International Journal of Technical Research and Applications e-ISSN: 2320-8163,
www.ijtra.com Volume 8, Issue 3 (MAY-JUNE 2020), PP. 06-09

AUTOMATIC PAYMENT USING FACE RECOGNITION

Mihir Sase[1], Mehul Shah[2], Neha Mugloo[3], Adarsh Shinde[4], Prof. Priyanka Gulhane[5],
Department Of Information Technology
[1,2,3,4,5] MIT College of Engineering, Maharashtra
Pune, India

***Abstract*— In this paper, we present a system that automates payment using face recognition. It uses eye blink detection as its prime method. The system keeps a track on the duration of user's eye blink. Once the blink is detected, the system captures the face from the frame, which can be used as a template for face recognition. Once the face has been recognized, the money is debited from his account (wallet) . The system is more secure and convenient for users than the traditional methods of authentication. It can effectively block spoofing attempts using other people's still images or pre-recorded videos as there are no offline templates required for the functioning of the system.**

***Index Terms*— Face recognition, eye blink, payment, authentication, spoofing.**

I. INTRODUCTION

"Automatic Payment using Face Recognition" uses deep learning (face recognition and eye blink detection techniques) so that customers can make their payments by having their faces scanned. Security is a primary concern of every individual and when it comes to making online transactions, everyone wants it to be secure and fast. However, common approaches to entering passwords through keyboard, mouse, touch screen or any traditional input device, are frequently vulnerable to attacks such as password snooping and shoulder surfing. The proposed system requires the user to be present in person and does not work if any pre-recorded videos are used for blink detection, thus making the system secure and efficient. Eye blink detection is useful in monitoring a human operator vigilance, e.g. driver drowsiness [1,2] in systems that ease communication for disabled people [3] or for anti-spoofing protection in face recognition systems [4,5].

The general architecture of the system is comprised up of two parts:

a) Android app (shown in figure 1),

b) Face recognition and Blink detector (shown in figure 2).

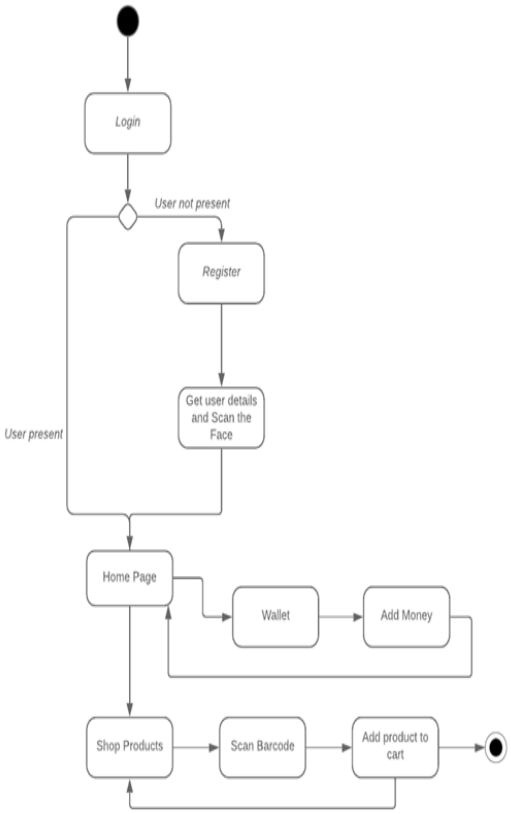

Fig. 1: Android app architecture

II. RELATED WORK DONE

S.No.	Reference Paper	Method proposed	Problems faced
1.	[5]	Eye blink detection using facial landmarks and Eye Aspect ratio.	Slightly sensitive to lighting conditions.
2.	[6]	This paper addresses the problem of Face Alignment for a single image.	Uses 5-point facial landmark detector . Not effective for detecting eye blink.
3.	[7]	Uses motion detection and template matching.	Time taken to compare concurrent frames for motion detection is more.
4.	[8]	Used for object detection in real-time.	Not effective for detecting tilted faces and sensitive to lighting conditions.

III. METHOD

The user has to download the app and register on the app using email ID, phone number and scan his face for authentication process. User can add money to the app wallet which will be used to debit the cart amount during checkout. When user visits the store he has to scan the product barcode to add the product to his cart using inbuilt barcode scanner.

Once this is done, the user goes for blink detection. As soon as the blink is detected, user's face at that instance is captured for face recognition. Once the captured face matches with the face(stored at the time of registration), the payment is considered successful.

A. Blink Detection

To check whether the blink is detected or not, we'll use a method called as the **Eye Aspect Ratio (EAR)**. The eye aspect ratio involves a simple calculation based on the ratio of distances between facial landmarks of the eyes. This method for eye blink detection is fast and easy to implement as compared to the other proposed methods. Each eye is represented by 6 *(x, y)*-coordinates, starting at the left-corner of the eye (assume the person is in front of you), and then going clockwise around the remaining eye (as shown in Figure 3) [5]

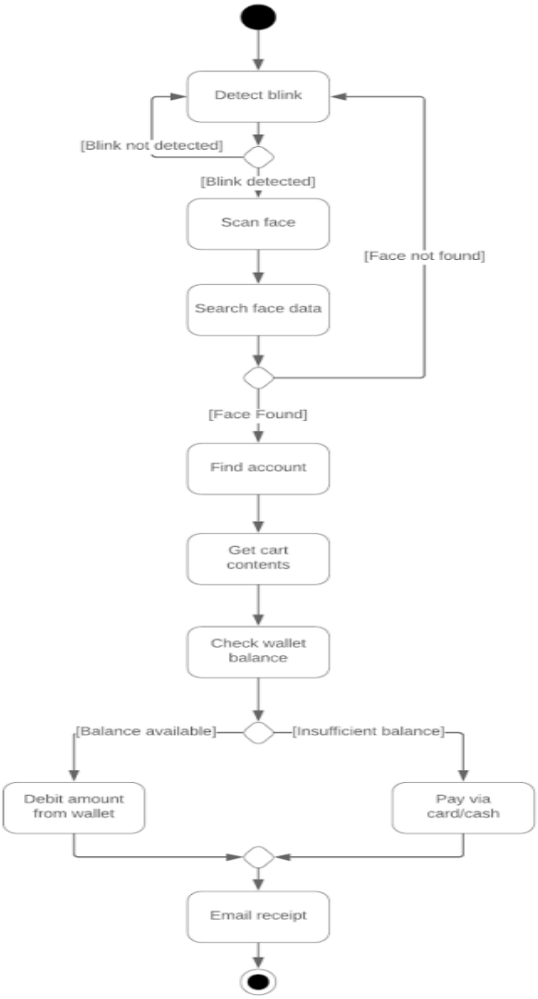

Fig. 2 : Blink detector system

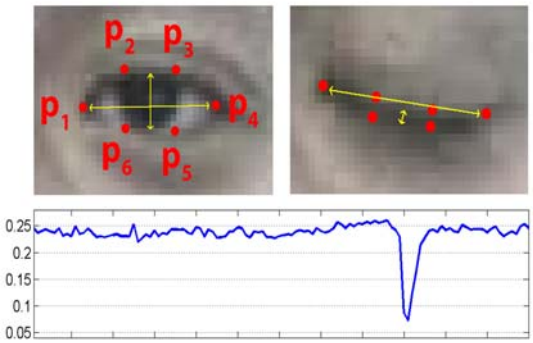

Fig.3: Open and closed eyes with landmarks. The eye aspect ratio EAR in Eq. (1) plotted for several frames of a video sequence. A single blink is present. Source: [5]

The EAR is calculated as:

$$\frac{\{\|p2-p6\|+\|p3-p5\|\}}{2\|p1-p4\|} \quad (1)\ [5]$$

p1, p4 are the horizontal landmarks and p2,p3,p5,p6 are the vertical landmarks.

The numerator of the equation computes the distance between the vertical eye landmarks while the denominator computes the distance between horizontal eye landmarks. The eye aspect ratio is approximately constant while the eye is open, but will rapidly fall to zero when a blink takes place.(Figure 3)

B. Face Detection

As soon as the blink is detected, the frame is captured and saved and is used for comparing with the faces stored in the database at the time of registration. We can apply facial landmark detection to localize important regions of the face, including eyes, eyebrows, nose, ears, jaw and mouth. The pre-trained facial landmark detector inside the dlib library is used to estimate the location of **68 (x,y)-coordinates** that map to facial structures on the face.(Figure 4) [6]

Fig. 4: Facial landmarks as estimated by the dlib face detector

The facial detector in dlib is based on HOG (Histogram of Oriented Gradient) features and SVM (support vector machine). HOG is a feature descriptor that is used to extract features from image data. The first step is to convert the RGB image to grayscale. While moving from pixel to pixel in an image, there comes a point when a pixel becomes dark or light as compared to its surrounding pixels. This sudden change is known as *gradient*. The complete image is broken down into smaller regions and for each region, the gradients are calculated. Once we get the gradients, we calculate the *gradient magnitude* and *gradient angle* for each of the pixels.

Fig. 5: Visualization of HoG features

The HOG generates a Histogram for each of these regions separately. The histograms are created using the **gradients** (gradient magnitude) and **orientations** (gradient angle which shows the direction of flow from light to dark regions) of the pixel values, hence the name '**Histogram of Oriented Gradients**'. This facial detector is the fastest method on CPU, works very well for frontal and slightly non-frontal faces and also works under small occlusion. It is a light-weight model as compared to other facial detectors. One major demerit is that it does not detect small faces as it is trained for minimum face size of 80×80. Thus, you need to make sure that the face size should be more than that. Also, it does not work for side face and extreme non-frontal faces, like looking up or down.

C. Face Recognition

Once the facial features have been extracted, next step is face recognition. Face Recognition here is carried out using face_recognition library in Python. The methods used are as follows:

- The face_encodings (face_image, known_face_locations=None) returns the multi-dimensional face encoding for each face in the image. The representation is an array with 128 floating point elements (i.e. a face vector).
- The compare_faces (known_face_encodings, face_encoding_to_check, tolerance) compares a list of face encodings against a candidate encoding to see if they match.
- The distance between two face encodings is determined by calculating the Euclidean distance of the given face encoding.
- The face recognition result depends on the **tolerance value** (the distance between faces to consider it a match)
- The tolerance value set here is 0.48. The faces are considered 'a match' if the Euclidean distance between the face vectors is 0.48 or less.

The face image captured has to be compared with all the registered persons' face (one-to-many matching) in the database(Figure 6). Once the faces match, the money is debited from the person's wallet and payment is considered successful.

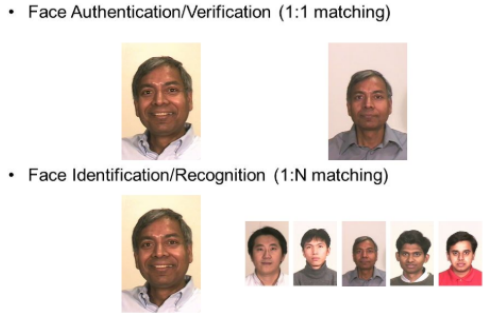

Fig. 6: One to many matching. [10]

IV. CONCLUSIONS AND FUTURE SCOPE

A system that automates payment using face recognition was presented. The method used facial landmarks to estimate the user's EAR. The system requires proper lighting conditions and can be beneficial for handicapped people with limited or no mobility of hands. Also, one need not remember passwords or various credit/debit card pins. With the recent trends in technology, the system can be useful in various supermarkets, shopping malls, etc. where payment is involved. . The proposed system performs face recognition for 130 images in 0.13 seconds, thereby making it quick and efficient. This calls for lesser employees at the store checkout center and prevents the customers from waiting in long queues and making the payment by a simple face scan.

REFERENCES

[1] Taner & Bilasco, Ioan Marius & Djeraba, Chaabane & Ihaddadene, "Drowsy driver detection system using eye blink patterns." 2010 International Conference on Machine and Web Intelligence, ICMWI ,2010

[2] Sahayadhas, Arun & Sundaraj, Kenneth & M, Murugappan."Detecting Driver Drowsiness Based on Sensors: A Review." Sensors Basel, Switzerland,2012

[3] D. Torricelli, M. Goffredo, S. Conforto, and M. Schmid. "An adaptive blink detector to initialize and update a view-based remote eye gaze tracking system in a natural scenario." Pattern Recogn. Lett., 30(12):1144–1150, Sept. 2009.

[4] G. Pan, L. Sun, Z. Wu, and S. Lao. "Eyeblink-based anti-spoofing in face recognition from a generic web camera." In ICCV, 2007.

[5] Soukupová and Čech, " Real-Time Eye Blink Detection Using Facial Landmarks".21st Computer Vision Winter Workshop,2016

[6] Vahid Kazemi and Josephine Sullivan, "One Millisecond Face Alignment with an Ensemble of Regression Trees". KTH, Royal Institute of Technology Computer Vision and Active Perception Lab Teknikringen 14, Stockholm, Sweden,2015.

[7] M. Chau and M. Betke "Real time eye tracking and blink detection with USB cameras". Computer Science Department Boston University Boston, MA 02215, USA, 2005

[8] P. Viola and M. Jones, "Robust real-time face detection," Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001, Vancouver, BC, Canada, 2001

[9] Noman Md. Talal Bin, Ahad Md. Atiqur Rahman,"Mobile-Based Eye-Blink Detection Performance Analysis on Android Platform",Frontiers in ICT,Vol. 5,2018

[10] https://medium.com/@krsatyam1996/face-recognition-using-openface-92f02045ca2a

[11] D. T., A. R. B., T. Sree Sharmila and J. S. Jennifer "Determining Response Credibility by Blink Count," International Conference on Recent Trends in Advance Computing (ICRTAC), Chennai, India, 2018.

[12] J. S. Jennifer and T. S. Sharmila, "Edge based eye-blink detection for computer vision syndrome," 2017 International Conference on Computer, Communication and Signal Processing (ICCCSP), Chennai, 2017.

[13] A. G. Mavely, J. E. Judith, P. A. Sahal and S. A. Kuruvilla, "Eye gaze tracking based driver monitoring system," IEEE International Conference on Circuits and Systems (ICCS), Thiruvananthapuram, 2017.