Blink Detection Based PIN Verification

[1]Vibha S Navale, [2]Suman R Kulkarni, [3]T Kishore

[1]Bachelor of Engineering
[1]Information Science and Engineering,
[1]RNS Institute of Technology, Bengaluru, India

***Abstract*—Blink Detection Based PIN verification is carried out because of the susceptible attacks that might be caused due to an authorized user entering a PIN in public places making it easy for attacks such as keyboard eavesdropping, thermal tracking, or screen electromagnetic emanations. Hence, to prevent these issues, eye tracking PIN verification is a way to conserve the safety of a system, and the natural interaction method which is based on tracking how we blink our eyes provides a promising solution to the system safekeeping and useableness as this type of verification with hands-off gaze-based PIN entry techniques leave no physical footprints behind and therefore offer a more secure PIN or password entry option. First, the face and the eyes are detected, then, using the results of the detection, it is used to identify if the eye is either open or closed. The smart camera allows the processing and collection of data onboard. Physical PIN entries are made more secure by using non-contact PIN-based verification, which is predicted to minimize the vulnerability of the overall process.**

***Index Terms*—Blink Detection, Eye tracker, PIN authentication, Haar Cascade, Facial Landmark algorithm.**

I. INTRODUCTION

Due to the extreme advancements in technology in recent years, the extent of launching cyberattacks has also rapidly expanded. As a result, traditional security and privacy mechanisms must be updated globally. Personal identification numbers (PINs) are the primary means of communication for authentication of any user across different applications in today's systems. Unfortunately, conventional methods of inputting PINs utilizing an input device such as a mouse or keyboard are vulnerable to intrusions such as shoulder surfing, acoustic keyboard eavesdropping, and screen electromagnetic emanations, all of which pose a threat to the user's security and privacy. The biometric authentication system, which is slowly acquiring significant importance and popularity worldwide, is one such modern authentication method that has piqued the interest of researchers worldwide. Unfortunately, putting biometric authentication into practice on a large scale remains a difficulty. Fulfilling all these requirements is a difficult task in and of itself. Most of the criteria listed above are highly dependent on how people deal with new technology, and past research has shown that user contact with the system has a significant impact on system performance. As a result, a powerful defense system is desirable. Recent research has suggested that gaze information can be used to supplement rather than replace existing interaction approaches as a realistic type of input. Numerous researchers have investigated gaze-based techniques, which are less predictable and thus less vulnerable.

To facilitate direct interaction, eye tracking has been developed. This might open the door for the introduction of new technologies and gadgets for potential users. It is becoming a popular way of interaction. The user's chosen target on the screen might be located using gaze coordinates so that a command can be executed more quickly. We hypothesize that the limitations and disadvantages mentioned for touch screen interfaces can be improved by adding gaze capabilities. As a result, targeting may be done using gaze, and selection could be made using touch, increasing efficiency, and preventing hand obstruction during targeting. The multi-modal approach will benefit the users with a user-friendly interface and aims to make the PIN entry secure and improve the accuracy of target selection and reduce unwanted selections. To improve accuracy, user-friendliness, effectiveness, and reliability, this study will explore how the direct mode of interaction could be optimized for PIN-entry, i.e., a multimodal technique of improving the prevalent touch-based interplay employing cues and context from gaze signals.

The purpose of the paper is to use a smart camera to input and recognize gaze-based PINs utilizing real-time eye detection and tracking. On-board data processing and collecting are possible with the smart camera. Non-contact PIN-based authentication adds an extra layer of protection to physical PIN input and is supposed to make the authentication process more secure.

II. METHODOLOGY

The commercial eye tracker is used in all prior eye tracking-based authentication systems to get a precise user's viewpoint, which necessitates a calibration phase. The user must enter the password, which results in a difference in login time and error rate between the prior systems with different designs. The method for discovering the user's password is not discussed in these systems because the commercial eye tracker identifies the user's point of view. This paper aims to provide a blink-based verification system called Eye Gesture Blink Password (EGBP).

The EGBP is the first verification system that does not require the use of a commercial eye tracker and does not require calibration. For the first time in this system, a strategy is provided that can estimate a user's password sequence without having a precise point of view. One of the most significant changes between the EGBP system and earlier systems is that the angle between the points extracted as the fixation is considered. The method is inexpensive because an eye tracker is not required. A calibration step is not required for the system. It decodes passwords using eye fixations and the angle between fixations.

Basic Architecture:

The system is designed on 2 levels, client level, and management level. Here at the client level, with access to the camera, the user's face will be recorded and trained using the Haar cascade algorithm. Which then will be turned into grayscale for further facial features

analysis such as eye detection, and blink detection. We make use of Facial landmark algorithms for facial feature recognition. The below figure represents this system.

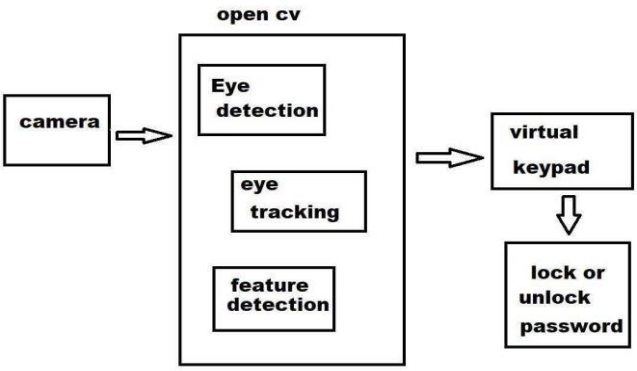

Fig. 1. Proposed System Architecture

A. Face Recognition

Face recognition can be obtained through the Haar Cascade method which detects the face. Haar cascade algorithm is the machine learning object detection algorithm used to identify objects in an image or video based on the concept of features.

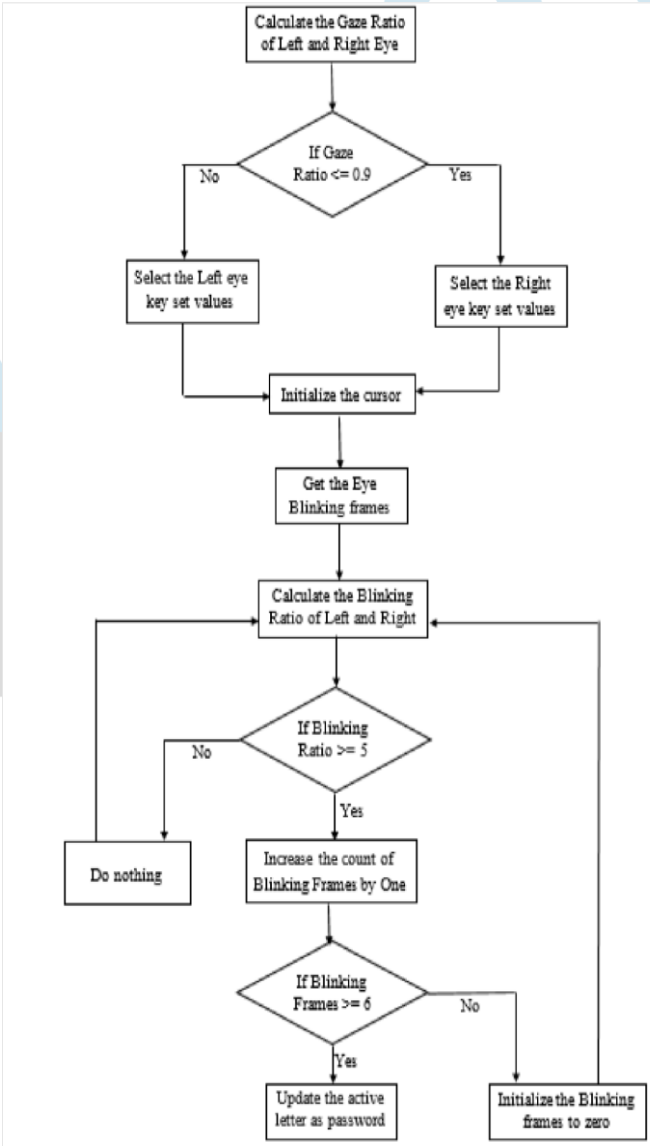

Fig. 2. Flowchart of Proposed System

Haar cascade algorithm has four steps:

1. Haar Feature Selection
2. Creating integral images

3. Ada boost training
4. Cascading Classifiers

1. Haar Feature selection:
Haar feature selection is a cascade classifier used in object recognition and classification. To train the classifier, the algorithm must first train with a large number of both positive (images of faces) and negative (images without faces) images. The feature is then derived from it. The Haar features are shown in the following figure:

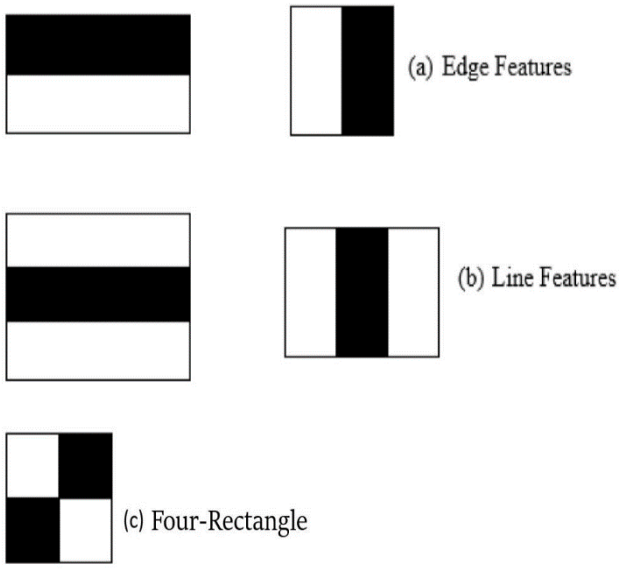

Fig. 3. Three different Haar features

Every feature is a single value that is acquired by deducting the sum of pixels under white and black rectangles from each other. Since the region around the eyes is darker than the other parts of the face, such as the cheeks or nose, the edge feature detection method is used to identify the eye region.

2. Creating Integral Images:
The images in which the pixel value at any (x, y) position is the sum of the all-pixel values present before the current pixel are called integral images. We can see what this is in the following example.

5	4	3	8	3
3	9	1	2	6
9	6	0	5	7
7	3	6	5	9
1	2	2	8	3

5	9	12	20	23
8	21	25	35	44
17	36	40	55	71
24	46	56	76	101
25	49	61	89	117

Fig. 4. Left: A regular image; Right: Its integral image

3. Ada Boost Training:
This method reduces dimensionality and potentiality while increasing execution time by eliminating the need to compute characteristics that are not known to increase the model's predictive capacity.
During this window, the specific size is moved over the image, and for each sub-section of the image, the Haar features are calculated. The difference is then compared to a learned threshold that separates non-object from objects.

4. Cascade Classifier:
It is composed of a collection of storage, where each storage is a group of weak learners. The learners are straightforward classifiers known as decision stumps. A process known as "boosting" is used to train each step. Boosting offers the possibility of training highly accurate classifiers by taking a weighted average of the decisions made by weak learners.

B. Eye Recognition

An eye detection module is used to detect the eye region in each image. This task is achieved by employing the Haar cascade algorithm.
The window location from the above module is considered to detect the key facial structure of the face and locate the facial structures with the specific (x, y) coordinates values. The coordinate values of both eyes are then considered, and a polygon is drawn around the region of the eye. Facial Landmark detector is used to achieve the above process.

Facial Landmark Algorithm:

1. Initially, we input the location of the window where the face and eye region are found.
2. The key facial features in the image are detected.
3. These features are then located with specific (x, y) coordinates.
4. The first (x, y) coordinate should begin with 1.
5. The last (x, y) coordinate should end with 68.

The polygon around the region of the eye is drawn by getting the coordinates value of the left and right eyes.

```
eye_region=np.array([(facial_landmarks.part(eye_points[36]).x,
facial_landmarks.part(eye_points[36]).y),
(facial_landmarks.part(eye_points[37]).x,facial_landmarks.part(eye_points[37]).y),
(facial_landmarks.part(eye_points[38]).x,facial_landmarks.part(eye_points[38]).y),
(facial_landmarks.part(eye_points[39]).x,facial_landmarks.part(eye_points[39]).y),
(facial_landmarks.part(eye_points[40]).x,facial_landmarks.part(eye_points[40]).y),
(facial_landmarks.part(eye_points[41]).x, facial_landmarks.part(eye_points[41]).y)],np.int32)
```

	x	y		x	y
36	403	321	42	495	320
37	415	313	43	508	311
38	430	313	44	521	311
39	443	326	45	533	316
40	430	326	46	523	321
41	415	326	47	509	322
Left eye values			Right eye values		

Fig. 5. Eye co-ordinate

C. Eye Tracking

Eye tracking is the process of detecting the position of the eye in a video frame. Tracking the eye is important for the development and areas of research such as visual systems, psychological, cognitive science, and product design. In this module, continuously the eye movement is tracked to obtain the Gaze Ratio and based on the gaze ratio the respective keyboard will be displayed. Then the eye blinking ratio will be calculated to update the respective digit as the PIN.

We can calculate the Gaze Ratio using the below formula:

Gaze Ratio of left eye = Number of white pixels on right side/Number of white pixels on left side

Gaze Ratio of right eye = Number of white pixels on right side/Number of white pixels on left side

Gaze Ratio = Gaze Ratio of left eye + Gaze Ratio of right eye/ 2

If Gaze Ratio <= 0.9 then we select the right keyboard otherwise we select the left keyboard.

The corresponding eye region is converted into a gray scale (Fig. 6) using the data from Fig. 5.

Fig. 6. Gray Scale Image

Fig. 7. The menu keyboard

Then the midpoint between the 37th and 38th coordinates is calculated and the midpoint of the 40th and 41st coordinates is calculated, and those midpoints are joined to divide the eye region into two parts.

Fig. 8. Vertical line is drawn over the image to divide the eye region

The gaze ratio of both left and right eye will be calculated and then the average of both the left and right eye is taken to obtain the Gaze ratio.

To calculate Blinking Ratio:

Blinking ratio of left eye = length of the horizontal line/ length of the vertical line

Blinking ratio of right eye = length of the horizontal line/ length of the vertical line

Blinking ratio = Blinking ratio of left eye + blinking ratio of right eye) / 2

If the Blinking ratio >=5, then we increase the blinking frames' value by one.

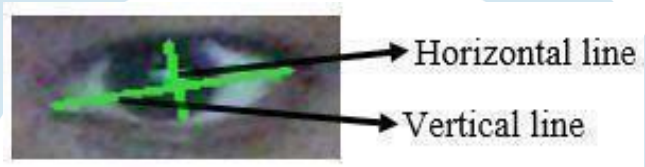

Fig.9. The horizontal and vertical lines drawn on the eye

Fig. 10. Horizontal and vertical line when the eye is blinked

III. Experimental Results

It was performed using Spyder IDE. The image frames are captured by the Webcam. Following that, the face is detected as mentioned in section A. The classifier divides the image frames into two groups, one for faces and one for non-faces. This method is used to extract the section that contains the face.

A. Datasets and Training Strategy

The datasets used in the model are the images of the user that are recorded using the user system's camera. These images are then trained using algorithms such as the Haar cascade and facial landmark algorithm for further processes.

Fig. 11. Dataset Folder

B. Face Recognition

Using the Haar cascade method after obtaining the images, we train the dataset to recognize if the user trying to access the device is the one whose images were obtained. We allow the user to turn on the camera via which the face will be recognized. The user here is asked to follow certain conditions such as having a clear background, and no other user in the frame of the camera.

Fig. 12. Face Capturing

C. *Blink Verification Test*

When the digits in the keypad appear in white and if it happens to be a digit in the user's PIN, then there must be some eye movement i.e., closing of the eyes, so that the highlight of the eye turns green, and the digit is selected.

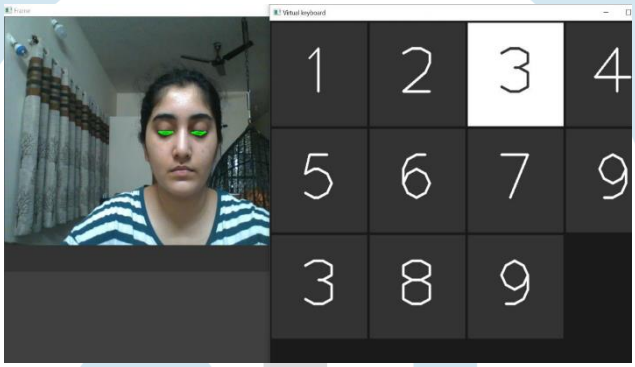

Fig. 13. Selecting digits via eye movement

D. *Password Verification*

If the PIN is verified, an audio plays which says the PIN matches and the prompt tells that the PIN is matched, and all the sensitive numbers and data are revealed. If the PIN doesn't match, the prompt says, 'PIN not matched'.

```
1
1
1
1
0
1
0
0
0
0
2
textl ['2', '5']
2 5
Enter ererer
<type 'str'>
not matched
0
```

Fig. 14. PIN not matched

IV. CONCLUSION

As a new application for gaze-based PIN recognition, a smart-camera-based eye-tracking system has been added. The system has been successfully tested with numbers and may be expanded to accept character and digit password combinations. The user password protects the user against attacks such as shoulder surfing and thermal tracking, and it can also help physically challenged people who are unable to use computers. Thus with certain algorithms that we have used, we can conclude the working of the model is accurate enough.

The future enhancement that can be done is making sure the system is even safer for military operations personnel so that small errors like forgetting your passwords and leakage of the same can be overcome by this method and can be used further for high-end security systems by making more improvements and research at a higher level where the users may be asked to sit almost 2-3 feet away from the camera.

REFERENCES

[1] V. Aharonson, V. Y. Coopoo, K. L. Govender and M. Postema, "Automatic pupil detection and gaze estimation using the vestibulo-ocular reflex in a low-cost eye-tracking setup", in SAIEE Africa Research Journal, vol. 111, no. 3, pp. 120-124, Sept. 2020, doi: 10.23919/SAIEE.2020.9142605.

[2] Khan, W.; Hussain, A.; Kuru, K.; Al-askar, H. Pupil Localisation and Eye Centre Estimation Using Machine Learning and Computer Vision. Sensors 2020, 20, 3785

[3] Ahmad Aljaafreh, Murad Alaqtash, Naeem Al-Oudat, Jafar Abukhait, and Ma'en Saleh, "A Low-cost Webcam-based Eye Tracker and Saccade Measurement System", in INTERNATIONAL JOURNAL OF CIRCUITS, SYSTEMS AND SIGNAL PROCESSING, Volume 14, 2020

[4] A. Siripitakchi, S. Phimoltares, A. Mahaweerawat, 2017, "Eye- Captcha: An Enhanced Captcha Using Eye Movement", 3rd IEEE International conference on Computer and Communications, pp. 2120 – 2126.

[5] Z. Li, M. Li, P. Mohapatra, J. Han, S. Chen, 2017, "iType: Using Eye Gaze to Enhance Typing Privacy", IEEE Infocom on Computer Communications, pp. 1-9

[6] C. Meng and X. Zhao, "Webcam-Based Eye Movement Analysis Using CNN," IEEE Access, vol. 5, pp. 19581 – 19587, 2017

[7] K. Krafka et al., "Eye tracking for everyone", in Proc. IEEE Conf. Comput. Vis. Pattern Recognition, Jun. 2016, pp. 2176_2184

[8] M. Khamis, F. Alt, M. Hassib, E.V. Zezschwitz, R. Hasholzner, A. Bulling, 2016, "GazeTouchPass: Multimodal Authentication Using Gaze and Touch on Mobile Devices", CHI Conference Extended Abstracts on Human Factors in Computing Systems, pp. 2156 -2164

[9] R. Revathy and R. Bama, 2015, "Advanced Safe PIN-Entry Against Human Shoulder Surfing," IOSR Journal of Computer Engineering (IOSR-JCE), vol 17, issue 4, ver.II, pp.9-15

[10] P. Kasprowski and K. Harezlak, "The second eye movement verification and identification competition," in Proceedings of the International Joint Conference on Biometrics (IJCB), Clearwater, FL, USA, 2014, pp 1-6

[11] D. Rozado, 2013, "Using Gaze Based Passwords as an authentication Mechanism for Password Input", 17th European Conference on Eye Movements (ECEM).

[12] G. Pan, L. Sun, Z. Wu, and S. Lao, "Eyeblink-based Anti-Spoofing in Face Recognition from a Generic Web camera", 2007 IEEE 11th International Conference on Computer Vision, 2007, pp. 1-8, doi: 10.1109/ICCV.2007.4409068

[13] P. Kasprowski, "Human identification using eye movements," Praca doktorska, Politechnika OEląska, 2004