Wasit Journal of Engineering Sciences

Journal homepage: https://ejuow.uowasit.edu.iq

Eye Movement Tracking Using Opencv Python

Mohammed S. Irshayyid[1], Basim k. J. Al – Shammari[1], Hayder Hussain Jasim[2]

Affiliations
[1]Department of Electrical Engineering, College of Engineering, Wasit University Wasit, Iraq
[2]Medical City Complex/ Specialised Surgeries Hospital/ Ophthalmology Center/Baghdad- Iraq
Correspondence
Mohammed S. Irshayyid,
Email:

mohammeds302@uowasit.edu.iq

Received 1-November-2022
Revised 5-December-2022
Accepted 1-August-2023

Doi: 10.31185/ejuow.Vol11.Iss2.393

Abstract:

In this study, we made a simple, low-cost algorithm for tracking eye movements and eye blinks in real-time and non-real-time. Several methods are being used right now. Show parts of the face, like the eyes, or the whole face. For this reason, OpenCV enables high-level programming to implement reliable and accurate detection algorithms like Haar Cascade. Pay attention to how hardware, like a computer, can only use a certain amount of resources (processing power). The system has been proven to work in tests with 15 people of different ages and backgrounds. These tests are done to see how the user and the device work together and ensure everything works correctly. The In the tests done, the system worked between 80% and 100% of the time. The results showed that Haar Cascade had a significant effect on the detection of faces in 100% of cases, while the eyes and pupil where they overlap (light and shade) were less effective. In addition to looking at how well these activities worked, the demo application also showed that user restrictions shouldn't stop people from enjoying and using a certain type of technology. The programme was written in C++, and the OpenCV library makes it work on Windows. This system has many uses in the real world and in science. By looking at the data from this algorithm from afar, for example, it can tell if someone has an eye disease or is tired. It can also help people who have physical or mental problems.

Keywords: Eye Control System, Face Detection, EMG, Eye Detection and Tracking, OpenCV.

الخلاصة: في هذه الدراسة ، أنشأنا خوارزمية بسيطة ومنخفضة التكلفة لتتبع حركات العين ومیض العين في الوقت الفعلي وغير الحقيقي. يتم استخدام عدة طرق في الوقت الحالي. أظهر أجزاء من الوجه ، مثل العينين أو الوجه كله. لهذا السبب ، يتيح OpenCV البرمجة عالية المستوى لتنفيذ خوارزميات كشف موثوقة ودقيقة مثل Haar Cascade. انتبه إلى كيف يمكن للأجهزة ، مثل الكمبيوتر ، استخدام قدر معين فقط من الموارد (طاقة المعالجة). لقد ثبت أن النظام يعمل في اختبارات مع 15 شخصًا من مختلف الأعمار والخلفيات. يتم إجراء هذه الاختبارات لمعرفة كيفية عمل المستخدم والجهاز معًا والتأكد من أن كل شيء يعمل بشكل صحيح. في الاختبارات التي تم إجراؤها ، عمل النظام بين 80٪ و 100٪ من الوقت. أظهرت النتائج أن Haar Cascade كان له تأثير معنوي في الكشف عن الوجوه في 100٪ من الحالات ، بينما كانت العيون والبؤبؤ عند تداخلهما (الضوء والظل) أقل فاعلية. بالإضافة إلى النظر في مدى نجاح هذه الأنشطة ، أظهر التطبيق التجريبي أيضًا أن قيود المستخدم يجب ألا تمنع الأشخاص من الاستمتاع واستخدام نوع معين من التكنولوجيا. تمت كتابة البرنامج بلغة C ++ ، ومكتبة OpenCV تجعله يعمل على Windows. هذا النظام له استخدامات عديدة في العالم الحقيقي وفي العلوم. من خلال النظر إلى البيانات من هذه الخوارزمية من بعيد ، على سبيل المثال ، يمكنها معرفة ما إذا كان شخص ما مصابًا بمرض في العين أو متعب. يمكن أن يساعد أيضًا الأشخاص الذين يعانون من مشاكل جسدية أو عقلية.

1. INTRODUCTION

When "eye tracking" is used in this context, it means figuring out where the user is looking. Most of the time, figuring out where someone is looking means figuring out who they are looking at. The thing that people look at [1-3] Eye tracking goes back to the 18th century. [2, 3] In 1792, Wells wrote about "after images," also called "ghost images." the way the eyes move [6]. During the 1800s, Javal (1879) and Lamare (1892) got eye movements that could be heard by using a The eyes and ears are linked together with a rubber band. In 1901, Dodge and Cline made the first small, unobtrusive radio. Eye movements (horizontal eye movements) are measured. Only) by taking pictures and using light reflections from the eyes. Jung took both vertical and horizontal eye measurements in 1939, Movements simultaneously with electrodes on the skin close enough to see [2].

Electrooculography is another name for this method. (EOG) is a test that measures the electric fields of the eyeball. Dipole [4, 5]. The method also led to the (theoretical) processing of gaze data in real time using analogue electronics. In the 1980s, small computers got strong enough to do eye tracking in real-time, which made it possible to use Eye trackers that use video (Video-OcculoGraphy) for people's interaction with a computer. Since the 1990s, there have been a lot of eye trackers That are being used more and more often. Price drops for the tracking systems grew in popularity, mostly for marketing research or studies of how things work. Scientists began to study how eye-trackers could help people and computers work together. Our research aims to use OpenCV and Python to track eye movement and calculate its frequency in real-time (camera) and non-real-time (video). In a simple and low-cost way and also where it is possible to monitor the condition of the eye from a distance. The rest of the paper is organized as follows. Section 2 presents the related work. Section 3 describes the research methodologies, followed by a discussion and conclusion.

2. Related Works

There are many studies on eye movement tracking, especially using Python's computer vision OpenCV library. We will be clarifying some of the previous studies in this field. Dhaval Pimplaskar et al. Robust and accurate algorithms for tracking the eyes in real time are fundamental problems in computer vision. Using initial centroid analysis, this paper suggested a way to figure out where and how the eyes look. Position of the eye between high and low blockage. This paper shows a way to determine if someone blinks their eyes in real time. Environment. Connected component and centroid methods are used to track eye blinking. Intel's open-source OpenCV platform [6], Shruti Mohanty et al. Drowsy driving causes thousands of deaths annually. Complex systems have been created to detect driver tiredness, but this research investigates a simpler, more effective alternative. This article uses Python and Dlib to detect drowsy drivers.

Dlib's form detector maps face landmarks and detect tiredness by monitoring eye and mouth aspect ratios. The suggested system's performance is evaluated using a public dataset and lab-captured real-time videos. The proposed system had a 96.71% video recognition accuracy [7]. Ahmad Aljaafreh et al. Eye movements are linked to cognitive processes, making them a useful research tool for studying human behaviour. Eye motions can reveal brain activities. This research uses a low-resolution webcam to create an eye tracker and saccades measurement tool. Using open-source software (Python), a consistent technique is created to record the eye location time series on a camera. Several algorithms extract high-level saccadic eye movement measures from raw gaze outputs. Ten normal and MS patients participate in a pilot study. The suggested system is quick, simple, and efficient for eye tracking and saccade assessment. Clinicians and doctors can use the instrument to diagnose neurological problems [8].

3. Research Methodology

Before we get into the specifics of an Eye tracking system, let's learn a bit about the eye and think about what we could do. There are three main parts to the eye, and shows in Figure 1.

Figure1 shows The main parts of the eye. The black circle in the middle is the pupil, Iris is the larger circle that can look different for each person, and Sclera always looks white. This article is a detailed guide on using Python and the OpenCV library to find and follow your students' movements. It is a step-by-step guide with a full explanation, so even people who have never done it can follow along. In our project, we need to get some important libraries, such as Idlib, OpenCV-python and NumPy. The system uses a computer camera or saved video to track eye movement through Python code. Figure 2 shows the flow diagram of the proposed eye-tracking system, During this algorithm, the head was fixed so as not to affect eye-pupil movement.

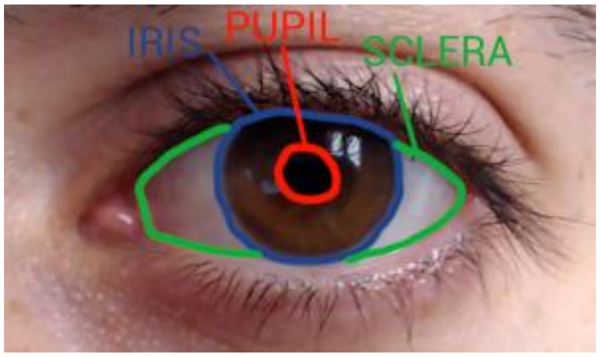


Figure 1: Parts of The Eye[9].

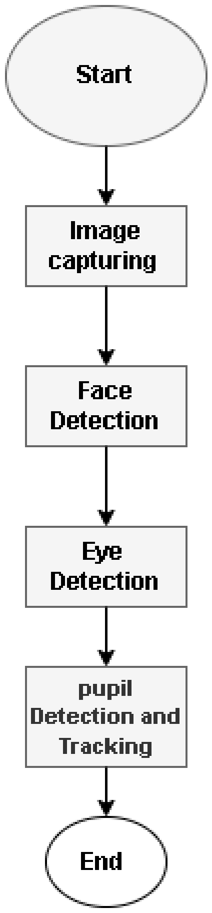


Figure 2: Block diagram of a typical eye-tracking system.

Each image captured by the computer's front-facing camera served as input to the procedure depicted in Figure 2, making this an iterative procedure. To ensure that this system may be integrated into existing and future applications, an open-source bookstore (OpenCV) was built. In this project, we built a structure incorporating tried-and-true practices and modern approaches. All of the phases described below are processed within this library.

3.1. Image Capturing

The main goal of this part is to make different frames based on what the computer's front camera records on video in real time, The specifications of the computer used are: Core i5 processor; GTX 1050 GPU; DDR4 RAM; Hard 512 SSD; and 720 cameras.In the next step, called "pre-processing," the image is changed from colour to grayscale, and the number of channels is cut from three to one. The image is also equalized to help with Detection. We use the cat colour function to turn each frame into a black-and-white image, Figure 3 shows.

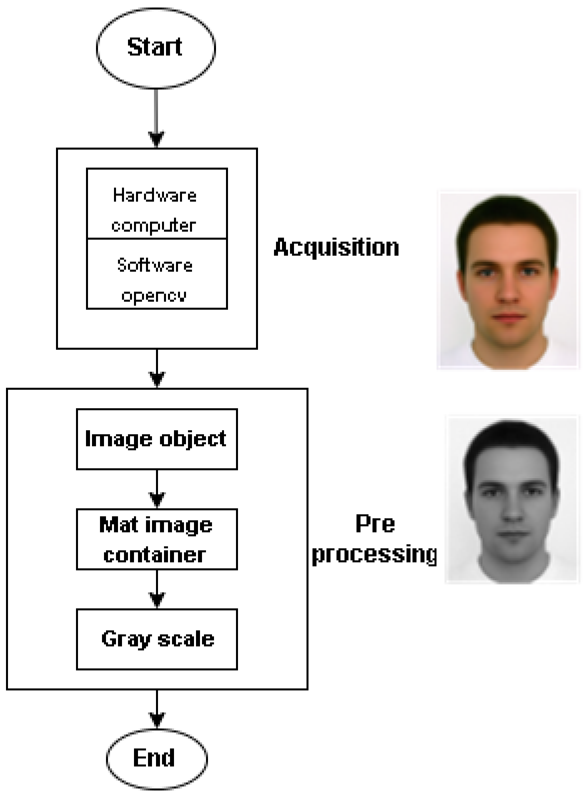
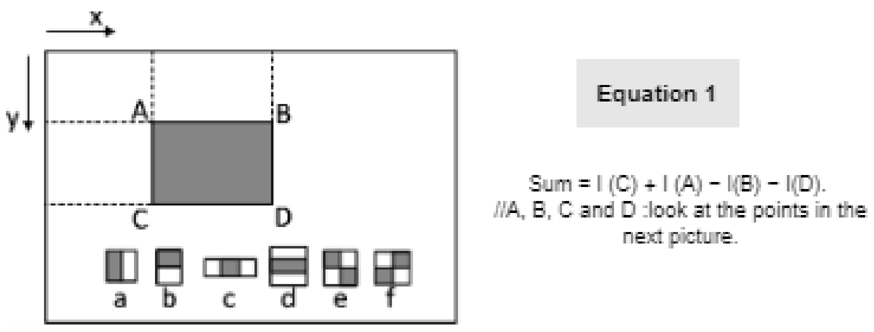


Figure 3: Acquisition and pre-processing operation.

3.2. Face Detection

Each image taken in part 1 starts to be worked on. The Haar Cascade object detector[10], which is trained to how track faces, is used to do this. Paul Viola and Michael Jones came up with the Haar Cascade method in 2001 [11]. It works very well. This is a machine-learning process in which the cascade function was taught from a large range of examples. There are images with faces that are good and images without faces that are bad [12, 13]. Once it's been after it has been taught, it is used to find things in pictures. In this project, the algorithm tracks the face and eyes [12-14]. The algorithm needs a lot of good and bad images to train the classifier. The summed area tables or integral images were one of the most important things that Viola and Jones did (see equation1). Integral images can be considered search tables with two dimensions that look like a matrix the same size as the original image. Each part of the whole picture comprises the sum of all the pixels in the upper left corner of the original image, which is where the element is.As you can see in Figure 4, this means that the sum of the rectangles in an image can be found in any place or at any size with just four searches.



Sum = I (C) + I (A) − I(B) − I(D).
//A, B, C and D :look at the points in the next picture.

Figure 4: Haar Cascade integral images.

Because of this system, Haar features of any size image can be used simultaneously, cutting down on processing time and making the system run better. Because of this, this type of template has been shown to work in the field of eye tracking, matching, and sorting techniques [15]. By getting the image matrix, you can get all the different face-tracking data, which will be looked at in the next step. At this point, more information is also gathered that lets you track your head. The next picture shows more about part 2. Lastly, it's important to discuss the algorithm made during this phase. This completes the whole process shown in Figure 5, including filtering.

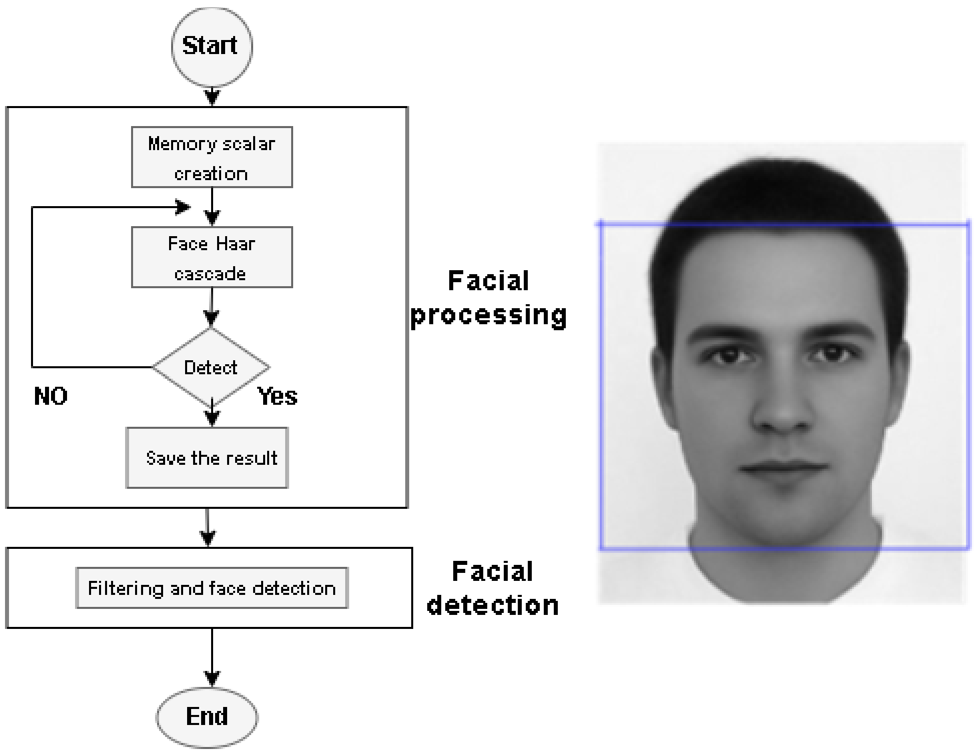


Figure 5: Daigram of Face Detection.

3.3. Eye Detection

In the third part, we start with the face detection matrix and change it so that processing only happens in the head's area of interest (ROI). Again, the same OpenCV resource is used to find both eyes. This time, a specially made Haar Cascade is used to find them. As a result, a matrix is made with both eyes. It was decided to work with only one eye so that the image matrix created by the Haar Cascade could be cut in half. This means that processing time could also be cut in half, which is important in real-time applications. Last but not least, this is the matrix that moves on to the next part. Figure 6 shows in detail the process and the result.

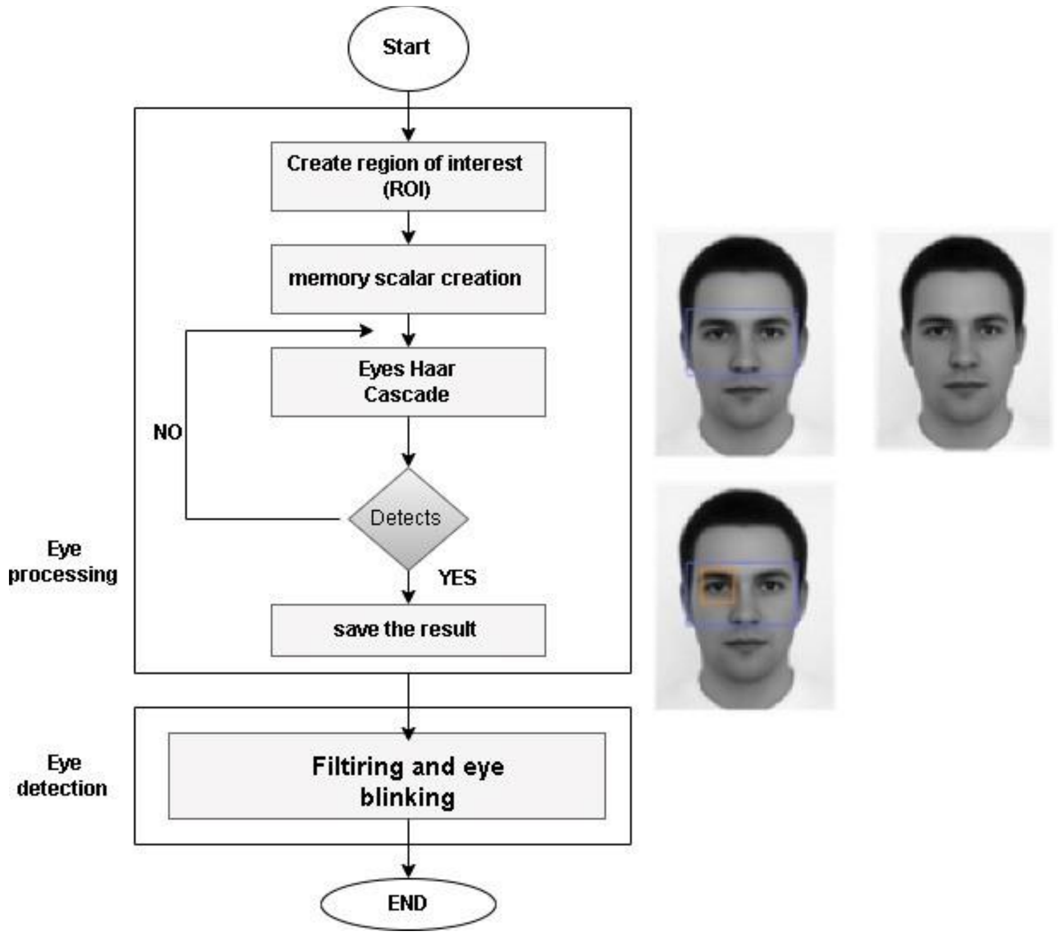


Figure 6: Diagram of Eye Detection.

During the eye detection phase, the algorithm that can tell when the eye blinks is made. The blink Control algorithm carries out the phase 3 process and its filtering stage: blink Control Algorithm: What It Does. This algorithm controls the blinking of the eyes. Eye open, Eye close, and How long the eye is closed for The call to a second method that filters the different states is also included. Figure 7 is a flow chart showing how the filtering method works. When the eye goes from open to closed, the meter starts counting, and when the eye goes from closed to open, it stops counting. This lets the meter figure out how long the eye is closed.

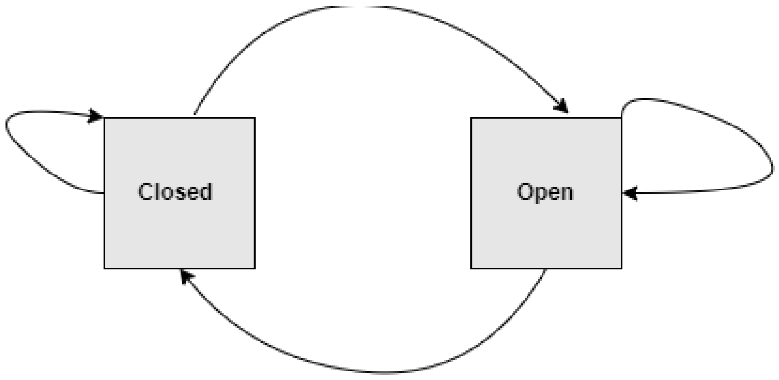


Figure 7: A flow chart for filtering the blink Control algorithm.

3.4. pupil Detection and Tracking

Due to the hardware requirements mentioned in the first stage, different methods [16] were tested, but some couldn't be used because of hardware limitations. The Hough transform Circles, often used to find circles, are a good example of this (such as the pupil). Figure 8 shows this is impossible because the image is not clear enough to find the pupil (circle).

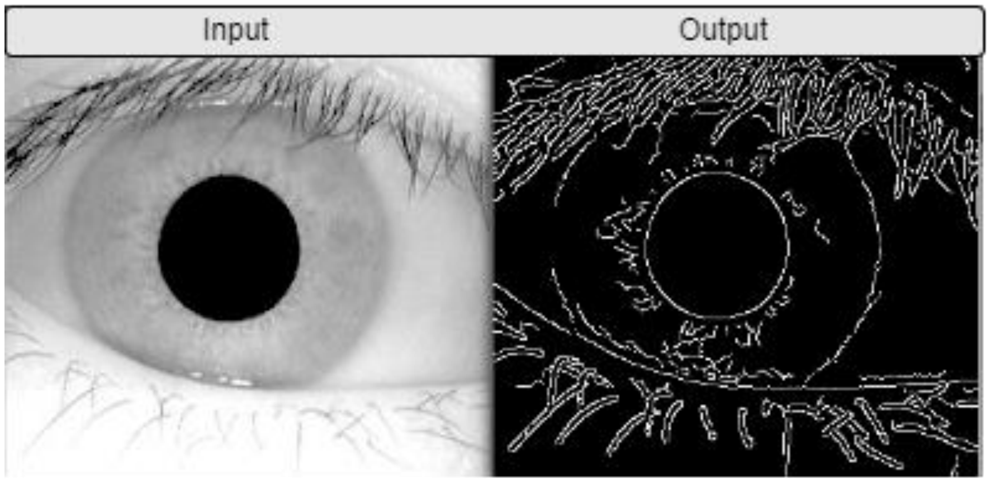


Figure 8: Eye region of interest.

Figure 8 shows that because of the quality of the camera, the Hough Circles did not change to be used correctly. When the signal was amplified, the low resolution and interference (eyelashes) made it hard to see. Bright pupil eye tracking is the process of capturing and processing eye images with a set of illuminators that are placed close to the optical axis of the camera. The light enters the eye, reflects off of the retina, and then exits the eye on axis and directly back into the camera. This causes the pupil to appear brighter in the image. It is the same phenomenon that causes the "red-eye effect" in photographs. The eye tracking algorithms identify the pupil by searching for a bright elliptical form in the image. An image so much that a circle could not be seen. In the end, it was decided to use the matrix values from the last step, with the darkest values coming first. The value of the eye's pupil is found. A system was made to figure out the direction of gaze that doesn't need to be calibrated every time it's used because it's made to work in the background with little user input. The following method was done with a tool that only needs to be set up once by the user. After this step, data for eye tracking is collected, and the goal comes at the end. That was explained when eye tracking began. The process is then repeated to find the pupil, as shown in Figure 9.

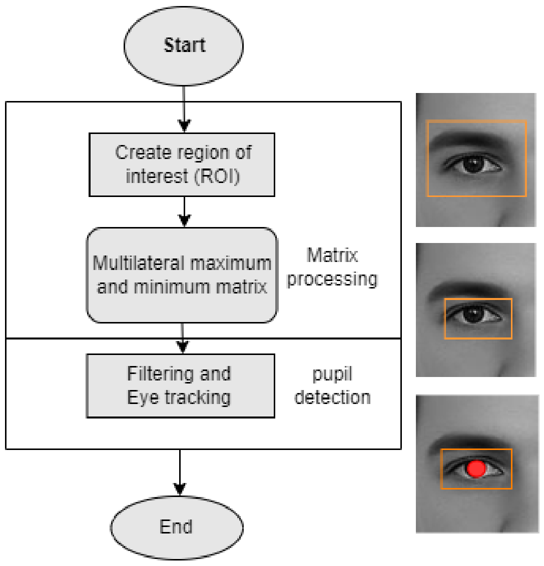

Start
Create region of interest (ROI)
Multilateral maximum and minimum matrix
Matrix processing
Filtering and Eye tracking
pupil detection
End

Figure 9: Diagram of pupil detection and Tracking.

To finish this last part of the fourth stage, a new algorithm was made that eye tracking in charge. Eye control Description The above algorithm is based on the coordinates of the pupil and the width of the eye's region of interest. Figure10 shows how this can determine the gaze direction (left, centre, or right).

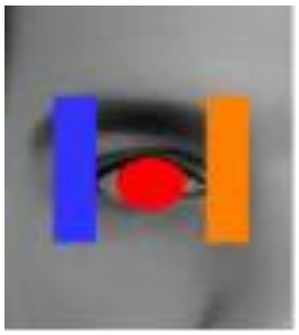


Figure 10: Processing margins for eye tracking.

Two margins were found, and They change depending on how big the user's eyes are. After they've If the central sensor has been set up, it is possible to tell if a person is looking left, right, or in the middle. The pupil's point goes past any of the lines. Also included is a call to a second method responsible for filtering the different positions. The flow chart in Figure 11 shows how the filtering method works. Events happen when the centre moves to the left and the centre to the right.

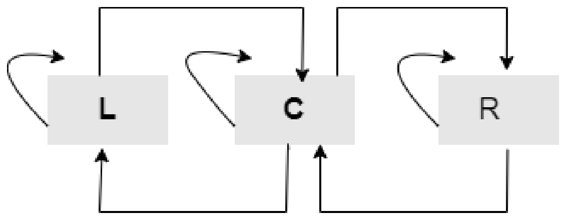


Figure 11: The Flow of the eye control algorithm filtering process.

4. Result from Analysis and Discussions

Initially, I applied this system to track eye movement in real-time to obtain the motion wave and its frequency to analyze the components of this wave. Also, I can track each eye separately or both in this system. I experimented with the left eye, As shown in the Figure 12.

(a)

(b)

(c)

(d)

(e)

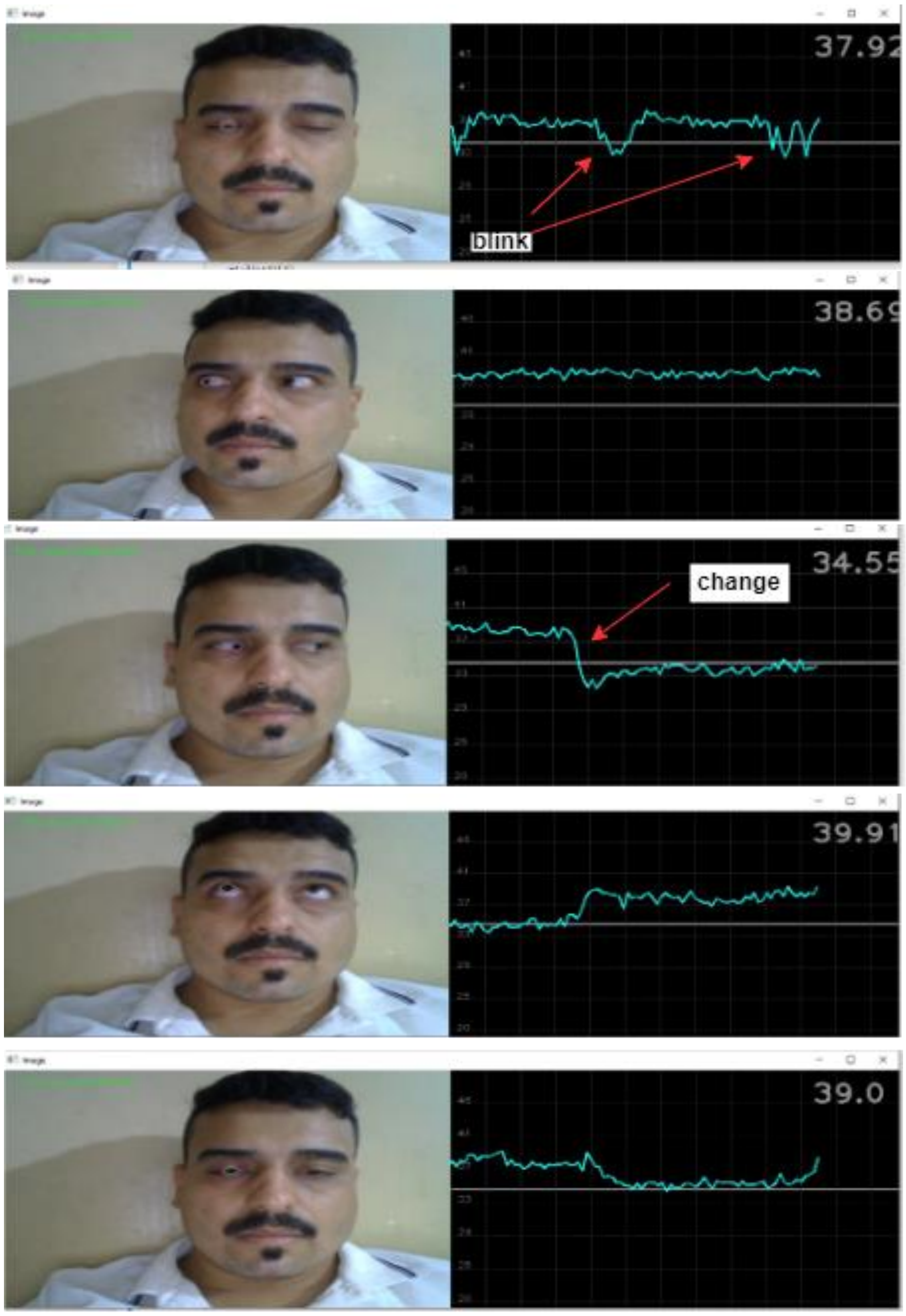


Figure 12: Real time eye movement tracking system test :(a) Blinking, (b) Left movement, (c) Right movement, (d) Up movement, (e) Down movement.

After completing the recording process, for the time being, we will get two waves, the first representing the eye movement waveform and the second representing its frequency, as shown in figure 13.

(a)

(b)

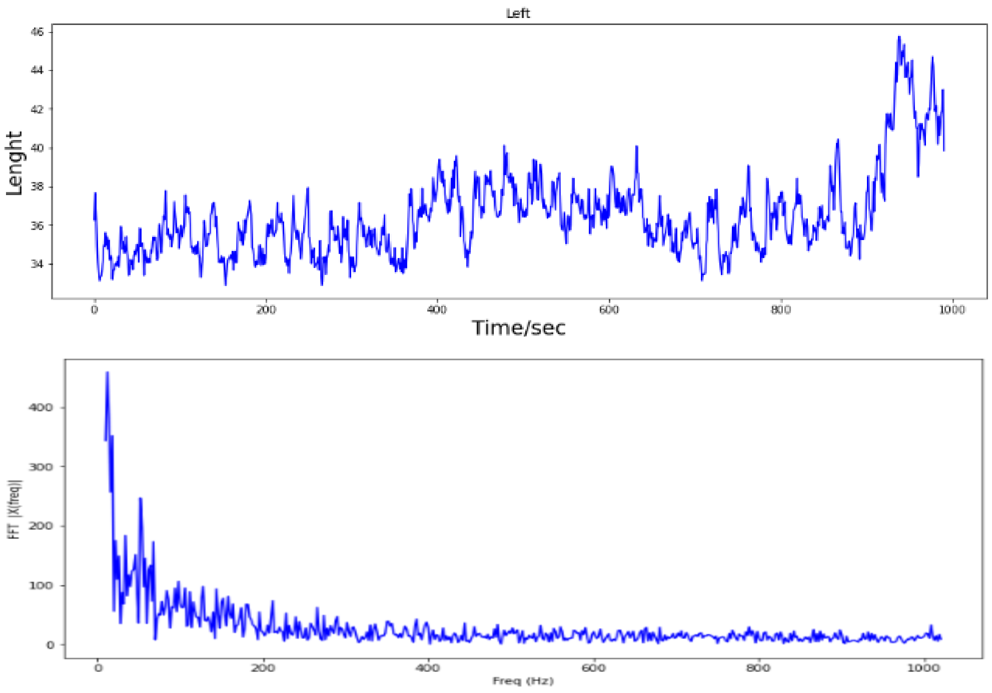


Figure 13: Output waveform for real-time eye movement system : (a) Time domain eye movement signal, (b) Frequency domain eye movement signal.

In the second part of the system testing process, I tested the tracking process in non-real-time through a video stored on the computer, which is very useful for remote study and analysis. I tracked each eye separately, as shown in Figure 14, VOG can sometimes be challenging, especially due to pupil detection problems (e.g., blinking, droopy eyelids, etc.) as compared with EOG, which has challenging It is affected by the interference of signals and cannot analyse the signal remotely.

(a)

(b)

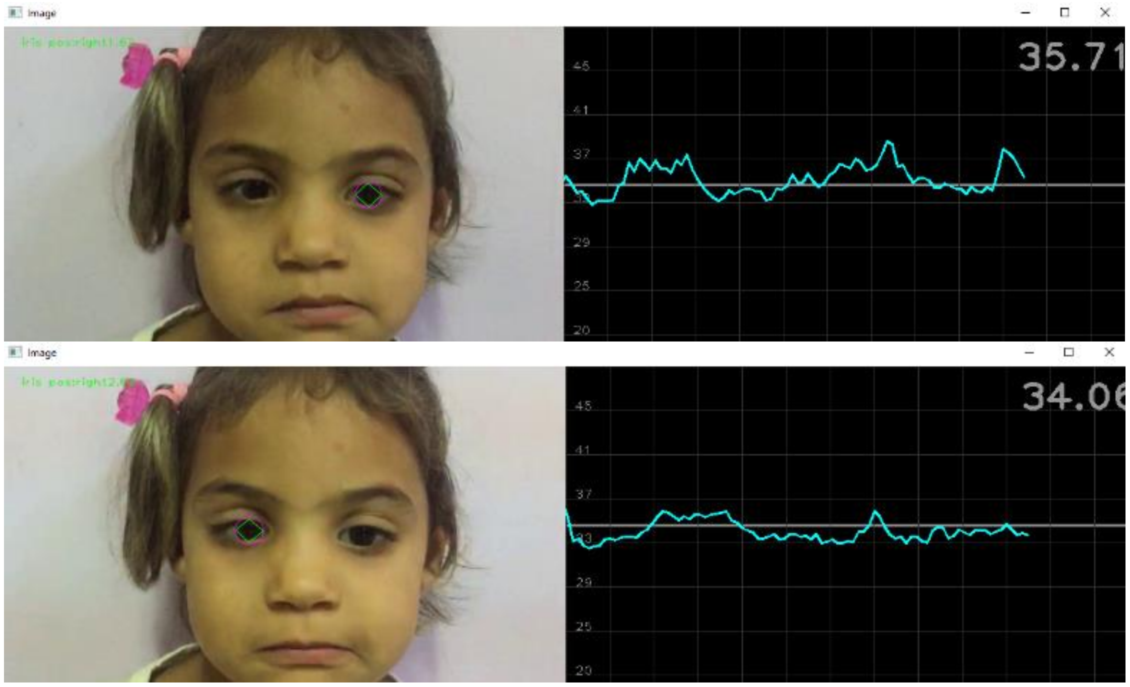


Figure 14: Non real-time eye movement tracking system test : (a) Right eye tracking, (b) Left eye track.

5. Conclusion

This paper describes our algorithm for real-time and non-real-time eye tracking. The Haar Cascade algorithm tracked, found, and applied filters to improve results. This approach works at 25–30 fps. OpenCV and the Single camera view extension power the C++ Windows app. Our instructions Video surveillance findings are good in bright light.The circular Hough transform is commonly employed for iris centre detection, a crucial step in eye tracking progression. When ambient light changes, the typical circular Hough transform method's accuracy drops. To solve this, a facial feature detector detects the eye as a region of interest (ROI), tracks the eyes, and classifies eye locations as left, right, or centre to identify the gaze. The classifier scans pixels to classify eye position for accuracy. Python implements OpenCV's eye-tracking method for portability. Iris detection and fixation position classification accuracy averaged 80% and 100%, respectively.

6. References

[1] K. Arai and R. Mardiyanto, "Eye-based HCI with full specification of mouse and keyboard using pupil knowledge in the gaze estimation," in 2011 Eighth International Conference on Information Technology: New Generations, 2011: IEEE, pp. 423-428.

[2] M. S. Kalas, "Real-time face detection and tracking using OpenCV," international journal of soft computing and Artificial Intelligence, vol. 2, no. 1, pp. 41-44, 2014.

[3] C. C. Singer and B. Hartmann, "See-thru: Towards minimally obstructive eye-controlled wheelchair interfaces," in The 21st International ACM SIGACCESS Conference on Computers and Accessibility, 2019, pp. 459-469.

[4] A. B. Usakli, S. Gurkan, F. Aloise, G. Vecchiato, and F. Babiloni, "On the use of electrooculogram for efficient human-computer interfaces," Computational intelligence and neuroscience, vol. 2010, 2010.

[5] H. S. Dhillon, R. Singla, N. S. Rekhi, and R. Jha, "EOG and EMG based virtual keyboard: A brain-computer interface," in 2009 2nd IEEE International Conference on Computer Science and Information Technology, 2009: IEEE, pp. 259-262.

[6] D. Pimplaskar, M. Nagmode, and A. Borkar, "Real-time eye blinking detection and tracking using OpenCV," technology, vol. 13, no. 14, p. 15, 2015.

[7] S. Mohanty, S. V. Hegde, S. Prasad, and J. Manikandan, "Design of real-time drowsiness detection system using dlib," in 2019 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE), 2019: IEEE, pp. 1-4.

[8] A. Aljaafreh, M. Alaqtash, N. Al-Oudat, J. Abukhait, and M. e. Saleh, "A low-cost webcam-based eye tracker and saccade measurement system," Int J Circuits Syst Signal Process, vol. 14, 2020.

[9] J. Zhu, E. Zhang, and K. Del Rio-Tsonis, "Eye anatomy," eLS, 2012.

[10] N. Grammalidis and M. G. Strintzis, "Head detection and tracking by 2-D and 3-D ellipsoid fitting," in Proceedings Computer Graphics International 2000, 2000: IEEE, pp. 221-226.

[11] R. Lienhart and J. May, "An extended set of haar-like features for rapid object detection," in Proceedings. International conference on image processing, 2002, vol. 1: IEEE, pp. I-I.

[12] P. I. Wilson and J. Fernandez, "Facial feature detection using Haar classifiers," Journal of Computing Sciences in Colleges, vol. 21, no. 4, pp. 127-133, 2006.

[13] D. A. Forsyth and J. Ponce, Computer vision: a modern approach. Prentice hall professional technical reference, 2002.

[14] Y.-l. Tian, T. Kanade, and J. F. Cohn, "Dual-state parametric eye tracking," in Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580), 2000: IEEE, pp. 110-115.

[15] R. Padilla, C. F. Costa Filho, and M. Costa, "Evaluation of haar cascade classifiers designed for face detection," International Journal of Computer and Information Engineering, vol. 6, no. 4, pp. 466-469, 2012.

[16] C. P. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," in Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271), 1998: IEEE, pp. 555-562.