International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 10 Issue: 03 | Mar 2023 www.irjet.net p-ISSN: 2395-0072

Remote User Authentication using blink mechanism - 'Iblink' with Machine Learning

Vaibhavi S Kuber[1], Shobha V[2], J M Chinmai Jyothy[3], Dr. Kanagavalli R[4]

[123]*Final Year Students, Department of CSE, Global Academy of Technology, Bangalore, Karnataka – 560098, India*
[4] *Associate Professor, Department of CSE, Global Academy of Technology,* Bangalore, Karnataka – 560098, India

---

Abstract - *There are many security requirements for common terminal authentication systems. One among them is that they are simple, rapid, and safe. Users must authenticate themselves using traditional knowledge-based techniques like passwords while being constantly offered with authentication strategies. However, these techniques are risky since they might be seen by unintended observers who utilize surveillance techniques like shoulder-surfing, which involves watching someone write a password, to acquire user authentication data. PINs are frequently used for user verification in a variety of applications, such as managing money at ATMs, authenticating electronic devices, unlocking mobile devices, and accessing doors. Although when PIN authentication is used, as it is in banking systems and gate access control, perfect identity verification is still challenging to accomplish. The researchers developed a three-layered security architecture that prevents shoulder surfing by requiring users to input their password by blinking their eyes at the appropriate symbols in the proper order Hands-free PIN authentication using eye blinks On the other hand, PIN input methods leave no physical traces and offer a more secure alternative to typing passwords. The three stages of this method are facial recognition, eye-blink verification, and a one-time password. By combining all of these layers, the system is set up with a secure architecture that guards against risks like thermal tracking and shoulder surfing. Our framework completely resists attacks like thermal tracking and shoulder surfing because there is no physical password entry.*

Key Words: **HAAR Cascade Classifier, Local Binary Pattern Histogram, Machine Learning, Shoulder Surfing, Password Authentication, Thermal Tracking.**

1. INTRODUCTION

People must constantly authenticate themselves using conventional knowledge-based techniques like passwords, hence one of the security requirements for typical terminal authentication systems is that they be simple, quick, and secure. Unfortunately, these techniques are risky because they could be observed by malicious watchers who utilize surveillance methods like shoulder-surfing—watching someone type a password on a keyboard—to gather user authentication information. Inadequate interactions between systems and users also cause security problems. The researchers came up with a three-layered security architecture to safeguard PIN numbers as a result. By blinking their eyes at the proper symbols in the proper order to enter the password, users make themselves immune to shoulder surfing. PINs, or personal identification numbers, are frequently employed in a number of applications, such as ATM money management, electronic transaction verification, personal device unlocking, and door access. Authentication is always required, even when PIN authentication is used, as in banking systems and gateway management an obstacle. According to European ATM Security, there were 26% more ATM fraud assaults in 2016 than there were in 2015. PIN input is subject to password assaults like shoulder surfing and thermal surveillance since an authorized user must enter the code in an open or public place. We made the decision to develop a real-time eye blink-based password authentication solution in order to counter risks from thermal tracking and shoulder surfing. To protect the user authentication system from shoulder surfing assaults. To fend off keyboard thermal tracking attempts. To offer three levels of security for user authentication. This approach makes it simpler and more secure for people who are physically challenged to access their account information. We may utilize this technique for user authentication in ATMs, Gmail, etc.

2. LITERATURE SURVEY

[1] Authors "Vani, A, Gowhar A R, Jeevika G S, Sangeetha D, and Vallabh Mahale" have implemented Real Time Eye Based Password Authentication. While determining whether an eye is open or closed, the HAAR Cascade classifier uses a technique called eye flicker identification with the help of the calculation of the histogram of situated gradients (HOG).Advantage of this system is shorter secret words because there is less likelihood of flickering, which leads to longer secret words and more security. The based on biometrics highlights Validation frameworks don't provide appropriate security; a protected and active framework is clearly needed.

[2] Authors "Sen, Udit, Vaibhav Bhawlkar, Vinay Yadav, Kavita Namdev, and Satyam Shrivastava" have implemented I SPEAK–SMART TYPING WITH BLINK. The software detects eye blinking and converts the motion into text using Computer Vision and HAAR-Cascades. This paper will be able to translate eye blinks into meaningful text and

recite the words aloud to make it easier for those who are paralysed and those who have speech and motor function problems to communicate. This is just a small project to help persons with paralysis and other communication challenges communicate better and more affordably.

[3] Authors "Rusanu, O. A., L. Cristea, M. C. Luculescu, P. A. Cotfas, and D. T. Cotfas" have implemented Virtual keyboard based on a brain-computer interface. In this research, a LabVIEW application was developed that aims to assist persons with impairments by giving them a reliable way to type using a virtual keyboard that can be used by blinking the eyes. The Divide and Conquer paradigm serve as the foundation for its operation. The switch command is connected to the transition (alternative highlighting) between rows, half rows, or keys and is represented by a single eye blink (characters). Two eye blinks are tied to the choose command and are connected to the previous discussion's alternate selection of the objects. The text field can also be used to enter a character. It lessens eves and shoulder surfing the development of a comparable mobile virtual keyboard powered by eye blinks will be the main goal of future research.

[4] Authors "Attiah, Afraa Z., and Enas F. Khairullah" have implemented Eye-Blink Detection System for Virtual Keyboard. On the basis of the well-known "68 points" of the face detection method, this is done. With this technique, a character is entered by blinking the eye, much like pressing the "Enter" key on a keyboard. The system was created with disabled individuals in mind. The outcomes demonstrate high levels of user satisfaction and demonstrate the system's value. Further improvements include adding Arabic support, making it available as a mobile application, and adding voice assistance, a night mode, shortcuts, and graphics to the virtual keyboard.

[5] Authors "Chakraborty, Partha, Dipa Roy, Z. R. Zahid, and Saifur Rahman" have implemented Eye gaze controlled virtual keyboard. The system employs the HoG descriptor by adhering to dlib in OpenCV to detect faces. The Oriented Gradients Histogram (HOG). It is based on Linear SVM and HoG characteristics. The technology proved highly effective at detecting eyes and eye blinking through glasses, and it even works for persons who are wearing eyeglasses. The suggested system in this paper gives those people who lack any other physical abilities—other than eye movement—a new dimension in their lives.

3. METHODOLOGY

The following algorithms are used:

- HAAR Cascade Classifier
- Local Binary Pattern Histogram(LBPH)

3.1 HAAR Cascade Face Detection:

The HAAR Cascade object identification method, which was introduced in the 2001 publication "Rapid Object Identification using a Boosted Cascade of Basic Characteristics," was based on characteristics created by Paul Viola and Michael Jones. A function that is a cascade classifier is developed by utilising a lot of positive and negative pictures in this machine-learning technique. Then it is used to locate objects in other images.

The Algorithm consists of 4 phases that is:

1. Selecting a HAAR feature
2. Component Picture Creation
3. Training of Adaboost
4. Cascading Classifiers

It can be trained to recognise virtually any object, even though it is best known for its capacity to recognise faces and body parts in photos. As an example, consider face detection. The method first needs a lot of positive pictures with faces and a lot of negative pictures without faces in order to train the classifier, After this the feature extraction should be done. The HAAR Features must be gathered in the first phase. An adjacent rectangular region in a detection window is taken into account by a HAAR feature, which then calculates the difference between the summation of the pixel intensities in each sector.

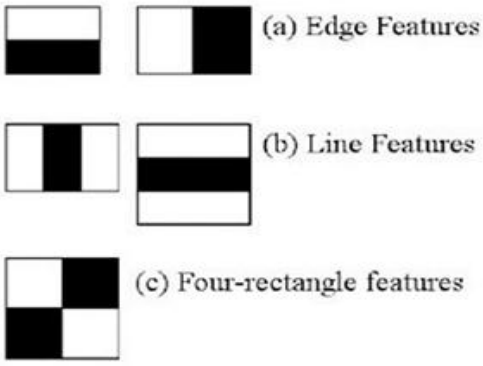


Figure 1 : HAAR features diagram

Step one is to obtain the HAAR Attributes. A HAAR feature analyses neighboring rectilinear areas in a detection window at a certain location, and it determines the discrepancy between these sums and the pixel intensities in each area. To create this, integral images were used. Nonetheless, the vast majority of the factors we calculated are unimportant. Examine the illustration below. The top row demonstrates two positive traits. It appears that the first characteristic picked is that the area around the eyes is frequently darker than the area around the forehead and cheekbones. The

second characteristic was chosen because the irises are darker than the nasal bridge. Nevertheless, applying the identical windows to the cheeks, or any other area, doesn't matter.

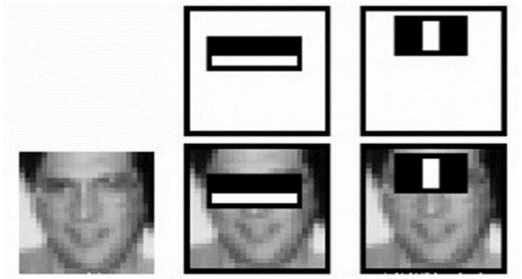


Figure 2: HAAR Characteristics diagram

Let's look at the facial recognition scenario. To train the classifier, a large number of positive photos with faces and a large number of negative images without faces are required. Then, we must infer characteristics from it.

Cascade Classifier:

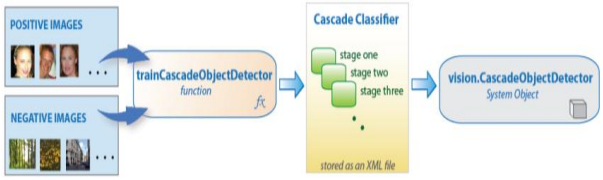


Figure 3: cascade classifier

The cascade classifier has numerous levels, and each level has a group of weak students. Decision stumps, which are simple classifiers, are the weak learners. Each stage of training uses the "boosting" method. Using a weighted average of the underachievers' classifications, A highly accurate classifier can be trained with the use of boosting. The area that the sliding window's present location designates is positively labelled or negatively by the classifier at each stage. Finding anything is indicated by a positive number, whereas finding nothing is shown by a negative number. The detector completes categorising this region if the label is false, then moves on to the next location. The region is promoted to the next level by the classifier if the label is positive. If the last step qualifies the area as positive, the detector indicates an object that was identified at the location with respect to the present window.Due to the arrangement of the phases, negative samples are not accepted as soon as practical. The supposition is that the majority of windows don't display the desired object. True positives, however, are uncommon and need to be looked into.

• False positives happen when a negative sample is incorrectly categorized as positive, and false negatives happen when a positive sample is incorrectly categorized as negative.

• A true positive occurs when a positive sample is accurately detected.

A low rate of false negatives, the cascade must perform well for each phase. If a stage wrongly classifies an object as negative and you are unable to fix the error, the classification process is interrupted. Yet, every step has a sizable risk of false positives. Although a nonobject is mistakenly marked as affirmative via the detector, the error can still be corrected. With an increase in the total false positive rate and total real positive rate decrease as the number of steps increases.

3.2 Local Binary Pattern Histogram (LBPH):

This algorithm was proposed in the year 2006. It can be discovered on a local binary operator that the user has access to. Due to its strong discriminative powers and computational simplicity, it is commonly employed in facial recognition.

To do this, the following procedures must be taken:

1. Creating a dataset
2. Acquiring faces
3. Extracting features
4. Classifying the data.

An element of OpenCV is the LBPH algorithm.

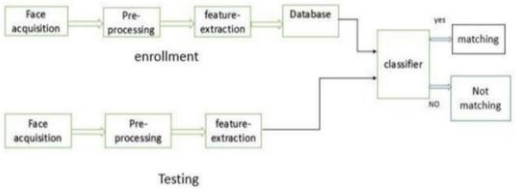


Figure 4: Local binary histogram pattern process flow diagram

Let's say an image with N x M dimensions is considered. It is divided into many regions of the same width and height, giving each zone a dimension of m x m.

Figure 5 : m x n dimensioned image of face

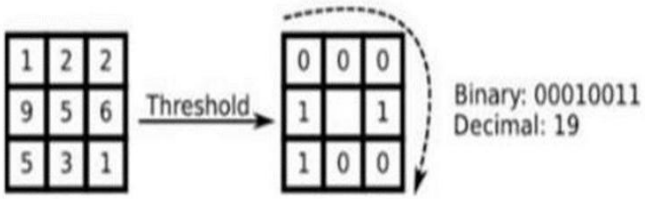
For each location, a local binary operator is employed. The LBP operator is specified in a 3 by 3 window.

$$LBP(x_c, y_c) = \sum_{p=0}^{P-1} 2^p s(i_p - i_c)$$

The intensity of the center pixel (Xc,Yc) in this image is "Ic," the intensity of the neighboring pixel is "In." This function compares a pixel to its eight nearest neighbors using the median pixel value as the threshold.

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

The neighbor value will be set to 1 if it is greater than or equal to the central value, else it is put to 0. It is set to 0 if not.



Using threshold, the m x n block's decimal representation is changed to binary. Subsequently, it was shown that a fixed neighborhood cannot encode information with scale variations. The algorithm was enhanced to incorporate various radius and neighbor counts, and it is now referred to as LBP.

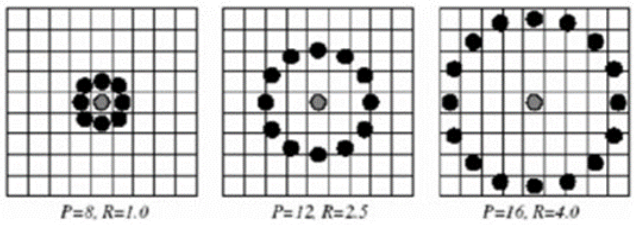


Figure 6: Better algorithm for neighbors and various radius values

The goal is to arrange any number of neighbors in a straight line on a circle of customizable radius.

The following neighborhoods are thus identified:

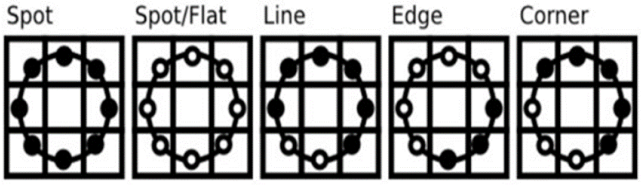


Figure 7 : Assigning arbitrary values to neighbors .

(Xc, Yc) are the given pair of coordinates, the neighboring pair position (Xp, Yp), which belongs to P, and is determined as follows: where R is the radius of circle and P is the count of sample points.

$$x_p = x_c + R\cos(\frac{2\pi p}{P})$$

$$y_p = y_c - R\sin(\frac{2\pi p}{P})$$

If a point's location on the circle does not match the image's coordinates, it is often interpolated using bilinear interpolation:

$$f(x,y) \approx \begin{bmatrix} 1-x & x \end{bmatrix} \begin{bmatrix} f(0,0) & f(0,1) \\ f(1,0) & f(1,1) \end{bmatrix} \begin{bmatrix} 1-y \\ y \end{bmatrix}$$

After an histogram of each region is extracted, all these histograms are concatenated to create a new and bigger histogram which represents the image which is known as a feature vector of image. The monotonic grey scale modifications cannot harm the LBP operator.

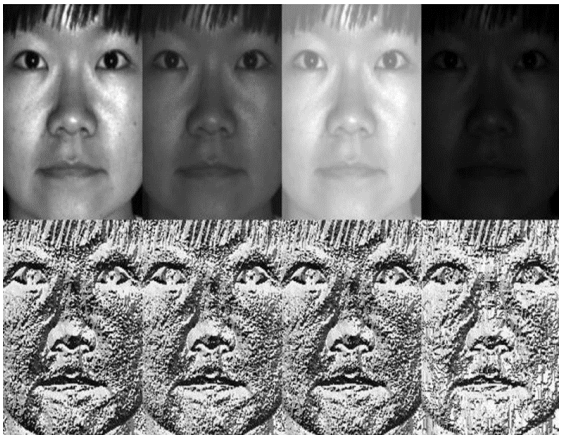


Figure 8 : Transformation of grayscale

Following the production of LBP values, the region's histogram is constructed by tallying the number of nearby LBP values that are similar.

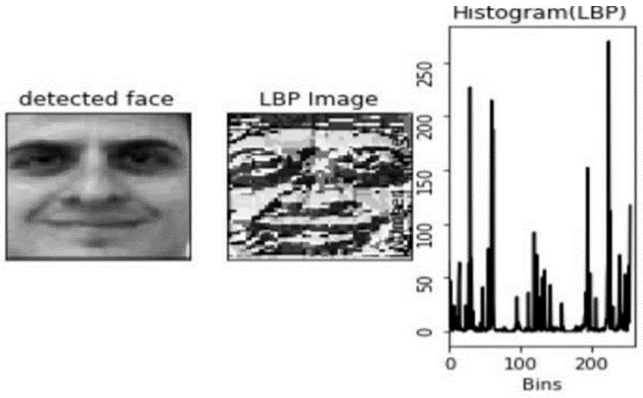


Figure 9: The creation of histograms

Then, the image with histogram that is closest to the histogram is returned after comparing the histograms of the testing image and the database's images. (Several methods can be used for this. To name a few, Euclidean distance, chi-square etc.)

By contrasting the testing image's features with those from the dataset, the Euclidean distance is found out. The matching rate is found by the smallest distance between the original and images.

$$d(a,b) = \sqrt{\sum_{i=1}^{n} |a_i - b_i|^2}$$

An ID of the image from the database is given as output only if recognition of the test image takes place.

LBPH module can recognize the front part and side part of faces, and it is unaffected with changes in light set up, making it most robust.

4. DESIGN AND DEVELOPMENT PROCESS

1. System Architecture:

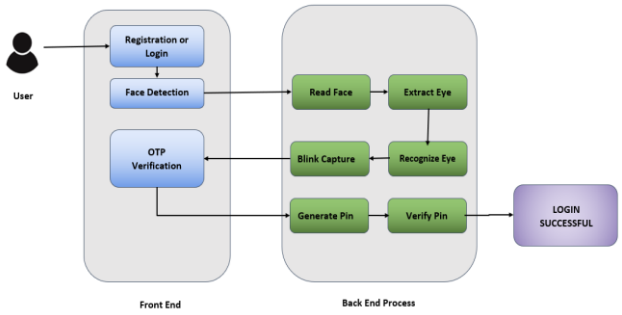
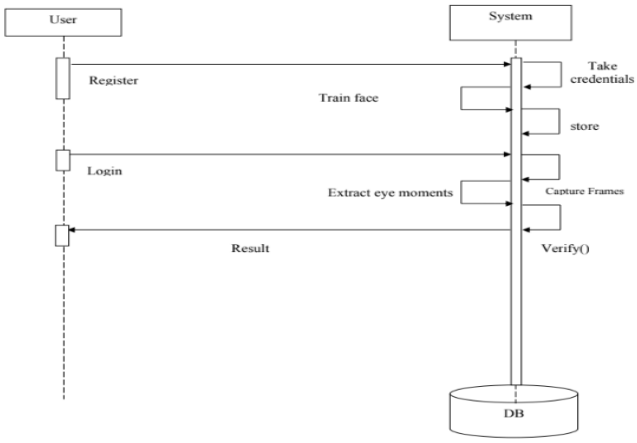


Figure 10 : System Architecture

A system's conceptual model, also referred to as its system architecture, determines the system's structure, behavior, and other attributes. With respect to our project, a user interacts with the front end of the project first. The user registers on the page in the first step if they do not already have an account; otherwise, they sign in to their account. After logging in, the user's face is detected using the LBPH approach. The OTP is verified when the face has been found. In this, a random one-time password made up of numbers is generated by the system and sent to the registered mobile number. The user is successfully signed in to the page once this OTP has been successfully entered.

2. Sequence Diagram:



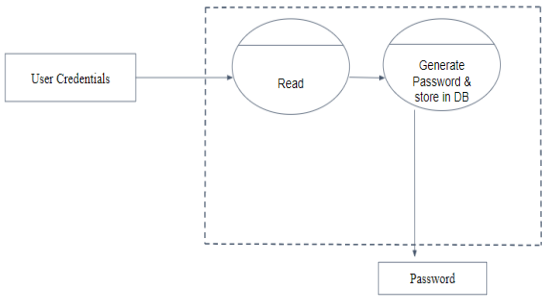
The model's structure, behavior, and process are described in the sequence diagram. With our project, a user first engages with the project's front end. The user checks in to his account in the first step if an account has already been created and registers on the page otherwise. After logging in, the user's face is detected using the local Binary Pattern Histogram (LBPH) approach. It will keep the user's credentials in the database after the user's face is recognized and it captures the frames to extract eye moments, which aids in subsequent logins. Verification is completed, and a conclusion is drawn.

3. Data Flow Diagrams

Level 0:

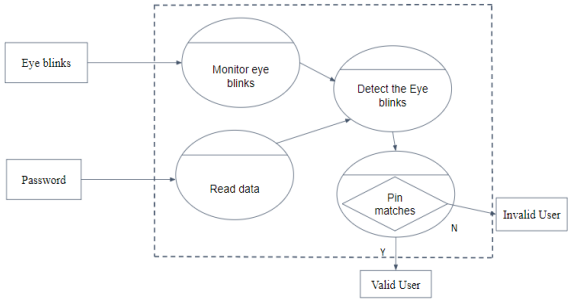
The project's overall process is described in Level 0. Users' eye blinks are being used as input. Here, the protection of user account information from shoulder surfing assaults is done. For this process, the system will use OpenCV.

Level 1:



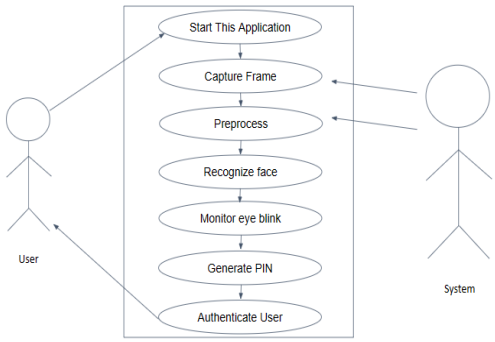
Then project's first step is described in Level 1. The system will read the user's credentials as input and in turn, this will generate a password.

Level 2:



The project's final step is described at Level: 2. Users' eye blinks and a randomly generated password are used as input. The system will watch the user's eye blink and verify their identity.

4. Use Case Diagram:



According to the use-case diagram, after the user launches the programme, the system will take a picture of the user's eyes and preprocess it. The computer will recognize the user's face, keep track of eye blinks, and then generate a pin by blinking a set of characters. The user is authenticated and can check whether they are authorized to log in at the final stage of verification.

5. RESULTS AND DISCUSSION

The three stages of the authentication layer, such as facial recognition, eye blink detection, and OTP generation, At the end results should be more precise and efficient, after consulting the few papers that are stated in the literature review.

6. CONCLUSION

Using a camera-based eye-blinking mechanism and the aforementioned methodologies, a special application for eyelid blink-based PIN detection has been developed. A nine-digit keypad has been used to successfully test the system and it may be made to accept password combinations that combine characters and numbers. The user's eye blink stability has an impact on the accuracy of the pins identified, therefore this must be considered.

Nowadays, real-time eye-blink computations and recording are completed before PIN identification.

ACKNOWLEDGEMENT

We would like to thank our HOD Dr.Kumaraswamy S, and Guide Dr. Kanagavalli R, Dept. of Computer Science and Engineering, Global Academy of technology, Bangalore for their guidance and support.

REFERENCES

[1] DVani, A, Gowhar A R, Jeevika G S, Sangeetha D, and Vallabh Mahale. 2022. "Real Time Eye Based Password Authentication". *Perspectives in Communication, Embedded-Systems and Signal-Processing - PiCES*, June, 20-22. http://pices-journal.com/ojs/index.php/pices/article/view/356

[2] Sen, Udit, Vaibhav Bhawlkar, Vinay Yadav, Kavita Namdev, and Satyam Shrivastava. "I SPEAK–SMART TYPING WITH BLINK."

[3] Attiah, Afraa Z., and Enas F. Khairullah. "Eye-Blink Detection System for Virtual Keyboard." In *2021 National Computing Colleges Conference (NCCC)*, pp. 1-6. IEEE, 2021.

[4] Rusanu, O. A., L. Cristea, M. C. Luculescu, P. A. Cotfas, and D. T. Cotfas. "Virtual keyboard based on a brain-computer interface." In *IOP Conference Series: Materials Science and Engineering*, vol. 514, no. 1, p. 012020. IOP Publishing, 2019.

[5] Chakraborty, Partha, Dipa Roy, Z. R. Zahid, and Saifur Rahman. "Eye gaze controlled virtual keyboard." *Int. J. Rec. Technol. Eng.(IJRTE)* 8, no. 4 (2019): 3264-3269.