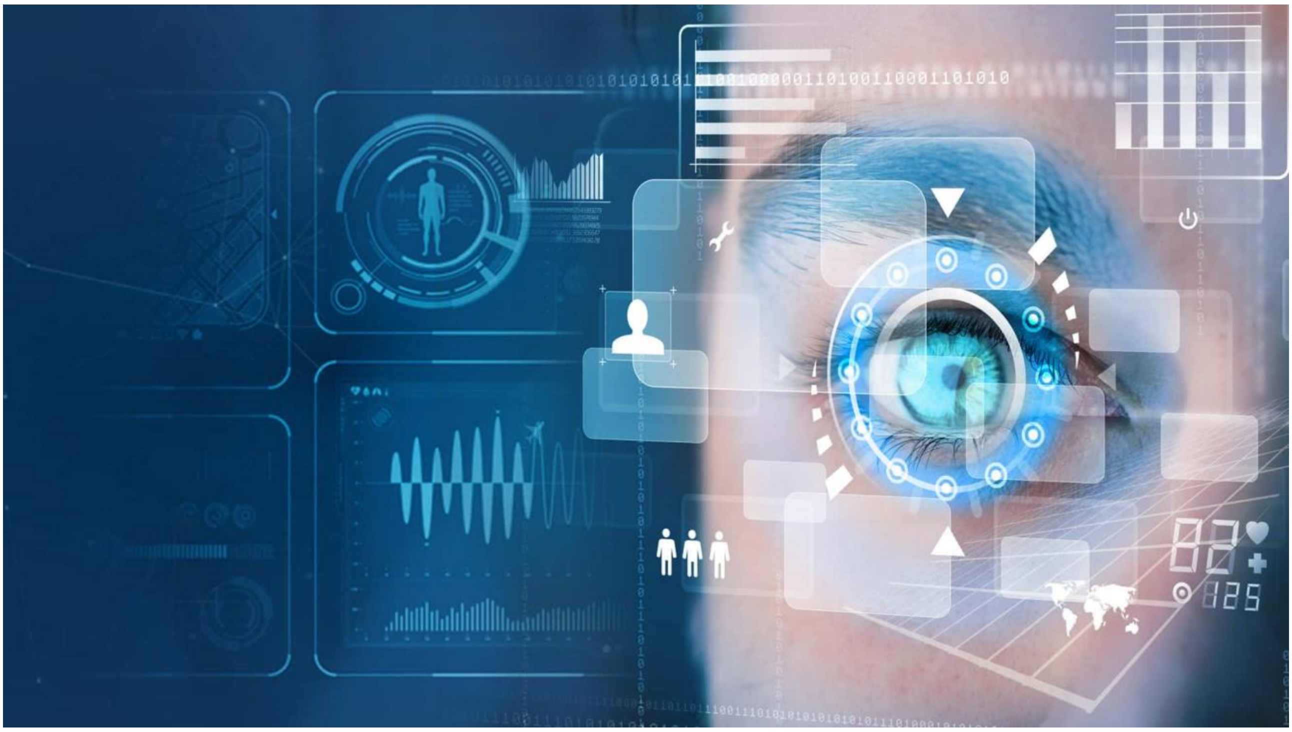
Smart Internz

Strain Analysis based on Eye Blinking



Team Members:

N. Sai Charan (20481A5437) (Team lead)

R. Sai Nithin (20481A5448)

T. Sita Venkata Sathwika (20481A5455)

1. INTRODUCTION

1.1 Overview: -

"Strain Analysis based on Eye Blinking" is a machine learning project focused on using eye blink patterns to detect and analyse strain and fatigue levels in individuals. Strain refers to the physical or mental stress experienced by a person due to extended periods of concentration, fatigue, or demanding tasks.

The project aims to develop an automated system that can non-intrusively monitor an individual's eye blinking behaviour and analyse it to provide insights into their level of strain or fatigue. The core assumption is that eye blinking patterns can reflect a person's cognitive state and stress levels, making it a valuable indicator in various domains like driving, aviation, healthcare, and more.

A neural network model is built which alerts the user if eyes are getting strained. This model uses the integrated webcam to capture the face (eyes) of the person. It captures the eye movement and counts the number of times a person blinks. If blink count deviates from the average value (if the number of blinks is less or more), then an alert is initiated by playing an audio message along with a popup message is displayed on the screen appropriately.

In the European Community more than 40% of the today's working population use computers in their daily work. Computer use is related with static work, constrained sitting and vision problems. For example, approximately 70% of computer workers worldwide are reported to having vision problems leading towards Computer Vision Syndrome. The number of computer-related jobs is expected to increase significantly in the next decade, along with the number of workplace-related illnesses.

One of primary causes of vision problems during computer use is insufficient eye movement, caused by long periods of gazing at computer screen. Distance between the screen and the user's head usually doesn't change much and as a result the muscles involved in adaptation of the eye are not exercised for long periods of time, leading to their weakening. This is usually accompanied by decrease in eye blinking frequency, which leads to excessive dryness of the eye surface (cornea and sclera) and can be harmful to the eye. Chronic dry eyes can eventually lead to scarring of the cornea and sight loss. With preventive measures like regular breaks, eye exercises and relaxational activities most of those disorders can be avoided. Unfortunately, the majority of population is reluctant to change their workplace habits until first signs of health issues appear. To help users become aware of the problem and assist them in prevention, we propose to use a simple monitor mounted camera (webcam) for capturing video of user at his workplace and estimating his eye blinking patterns. When potentially harmful behaviour is detected, the user can be alerted and informed about suitable actions.

1.2 Purpose: -

The purpose of the project "Strain Analysis based on Eye Blinking" is to develop a machine learning system that can assess and analyse an individual's mental and physical strain levels using their eye blinking patterns. The project aims to achieve the following objectives:

1. Non-Intrusive Strain Monitoring: The project seeks to provide a non-intrusive method for monitoring an individual's stress or strain levels. Traditional methods of measuring stress may involve self-reporting or physiological measurements, which can be intrusive and may affect the natural behaviour of the person being monitored. Monitoring eye blinking patterns offers a less intrusive approach to understanding an individual's strain levels.
2. Real-Time Assessment: The system aims to provide real-time assessment of strain levels. This allows for timely interventions and support, especially in situations where immediate feedback can be beneficial, such as in workplace settings, educational environments, or during medical procedures.
3. Objective Strain Measurement: Eye blinking patterns can serve as a potential physiological indicator of stress or strain. By leveraging machine learning techniques, the project aims to objectively quantify strain levels based on these patterns, reducing the reliance on subjective self-assessment.
4. Insights for Well-Being Improvement: The analysis of strain patterns can provide valuable insights into an individual's well-being. Understanding when and how strain levels increase can help identify triggers and potential health implications related to stress. This information can be used to implement strategies to improve overall well-being and manage stress more effectively.
5. Potential Applications in Different Fields: The project's purpose is to explore the wide range of applications where strain analysis based on eye blinking can be beneficial. These applications include workplace health and productivity, healthcare and patient well-being, educational environments, and mental health support.
6. Advancing Research in Stress Analysis: The project contributes to the ongoing research in stress analysis and its correlation with physiological responses. The findings and methodologies developed in this project may pave the way for further advancements in the field of mental health monitoring and stress management.

2. LITERATURE SURVEY

2.1 Existing problem: -

Accurately capturing eye blinking data can be challenging. Traditional methods like manual observation or video recording might not provide the required precision for strain analysis. Advanced technologies such as eye-tracking systems or specialized sensors may be needed to obtain more accurate data.

Eye blinking patterns can vary significantly between individuals and even in the same individual under different conditions. This variability can complicate the analysis and interpretation of strain-related data. External factors, such as lighting conditions, screen brightness, and environmental stress, can influence blinking behavior. Understanding and controlling these factors are crucial for accurate strain analysis.

Defining and identifying the exact indicators of strain in eye blinking data can be challenging. Researchers need to identify relevant metrics and patterns that correlate with eye strain and differentiate them from normal variations.

Strain analysis of eye blinking might work well in controlled laboratory settings, but its application in real-world scenarios, such as during extended computer usage or driving, might introduce additional challenges.

2.2 Proposed solution: -

The eye blink is a fast closing and reopening of a human eye. Each individual has a little bit different pattern of blinks. The pattern differs in the speed of closing and opening, a degree of squeezing the eye and in a blink duration. The eye blink lasts approximately 100-400 ms. We propose to exploit state-of-the-art facial landmark detectors to localize the eyes and eyelid contours. From the landmarks detected in the image, we derive the eye aspect ratio (EAR) that is used as an estimate of the eye opening state. Since the per frame EAR may not necessarily recognize the eye blinks correctly, a classifier that takes a larger temporal window of a frame into account is trained.

Blink detection method:

For every video frame, the eye landmarks are detected. The eye aspect ratio (EAR) between height and width of the eye is computed.

Eye Aspect Ratio Eye Aspect Ratio is proposed by the Czech Technical University [26,27]. It is a scalar quantity obtained by detecting a face from an image, finding the Euclidean distance of the corresponding eye coordinates, and substituting it into the following formula.

$$\mathrm{EAR} = \frac{||p_2 - p_6|| + ||p_3 - p_5||}{2||p_1 - p_4||}$$

where p1, . . ., p6 are the 2D landmark locations, depicted in Fig. 1. When the eye is open, the value is almost constant, but it approaches zero when the eye is closed. This value does not depend on head posture or distance, and there are small individual differences when the eyes are open. Since eye blinking is performed by both eyes synchronously, the EAR of both eyes is averaged. Due to the simplicity of the calculation formula, it has excellent real-time performance and shows high robustness. However, although we are trying to classify irregular movements such as eye thinning, absent stretching, or only eyes looking down, detection is difficult at the threshold. Also, although it is claimed to have high and straightforward detection accuracy, it causes a loss of accuracy when there is a sudden movement of the face or when the face is exceptionally far away from the camera. The resolution degradation due to distance could be solved by converting face images to higher resolution using super-resolution techniques, but this has not been investigated.

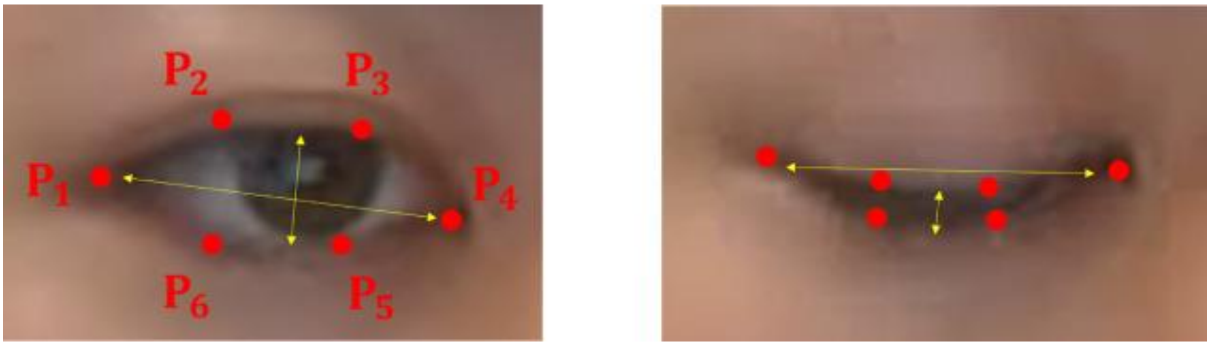


Fig-1

EAR is mostly constant when an eye is open and is getting close to zero while closing an eye. It is partially person and head pose insensitive. Aspect ratio of the open eye has a small variance among individuals and it is fully invariant to a uniform scaling of the image and in-plane rotation of the face. Since eye blinking is performed by both eyes synchronously, the EAR of both eyes is averaged. An example of an EAR signal over the video sequence.

Importing Necessary Libraries:

The first step is usually importing the libraries that will be needed in the program.
The required libraries to be imported to Python script are:

- SciPy
- Imutils
- Numpy
- Argparse
- Time
- Datetime
- Dlib
- Opencv
- Google text to speech
- Playsound
- Tkinter

Defining Necessary Functions:

We are defining necessary functions that can be used in the latter part of the program. The necessary functions are

1. **playaudio(text):**
   In this function, we are translating the text input to a speech by using gTTS and saving the translated speech to the output1.mp3 file. We are returning the output1.mp3 to the calling function.

2. **popupmsg(msg):**
   Creating an instance of Tk initializes the interpreter and creates the root window.

3. **eye_aspect_ratio(eye):**
   We can Calculate Eye Aspect Ratio:
   - Compute the Euclidean distances between the two sets of vertical eye landmarks (x, y)-coordinate.
   - Compute the Euclidean distance between the horizontal eye landmark (x, y)-coordinates.
   - Compute the eye aspect ratio using the above formula and then return ear to the calling function.

Construct Argparser:

We are using argparse library to parse command-line arguments.ArgumentParser() is a predefined class

Flags:

--shape-predictor: This is the path to dlib's pre-trained facial landmark detector.

--video: This is used to access a video file that is residing on the disk. If you want to use a webcam, you can simply omit this.

Defining Important Constants

Let us define two important constants

- **EYE_AR_THRESH**
  - By using ear value, we determine if a blink is taking place in the video stream.
  - A "blink" is registered when ear value falls below a certain EYE_AR_THRESH and then rises above the **EYE_AR_THRESH.**
  - We default it to a value of 0.3 as this is what has worked best for my applications, but you may need to tune it for your application.
- **EYE_AR_CONSEC_FRAMES**
  - This value is set to 3 to indicate that three successive frames with an eye aspect ratio less than EYE_AR_THRESH must happen for a blink to be registered.

Let us initialize two counters for counting the number of blinks.

- **COUNTER**
  - It is the total number of successive frames that have an eye aspect ratio less than EYE_AR_THRESH
- **TOTAL**
  - It is the total number of blinks that took place while the script has been running.

Get The Face Land Marks Using Dlib

We first load the detector using the g**et_frontal_face_detector(**) and facial landmark predictor dlib.shape_predictor from dlib library.

Capturing The Input Frames

The average number of blinks per minute should be at least 10-14. Hence, setting the eye_thresh default value to 10. But this can be modified based on the application to improve accuracy.
The function **datetime.datetime.now()**.minute gives the minute at that instant.
There are two ways we can capture the input video
1.using in-built webcam
2. using video file residing on the disk

Converting Frames to Grayscale Channels

cv2.cvtColor(frame, flag): is used for colour conversion.
cv2.COLOR_BGR2GRAY: The flag is used to convert the coloured image to grayscale.

Detect links

The aspect ratio will be approximately constant while the eye is open, and it will quickly fall to zero when a blink occurs. We need to determine the threshold for a blinking ratio that is near to zero.

Alerting The User

Calculate the average number of blinks:
Calculating the average number of blinks per minute by using the below code.

Initiate the alarm and popup message:
If the total blink count is less or more than the average blink count for the stipulated time(calculated for every minute incrementally), then an alert is initiated using audio and popup messages to take rest by calling the playaudio and popup functions.

Display The Result

The cv2.putText function displays the number of blink and ear on the OpenCV window once a blink count is detected.

The cv2.imshow() function always takes two more functions to load and close the image. These two functions are cv2.waitKey() and cv2.destroyAllWindows(). Inside the cv2.waitKey() function, you can provide any value to close the image and continue with further lines of code.

Run The Application

To access the built-in webcam execute the following command in anaconda prompt:
python app_eye.py --shape-predictor shape_predictor_68_face_landmarks.dat
To access video file residing on the disk execute the following command in anaconda prompt
python app_eye.py --shape-predictor shape_predictor_68_face_landmarks.dat –video filename.mp3

3. EXPERIMENTAL INVESTIGATIONS

Dataset:

A serious problem exists in evaluating the performance of blink detection systems. There are no standardized evaluation videos with which researchers can compare their results. For this reason, most of the research to date has generally been conducted by evaluating algorithms on videos that have not been standardized and taken by individuals.

Evaluation Videos:

All videos were shot using the front camera on the surface pro5 at a resolution of 640 × 480 pixels at 30fps with a length of about 11k frames per person. The images show the upper part of the body almost facing the camera, allowing us to see the appropriate body movements during VDT work. Also, those with bangs over their eyes had their hair trimmed to ensure face detection. In addition, the correct labels of SBR were counted manually after we visually checked all the videos. Furthermore, only complete spontaneous blinks were counted, so incomplete blinks were ignored.

Subject:

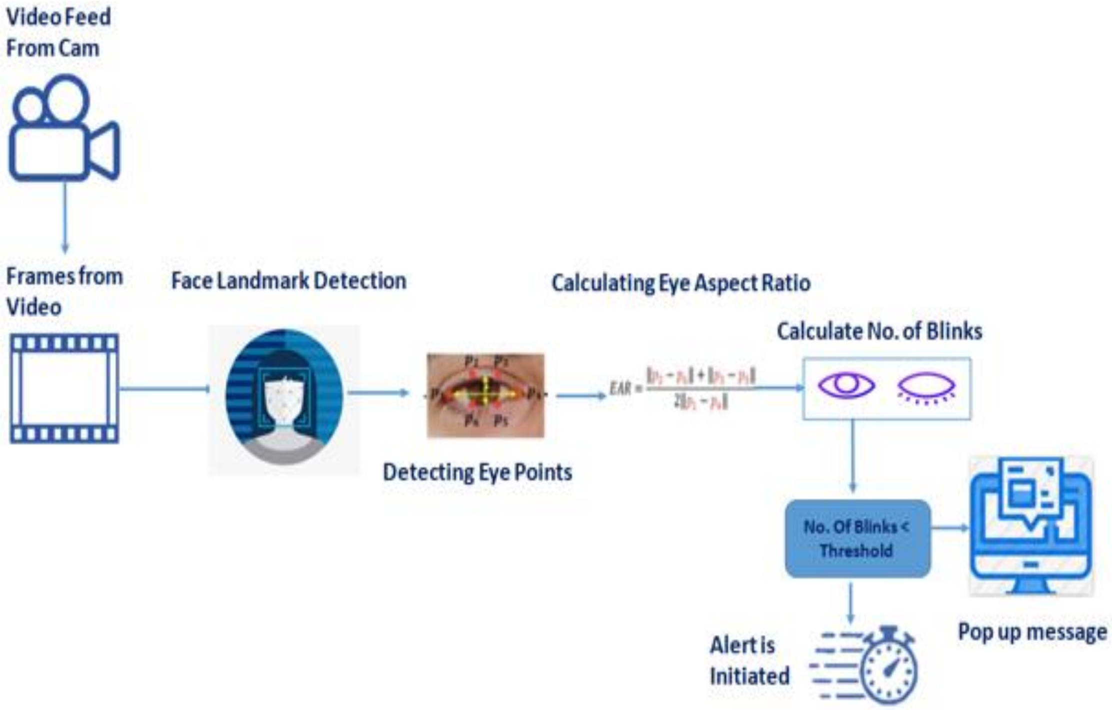
The subjects were four subjects in their 20 s who handled information equipment daily. In order to keep the experimental conditions, the same, the experiment was conducted within 5 hours of waking up with the minimal accumulation of eye fatigue.

Experimental environment:

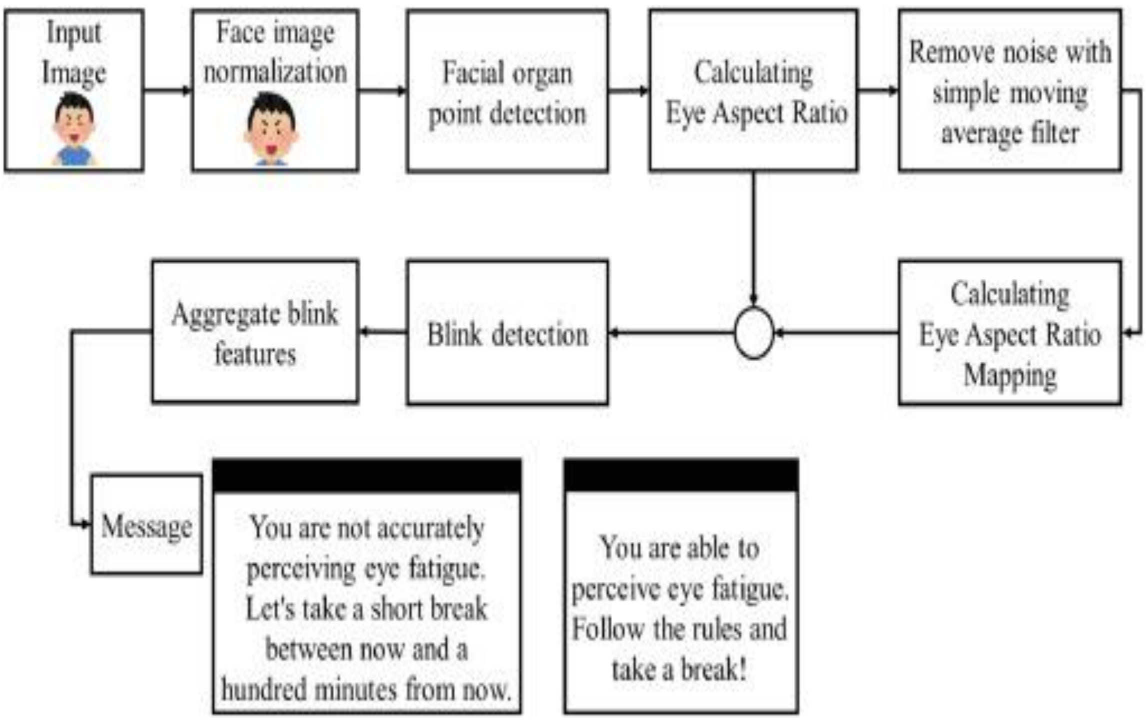
In this experiment, we worked indoors, shutting out the outside light to not depend on the weather conditions of the day. The room temperature was set at 26 ° Celsius, and the air conditioning was turned off before the start of the experiment in order to prevent the eyes from drying out. The luminous intensity of the display was set to the highest luminance that could be set. All the subjects were tested one by one in the same room, and no outsiders were allowed in the room while they were working to elicit their concentration.

4.THEORETICAL ANALYSIS

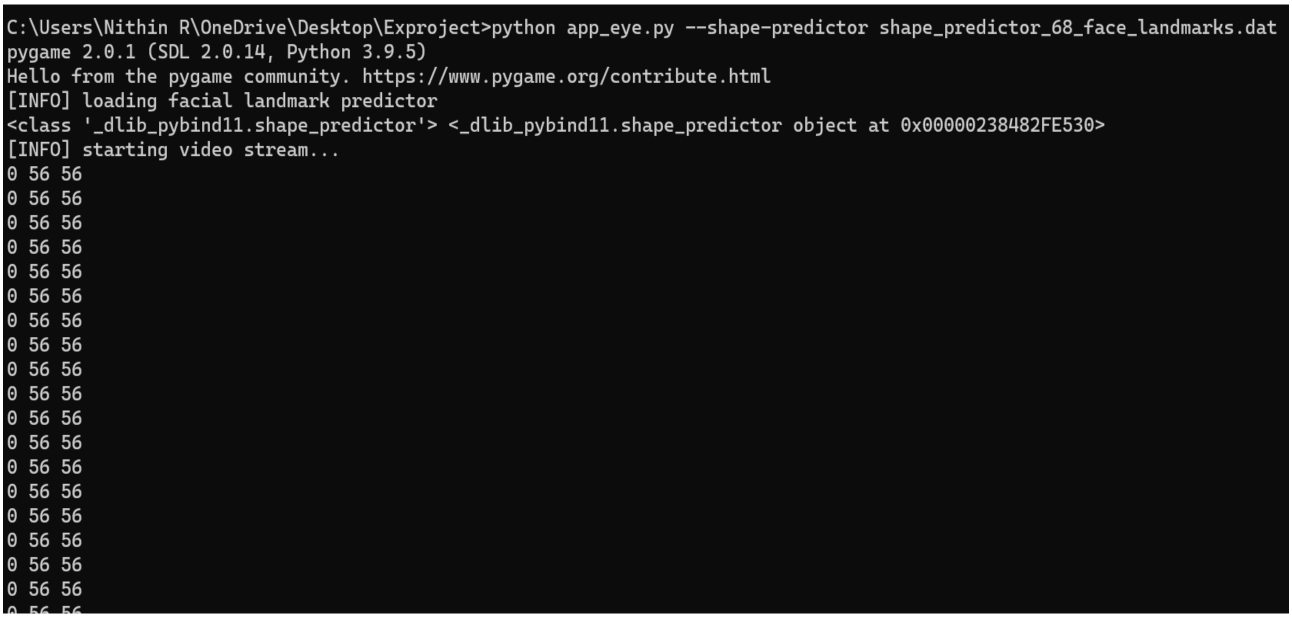
4.1 Block diagram: -

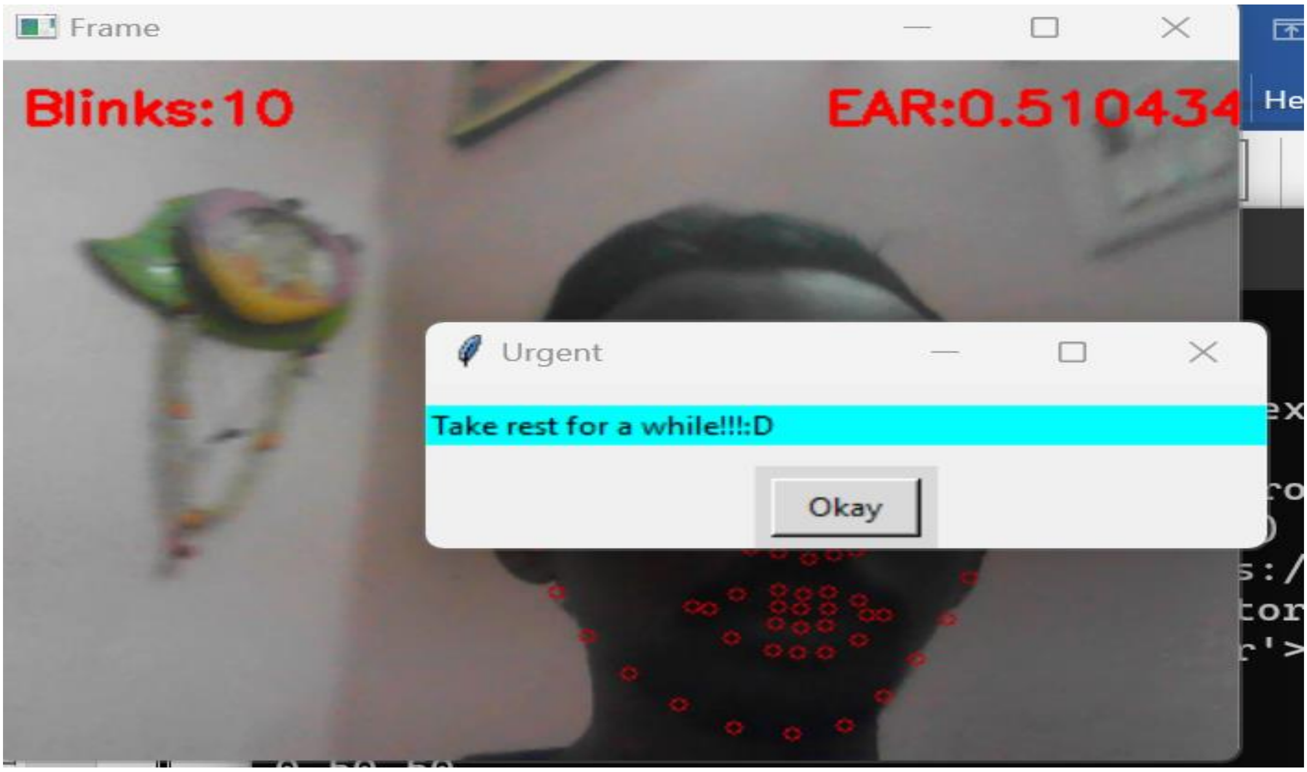


5. FLOWCHART



6. RESULT


C:\Users\Nithin R\OneDrive\Desktop\Exproject>python app_eye.py --shape-predictor shape_predictor_68_face_landmarks.dat
pygame 2.0.1 (SDL 2.0.14, Python 3.9.5)
Hello from the pygame community. https://www.pygame.org/contribute.html
[INFO] loading facial landmark predictor
<class '_dlib_pybind11.shape_predictor'> <_dlib_pybind11.shape_predictor object at 0x00000238482FE530>
[INFO] starting video stream...
0 56 56
0 56 56
0 56 56
0 56 56
0 56 56
0 56 56
0 56 56
0 56 56
0 56 56
0 56 56
0 56 56
0 56 56
0 56 56
0 56 56
0 56 56
0 56 56
0 56 56
0 56 56


Frame
Blinks:10
EAR:0.510434
Urgent
Take rest for a while!!!:D
Okay

7. ADVANTAGES & DISADVANTAGES

Advantages:

1. **Non-Intrusive Monitoring:** Eye blinking analysis provides a non-intrusive method to assess an individual's strain levels without the need for intrusive physiological sensors or self-reporting.

2. **Real-Time Assessment:** The project's potential to provide real-time strain analysis enables timely interventions and support, which can be valuable in managing stress in various scenarios.

3. **Objective Strain Measurement:** By leveraging machine learning, the project aims to provide an objective and quantitative measurement of strain levels, reducing reliance on subjective self-assessment.

Disadvantages:

1. **Individual Variability:** Eye blinking patterns vary significantly among individuals, making it challenging to create a generalized model that accurately assesses strain for diverse populations.

2. **Data Collection Challenges:** Collecting a diverse and representative dataset of eye blinking patterns, especially in real-world scenarios, can be time-consuming and resource-intensive.

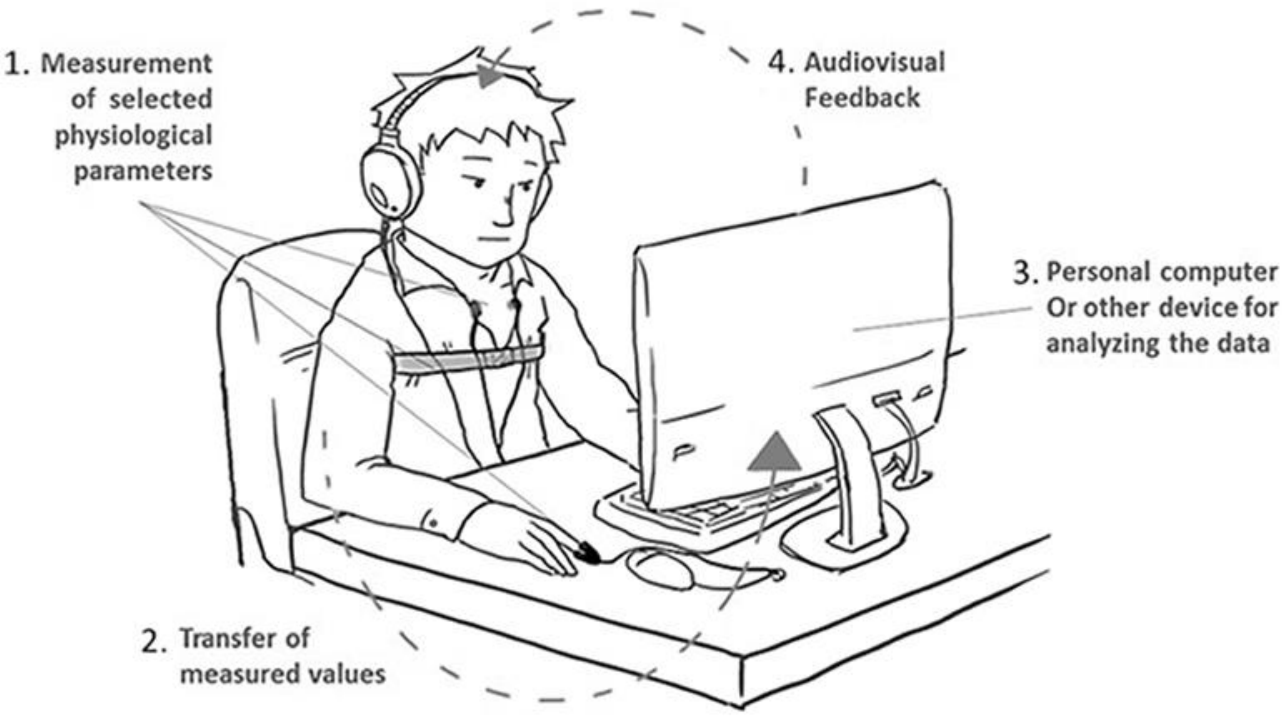
3. **Context Dependency:** Strain levels are context-dependent, and eye blinking alone may not capture all aspects of an individual's stress response in complex scenarios.

4. **Ethical Considerations:** Analysing and storing physiological data, such as eye movements, raises privacy and ethical concerns, requiring stringent data protection and informed consent.

8. APPLICATIONS

- **Healthcare and Patient Monitoring**

- **Biofeedback and Stress Management**



9. CONCLUSION

In conclusion, the "Strain Analysis based on Eye Blinking" project represents a valuable contribution to the domain of stress monitoring and well-being assessment. By combining computer vision, machine learning, and real-time analysis, the project offers practical applications with the potential to positively impact various aspects of people's lives, from workplace productivity to mental health support. As research in this field continues to advance, the project's findings and methodologies will contribute to ongoing efforts in stress management and human-computer interaction, fostering healthier and more productive living environments.

We presented an approach for video-based detection of eye-blinks that could be used to alert the computer user if potentially dangerous blinking behavior is detected. We believe this approach could be much more effective than currently available preventive software, which is usually based on keyboard and mouse activity. The proposed twolevel analysis of eye’s normal flow enables us to detect blinks as well as various other kinds of eye movements. To improve the performance, we plan to use GPU-based implementation of optical flow estimation instead of simple normal flow calculation. Additionally, an upgrade to the Lucas-Kanade tracker would allow for better tracking of fast, significant face movements.

10. FUTURE SCOPE

"Strain Analysis based on Eye Blinking" holds significant potential for future advancements and applications. As technology and research progress, the project's future scope can encompass several areas of development and expansion:

- Cognitive Load Assessment
- Biofeedback and Neurofeedback Applications
- Real-World Deployment and Validation

11. BIBLIOGRAPHY

[1] T. Morris, P. Blenkhorn and F. Zaidi. Blink detection for real-time eye tracking. Journal of Network and Computer Applications, vol. 25, num. 2, pp. 129-143, April 2002.

[2] S. Sirohey, A. Rosenfeld and Z. Duric. A method of detecting and tracking irises and eyelids in video. Pattern Recognition, vol. 35, num. 6, pp. 1389-1401, 2002.

[3] Ric Heishman and Zoran Duric. Using image flow to detect eye blinks in color videos. Proceedings of the Eighth IEEE Workshop on Applications of Computer Vision, pp. 52, 2007.

[4] Michael Chau and Margrit Betke. Real Time Eye Tracking and Blink Detection with USB Cameras. Boston University Computer Science Technical Report No. 2005-12, May 2005.

[5] Gang Pan, Lin Sun, Zhaohui Wu and Shihong Lao. Eyeblink-based anti-spoofing in face recognition from a generic webcamera. The 11th IEEE International Conference on Computer Vision (ICCV'07), Rio de Janeiro, Brazil, October 2007.

[6] Javier Orozco, F. Xavier Roca and Jordi Gonzalez. Real-time gaze tracking with appearance-based models. Machine Vision and Applications, 2008.

[7] Rainer Lienhart and Jochen Maydt. An Extended Set of Haar-like Features for Rapid Object Detection. IEEE ICIP 2002, vol. 1, pp. 900-903, Septembre 2002.

[8] Jean-Yves Bouguet. Pyramidal Implementation of the Lucas Kanade Feature Tracker. Intel Corporation, 2002.

[9] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. European Conf. on Computer Vision, vol. 1, pp. 430-443, May 2006.

[10] The BioID Face Database, HumanScan AG, Switzerland, http://www.bioid.com/downloads/facedb/index.php.

[11] The Color FERET Database, Version 2, National Institute of Standards and Technology, http://face.nist.gov/colorferet/.

[12] N. A. Fox, B. A. O'Mullane and R. B. Reilly. The Realistic Multi-modal VALID database and Visual Speaker Identification Comparison Experiments. Proc. of the 5th International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA-2005), New York, July 20-22, 2005.

[13] Talking Face Video. Face&Gesture Recognition Working group, IST-2000-26434, http://www-prima.inrialpes.fr/FGnet/data/01-TalkingFace/talking_ face.htm