ISSN: 2508-7894 © 2022 KODISA & KAIA.
KJAI website: http://acoms.kisti.re.kr/kjai
doi: http://dx.doi.org/10.24225/kjai.2023.11.1.9

# Development of a Non-contact Input System Based on User's Gaze-Tracking and Analysis of Input Factors*

**Jiyoung LIM[1], Seonjae LEE[2], Junbeom KIM[3], Yunseo KIM[4], Hae-Duck Joshua JEONG[5]**

**Received: Febrary 26, 2023. Revised: Febrary 28, 2023. Accepted: March 04, 2023**

## Abstract

As mobile devices such as smartphones, tablets, and kiosks become increasingly prevalent, there is growing interest in developing alternative input systems in addition to traditional tools such as keyboards and mouses. Many people use their own bodies as a pointer to enter simple information on a mobile device. However, methods using the body have limitations due to psychological factors that make the contact method unstable, especially during a pandemic, and the risk of shoulder surfing attacks. To overcome these limitations, we propose a simple information input system that utilizes gaze-tracking technology to input passwords and control web surfing using only non-contact gaze. Our proposed system is designed to recognize information input when the user stares at a specific location on the screen in real-time, using intelligent gaze-tracking technology. We present an analysis of the relationship between the gaze input box, gaze time, and average input time, and report experimental results on the effects of varying the size of the gaze input box and gaze time required to achieve 100% accuracy in inputting information. Through this paper, we demonstrate the effectiveness of our system in mitigating the challenges of contact-based input methods, and providing a non-contact alternative that is both secure and convenient.

**Keywords :** Eye-tracking, Intelligent gaze-tracking, Face Recognition, Gaze, Security, Non-contact input system

**Major Classification Code**: C45, C88, C91, L86, O33

## 1. Introduction

Traditionally, information has been entered using keyboards and mouses. However, the digital age, where mobile devices such as smartphones and tablets are widely used, requires alternative input methods. Even on mobile devices, character and number string passwords have been used for simple user authentication and early smartphone lock screen security using soft keyboards. However, such passwords can be difficult to remember if they are complex and different for each service, leading to weak security (Kim, & Kwon, 2016).

* This study was supported by UISP (University Innovation Support Project) of Korean Bible University in 2022

1 First Author. Professor, Dept. of Computer Software, Korean Bible University, Korea. Email: jylim@bible.ac.kr

2 Second Author. Undergraduate student, Dept. of Computer Software, Korean Bible University, Korea. Email: dltjswo0323@bible.ac.kr

3 Third Author. Undergraduate student, Dept. of Computer Software, Korean Bible University, Korea. Email: estjunbeom@bible.ac.kr

4 Fourth Author. Undergraduate student, Dept. of Computer Software, Korean Bible University, Korea. Email: kys051348@bible.ac.kr

5 Corresponding Author. Professor, Dept. of Computer Software, Korean Bible University, Korea. Email: joshua@bible.ac.kr

To address this issue, various password input methods have been studied, including unlocking PIN numbers made by combining numbers from 0 to 9, creating user-specific patterns, and using user's body information such as fingerprint recognition. However, these methods may not be useful for individuals who are restricted from using their bodies due to innate or acquired factors (Kang, Kim, Seo & Kim, 2021). To overcome these limitations, researchers have been studying password input methods that do not require physical contact, such as gaze, gesture, and face recognition. In particular, gaze-based input methods have received attention due to their non-contact nature.

In this paper, we propose a gaze-based input method that allows users to input simple information without physical contact. Our method enables users to enter information such as simple passwords and mouse clicks effectively, even in environmental factors where they cannot use their hands. Moreover, our proposed method is effective in solving security problems such as shoulder surfing attacks.

The contribution of this paper is twofold. Firstly, we demonstrate the relationship between the gaze input box, gaze time, and average input time. Secondly, we present experimental results on the size range of the gaze input box and gaze time, which show conditions of 100% accuracy.

The remainder of this paper is organized as follows: Section 2 provides an overview of related works, Section 3 introduces our proposed gaze input method, Section 4 presents the development and performance evaluation, and finally, Section 5 concludes the paper with remarks and future directions.

## 2. Related Works

We introduce various input method studies and points to supplement related studies.

### 2.1 Input Method Using Body Part

The input method using the body commonly inputs a number-based pattern using physical contact with the body parts. Commonly, body-based input methods require physical contact with body parts to enter numeric patterns. For instance, a user can enter a numeric password from 1 to 9 using a single touch, as shown in Figure 1 (Ju & Seo, 2012). The patterns in Figure 1 correspond to the numbers 5 and 7, respectively. By combining single touch patterns to form complex passwords, users can make it challenging for attackers to guess or force the password. The user has a total of 54 different numeric patterns available, with a 4-digit password requiring a total of 8,503,056 possible combinations.

Recent studies have explored multi-touch technology for password input methods, which can recognize several touch points when a user interacts with a touch screen or touchpad (Ju & Seo, 2012). This technology supplements the limitation of existing touch technology, which can only recognize one touch point. In a 4-digit password, all input values must be entered in multiple dimensions, rather than one-dimensional input. Entered values are classified using parentheses, and list types are used instead of character types to enhance security.

However, inputting passwords through physical contact poses risks in the current pandemic era, such as COVID-19, as users could be exposed to viruses that remain on the screen. Additionally, physical contact methods are vulnerable to shoulder surfing attacks (Lim, Kim, & Kim, 2012), where attackers use fingerprints or traces to indirectly or directly identify a user's password. Therefore, many researchers have recently studied non-contact password input methods, such as gaze, gesture, and face recognition, as an alternative to physical contact.

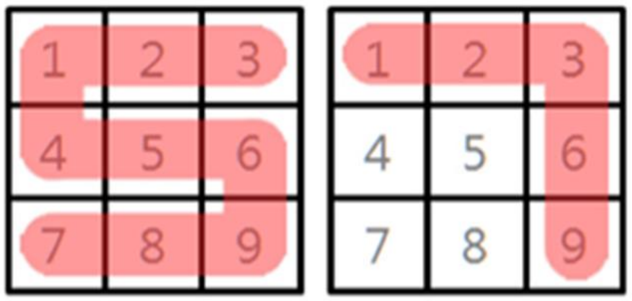

**Figure 1:** Numeric Password System through Pattern

### 2.2 Input Method Using Eye

A proposed technique for inputting a four-digit decimal password through a user's blink was introduced (Lee, 2022). The Haar cascade classifier (Puttemans, 2021) provided by OpenCV, an image processing library, was utilized to detect the user's eyes. The proposed method employed gaze-tracking technology to recognize the user's blink in real-time and input the password.

The number of input methods using blinks can be largely divided into three categories: when both eyes are opened, when only one eye is closed, and when both eyes are closed. If only one eye is closed, it can be further classified into two categories, such as when only the right eye is closed and when only the left eye is closed. When entering a number with only one eye closed, it is used as a method of determining which number to enter in a specific place.

The blink password input method allows the system to distinguish between numbers by the sounds it produces without a separate input interface. To enter a single number, the user can blink in the number section of their choice in the number section from 0 to 9, which is then divided by sound. The user can repeat this process four times to enter a four-digit password.

Some researchers conducted a study on the PIN number input method based on gaze-tracking technology. They modified Gaze Gestures, a password input method based on eye-tracking (De Luca, Weiss, & Drewes, 2007). When the user makes each gesture for a number using their gaze, a corresponding PIN number is entered for each gaze gesture.

When inputting a gaze gesture, a specific button should be pressed, and the next gesture can then be input. This method compensates for the shortcomings of previous studies as it continuously analyzes the user's gaze. However, there is a limitation that eye-tracking technology using blinking can cause problems such as Midas' hand (Jacob, 1995), which generates unnecessary input regardless of the user's intention. The gaze gesture also has a limitation in that it is only input when the user presses the button and requires physical contact.

## 2.3 Gaze Tracking Technology

Gaze-based technology has been utilized in various applications, and the development of gaze-tracking machines has continued to evolve up to the present (Seo, 2016). One of the representative open-source libraries used for facial recognition and gaze tracking is Dlib (Heo, Kim, & Lee, 2021; Boyko, Basystiuk, & Shakhovska, 2018), which is based on the C++ language and includes various machine learning algorithms for facial recognition.

The gaze-tracking method via the Dlib library operates based on Facial Landmark Estimation (FLE) (Park, 2019). As shown in Figure 2, FLE extracts 68 characteristic descriptors from the user's face based on the brightness value of the pixel, estimates the eyes, nose, and mouth, and connects the eye coordinates of the feature points in a vertical and horizontal direction to determine the intersection point as the pupil. However, if the gaze moves, the position of the pupil might not be in the center, making it inappropriate to judge solely based on the eye shape resulting from FLE.

WebGazer (Papoutsaki, Sangkloy, Laskey, Daskalova, Huang, & Hays, 2016), is one of the gaze-tracking technologies that use artificial intelligence technology to learn and approximate gaze through user interaction. It combines computer vision, image processing, and machine learning technologies to track users' gaze movements through webcams.

WebGazer goes beyond the capabilities of Dlib technology and adds several assumptions to improve performance. Dlib simply assumes that pupils will be at the vertical and horizontal junctions of the eye contour, but WebGazer assumes that after detecting the eye area, the pupil is in the center of the eye, the iris is darker than the surrounding area, and finally, the iris is circular.

Although the three assumptions of WebGazer are not always accurate, they persist long enough to obtain reliable accuracy from real-time results. When recognizing gaze in WebGazer, it may fail to locate the pupil in the actual appearance of the eye. To resolve this issue, the WebGazer model changed the image of each eye to 6×10 and included TurkerGazer (Martin, 2012).

## 3. Proposed Gaze Input Method

The overall flow of the proposed system is as follows. It recognizes the user's face through the user's webcam and extracts the user's gaze. The user then moves his gaze to enter the information. If the information input through the gaze is wrong, the input information is initialized and returns to the part where the gaze is input.

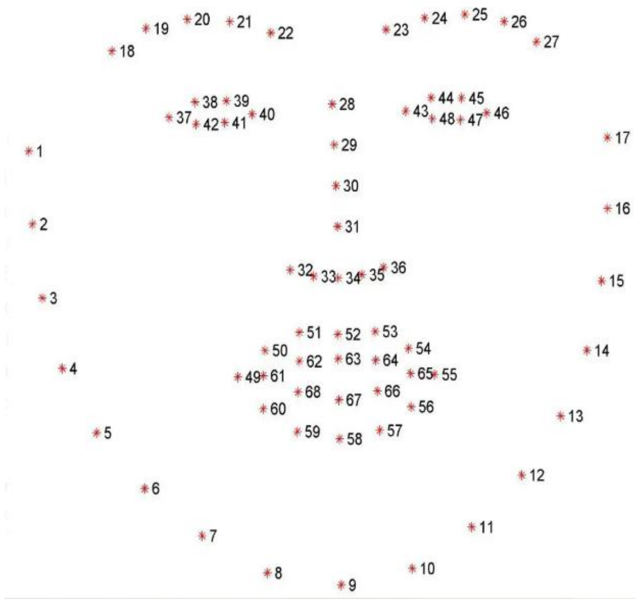

**Figure 2:** Facial Landmarks

## 3.1 Extraction of User Gaze

In this paper, we use the WebGazer Js-object Detect library (Mathias, 2014) to track the user's gaze to detect pupil and user's facial shape as shown in Figure 3.

After detecting the face shape, it was designed to input the user's pupil coordinates into the WebGazer model in milliseconds. As a result, the system outputs the predicted gaze coordinates in real time.

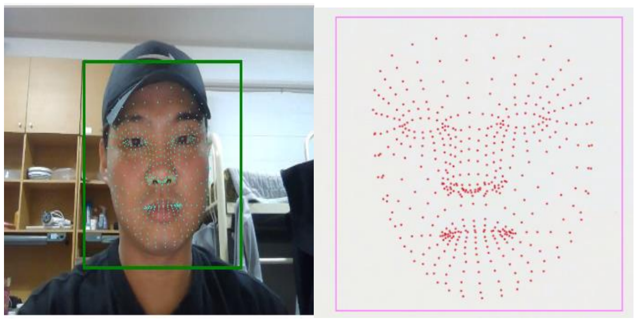
**Figure 3:** Face recognition result

### 3.2 Input of User Gaze

In this section, as shown in the flowchart of Figure 4, the gaze input method is described assuming that the user inputs password information through the gaze.

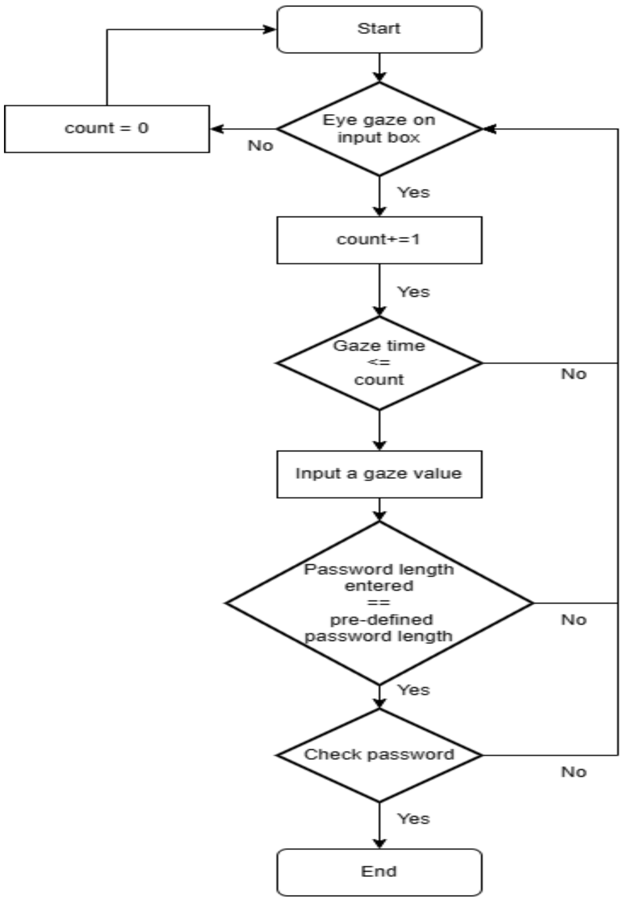

**Figure 4:** Gaze Input Algorithm

Since the user's gaze does not stay at the gaze extraction point but continues to move, it is very difficult to keep the gaze fixed at one point. Therefore, the gaze input box was virtually set on the computer screen, and the time the gaze stayed in the input box was counted and accepted as input.

When the user stares at the gaze input box and the gaze value is recognized, the count increases in milliseconds. If the gaze value is not recognized in the gaze input box and the count time is 0, the count of the system does not increase and only stops at the step of extracting the gaze.

If the user's gaze recognition count value is equal to the preset gaze time, the value of the corresponding gaze input box in the gaze is input. If the count time is less than the gaze time, such as when the gaze is directed to another place in the middle, the value of the gaze input box that remained in the user's gaze is not input and the count is initialized.

After completing the user gaze input, if the entered password length matches the user's pre-specified password length, proceed to password verification, and if the length does not match, go back to entering the password.

## 4. Development and Performance Evaluation

### 4.1 Development Environments

As shown in Table 1, the user's gaze input algorithm was implemented in JavaScript, a webcam was used to recognize the user's face, and the type of webcam was a laptop's default webcam. The interface through which the user inputs gaze and confirms input is implemented in HTML and CSS.

**Table 1:** Development environments

| | |
|---|---|
| **OS** | Windows 11 |
| **CPU** | Intel(R) Core(TM) i5-1135G7 |
| **RAM** | 16GB |
| **Camera** | HD UVC WebCam |
| **IDE** | Visual Studio Code |
| **Language** | HTML5, CSS3, JavaScript |

### 4.2 Gaze Input Interface

Figure 5 shows the gaze input interface implemented in this paper. It was implemented by arranging gaze input boxes for users on the top, bottom, left, and right sides of the experimental screen. The user may input a combination of four input boxes using a gaze.

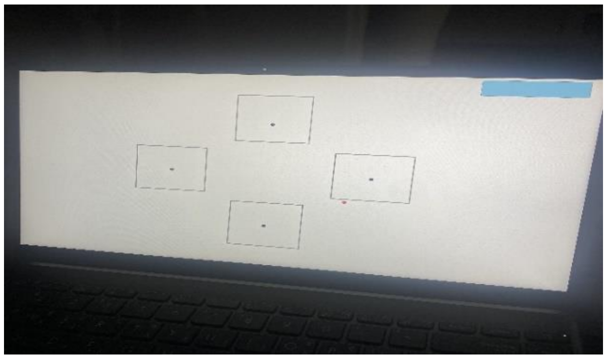

**Figure 5:** Gaze input interface

As shown in Figure 5, it is implemented so that the user can check whether the gaze is recognized on the upper right side of the screen. When a user enters information using gaze, only the number of gazes recognized by * mark can be checked so that people other than the user cannot check the information entered by the user.

### 4.2.1 The size of the gaze input box

In this section, we conducted experiments to determine the minimum and maximum size of the gaze input box that can accurately receive the user's gaze. The experiments were performed on a window screen with a size of 1920 X 1080 by varying the size of the gaze input box from 40 to 300 pixels, as shown in Table 1. The examination time was fixed at 1 and 1.5 seconds, and four consecutive random gaze inputs were assumed to be one input. We repeated each input ten times and obtained the experimental accuracy.

**Table 2:** Input accuracy according to the size of the gaze input box

| Box Size | Gaze Time | accuracy |
|---|---|---|
| 40pixels | 1.0seconds | 60% |
| 40pixels | 1.5seconds | 60% |
| 70pixels | 1.0seconds | 60% |
| 70pixels | 1.5seconds | 70% |
| 100pixels | 1.0seconds | 60% |
| 100pixels | 1.5seconds | 60% |
| 120pixels | 1.0seconds | 80% |
| 120pixels | 1.5seconds | 70% |
| 150pixels | 1.0seconds | 100% |
| 150pixels | 1.5seconds | 100% |
| 200pixels | 1.0seconds | 100% |
| 200pixels | 1.5seconds | 100% |
| 250pixels | 1.0seconds | 100% |
| 250pixels | 1.5seconds | 100% |

Table 2 shows that when the width and height of the gaze input box are less than 150 pixels each, the gaze extraction point shakes, leading to an error in the user's desired information input. However, if the size of the gaze input box is too large, it may result in input to two adjacent gaze input boxes when it exceeds 250 pixels on a 1920 X 1080 screen, which is about 23% of the ratio.

We recommend selecting a large gaze input box to prevent double input, but it should not exceed 150 pixels unless necessary. 13-inch Tablets and 20-inch monitors yielded the same results as 15-inch laptop monitors. In summary, the optimal size of the gaze input box should be selected based on the screen size and the accuracy required for the task.

### 4.2.2 Gaze Time

In the gaze time experiments, presented in Table 3, we varied the gaze time between 0.6 and 2.0 seconds, while testing the input accuracy assuming that four consecutive gaze inputs constitute one input. To achieve the experimental accuracy, we fixed the gaze input box to 150, 200, and 250 pixels. As the input time becomes too long, we excluded gaze times exceeding 2 seconds.

Our results demonstrate that gaze times of 0.6 and 0.8 seconds are insufficient for accurate information input, with an input accuracy that is less than 100%. On the other hand, stable and reliable experimental results were achieved between 1.0 and 2.0 seconds of gaze time. However, we observed that the user's gaze movement sometimes causes inaccuracies, which can be improved by setting a longer test time for the experiment.

**Table 3:** Accuracy with gaze time changes

| Box Size | Gaze Time | accuracy |
|---|---|---|
| 150pixels | 0.6seconds | 90% |
| 150pixels | 0.8seconds | 100% |
| 150pixels | 1.0seconds | 100% |
| 150pixels | 1.5seconds | 100% |
| 150pixels | 2.0seconds | 100% |
| 200pixels | 0.6seconds | 70% |
| 200pixels | 0.8seconds | 90% |
| 200pixels | 1.0seconds | 100% |
| 200pixels | 1.5seconds | 100% |
| 200pixels | 2.0seconds | 100% |
| 250pixels | 0.6seconds | 60% |
| 250pixels | 0.8seconds | 90% |
| 250pixels | 1.0seconds | 100% |
| 250pixels | 1.5seconds | 100% |
| 250pixels | 2.0seconds | 100% |

### 4.2.3 Average Input Time

Table 4 assumes that four consecutive random gaze inputs are one input and shows the average time required for 10 inputs. Analysis of the gaze time seems to be a natural result, but the shorter the gaze time when the user looks at the gaze input box, the less the average time it takes.

**Table 4:** Average gaze input time

| Box Size | Gaze Time | Average Time |
|---|---|---|
| 150pixels | 1.0seconds | 12.085seconds |
| 150pixels | 1.5seconds | 15.786seconds |
| 150pixels | 2.0seconds | 17.864seconds |
| 200pixels | 1.0seconds | 5.928seconds |
| 200pixels | 1.5seconds | 8.858seconds |
| 200pixels | 2.0seconds | 11.043seconds |
| 250pixels | 1.0seconds | 5.529seconds |
| 250pixels | 1.5seconds | 6.735seconds |
| 250pixels | 2.0seconds | 9.258seconds |

Assuming that gaze input is accurate, the average input time decreases as the size of the gaze input box increases. This is because the movement time between gaze input boxes is reduced. However, in many cases, the user's gaze extraction point on the screen may be different from the user's intention due to the characteristics of the human gaze. While this may not be an issue when entering a single piece of information, it is recommended to increase the size of the gaze input box when entering multiple pieces of information. Therefore, an interface should be established to allow the size of the gaze input box to be changed according to the user's needs when entering information using gaze tracking.

Our proposed gaze input method has the advantage of allowing the user to input information using only their gaze without any physical activity. Additionally, the gaze input method is immune to shoulder surfing attacks, and even if an attacker observes the user's gaze, it is difficult to determine the input value. It can also be used as a web surfing mouse function, allowing users to perform four frequently used functions: up and down scrolling, and back/forwarding.

Our gaze input method can be optimized for each user by flexibly modifying the size of the box, gaze time, and number of input information. The size of the gaze input box should be at least 150 pixels and should be less than about 23% of the short side of the monitor for tablets or desktop monitors. Larger input boxes lead to less error or input time. In terms of gaze time, it can be used for more than 1 second, but should be adjustable depending on the user's available state.

A more precise gaze-tracking input system could be developed if WebGazer learns the characteristics of the human gaze that continue to fluctuate, even when we believe we are fixated on a particular point.

## 5. Conlcusion

In conclusion, this paper proposed a novel non-contact input method using intelligent gaze-tracking technology. Our method allows users to enter information through gaze without the need for physical contact, making it an ideal solution for individuals who face restrictions in using their bodies. Furthermore, our method can prevent security threats like shoulder surfing attacks, making it a reliable solution in securing sensitive information.

The main contribution of this paper lies in identifying the optimal size of the gaze input box and the gaze time required to achieve 100% accuracy in inputting information. Our experimental results demonstrate that our proposed approach is both reliable and practical.

Moving forward, we plan to improve our system by developing algorithms that can correct unintended swaying gaze to enhance the accuracy of our approach. This would further improve the usability of our method, enabling users to input a broader range of information conveniently.

## References

Boyko, N., Basystiuk, O., & Shakhovska, N. (2018, August). Performance evaluation and comparison of software for face recognition, based on dlib and opencv library. In *2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP),* 478-482.

De Luca, A., Weiss, R., & Drewes, H. (2007, November). Evaluation of eye-gaze interaction methods for security enhanced PIN-entry. In *Proceedings of the 19th australasian conference on computer-human interaction: Entertaining user interfaces,* 199-202.

Heo, S. Y., Kim, K. M., & Lee, W. J. (2021). Design and Implementation of Visitor Access Control System using Deep learning Face Recognition. *Journal of digital convergence, 19(2),* 245-251.

Jacob, R. J. (1995). Eye tracking in advanced interface design. Virtual environments and advanced interface design, 258, 288.

Ju, S. H., & Seo, H. S. (2011). Password based user authentication methodology using multi-input on multi-touch environment. *Journal of the Korea Society For Simulation*, 20(1), 39-49.

Ju, S. H., & Seo, H. S. (2012). A study on User Authentication Technology of Numeric based Pattern Password. *Journal of the Korea society of computer and information*, 17(9), 65-73.

Kang, M., Kim, B., Seo, J., Kim, K. (2021). A Study on the Feasibility of IoT and AI-based elderly care system application. *Korean Journal of Artificial Intelligence,*9(2), 15-21.

Kim, S. Y., & Kwon, T. (2016). A Case Study of Password Usage for Domestic Users. *Journal of the Korea Institute of Information Security & Cryptology*, 26(4), 961-972.

Lee, S. H. (2022). An Input Method for Decimal Password Based on Eyeblink Patterns. *Journal of the Korea Institute of Information and Communication Engineering*, 26(5), 656-661.

Lim, S. M., Kim, H. J., & Kim, S. K. (2012). Designing Password Input System Resistant on Shoulder Surfing Attack with Statistical Analysis. *Journal of the Institute of Electronics and Information Engineers*, 49(9), 215-224.

Martin, T. (2012). js-objectdetect. Retrieved Januray 13, 2023, from https://github.com/mtschirs/js-objectdetect

Mathias, A. (2014). clmtrackr. Retrieved Januray 13, 2023, from https://github.com/auduno/clmtrackr

Papoutsaki, A., Laskey, J., & Huang, J. (2017, March). Searchgazer: Webcam eye tracking for remote studies of web search. In *Proceedings of the 2017 conference on conference human information interaction and retrieval,* 17-26.

Park, K. N. (2019). A morphing method using control lines of facial landmarks. *The Journal of Digital Contents Society*, 20(2), 443-450.

Puttemans, S. (2021). Object Detection using Haar cascades. Retrieved Januray 12, 2023, from https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html

Seo, E. S. (2016). mobile eye tracker and for use of the same for revitalizing studies on eye tracking. *The Journal of the Korea Contents Association*, 16(12), 10-18.