Eyesight Protector

Chun Ting Chan1, \* and Ching Fung Lai2, #

*1 St Paul’s Co-educational College, Hong Kong*

*2 Po On Commercial Association Wong Siu Ching Secondary School, Hong Kong*

*\*Contact:* [*jchanyw88@gmail.com*](mailto:jchanyw88@gmail.com)

*#Contact:* [*std391050@pocawsc.edu.hk*](mailto:std391050@pocawsc.edu.hk)

***Abstract*— The problem of myopia is highly common in Hong Kong. Due to the increasing popularity of electronic devices, researchers have proven that that users with poor usage habits have a higher tendency to be affected by myopia. Therefore, an eyesight protector is designed to remind the user to maintain a proper distance from the monitor and go to rest when they spend too long in front of the computer. The project uses the technology of facial detection (with the module of OpenCV) and calculates the distance with a simple self-designed algorithm. It aims to measure the distance of the user from the monitor, and to remind users when they have been using the computer for too long. Using the camera to do such work is better than using the radar detector, which is a norm, as it is more convenient and suitable for home use.**

1. Introduction
   1. *Problem Identified*

Myopia is a serious health concern for the youths of Hong Kong. Myopia affects more than 10% of children aged 6 years old, and up to 45% of children aged 9 years old.

Therefore, it is proven that myopia is a serious problem among young children in Hong Kong. Many children have had myopia since a very young age. There are assorted reasons behind the problem. One of them is proven to be due to poor habits of students when doing near work[1]. With the increasing prevalence of electronic devices, it is very reasonable for students to spend a significantly longer time on electronic devices, which is a kind of near work. Therefore, the effect of poor habits when using electronic devices is growing notably.

* 1. *Objective*

Unhealthy computer usage habits are a common cause of myopia, but it could be prevented by simply, timely reminders. In view of this, we aim to alleviate the problem by taking advantage of the often overlooked and underused webcams to alert the users whenever they are too close to the screen, thus nurturing proper usage habits.

1. Methodology
2. *Targeted Quantities to be Measured*

The project has the targets of reminding the users when they are too close to the screen and when rest is needed. The quantities in question are stated in table 1.

TABLE I  
Concerned Quantities

|  |  |  |  |
| --- | --- | --- | --- |
| **Quantity Nos.** | **Quantity** | **Data type of quantity** | **Purpose of the quantity** |
| I | User’s distance from the screen | Float | It determines whether the user is too close to the computer screen. |
| II | Existence of the user’s face on the camera | Boolean | It determines whether the user is away from the screen. |
| III | The user’s time span away from the camera | Float | It determines how long the user has rested, staying away from its computer. |

1. *Method of Extracting the Quantities*

*1) Categorizing quantities:* The quantities are then categorized into two groups, by determining whether they could be directly measured or not. To determine it, there are two constraints to be kept, including (i) the camera could only capture objects in its field two-dimensionally (namely only the x-axis and y-axis); and (ii) due to storage space limits, the program could not save captured photos, and instead could only extract the needed quantities from the photos.

Following the above constraints, quantity I is impossible to be extracted directly as the camera could only capture two-dimensionally, i.e., the x-axis and y-axis, while the quantity required is a z-axis quantity, which is out of range for the camera to determine. Also, quantity III has the same problem as the program cannot store pictures captured. Hence, it is impossible for the program to compare each picture captured and then measure the time span the user is away from the computer.

*2) Method to extract quantities that could not be measured directly:* Therefore, there is a need to search for methods to measure quantity I and quantity III. Calculations are required to determine such quantities.

*a) Quantity I:* For quantity I, a simple algorithm based on simple geometry is developed based on the diagram at fig. 1.

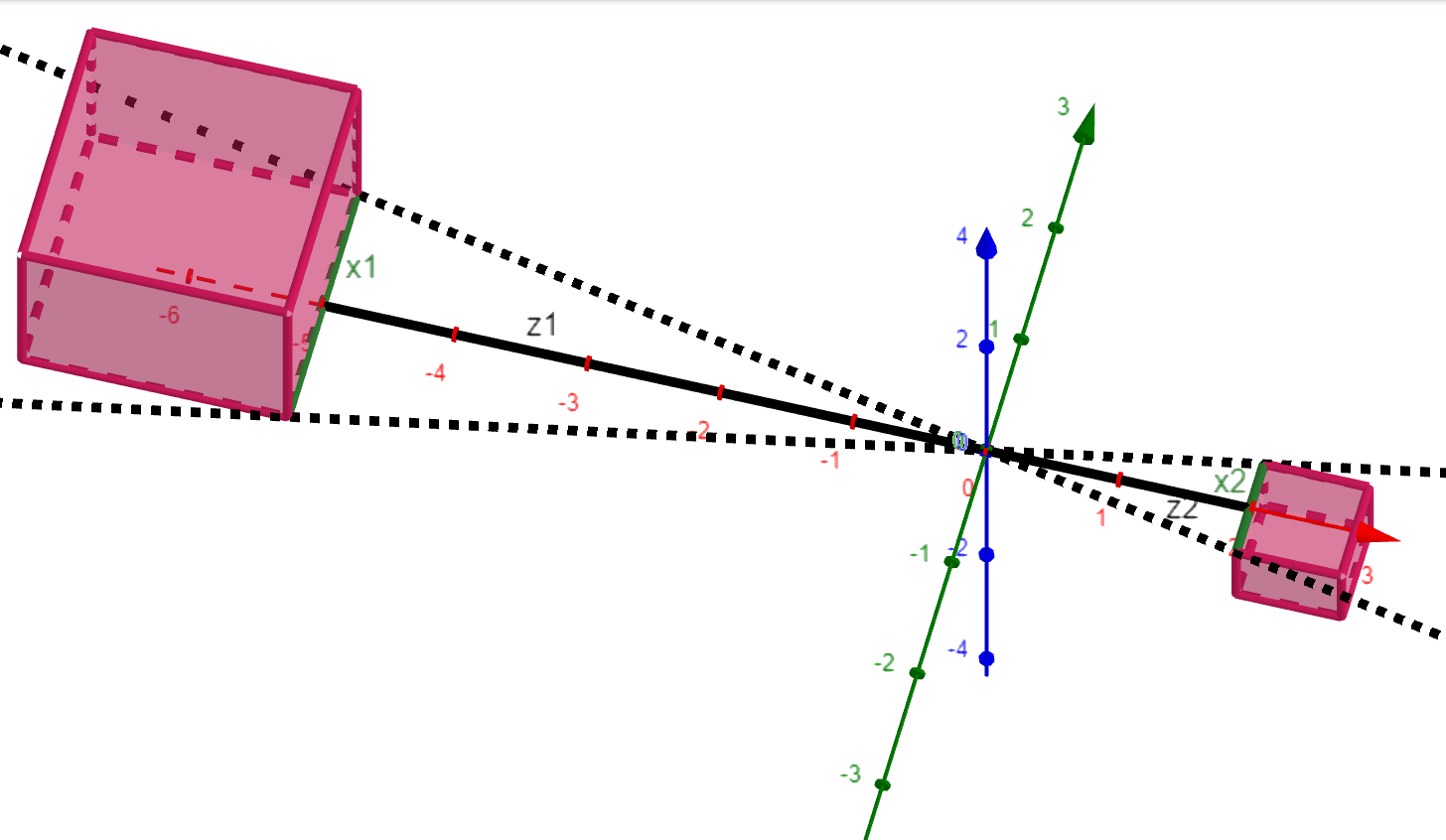


Fig. 1 A diagram showing all the unknowns (with the left side as the real side and right side as the virtual side)

The user is required to insert value of (the actual face width of the user) and (the distance of the user from the camera currently). This aims to find (the length between the camera lens and its receptor screen) according to the following equation.

Then, by utilizing the fact that both the result of and (the length between the camera lens and its receptor screen) are constants, a constant for the use of the program is deduced.

(the virtual “width” of the face) could be measured through the module “OpenCV” to identify a face, and finally, the actual distance of the user from the camera can be calculated with the equation below.

*b) Quantity III:* As quantity II is measurable as a Boolean object, by recording the point of time at which it first became “False” and when it turned back to “True”, the time difference can be used to determine the length of time the user is away from the camera.

*2) Quantity II – Determining whether faces exist:* The Python module “OpenCV” is used as a pre-trained model for utilization. Then it is utilized to determine whether the face exists, then return a few other quantities. The returned message of whether a face exists is then used to initiate the Boolean object (i.e. quantity II). However, a limitation lies in the fact that when many people are within the screen, the program lacks the ability to differentiate the user’s face accurately.

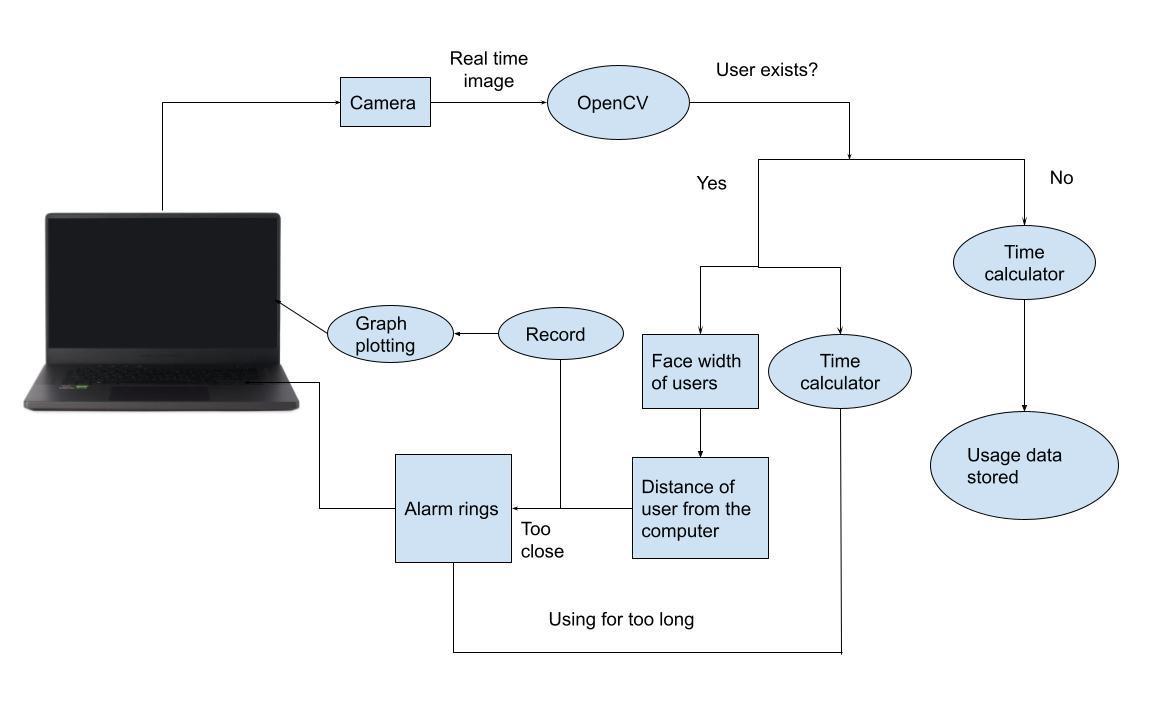


Fig. 2 A diagram showing the logic of methodology.

1. *Extra Improvement – Facial Recognition*

As the program goes through testing, it is found that the computer lacks ability to differentiate the actual user with other passerby. By experiment, the probability that the program successfully determines the correct distance (“accuracy”) within a 500 seconds period is plotted against the time span between each passerby. The graph is shown in figure 3.

一張含有 桌 的圖片

自動產生的描述

Fig. 3 A graph showing the effect of passerby on accuracy.

When there is a passerby every 20 seconds, the accuracy goes under our expectation. This gravely decreases the versatility of the program. For example, when users use their computer in a public area, the program may work poorer than expected. As a result, a facial recognition system is created, to differentiate users from the passerby.

At the calibration stage, a variable number of photos of the user are taken. It is used to generate a make-shift database to determine whether a face captured by camera is the user. Then, during operation, the program identifies the user and labels who is the user and who is an unknown stranger in the video feed. The quantities in question are then taken from the actual user, giving accurate results. The accuracy has increased significantly (from 90.6% to 96.3%) after this model is implemented.

1. Functions
2. *Methods of Recording the Data*

To allow users to understand more about their computer usage habits, the accumulated data of quantity III will be stored in a list, as well as it will be exported to another file which will then plot for the user an instantaneous graph of their distance from the screen. This helps users to analyze their habits, for example, whether they have the tendency to move closer and closer to the screen as they use their computers.

Also, to allow users to analyze their rest pattern, a dictionary including all their rest information will also be provided and printed. The user could then check if they have adequate rest when using the computer. However, if the user returns within 10 seconds, the period will be neglected as it may be caused by some error other than the user has really gone to rest.

1. *Reminding the User*

In the program, the most important part is to remind the user that they are too close to the camera, as well as when they need to rest. Therefore, a few methods are used to remind the user in both scenarios.

When the user is too close to the camera, an alarm, constituted by a 853 Hz and a 960 Hz sound, will go off. It is set to be intentionally noisy and unbearable due to acoustic reasons. Therefore, the user will have a higher tendency to get notified immediately. At the same time, the words “Too close!” in red will show on the screen, warning the user both by visual and auditory means.

When the user is using the computer longer than the time span they have set, a softer alarm will ring, and the sentence “Go get some rest!” will show on the screen, reminding users to rest. As they leave the screen, the program will automatically record that they have gone to rest, utilizing the function mentioned in part A.

1. Results
2. *General Accuracy*

The general accuracy is tested by allowing users to use the program while they are using their computer as usual. There are a total of 15 people who have tested the program, with an accumulated total usage time of the program of 493 minutes. In the 493 minutes, it is calculated that the users’ action is successfully identified in 475 minutes, therefore, the successful rate is approximately 96.3%.

By analyzing through the screen recording returned by the test users, the major cause of the inaccuracy is caused by the angle of rotation of the head and the impact of over-exposure or under-exposure to light.

1. *Angle of Rotation with Accuracy*

The angle of rotation of the user’s head affects the accuracy of our program. To compare the accuracy effect of angle of rotation, experiment is conducted by eliminating all other effects, only remaining the angle of rotation, and the result is plotted as a graph, at fig. 2.

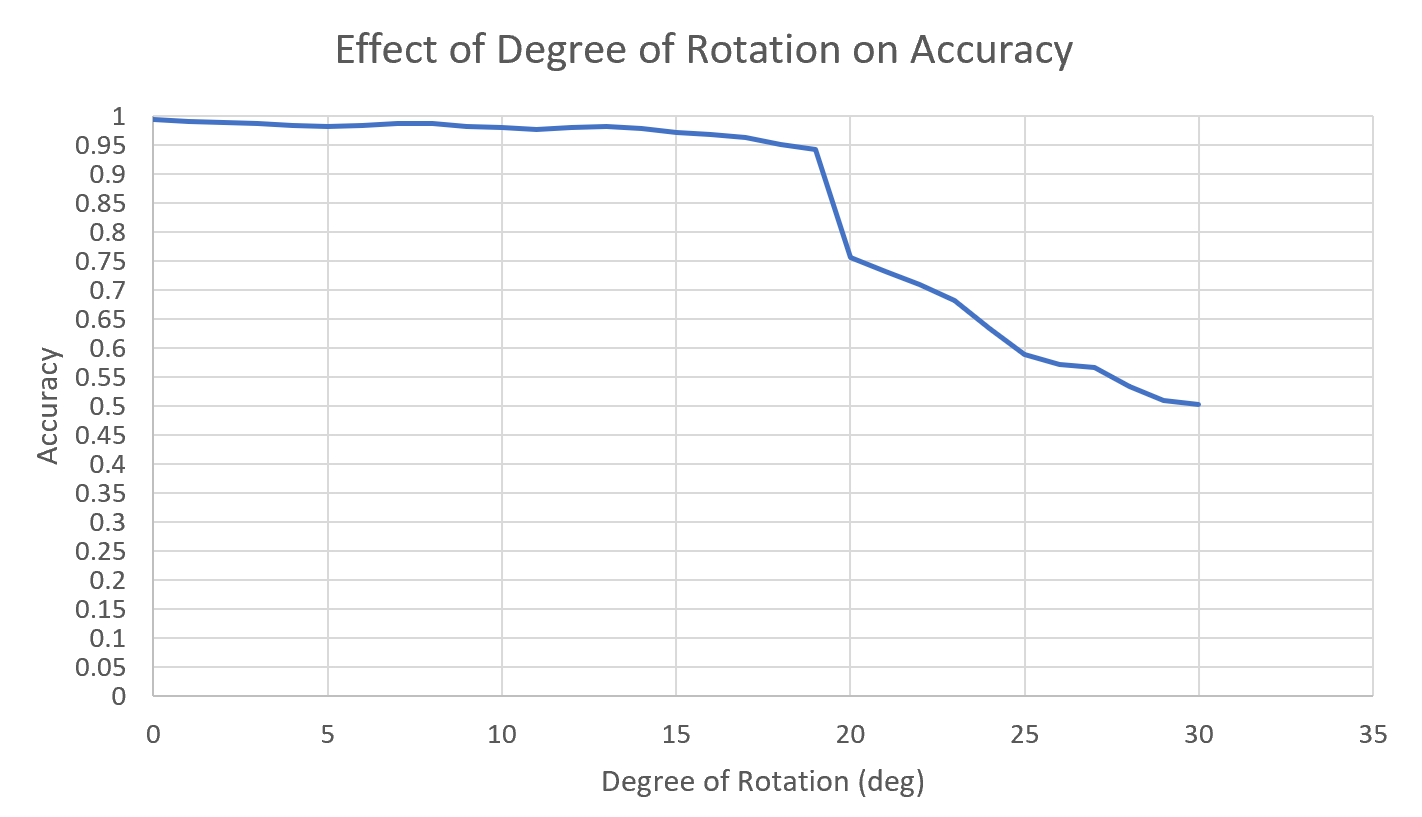


Fig. 4 A graph showing the effect of degree of rotation on accuracy.

After comparing with the control setup, with accuracy of 96.3%, it is concluded that the expected accuracy could only be maintained when the user is maintaining its head forming an angle <16° with the perpendicular line from the screen. However, this should not be considered as a main defect, as the user will be unable to read the screen if they rotate too much from the axis.

1. *Exposure to light with Accuracy*

To experiment the effect of light intensity on the accuracy of the program, the camera quantity ISO is utilized. For brighter areas, a lower ISO value is needed to get a clear image, while in darker regions, a higher ISO value is needed. By achieving the optimal ISO value through a separated auto-focused camera while running the program, the light intensity of the area could be measured indirectly. The result is shown in the graph, at fig. 3. However, it should be noted that the camera on a laptop itself does not have the ability to change the ISO value.

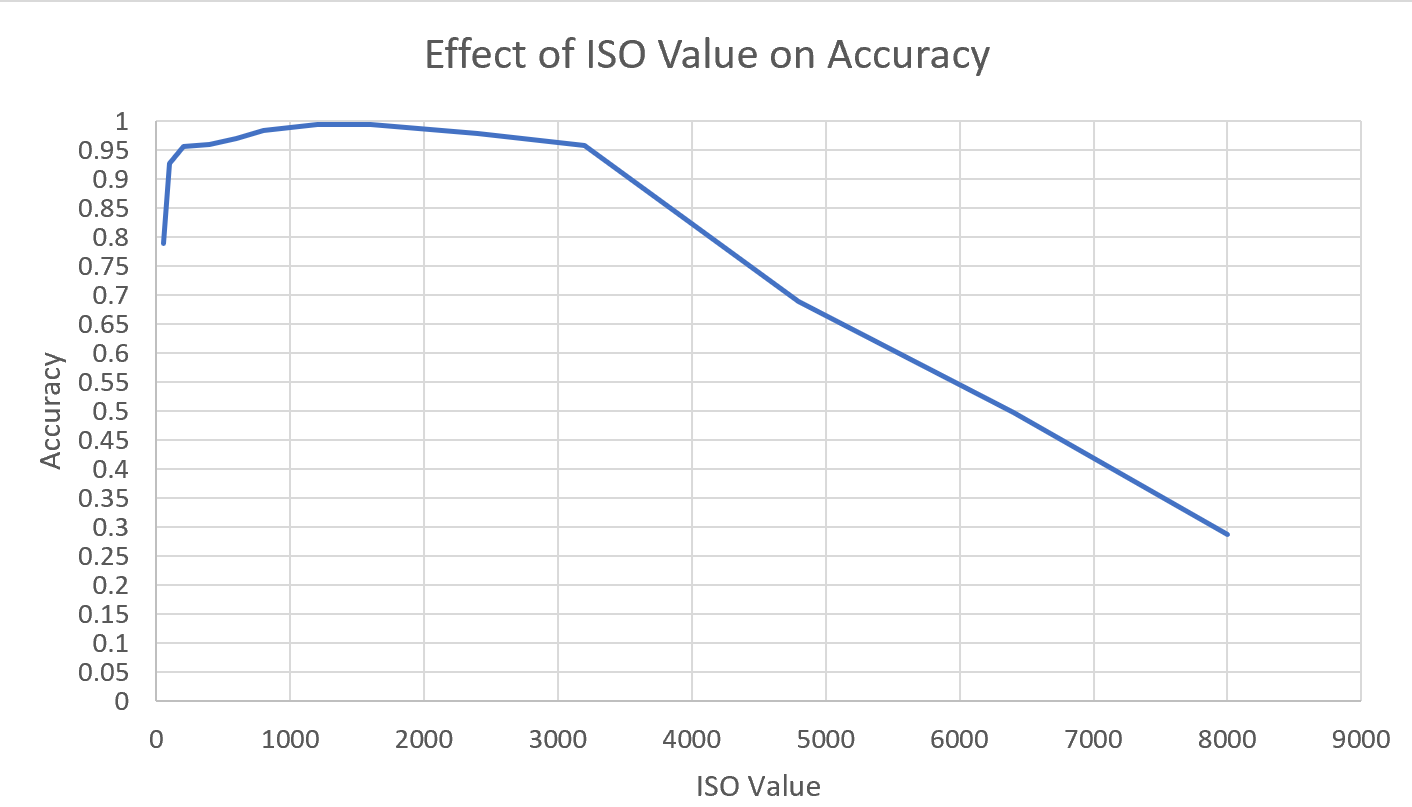


Fig. 5 A graph showing the effect of ISO value on accuracy.

It is concluded that the program works up to standard with an ISO value approximately around 150 < ISO Value < 3200. However, when the light intensity of the environment is too low, i.e., requires a very large ISO value, the camera could not absorb enough light to produce a clear enough image to identify whether a person exists. However, as it is very harmful to read from a computer screen in a very dark environment, it is expected users will not be using the program in such a dark environment.

1. *Effect on Users*

To experiment the effect on user’s habits with our program, we used the percentage of users that goes off the alarm during each minute. When the number decreases, it proves that more users could maintain a good posture during their use of computers.

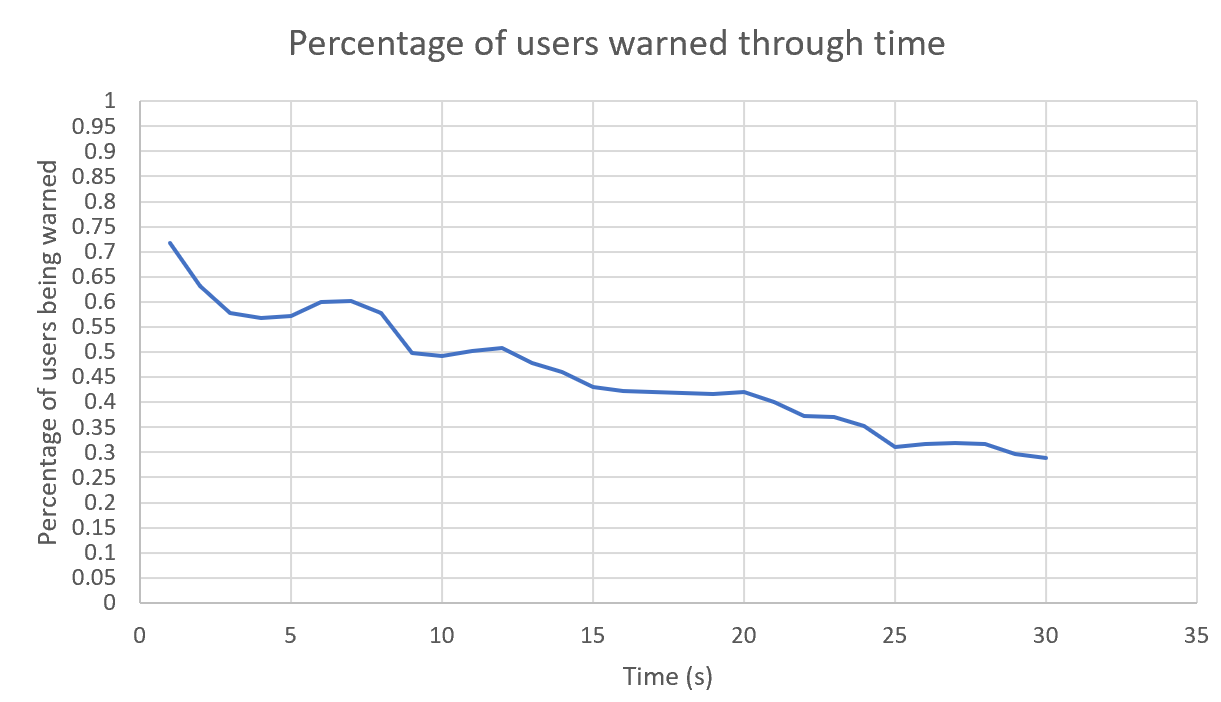


Fig. 6 A graph showing the percentage of users warned through time.

It could be deduced that the general trend is decreasing. More and more users could maintain good posture through time, and are less dependent on reminders. This proves that in the short-term, the program has a good ability to correct users’ habits. However, although we believe that through constant reminders, a long-term habit can be nurtured, more tests are required to prove the actual figures.

1. Conclusions

The program works currently with a few constraints, including the angle of rotation of the user’s head and over-exposure to light. These all lead to the difficulties for the program to identify the person successfully. Therefore, the current version of the program may underperform when serving users that have to constantly interact with others in the surroundings during usage of the computer. Further development, such as to include another camera to deduce the user’s distance, could help the program to improve its performance. However, we believe the product still has advantages over using the radar for the job, as it is more convenient and does not require extra installation and power source.

Acknowledgment

We would like to thank Dr. Ray Cheung for his guidance throughout the whole GEF program, as well as TAs from the GEF program for assisting our learning. We would also like to thank our parents for having trust in us, providing us support throughout the learning process.

References

1. Jenny M. Ip, Seang-Mei Saw, Kathryn A. Rose, Ian G. Morgan, Annette Kifley, Jie Jin Wang, Paul Mitchell; Role of Near Work in Myopia: Findings in a Sample of Australian School Children. Invest. Ophthalmol. Vis. Sci. 2008;49(7):2903-2910. doi: <https://doi.org/10.1167/iovs.07-0804>.
2. Kim, Donghee & Han, Kyungsik & Sim, Jeong & Noh, Youngtae. (2018). Smombie Guardian: We watch for potential obstacles while you are walking and conducting smartphone activities. PLOS ONE. 13. e0197050. 10.1371/journal.pone.0197050.
3. Sriyanti, I & Aliyana, P & Marlina, L & Jauhari, J. (2020). Light Intensity Analysis Using Smartphone’s Light Sensor. Journal of Physics: Conference Series. 1467. 012056. 10.1088/1742-6596/1467/1/0120