Chapter 1

INTRODUCTION

Many embedded systems have substantially different designs according to their functions and utilities. In this design, structured modular design methods are adopted, and the system is mainly composed of a single microcontroller. The sleepiness detection system is important to lecturers to monitor the sleepy students in the class. The students cannot give proper attention to the teacher in the classroom. Leading to the problem in the classroom, many students have been seen to be sleeping during a lecture class.

The teacher might need to observe while lecturing to provide optimum learning to students. Observation in the classroom is difficult to handle, due to the limitation of lecturers at the university. The main intention is to develop a sleepiness detection system. This aids the lecturer to monitor the sleepy student during a lecture in the classroom. It mainly focuses on designing a system that can monitor the open and closed state of the student's eyes in real-time.

The main aim is to detect sleepiness among students in the classroom by monitoring the eyes. For this, it involves a sequence of images of the student's face and sees of the eye's movements and blinking patterns. Then, this is programmed to localize the eyes of the student, which involves screening the entire image of the face and then determining the position of the eyes by a self-developed image processing algorithm. Once the position of the eyes is located by the system, it analyzes and determines whether the eyes are open or closed. Another objective of this work is to evaluate the performance and limitations of the system in detecting sleepy students. This is placed in the classroom in front of the student and then it will determine whether the student is sleepy or not.

1.1 Problem Definition

Detection of sleepiness with the help of a webcam to detect signs of sleepiness in a person's eyes in real-time.

1.2 Project Idea

- Choosing a suitable software for image processing.
- Design a suitable algorithm for fatigue recognition.
- Design a system to activate an alarm.
- Design a prototype for a student sleepiness detection system.
- Testing and result analysis.

1.3 Motivation

Academic Performance Improvement: By detecting signs of sleepiness and distraction, the system can help students maintain focus and attentiveness, leading to improved academic performance. The timely delivery of engaging memes can serve as a fun and effective way to keep students alert and actively engaged during learning activities.

Educational Innovation: Integrating memes into the alert system represents an innovative approach to addressing students' sleepiness and distraction. This innovative use of technology can contribute to the advancement of educational practices and student engagement strategies, promoting a more dynamic and effective learning experience.

1.4 Scope

The project's sleepiness detection system can identify students who are sleepy and, depending on the circumstances, will use an alarm to notify the student. The student sleepiness detection technology being used in this study intends to be widely accessible and compatible with various kinds.

1.5 Literature Survey

Paper	Author	Description
The Sleepiness of International Students in Online Chinese Classroom: An Experimental Study Based on Hybrid Technology of Facial Feature Point Distribution Model [2022]	Tianya Wang, Dong Ge Chun Hong Qi	● Hybrid technology of facial feature point distribution model. ● The hybrid technology used be a useful tool for future research in monitoring students' engagement and attention levels in online classrooms.
A deep learning based online classroom fatigue monitoring system for students [2022]	Zijie Wang, Liu Long, Yunwen Lang, Yuanxia Ji, Jie Xie, Shaokun Lu	● Uses the improved YOLOv5s target detection model ● The method uses three indicators: blinking frequency, yawning frequency, and nodding drowsiness frequency to detect and alert students to fatigue according to the set thresholds.

Paper	Author	Description
Driver Drowsiness Detection by Yawn Identification Based on Depth Information and Active Contour Model	Mina Zohoorian Jafari Yazdi Mohsen Soryani	• Used viola-jones algorithm. • Using the depth image, an open or closed mouth was detected with an accuracy of 86%.
Deep CNN: A Machine Learning Approach for Driver Drowsiness Detection Based on Eye State[2019]	Venkata Rami Reddy Chirra, , Srinivasulu Reddy Uyyala, , Venkata Krishna Kishore Kolli	• Smart cameras using computer vision technologies can identify drowsiness in drivers and alert them. • Viola Jones algorithm is being used here. • The system alerts the driver with an alarm when they are in a sleepy mood.
Dank Learning: Generating Memes Using Deep Neural Networks [2018]	Abel L. Peirson V, E. Meltem Tolunay	• Humor varies across people and cultures, making it difficult to capture. • One example of a new idea is to train on a dataset that includes the break point in the text between upper and lower for the image.

Table : 1.5.a Literature survey table

Chapter 2

PROJECT DESIGN

2.1 Hardware and Software Requirements

Hardware:

- Cameras or sensors for monitoring students' behavior and facial expressions.
- Computing devices (e.g., servers, workstations, or cloud -based infrastructure for data processing and analysis.
- Displays or screens for presenting memes as alerts.

Software:

- Operating system (OS) for the computing devices, preferably a stable and widely compatible OS.
- Programming languages such as Python for implementing machine learning algorithms and system logic.
- Libraries and frameworks for computer vision, such as OpenCV, to process and analyze the data from the cameras or sensors.
- Machine learning frameworks like TensorFlow or PyTorch for training and deploying sleepiness and distraction detection models.
- Database management systems (DBMS) for storing and managing the dataset and system logs.

2.2 Dataset Design:

- Collect data using cameras or sensors in a controlled environment, such as a classroom or a designated study area.
- Record video footage of students exhibiting various levels of alertness, including instances of sleepiness and distraction.
- Annotate the dataset with labels indicating the presence of sleepiness or distraction in each recorded instance.
- Ensure diversity in the dataset by including students of different ages, genders, and ethnicities, as well as various lighting conditions and classroom settings.
- Validate the dataset to maintain accuracy and consistency in the labeled instances.

2.3 Hours Estimation:

- Software development and integration: 10-20 hours
- Documentation and reporting: 4-6 hours

Chapter 3

MODULE DESCRIPTION

3.1 Module Description

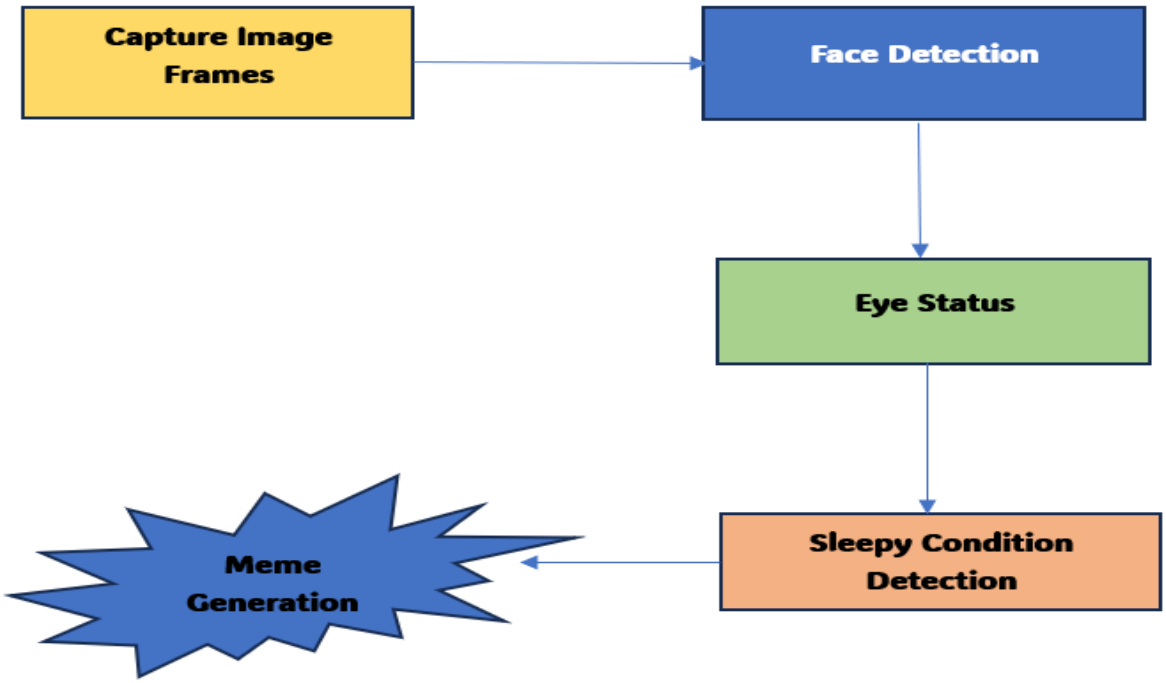

Fig. 3.a . Modules operational flow diagram

Captures image frames: The system captures image frames from a camera installed in the classroom or any other location where students are present.

Face detection: The system performs face detection on the captured image frames to identify the presence of a face.

Eyes status: If a face is detected, the system checks the eyes status to determine if they are open or closed.

Sleepy condition detection: If yawning is detected, the system checks for sleepy condition detection. This is done by analyzing various parameters such as head position, eye movement, and facial expressions to determine if the student is feeling drowsy.

Meme generations: If the sleepy condition is detected, the system generates memes and stores them in a database. These memes can be used to alert the student and help them stay awake.

Chapter 4

RESULTS AND DISCUSSION

4.1 Code :

Main.py > ...

```python
def play_alert_video():
    # Get a list of all meme videos in the dataset folder
    dataset_folder = 'C:/Users/adiko/Downloads/edi/memealert/meme_data'
    meme_videos = [file for file in os.listdir(dataset_folder) if file.endswith('.mp4')]
    meme_video_path = os.path.join(dataset_folder, random.choice(meme_videos))
    audio_file_name = os.path.splitext(meme_video_path)[0] + '.wav'
    # Check if the audio file exists
    if not os.path.exists(audio_file_name):
        print("Audio file not found for the meme video:", meme_video_path)
        return
    # Initialize Pygame mixer for audio playback
    pygame.mixer.init()
    alarm_sound = pygame.mixer.Sound(audio_file_name)
    cap = cv2.VideoCapture(meme_video_path)
    frame_rate = cap.get(cv2.CAP_PROP_FPS)
    # Get the duration of the audio file
    audio_duration = alarm_sound.get_length()
    while cap.isOpened():
        ret, frame = cap.read()
        if not ret:
            break
        cv2.imshow('Meme', frame)
        pygame.mixer.Sound.play(alarm_sound)
        delay = int(1000 / frame_rate)  # Delay in milliseconds
        pygame.time.delay(int(delay * 0.8))
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
    cap.release()
    # alarm_sound.stop()
    cv2.destroyWindow('Meme')
```

Fig. 4.1.a : code section for play alert video

In this code snippet, the play_alert_video() function is defined, which takes the path of the alert video file as input. The function uses the OpenCV library to load and display the video frames, while the playsound library is utilized to play the alert sound simultaneously. The video is displayed in a window titled 'Alert Video', and the function continues to run until the video ends or the user presses the 'q' key to quit. This section of the code ensures that an alert video is promptly played when the Sleepiness Detection System identifies drowsiness, effectively capturing the user's attention and promoting alertness.

```
def blinked(a, b, c, d, e, f):
    up = compute(b, d) + compute(c, e)
    down = compute(a, f)
    ratio = up/(2.0*down)
    if (ratio > 0.25):
        return 2
    elif (ratio > 0.21 and ratio <= 0.25):
        return 1
    else:
        return 0
while True:
    _, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = detector(gray)
    face_frame = frame.copy()
    for face in faces:
        x1 = face.left()
        y1 = face.top()
        x2 = face.right()
        y2 = face.bottom()
        cv2.rectangle(face_frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
        landmarks = predictor(gray, face)
        landmarks = face_utils.shape_to_np(landmarks)
        left_blink = blinked(landmarks[36], landmarks[37],
                             landmarks[38], landmarks[41], landmarks[40], landmarks[39])
        right_blink = blinked(landmarks[42], landmarks[43],
                              landmarks[44], landmarks[47], landmarks[46], landmarks[45])
        if (left_blink == 0 or right_blink == 0):
            sleep += 1
```

Fig. 4.1.b : Blinked Function Code part 1

```
            active = 0
            if (sleep > 6):
                status = "SLEEPING !!!"
                alarm_sound.play()
                play_alert_video()
                color = (255, 0, 0)
        elif (left_blink == 1 or right_blink == 1):
            sleep = 0
            active = 0
            drowsy += 1
            if (drowsy > 6):
                status = "Drowsy !"
                # for i in range(5):
                #     win32api.Beep(random.randint(37,10000), random.randint(750,3000))

                color = (0, 0, 255)
        else:
            drowsy = 0
            sleep = 0
            active += 1
            if (active > 6):
                status = "Active :)"
                color = (0, 255, 0)
        cv2.putText(frame, status, (100, 100),
                    cv2.FONT_HERSHEY_SIMPLEX, 1.2, color, 3)
        for n in range(0, 68):
            (x, y) = landmarks[n]
            cv2.circle(face_frame, (x, y), 1, (255, 255, 255), -1)
    cv2.imshow("Frame", frame)
    cv2.imshow("Result of detector", face_frame)
```

Fig. 4.1.c : Blinked Function Code Part 2

The given code snippet in 4.1.b and 4.1.c defines a function blinking_ratio(a, b, c, d, e, f) that calculates the ratio of eye closure to detect blinks based on the facial landmarks obtained from the Dlib library. The function takes six points representing specific facial landmarks as input and computes the ratio of the upward movement of the eyelids to the downward movement of the eyes. If the calculated ratio indicates a blink, the function returns a value of 2. If the ratio suggests partial closure, it returns 1. Otherwise, if the eyes are open, the function returns 0.

The code also contains a while loop that continuously captures video frames using the OpenCV library. It detects faces in the frames and draws rectangles around the detected faces. For each detected face, the code calculates blink ratios for both eyes using the blinking_ratio function. Based on these ratios, the code determines whether the person is sleeping, drowsy, or active. If the person is identified as drowsy or sleeping for a certain duration, an alert video is played to capture the individual's attention. The status (sleeping, drowsy, or active) is displayed on the video frame along with corresponding text color. The loop continues until the user presses the 'Esc' key (key code 27), at which point the program exits.

4.2 Outputs :

The output window shows two screens, the left screen shows the state and the right screen shows the landmarks made by dlib library .

Figure 4.2.a In the active state, the system observes a consistent eye opening ratio exceeding 0.25, indicating alertness. It displays "Active :)" on the screen in vibrant green, confirming the person's attentive and awake condition, ensuring a safe and productive environment.

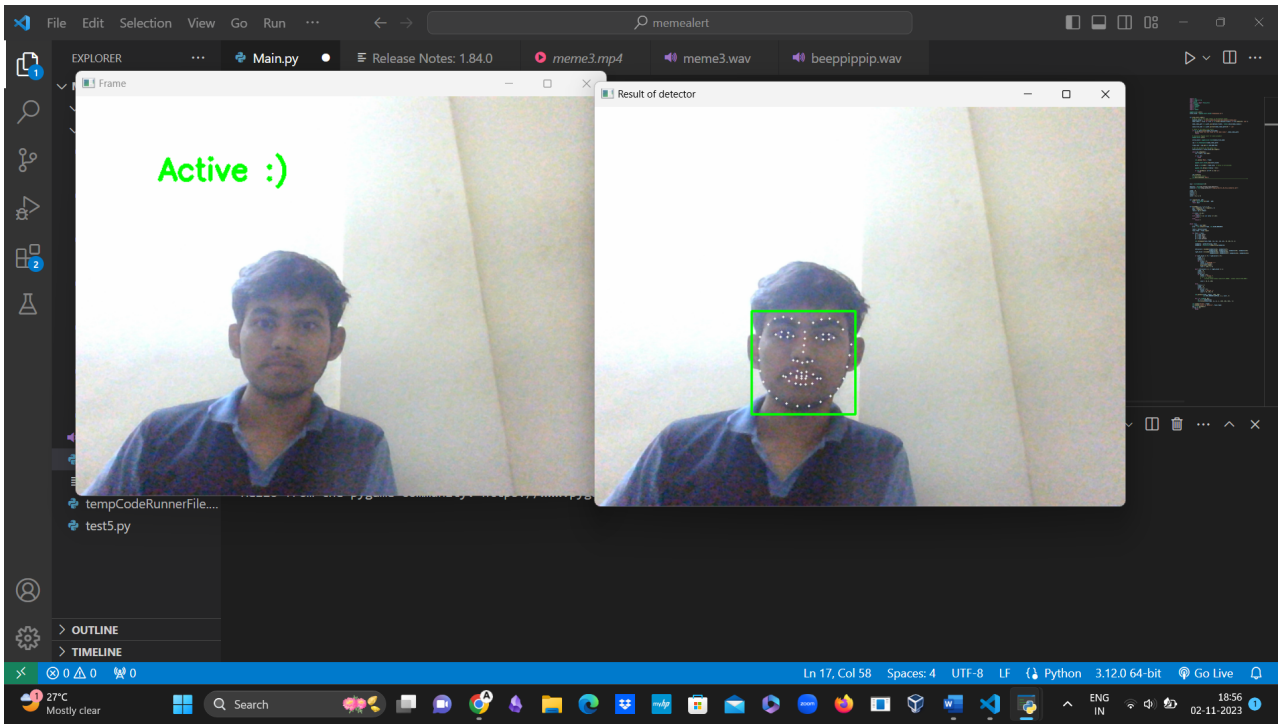

Fig. 4.2.a : active state output

In Figure 4.2.b, the system detects a drowsy state when the eye closure ratio falls between 0.21 to 0.25. It prompts a warning message ("Drowsy !") on the screen in red, indicating reduced alertness and the need for attention, enhancing safety awareness in real-time situations.

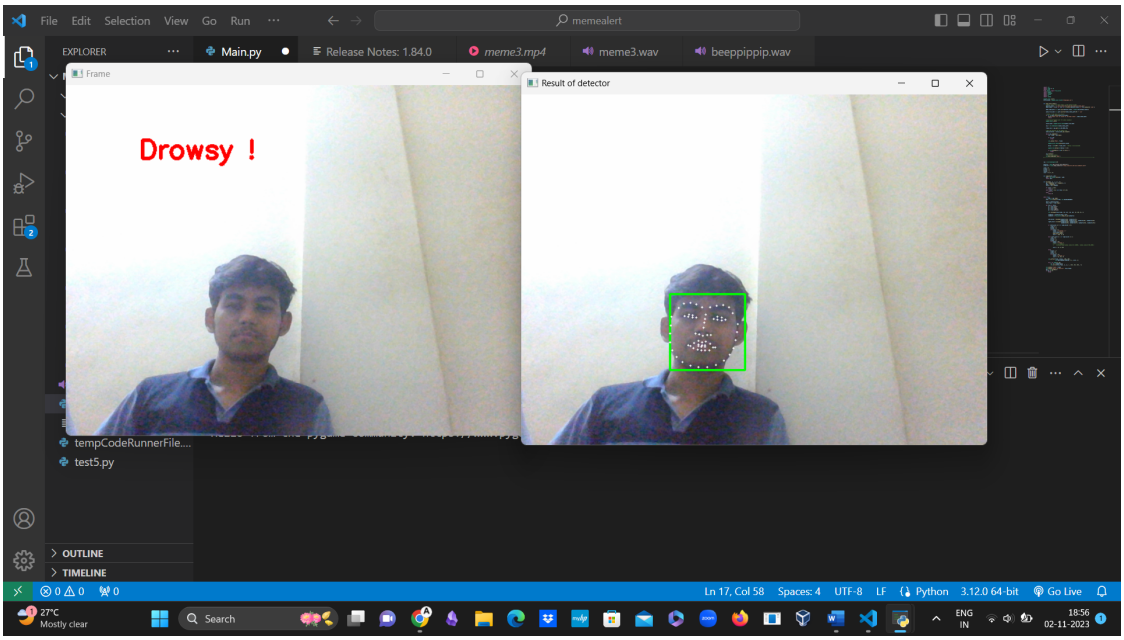

Fig. 4.2.b : drowsy state

In Figure 4.2.c, the system detects a sleeping state when the eye closure ratio is below 0.21, indicating complete or near-complete eye closure. An alarm is triggered, displaying "SLEEPING!!!" in blue, prompting immediate intervention for the individual's safety.

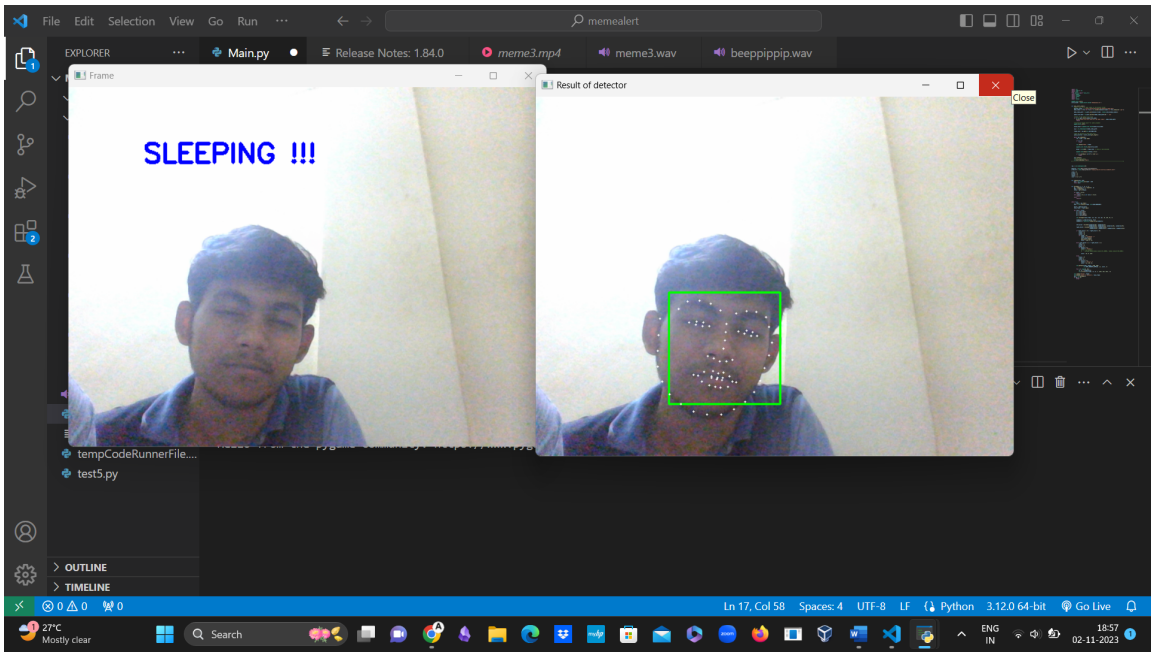

Fig. 4.2.c : Output for the state

In Figure 4.2.d, the system deploys a meme alert if it detects sleepiness persisting for

over 6 frames. This engaging response captures attention, encouraging alertness in a lighthearted manner and enhancing safety.

Fig. 4.2.d : meme alert

Chapter 5

CONCLUSION

In conclusion, the computer vision project described is designed to monitor a user's facial expressions and eye blinks to determine their state, whether it's sleepy, drowsy, or active. When the system detects that the user is drowsy, it triggers an alert in the form of a meme video and sound to help the user stay awake and alert. The project utilizes OpenCV for image processing, Dlib for face detection and landmark detection, and various techniques to analyze eye blinks and determine the user's state. When the user is detected as being drowsy, the project plays a random meme video to grab their attention and keep them awake. The project combines computer vision with multimedia elements to provide an interactive and engaging solution for addressing drowsiness. Further improvements and refinements can be made, such as enhancing the accuracy of drowsiness detection, customizing the meme selection process, and implementing safety features to ensure the user's well-being while using the system. This project demonstrates the potential of computer vision in addressing real-world problems and promoting user engagement and safety.

Chapter 6

REFERENCES

1. Wang, Tianyu & Ge, Dong & Qi, Chunhong. (2022). The Sleepiness of International Students in Online Chinese Classroom: An Experimental Study Based on Hybrid Technology of Facial Feature Point Distribution Model. International Journal of Human-Computer Interaction. 1-10. 10.1080/10447318.2022.2133673.

2. Y. Chen, D. Zhao and G. He, "Deep Learning-Based Fatigue Detection for Online Learners," 2022 5th International Conference on Pattern Recognition and Artificial Intelligence (PRAI), Chengdu, China, 2022, pp. 924-927, doi: 10.1109/PRAI55851.2022.9904096.

3. Li, Deguang & Cui, Zhanyou & Cao, Fukang & Cui, Gaoxiang & Shen, Jiaquan & Zhang, Yongxin. (2022). Learning State Assessment in Online Education Based on Multiple Facial Features Detection. Computational Intelligence and Neuroscience. 2022. 1-16. 10.1155/2022/3986470.

4. Yazdi, Mina & Soryani, Mohsen. (2019). Driver Drowsiness Detection by Yawn Identification Based on Depth Information and Active Contour Model. 1522-1526. 10.1109/ICICICT46008.2019.8993385.

5. Ch, Venkata & Reddy, U. Srinivasulu & KishoreKolli, Venkata. (2019). Deep CNN: A Machine Learning Approach for Driver Drowsiness Detection Based on Eye State. Revue d'Intelligence Artificielle. 33. 461-466. 10.18280/ria.330609.

6. Peirson, A.L., & Tolunay, E.M. (2018). Dank Learning: Generating Memes Using Deep Neural Networks. *ArXiv, abs/1806.04510*.

7. Park, Sanghyuk, et al. "Driver drowsiness detection system based on feature representation learning using various deep networks." Asian Conference on Computer Vision. Cham: Springer International Publishing, 2016.

8. Tabrizi, Pooneh R., and Reza A. Zoroofi. "Open/closed eye analysis for drowsiness detection." 2008 first workshops on image processing theory, tools and applications. IEEE, 2008.

9. Panicker, Arun D., and Madhu S. Nair. "Open-eye detection using iris–sclera pattern analysis for driver drowsiness detection." Sādhanā 42 (2017): 1835-1849.

10. Ngxande, Mkhuseli, Jules-Raymond Tapamo, and Michael Burke. "Driver drowsiness detection using behavioral measures and machine learning techniques: A review of state-of-art techniques."

11. Pandey, Nageshwar Nath, and Naresh Babu Muppalaneni. "A novel algorithmic approach of open eye analysis for drowsiness detection." International Journal of Information Technology 13 (2021): 2199-2208.

12. Liu, Danghui, et al. "Drowsiness detection based on eyelid movement." 2010 second international workshop on education technology and computer science. Vol. 2. IEEE, 2010.

13. Pandey, Nageshwar Nath, and Naresh Babu Muppalaneni. "A novel algorithmic approach of open eye analysis for drowsiness detection." International Journal of Information Technology 13 (2021):2199-2208.

14. Flores, Marco Javier, Jose Maria Armingol, and Arturo de la Escalera. "Real-time drowsiness detection system for an intelligent vehicle." 2008 IEEE Intelligent Vehicles Symposium. IEEE, 2008.

15. Purnamasari, Prima Dewi, et al. "Eye based drowsiness detection system for driver." Journal of Electrical Engineering & Technology 17 (2022): 697-705.

16. Flores, Marco Javier, Jose Maria Armingol, and Arturo de la Escalera. "Real-time drowsiness detection system for an intelligent vehicle." 2008 IEEE Intelligent Vehicles Symposium. IEEE, 2008.

17. Sahayadhas, Arun, Kenneth Sundaraj, and Murugappan Murugappan. "Detecting driver drowsiness based on sensors: a review." Sensors 12.12 (2012): 16937-16953.