



# International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

*(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)*



**Impact Factor: 8.699**

**Volume 14, Issue 4, April 2025**

# Eye Blink Detection

**Aryapratap Santosh Satpute, Jambhale Adinath Santosh, Tushar Ananda Masavekar, Sudarshan  
Sadashiv Farakate, Shreyash Satapa Patil**

Students of Computer Engineering Department, K.P.Patil Institute of Technology, Mudal, Maharashtra, India

**S. S. Patil**

Lecturer in Computer Engineering Department, K. P. Patil Institute of Technology (Polytechnic), Mudal, Mudal,  
Maharashtra, India

**ABSTRACT:** The online Examination portal is a web-based application for technical evaluation. The online examination portal not only replaces paperwork but also releases the workload of faculty. Most of the e-examination (Online Examinations) have fixed number of questions without randomizations, they have pool scalability. Online examination system is increasing rapidly with the change in scenario due to the pandemic, also it seems to be an easy, flexible, and secure methodology to carry the examinations effectively.

**KEYWORDS:** online exam," "e-exam," "digital assessment," "remote proctoring," "online testing," and "automated grading".

## I. CONTENT

### 1. Introduction:

The Eye Blink Detection System is an innovative application designed to monitor and detect the state of a person's eyes—specifically, whether they are open or closed—using a webcam. This system is particularly useful in various scenarios, such as enhancing user safety during prolonged computer use, assisting individuals with certain medical conditions, or even serving as a tool for research in human-computer interaction.

The application captures a continuous video feed from the webcam, allowing for real-time analysis of eye movements. This ensures that the system can promptly detect when the eyes are closed for an extended period. Upon detecting that the eyes remain closed for a predetermined duration, the system triggers an alert, which can be both audio and visual, ensuring that the user is made aware of their eye state. This feature is crucial in helping to prevent fatigue or potential health issues associated with prolonged eye closure.

To achieve accurate detection, the system employs advanced algorithms to analyse facial landmarks, specifically focusing on the eye region. By accurately identifying the position of the eyes, the application can effectively calculate the Eye Aspect Ratio (EAR), a critical metric used to determine whether the eyes are open or closed.

Designed with user experience in mind, the application features a graphical user interface (GUI) that displays the video feed, current status updates, and control buttons, making it accessible for users of all technical backgrounds. The Eye Blink Detection System leverages a combination of powerful libraries and technologies, including OpenCV for video processing, Dlib for facial landmark detection, and Pyttsx3 for text-to-speech notifications, resulting in a robust and efficient system capable of performing complex tasks with ease.

The Eye Blink Detection System is an innovative application designed to monitor and detect the state of a person's eyes—specifically, whether they are open or closed—using a webcam. This system is particularly useful in various scenarios, such as enhancing user safety during prolonged computer use, assisting individuals with certain medical conditions, or serving as a tool for research in human-computer interaction. By providing real-time monitoring, the application aims to address issues related to eye fatigue and drowsiness, which can have significant implications for health and productivity.

## 2. Related Work:

The Eye Blink Detection System addresses several critical issues related to eye health and user safety in environments where prolonged screen time is common. As individuals increasingly engage with digital devices for work, entertainment, and communication, the risk of eye strain, fatigue, and related health problems has escalated. The inability to monitor eye states effectively can lead to negative consequences, including decreased productivity, impaired focus, and long-term vision issues.

One of the primary challenges is the difficulty in recognizing when a person's eyes are closed for an extended period, which can occur during moments of drowsiness or distraction. Traditional methods of monitoring eye states, such as manual observation or relying on user self-reporting, are often ineffective and prone to error. This lack of real-time feedback can result in users remaining unaware of their eye condition, leading to potential health risks such as dry eyes, headaches, and reduced cognitive performance.

Additionally, there is a growing need for automated systems that can provide timely alerts to users when they exhibit signs of fatigue or drowsiness. Current solutions may not offer the necessary responsiveness or accuracy, making it essential to develop a reliable system that can detect eye closure and provide immediate feedback. The Eye Blink Detection System aims to fill this gap by utilizing advanced computer vision techniques to monitor eye states continuously and alert users when necessary.

In summary, the problem definition for the Eye Blink Detection System encompasses the need for an effective, real-time monitoring solution that can detect eye closure, provide alerts to prevent fatigue-related issues, and ultimately promote better eye health and user safety in an increasingly digital world. By addressing these challenges, the system seeks to enhance user experience and well-being in various settings, from personal computing to professional environments.

In summary, the problem definition for the Eye Blink Detection System encompasses the need for an effective, real-time monitoring solution that can detect eye closure, provide alerts to prevent fatigue-related issues, and ultimately promote better eye health and user safety in an increasingly digital world. By addressing these challenges, the system seeks to enhance user experience and well-being in various settings, from personal computing to professional environments.

## 3. Applications of Eye Blink Detection

### 3.1 Driver drowsiness detection

- As the name suggests, this application is very useful in building real-world applications like detecting whether the driver is sleepy or not while detecting the eye moment or blink.



**Fig. 3.1 Driver Drowsiness Detection**



### 3.2 Iris tracking

- This is another use case while we can also track the iris movement for building an AR kind of a9

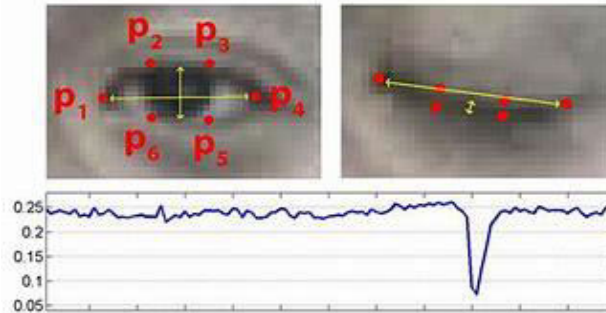


Fig. 3.2 Iris Tracking

### 3.3 Virtual gaming

- We are in the age of virtual reality evolution, and by far, mostly VR-powered games are either hand or body movement driven, but we can also build eye movement driven games.



Fig. 3.3 Virtual Gaming

- Eye blink sensors can be used to control games, providing a more immersive experience. For example, a player can blink to shoot or jump in a game.

### 3.4 Accessibility

- For individuals with disabilities, eye blink sensors can be used to control assistive devices, such as wheelchairs or computers.

## II. METHODOLOGY

- **Introduction to Methodology**

Overview of the research design and methodology. Importance of the chosen methods in achieving research objectives.

- **Data Collection**

Description of the data sources used for eye blink detection. Methods of data acquisition (e.g., video recordings, sensors).

- **Pre-processing of Data**

Steps taken to clean and prepare the data for analysis. Techniques used for noise reduction and normalization.

- **Feature Extraction**  
Identification of key features relevant to eye blink detection. Methods used for extracting features from the data (e.g., image processing techniques).
- **Model Selection**  
Criteria for selecting the appropriate models for detection. Overview of algorithms considered (e.g., machine learning models).
- **Training the Model**  
Description of the training process for the selected model. Details on the training dataset and validation techniques.
- **Testing and Validation**  
Methods used to test the model's performance. Metrics for evaluating accuracy and reliability (e.g., precision, recall).
- **Implementation**  
Steps taken to implement the eye blink detection system. Tools and technologies used in the implementation phase.
- **Real-time Processing**  
Techniques for enabling real-time detection of eye blinks. Challenges faced in achieving real-time performance.
- **User Interface Design**  
Overview of the user interface developed for the system. Considerations for user experience and accessibility.

### 1. Experimental results:-

The implementation of the proposed system faced several areas of difficulty before achieving the following advantages:

1. Identifying face and eye location.
2. Detecting blinks of the eyes with glasses or without.
3. Being able to detect the information of eye (open or closed) in the real-time environment.

Performance evaluation is very crucial and tedious task and at the end of the development process it is significant to show that the final proposed system achieves an acceptable level of performance and that it represents a significant improvement over existing systems or not. So, we test our proposed system with two environments: the offline and the online.

The System is developed by a MATLAB R2012a. This implementation is running on Windows 7 32bit operating system with an Intel® core2TM Duo CPU (2.20GHz) and 3GB of RAM. The camera in the proposed system is HP WebCam101 that built in camera embedded in the used laptop. Webcam with video size equal 640 x 480. For evaluating our proposed system, we select the two International Journal of Computer Applications (0975 – 8887) Volume \*– No.\*, \_\_\_\_\_ 2013 5 measurements are usually used in the evaluation process "Recall" and "Precision". Recall: is the average detection rate, which is calculated by the equation (1):

$$\text{Recall} = \frac{TP}{TP+FN} \quad (1)$$

Where TP is the number of true positives (implies correct detection of eye-state means open eye returned as open eye) and FN is the number of false negative (closed eye returned as open eye).

Precision: is calculated by the equation (2):

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$

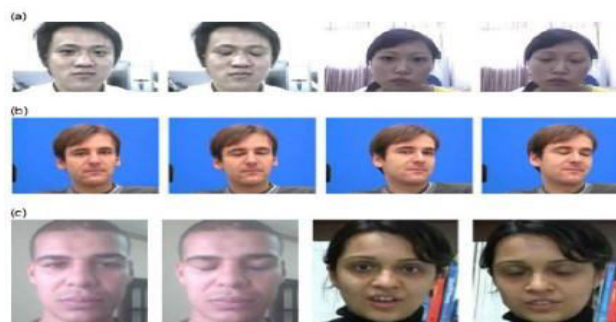
Where TP is the number of true positives (implies correct detection of eye-state) and FP is the number of false positives (implies an open eye returned as closed eye). In order to evaluate the performance of the proposed system several experiments have been designed. These experiments are divided into two sets: Offline tests, and Real time tests.

**2. The Offline Tests**

The proposed system was tested on people with glasses or without glasses. First, we start the testing process with people without glasses. Second, for testing process with people who is wearing glasses.

4.1.1 The testing process with people without glasses:

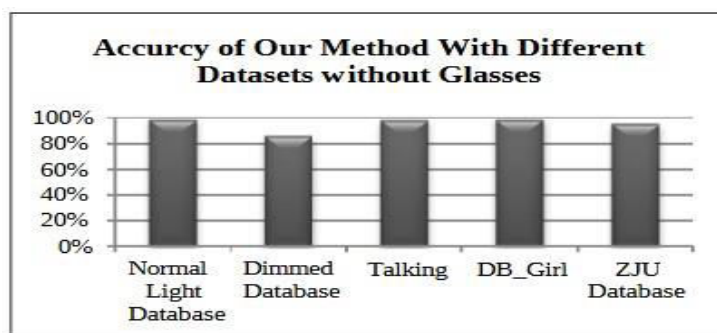
In the testing process five different databases for people who not wearing glasses are used. The first two databases are our own database, which are captured from laptop webcam and the spatial resolution is 640×480 pixels. These two databases are for different people. Their ages are between 25 and 35. The number of frames in the first video is 120 frames. The number of frames in the second video is 192 frames. The third database is the Talking Face video database ([http://www.prima.inrialpes.fr/FGnet/data/01TalkingFace/talking\\_face.html](http://www.prima.inrialpes.fr/FGnet/data/01TalkingFace/talking_face.html) .Last Visit: July, 2014) which contains one video having 999 frames captured from one Caucasian male whose age is estimated as 20s or 30s. It includes 61 blinks in all. The fourth database is self-collected dataset, for female with age around 25 years, which contains video having 149 frames. The fifth database is the ZJU Eye blink Database ([http://www.cs.zju.edu.cn/~gpan/database/db\\_blink.html](http://www.cs.zju.edu.cn/~gpan/database/db_blink.html) . Last Visit: July, 2014) which comprises 20 short videos having 2950 frames (320 × 240 pixels) captured from 20 Chinese people (13 males and 7 females, dark brown eye colors, ages are estimated as 20s) at a capturing speed of 30frames/s. They include 255 blinks at all. Examples of databases that used in the proposed system are shown in Figure 12.



**Fig 12:** Examples of detecting eye blink on different data sets. (a) Some cases from used database in the ZJU Database. (b) Talking Database. (c) Our own Database.

This part begins by testing the accuracy of the proposed system to know the total accuracy that achieved by the proposed system. The result is shown in Figure 13 based on the equation (3):

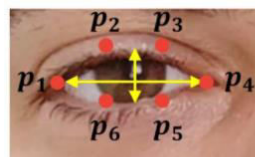
$$\text{Accuracy} = \frac{\text{Number of correcting case}}{\text{corrected} + \text{incorrected}}$$



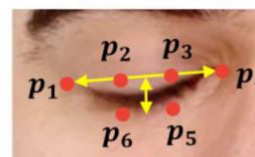
**Fig 13:** Accuracy of The Proposed Method

This experiment demonstrates the performance of the proposed system under different illumination conditions. The Table I lists the result of "Recall" and "Precision" that obtained using the different databases: ZJU, Talking, and our own databases. The results are varied from dataset to another as a result of the resolution of the camera and the strength of the ambient lighting. In the dataset "Our Own Database1" the web cam with low resolution was used. The result of the recall was 94% and the result of precision was 100%. In the dataset "Our Own Database2" the web cam with low resolution was used. In this dataset the ambient lighting decreased so, the images became darker than the first dataset "Our Own Database1", and the result of recall became 66% and the result of precision became 99%. With the dataset "DB Girl" give result to recall was 100% and with precision 99% because the resolution of cam better than our web cam and the lighting conditions more better. The result of recall and precision are 96% and 96%, respectively, with "Talking" data set. The result of recall and precision are 95% and 98%, respectively, with "ZJU Database" data set. The proposed system achieved an average Recall of 90.5% and Precision of 98%.

### 3. Function to Calculate Eye Aspect Ratio (EAR)



Open eye will have  
more EAR



Closed eye will  
have less EAR

`def eye_aspect_ratio(eye):`

```
A = dist.euclidean(eye[1], eye[5])
B = dist.euclidean(eye[2], eye[4])
C = dist.euclidean(eye[0], eye[3])
```

```
ear = (A + B) / (2.0 * C)
return ear
```

#### Code breakdown:

- 1) First, we will get the **Euclidean distance** between the 2 coordinates of the eyes.
- 2) While grabbing the coordinates, we will first have the **vertical eye landmarks**.
- 3) Then we will have the **horizontal eye landmarks** using the same algorithm.
- 4) After grabbing the coordinates, we will calculate the **Eye Aspect ratio**.
- 5) Then, at last, we will **return the EAR**.

- **Main Logic**

```
while True:
```

```
if fileStream and not vs.more():
    break
```

```
frame = vs.read()
frame = imutils.resize(frame, width = 450)
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

```
rects = detector(gray, 0)
```

```
for rect in rects:
```

```
    shape = predictor(gray, rect)
    shape = face_utils.shape_to_np(shape)
```

```
leftEye = shape[lStart:lEnd]
rightEye = shape[rStart:rEnd]
left EAR = eye_aspect_ratio(left Eye)
right EAR = eye_aspect_ratio(right Eye)

ear = (left EAR + right EAR) / 2.0

leftEyeHull = cv2.convexHull(left Eye)
rightEyeHull = cv2.convexHull(right Eye)

cv2.drawContours(frame, [leftEyeHull], -1, (0, 255, 0), 1)
cv2.drawContours(frame, [rightEyeHull], -1, (0, 255, 0), 1)

if ear < EYE_AR_THRESH:
    COUNTER += 1

else:
    if COUNTER >= EYE_AR_CONSEC_FRAMES:
        TOTAL += 1
    #Reset the eye frame counter

COUNTER = 0

cv2.putText(frame, "Blinks: {}".format(TOTAL), (10, 30), cv2.FONT_HERSHEY_COMPLEX, 0.7, (0, 0, 255), 2)
cv2.putText(frame, "EAR: {:.2f}".format(ear), (300, 30), cv2.FONT_HERSHEY_COMPLEX, 0.7, (0, 0, 255), 2)

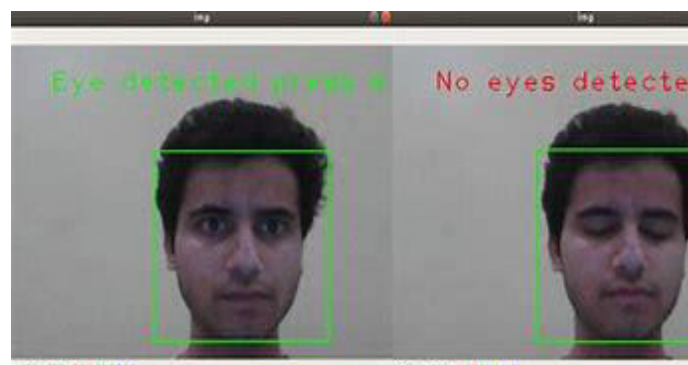
cv2.imshow("Frame", frame)
key = cv2.waitKey(12) & 0xFF

if key == ord("q"):
    break

cv2.destroyAllWindows()
vs.stop()
```

• **OUTPUTS:-**

1)





2)



### III. CONCLUSION

Eye Blank Detection System using OpenCV is a useful tool for real-time detection of eye blinks from a video stream. It utilizes facial landmark detection and EAR computation to detect eye blinks by setting a blink threshold and tracking consecutive frames where the eyes are closed. Eye tracking has proven itself to be valuable in many research areas. This study discussed the eye tracking techniques and applications with a focus on modern approaches such as ML, IoT, and cloud computing.

A classification of different ML algorithms that use eye tracking data along with the research goal and the strengths and limitations of the algorithm are presented. Firstly, we saw the real-world application of **blink detection application** then we saw what we would be doing in a nutshell. The main key takeaway from this article is to **segment the eyes** by using their coordinates. We have also learned about the concept of **Euclidean distance** and its formulae using the specific library. Along with that, we also came across the concept of **Eye Aspect Ratio (EAR)**, which is the soul of this application. We also learned how **the dlib** library could detect the landmarks of the face and, along with that, read the video files as well as live streaming. In addition, we summarize different lists that cover eye tracking applications and techniques along with the algorithms.

### REFERENCES

- [1]C. Chou, W. Peng and Y. Hung. 2008. Real-Time Three-Stage Eye Feature Extraction. Technical Report, National Taiwan University, Taipei, Taiwan.
- [2]B. Ashtiani and I.S. MacKenzie. 2010, BlinkWrite2 an improved text entry method using eye blinks. In Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications, ETRA'10, pp.339–345, New York, NY, USA, ACM.
- [3]M.Barrett, H.Skovsgaard, and J.SanAgustin 2009. Performance evaluation of a Low-Cost gaze tracker for eye typing. In Proceedings of the 5<sup>TH</sup> Conference on Communication by Gaze Interaction, COGAIN'09, pp.13–17.
- [4]M.Porta, A.Ravarelli, and G.Spagnoli. 2010. courser, a contextual eye cursor for general pointing in windows environments. In Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications, ETRA '10, pp.331–337, New York, NY, USA.
- [5]O. Tuisku, P.Majaranta, P. Isokoski, and K. Rähkä. 2008. Now dasher! dash away! : longitudinal study of fast text entry by eye gaze. In Proceedings of the 2008 Symposium on Eye-Tracking Research & Applications, ETRA'08, pp.19–26, New York, NY, USA. ACM. ACMID:1344476.



INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  [ijirset@gmail.com](mailto:ijirset@gmail.com)



[www.ijirset.com](http://www.ijirset.com)

Scan to save the contact details