the device when the mobile device is not performing a task. When an interaction is initiated, the ocular signature is requested to be unlocked. The screen lock will remain in sleep until the user allows the time determined by the mobile device to go into idle state to reach its limit or the user uses the off button to put it into idle state. When interacting with the phone again, the screen will be locked again.

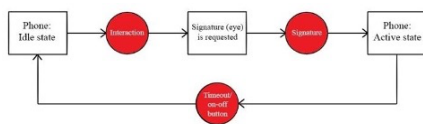


Figure 2. Diagram of locking-unlocking process of mobile device.

As part of the functionality of the application, there is a button that allows the user to run a visual calibration, this is done to manage an acceptable precision of the signature. In addition, the touch interaction with the mobile device screen remains active. This is done in case a situation arises such as forgetting the ocular signature, a bad calibration that does not allow ocular detection or possible obstructions that the camera has or cannot focus well or caused by external factors such as lack of light. Users wearing glasses are treated as special cases during the testing phase. This is necessary to know if the reflection of the glass does not generate recognition problems for the algorithm. Since eye movement varies from person to person, it is expected that the signature does not ask for a full accuracy, therefore calibration is asked for the signature.

The first element consisted in the development of the software document. It was proposed mainly to be able to formalize all the elements that were previously seen in the requirements phase and to be able to take a better control of development progress. These elements consist of the purpose, scope, glossary, risk analysis, and user characteristics, while the other part of the document presents the use case diagrams, as well as the functional and non-functional requirements, being those of greater weight, since they were those that were analyzed and created, through the previous phase, making clear, the functionality of the project worked.

Figure 3 is a representation of the recognition algorithm. This use case diagram shows how the recognition process was carried out where the screen lock system interacted with the algorithm, while the mobile device, being the secondary actor, receives the access granted through the recognition algorithm.

During this project a mobile phone Huawei Mate 20 lite was used with the following features: OS: Android 8.1 (Oreo), updated to Android 10, EMUI 10.0; CPU: Octa-core (4x2.2 GHz Cortex-A73 y 4x1.7 GHz Cortex-A53); front camera: 24 MP, f/2.0, 26mm (amplitude), 1/2.8", 0.9µm, 2 MP, depth sensor; screen size: 6.3 in; resolution: 1080 x 2340 px, 19.5:9 ratio; GPU: Mali-G51 MP4 ;RAM: 4GB.

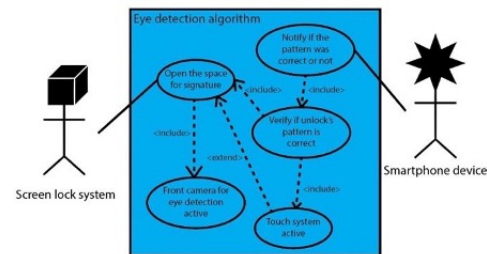


Figure 3. Diagram of locking-unlocking process of mobile device.

4 Implementation

The Open-Source Computer Vision Library (OpenCV, version 2.4.6) was used. It provides a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products [8]. This tool was used because it is composed of a set of libraries with special algorithms for face detection and recognition, object identification, movement tracking cameras, tracking of targets in movement as it was required here: the tracking of the movement of the users' eyes. The specific OpenCV library [8] used was Haar feature-based cascade classifiers. The object detection using this library is an effective method proposed by Paul Viola and Michael Jones in [15]. It is an approach based on machine learning where the cascade function is

trained from a series of positive and negative images. Then can be used to detect objects in other images.

To better understand and adapt the functionality of the Haar cascade classifier to our work, Controlling Android Device with Eye Tracking [18] was a reference. It suggests that the classifier must be related to the training output of hundreds of images, both positive and negative [17]. For example, to detect an image of a car, on the positive side, thousands of images are taken of different car models as well as their markings, after that, the car locations are manually tagged in each image. On the downside, most of the same set of positive images are taken or much more, and those images should not contain cars. Then, the classifier can be trained using a specific algorithm.

After the classifier has been trained, it can be applied to an image input. Classifier will give a boolean result where it will say if there is a car or not. Since the image input could be much longer than the training images, the search window will hover over the image input to detect possible multiple occurrences.

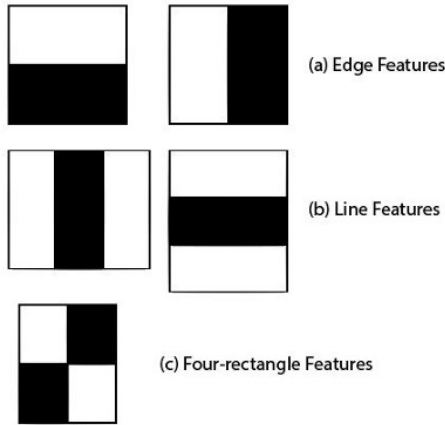


Figure 4. Scheme of *haar* features [18].

The classifier is designed in such a way that you can easily adjust the size of the images to be able to find the objects of interest posed in their different sizes. To use this method applied in face detection and eye tracking, classifier training requires a large series of positive and negative images of both faces as well as eyes. Afterwards, it is required to label the characteristics of each image. Each characteristic represents an individual value that is obtained by subtracting the sum of the pixels under a white rectangle from the sum of the pixels under a black rectangle. Figure 4 shows the representative scheme of the subtraction. Then, it is possible to have the interpretations of sizes and locations of each image in use to calculate characteristics. For each calculation, it is required to find the sum of pixels under the black and white areas. Some authors have presented complete images, thus simplifying the calculation of pixels, improving the speed of the calculation. Unfortunately, most of these computations are irrelevant. This situation is explained in Figure 5.

The first feature selected shows that the eyes are usually darker than the nose and cheeks. In the second case, the characteristics focus on the eyes, the area being darker than the bridge of the nose. The problem here is that the application window that is grabbing other zones will become relevant in the classifier. Adaboost was used as a meta-algorithm. It is applied in each characteristic to find the best threshold with which the classifier will be able to classify the faces in positive and negative.

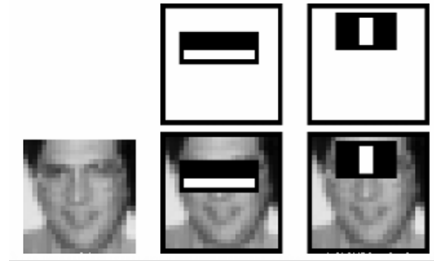


Figure 5. Scheme of *haar* features [18].

In practice, since most of the image will be a faceless region, the algorithm will first check for a face in the window; if it does not find a face, this window is discarded in the first phase and the algorithm will not process it again; this way, you can have more time to review regions of the face. This is how the cascade classifier is generated. Instead of applying all 6,000 features in one window, it groups the features into different levels and applies them one by one, so that if a window fails at the first level, it is discarded. If you pass the first level, the second level is applied and the process continues to the point where if a window passes through all the levels, there is a face region. For the structure of the algorithm, the English Real-time gaze Tracking Algorithm with head movement Compensation (RTAC) was used. This algorithm detects the pupil even with head movements. It is based on the English Pupil Center Corneal Reflection (PCCR) [19]. To calculate the gaze, the system takes the vector from the pupil flare -reflection of infrared light in the eye- and calculated by a mapping function; . Therefore, the flare of the pupil becomes the 2D vector between the flare and the pupil. Then the algorithm uses the mapping function to locate the vector in the direction of the 3D gaze; the mapping function is as follows:

$$\theta_h = b_{\theta_h}V_x + a_{\theta_h} \text{ and } \theta_v = b_{\theta_v}V_y + a_{\theta_v}$$

Here, θ_h is the angle between the gaze and the horizontal direction and θ_v is the angle between the gaze and the vertical direction. Coefficients a and b are estimated using the pupil flare vectors, the changes of position of the head in the horizontal and vertical planes [19]:

$$\delta I_x = (b_{2x}\delta L_h + a_{2x})\delta d^2 + (b_{1x}\delta L_h + a_x)\delta d + (b_{0x}\delta L_h + a_{0x})$$

and

$$\delta I_y = (b_{2y}\delta L_h + a_{2y})\delta d^2 + (b_{1y}\delta L_h + a_y)\delta d + (b_{0y}\delta L_h + a_{0y})$$

Next phase consisted of the implementation on the Android Studio platform (OpenCV-2.4.6-android-sd [8]).

5 Results and Discussion

An experiment was performed to measure the effectiveness of the algorithm. A total number of 20 university students (10 men and 10 women) were recruited by sending an invitation by a social network. Half of them were asked to try to cause some interference to the algorithm performance by wearing glasses. The idea was to "trick" the system. However, to test the robustness of our system, extra test conditions would be necessary (eg. test with a video of the cellphone user played in front of the cellphone camera). Another aspect considered was the lighting factor. If the user was not in an environment with enough lighting, the algorithm does not recognize the environment and therefore the eye cannot be

detected. Similarly, excess lighting could also cause a lack of facial recognition. The experiment consisted of following a linear pattern guided by the eye movement.

The test was carried out in spaces where natural as well as artificial lighting was present. Type of lighting is a crucial factor that affects the algorithm. Table 1 resumes the experimental conditions that were tested.

Table 1. Experimental conditions during the testing phase.

<i>Gen.</i>	<i>Glass</i>	<i>Nat.</i>	<i>Art.</i>	<i>NL</i>	<i>AL</i>	<i>Noise</i>
M/F	Y/N	Y/N	Y/N	Y/N	Y/N	Y/N
Tester [1...20]

First column refers to participant gender (*Gen.*). Columns from 2 to 7 are fields of binary type. *Glass* indicates if the participant was wearing glasses or not. *Natural (Nat.)* and *Artificial (Art.)* refers to the type of lighting. Next two columns refer to if that signature was successfully accepted under *NL* and *AL* conditions. Finally, the last column indicates if the participant was asked to try to produce interference (*Noise*) during the test.

Firstly, for the eye detection in *NL* condition, different situations were found. *NL* is a variable that cannot be controlled, thereby causing a discrepancy in some cases. In the case of female users, there was a success rate of 100% who do not wear glasses, while the users who wear glasses, there was a success rate of 40% and a failed of 60%. In the case of male not wearing glasses, a success rate of 100% was obtained, while the users with glasses, a success rate of 60% and a failed of 40%.

In the case of *AL* condition, users who do not wear glasses were not affected by light, however, in the case of users wearing glasses, the reflection caused by the light directly affected the front camera of the mobile device blocking the good performance of the algorithm and with it, not being able to perform an ocular detection. A repetitive pattern was observed in both schemes. Since *AL* is a variable that can be controlled, we made an adjustment to the environment for the benefit of the algorithm at the time of eye detection. For the female as well as male users who did not wear glasses, a success rate of 100% was obtained, however, the users who wear glasses, they registered a total failure of 100% in eye detection. Table [2] summarizes the results.

Table 2. Eye detection during the testing phase.

<i>Gender</i>	<i>Glasses</i>	<i>Lighting</i>	<i>Success</i>	<i>Failed</i>	<i>Success (%)</i>
Female	No	Natural	5	0	100
Female	Yes	Natural	2	3	40
Male	No	Natural	5	0	100
Male	Yes	Natural	3	2	60
Female	No	Artificial	5	0	100
Female	Yes	Artificial	0	5	0
Male	No	Artificial	5	0	100
Male	Yes	Artificial	0	5	0

A successful ocular signature -after the ocular detection- implies that the algorithm allows the ocular signature. Evaluation pointed out that all the users who did not wear glasses were able to perform the ocular signature, while the users with glasses had initial complications. However, in some cases the signature was successful. In the case of female users, a success rate of 40% and a failed of 60%. For male users with glasses, a success rate of 60% while a failed rate of 40%. Although the *NL* factor, it is an element that cannot be controlled, did not generate difficulties during the test for users who do not wear glasses, while users with glasses generated certain difficulties. When trying to perform eye movements, there were situations in which contact with the algorithm was lost and eye movement failed, while in other cases the movement was maintained stable. For the case of signature with *AL*, all users who did not wear glasses were able to perform the eye signature. However, all users with glasses failed to perform the eye signature. We point out that artificial light is a crucial factor with a direct effect at the time of performing the ocular signature. In the case of users who do not wear glasses, it was previously commented that as the lighting is more controllable, there was no difficulty when carrying out the test. Regarding users with glasses, our algorithm could not perform any ocular detection. From this step there was no recognition at all. Finally, Table [3] summarizes these results.

Table 3. Ocular signature during the testing phase.

<i>Gender</i>	<i>Glasses</i>	<i>Lighting</i>	<i>Success</i>	<i>Failed</i>	<i>Success (%)</i>
Female	No	Natural	5	0	100
Female	Yes	Natural	2	3	40
Male	No	Natural	5	0	100
Male	Yes	Natural	3	2	60
Female	No	Artificial	5	0	100
Female	Yes	Artificial	3	2	60
Male	No	Artificial	5	0	100
Male	Yes	Artificial	2	3	40

It is important to point out that in some situations the behavior of the algorithm was compromised to failure during the testing phase. For example, in certain cases, some false negatives were presented. This refers to the moments in which the algorithm did not allow access to users, due to lighting situations or by using glasses. Additionally, in several attempts, the algorithm lost the ocular detection by pausing the movement of the cursor and on other occasions the movement became unstable. We realize, that due the COVID-19 pandemic, an extra condition would be interesting to test when a user is wearing a mask. Here, we trust that our recognition system would still identify the gaze while eyes are uncovered. A usability testing questionnaire was administered to the testers.

Testing results show that the implementation of eye recognition in a mobile device is possible. The detection of the pupils was achieved thanks to the *Haar* Cascade model of OpenCV [8]. By using the appropriate libraries, it was possible to perform an ocular detection through the front camera of the mobile device. It is important to point out that although the designed algorithm

appears not adequate for those users who wear glasses, it is possible to improve the quality of our algorithm. One interesting possibility is to reinforce it by considering convolutional neural networks as is presented [1]. Moreover, the algorithm might reach the internal system of the mobile device and be used as a new security model and there would be the possibility to add it as an additional method in the security section of mobile devices.

6 Acknowledgments

To all university participants who took part of the testing phase.

7 References

- [1] Andronicus A. Akinyelu and Pieter Blignaut. Convolutional Neural Network-Based Methods for Eye Gaze Estimation: A Survey, 142581–142605, 2020. <https://doi.org/10.1109/ACCESS.2020.3013540>.
- [2] A. Wojciechowski and K. Fornalczyk. Single web camera robust interactive eye-gaze tracking method. *Bulletin of the Polish Academy of Sciences Technical Sciences* 63, 879–886, 2015. <https://doi.org/10.1515/bpasts-2015-01009>.
- [3] C. Holland and O. Komogortsev. “Complex Eye Movement Pattern Biometrics: The Effects of Environment and Stimulus.” *IEEE Transactions on Information Forensics and Security*, vol. 8, 2013.
- [4] C. Katsini, Y. Abdrabou, G. Raptis, M. Khamis, and F. Alt. *The Role of Eye Gaze in Security and Privacy Applications: Survey and Future HCI Research Directions*, 2020.
- [5] C. Song, A. Wang, K. Ren, and W. Xu. *EyeVeri: A secure and Usable Approach for Smartphone User Authentication*, 2016.
- [6] D. Liu, Dong B., X. Gao, and W. Haining. *Exploiting eye tracking for smartphone authentication*, 2015.
- [7] *Edita fácil. Uso de los dispositivos móviles en el mundo*, 2015.
- [8] G. Bradski. *The OpenCV Library*. Dr. Dobb’s Journal of Software Tools, 2000.
- [9] Ibrahim Furkan Ince and Jin Woo Kim. A 2D eye gaze estimation system with low-resolution webcam images. *EURASIP Journal on Advances in Signal Processing*, 1-40, 2011. <https://doi.org/10.1186/1687-6180-2011-40>
- [10] N. Escobar. (2015, Abril 28). ¿Qué tan seguros son los sistemas actuales de bloqueo para móviles? ((2015, Abril 28)).
- [11] O. Komogortsev and C. Holland. “Biometric identification via eye movement scan paths in reading.” in *International Joint Conference on Biometrics Compendium*, 2011.
- [12] O. Komogortsev and C. Holland. “2D Linear Oculomotor Plant Mathematical Model: Verification and Biometric Applications.” *ACM Transactions on Applied Perception (TAP)*, vol. 10, 2013.
- [13] P. Domínguez. *En qué consiste el modelo de cascada*, 2020.
- [14] P. Kasprowski and J. Ober. “Eye movements in biometrics.” in *European Conference on Computer Vision (ECCV)*, 2004.
- [15] P. Viola and M. Jones. Robust real-time face detection. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, Vol. 2. 747–747. <https://doi.org/10.1109/ICCV.2001.937709>
- [16] R. Bednarik, T. Kinnunen, and P. Franti. “Eye-Movements as a Biometric.” *Image Analysis*, vol. 3540, 2005.
- [17] R. Lienhart and J. Maydt. An extended set of Haar-like features for rapid object detection. In *Proceedings. International Conference on Image Processing*, Vol. 1. I–I, 2002. <https://doi.org/10.1109/ICIP.2002.1038171>
- [18] S. Hasan, T. Alam, T. Hasan, S. Tasnim, and S. Tanvir. (2016). *Controlling Android Device with Eye Tracking*, 2016.
- [19] Y. Huang, Z. Wang, and A. Ping. *Non-contact Gaze Tracking with Head Movement Adaptation based on Single Camera*, 2009.



© 2021 by the authors. This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.