International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 09 September 2025
Page No: 3432-3438
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0503

Real-Time Driver Drowsiness Detection Using Eye Aspect Ratio and Facial Landmark Analysis

Taufiya Fathima[1], Dr. H Girisha[2]
[1]II Sem M. Tech, Department of CSE, RYM Engineering College, RYMEC, VTU, Belagavi, India.
[2]Professor, Department of CSE, RYM Engineering College, RYMEC, VTU, Belagavi, India.
Emails: *taufiyafathima710@gmailcom[1], hosalligiri@gmail.com[2]*

Abstract
Driver drowsiness detection is crucial to preventing road accidents caused by fatigue. This paper proposes a non-intrusive, real-time system based on facial landmark detection and the Eye Aspect Ratio (EAR) using a standard webcam. The system continuously monitors eye activity and triggers an alert when signs of drowsiness are detected, measured by sustained low EAR values. The approach integrates a pre-trained face detector, facial landmark predictor, and an EAR-based thresholding mechanism to determine eye closure. High accuracy is demonstrated by the experimental results in detecting drowsiness, making the system suitable for embedded or mobile deployment in automotive applications.

Keywords: *Computer Vision Drowsiness detection, Eye Aspect Ratio (EAR), Facial landmarks, OpenCV*

1. Introduction
Drowsiness while operating vehicles or heavy machinery significantly increases the risk of accidents. According to road safety authorities, a considerable percentage of fatal crashes are linked to driver fatigue. Conventional solutions involve physiological sensors or wearable devices, which may be intrusive or uncomfortable. This study proposes a contactless, image-based method that leverages facial landmarks to detect eye closure, using the Eye Aspect Ratio (EAR) as a quantitative measure [1][2]. When EAR remains below a defined threshold for a specific duration, an alert is generated to awaken the subject.

2. Related Work
Various drowsiness detection methods have been explored, including EEG/ECG, behavioral analysis, and computer vision. Physiological sensors provide accuracy but are impractical in real-world driving [3]. In addition to physiological methods like EEG, video-based yawning datasets such as YawDD [2] have supported development of robust visual drowsiness detection systems. Vision-based systems are more viable. T. Soukupová and J. Čech introduced the EAR as a robust indicator for eye state detection in their blink detection model [4]. Several follow-up studies have utilized this metric in driver monitoring systems [1], [5], [6]. Unlike machine learning-based classification approaches, EAR relies on geometric relationships between eye landmarks, making it computationally efficient and suitable for real-time implementation without a large training dataset [7], [8]. Table 1 gives the literature survey

3. Methodology
The proposed system consists of several components: webcam capture, facial landmark detection, EAR computation, and alert logic. It is designed to run in real-time, continuously monitoring the driver's eye state to detect signs of drowsiness. The system integrates computer vision and audio feedback to prevent potential fatigue-induced accidents. The following pseudocode summarizes the core operational flow of the proposed drowsiness detection system.

Algorithm 1: EAR-Based Drowsiness Detection
1. Start webcam feed
2. Load face detector and facial landmark predictor
3. Initialize frame_counter = 0
4. While the webcam is open:
   a. Read a frame from the webcam
   b. Resize and convert frame to grayscale
   c. Detect faces in the frame
   d. For each detected face:

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 09 September 2025
Page No: 3432-3438
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0503

Table 1 Comparison Table of Literature Survey

Author	Method/Technique Used	Dataset Used / Key Findings
K. Satish et al. [1]	EAR using HOG + Dlib facial landmarks; hand pressure sensing via Arduino + load cell.	Live video and sensor data. Triggers alert only if both EAR and hand pressure thresholds are crossed.
Inakollu Kiran Kumar et al.[5]	Haar cascade for eye detection, ELR computed using Euclidean distances, triggers alarm below threshold.	Real-time webcam video. Cost-effective, Python-based, suitable for integration with GPS/sensors.
Radhika Gandhi et al. [6]	PERCLOS and EAR analysis using facial landmarks on Raspberry Pi with Python.	Raspberry Pi camera feed. Audio alert via speaker module; suitable for embedded environments.
Belal Alshaqaqi et al.[7]	PERCLOS-based detection using Hough transform for iris, symmetry-based face and eye detection.	IR camera input. Accurate under day/night; real-time capable using visual symmetry detection.
Aryan Ritesh et al. [8]	CNN-based real-time Android app using OpenCV and facial features + mobile accelerometer fusion.	Custom camera + sensor data. Achieves 89% accuracy, 91% precision; deployable via smartphone app.
Lie Yang et al. [9]	Two-branch multi-head attention (TB-MHA) model with center loss, facial region CSP filtering.	YawDD, NTHU-DDD, VBDDD. Outperforms baselines using local feature focus; robust to lighting variations.
Wanghua Deng & Ruoxue Wu [10]	MC-KCF (CNN + KCF) face tracking, eye-angle CNN classifier, yawning duration analysis.	CelebA, YawDD, and custom data. Achieves 92% accuracy; supports deployment on cloud and mobile.
Hemant K. Dua et al. [11]	Mobile-based system with camera-based eyelid magnitude tracking; alarm via Bluetooth signal.	Smartphone camera input. Detects eye closure > 3 sec; real-time, low-light-capable mobile solution.

- Predict facial landmarks (68 points)
- Extract left and right eye landmarks
- Calculate EAR for both eyes
- EAR = (left_EAR + right_EAR) / 2.0
- If EAR < threshold (e.g., 0.25): Increment frame_counter by 1
- If frame_counter >= threshold_frames
- (e.g., 20): Display ALERT message - Play alarm sound
- Else: (EAR ≥ threshold → eyes open)
- Reset frame_counter = 0
- Display the current frame with visual overlays
- If 'q' key is pressed: Exit loop

5. Release webcam and close all windows

Figure 1 shows the flow of operations in the proposed driver drowsiness detection system. Each block corresponds to a component in the pseudocode, emphasizing the sequential decision-making and real-time alerting mechanism. The proposed drowsiness detection system operates through a sequence of real-time vision-based modules. The implementation is guided by the pseudocode and corresponding flowchart described earlier. Each stage of the process is detailed below.

3.1 Webcam Initialization Discussion

The system initiates a live video stream using OpenCV's VideoCapture function, enabling frame-by-frame acquisition from the default webcam.

3.2 Loading Detection Models

A pre-trained frontal face detector and a 68-point facial landmark predictor from the Dlib library are loaded. These models facilitate the identification of facial structures required for eye analysis. Dlib provides a robust pipeline for facial analysis, utilizing a Histogram of Oriented Gradients (HOG)–based face detector in conjunction with an ensemble of regression trees for accurate facial landmark localization [12]. The 68-point landmark configuration enables precise detection of key facial regions such as the eyes, nose, mouth, and jawline, which are essential for computing the Eye Aspect Ratio (EAR) used in drowsiness detection [9].

3.3 Counter Initialization

A frame counter variable is initialized to zero. This variable tracks the number of consecutive frames in which the eyes are detected to be closed. [10]

3.4 Frame Processing Loop

The system enters a continuous loop in which each frame is processed as follows:

3.4.1 Frame Acquisition and Preprocessing

Each frame is read from the video stream, resized for efficiency, and converted to grayscale to enhance detection speed and accuracy. [11]

3.4.2 Face Detection

Dlib's HOG-based face detector is applied to locate faces within the frame.

3.4.3 Landmark Prediction

For each detected face, 68 facial landmarks are predicted. The eye regions are isolated using predefined landmark indices.

3.4.4 EAR Computation

The Eye Aspect Ratio (EAR) is computed for both eyes using six landmark points per eye. The EAR is facial structures required for eye analysis calculated as:

$$EAR = \frac{||p2-p6||+||p3-p5||}{2.||p1-p4||} \quad - (1)$$

These points correspond to the vertical and horizontal distances between specific eye landmarks. The average EAR is computed from both eyes:

$$EAR_{avg} = \frac{EAR_{left}+EAR_{rigth}}{2} \quad - (2)$$

This ratio decreases significantly when the eyes close, making it an effective metric for drowsiness detection [1], [5]. As illustrated in Figure 2, the six key eye landmarks (p_1 to p_6) are used to compute the Eye Aspect Ratio. The horizontal and vertical distances between these landmarks are visually represented, along with a sample EAR plot showing a temporal drop corresponding to eye closure or drowsiness. [12-15]

3.4.5 Drowsiness Condition Evaluation

If the average EAR falls below a threshold (e.g., 0.25), the frame counter is incremented. If the counter exceeds a defined frame threshold (e.g., 20), an alert is generated. This includes a visual warning on the screen and an audio alarm played using pygame. mixer. (Figure 2)

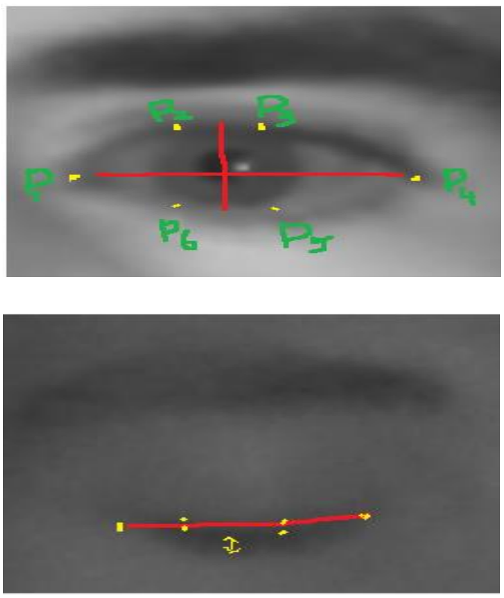

Figure 2 Ear Over Time (Noticeable Drop in Ear Indicates Eye Closure

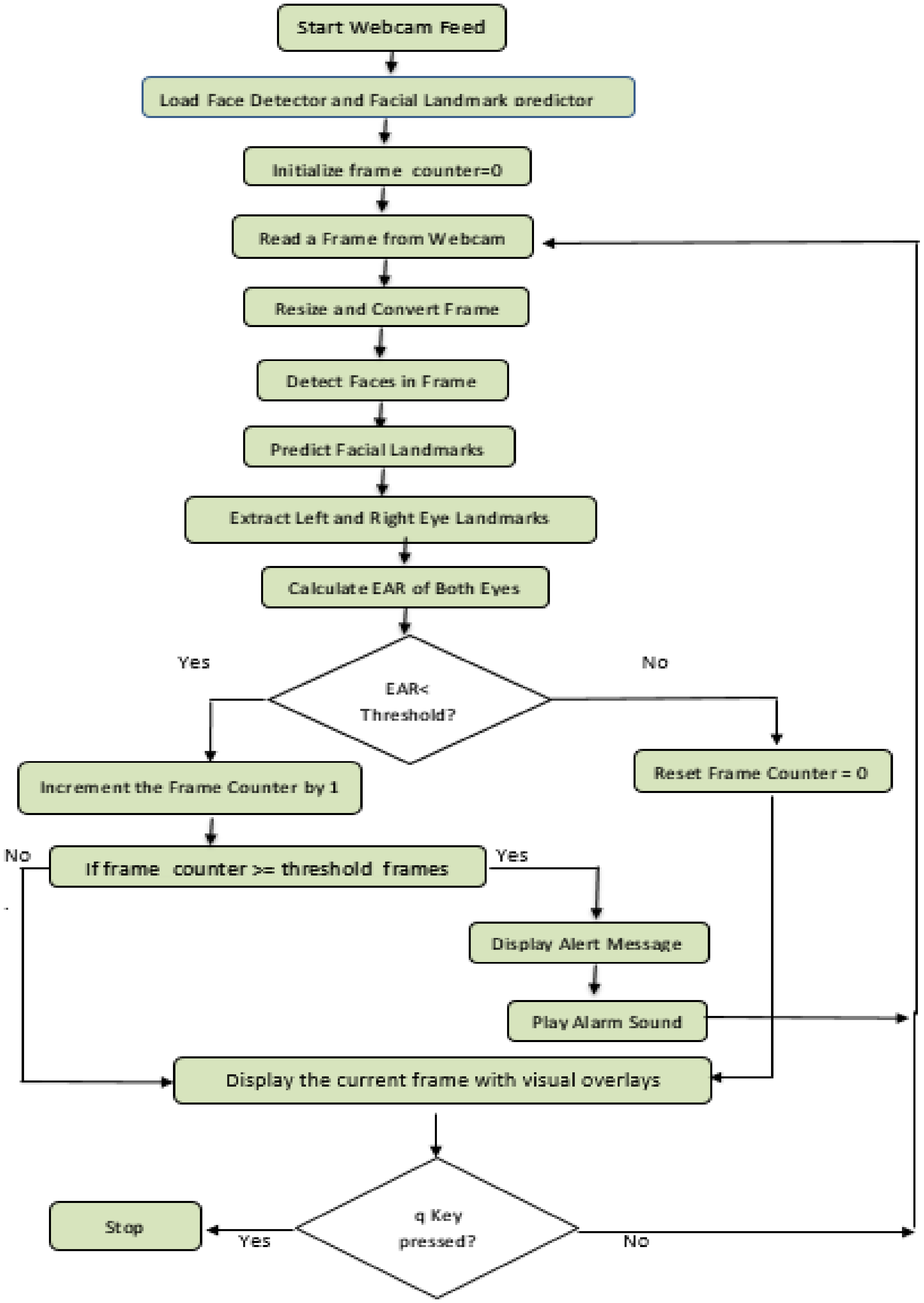

Figure 1 **Flowchart of The Proposed System**

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 09 September 2025
Page No: 3432-3438
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0503

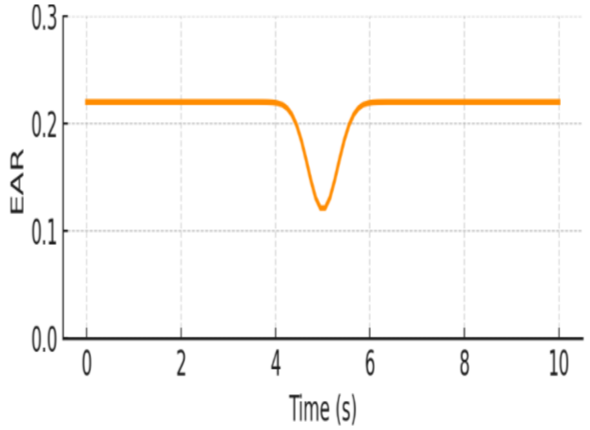

Figure 2 EAR Over Time (Noticeable Drop in EAR Indicates Eye Closure)

3.4.6 Reset Logic
If the EAR is above the threshold, the frame counter is reset to zero. This prevents false triggering due to normal blinks.

3.4.7 Frame Display
The processed frame is displayed with overlays showing eye contours and alert messages.

3.4.8 Loop Termination Condition
The loop continues until the user presses the 'q' key.

3.4.9 Resource Deallocation
Upon termination, the video capture object is released, and all OpenCV display windows are closed to ensure proper cleanup.

4. Experimental Setup

4.1 Hardware and Software Configuration
The proposed system was developed and evaluated on a laptop equipped with a 720p integrated webcam. The machine operated on Windows 10 and was powered by an Intel Core i5 processor. The implementation was carried out in Python, utilizing several open-source libraries for computer vision and facial landmark detection. Specifically, OpenCV 4.x was used for image acquisition, preprocessing, and visualization tasks. The Dlib library, along with its 68-point facial landmark predictor, was employed for facial feature localization [12]. Additionally, the pygame library was integrated to handle audio playback for alert notifications. This setup enabled real-time processing with minimal latency, suitable for continuous monitoring applications.

4.2 Parameter Configuration
For effective drowsiness detection, the system relies on a set of tuned threshold values and runtime parameters. The Eye Aspect Ratio (EAR) threshold was empirically set to 0.25, below which an eye is considered closed. To distinguish between normal blinking and prolonged closure, a frame-based counter was used. If the EAR remains below the threshold for 20 consecutive frames, the system triggers both visual and auditory alerts [10]. Furthermore, each frame captured from the webcam was resized to a fixed width of 450 pixels. This resizing helped standardize the computational load and ensured consistent detection performance across varying screen resolutions and hardware capabilities.

5. Results And Discussion

5.1 Accuracy
The drowsiness detection system was evaluated across multiple users and environments, including varying ambient lighting conditions. The system demonstrated a high degree of accuracy in identifying prolonged eye closure. It consistently triggered visual and audio alerts when the user's eyes remained closed for more than approximately one second, corresponding to the predefined threshold of 20 consecutive frames. Additionally, the system was able to differentiate between normal blinks and actual signs of drowsiness, thereby minimizing false positives [6],[11] [13][14].

5.2 Real-Time Performance
The application achieved real-time performance on a mid-range laptop equipped with an Intel i5 processor. During testing, the system maintained a processing speed between 15 to 25 frames per second (FPS), depending on the complexity of the scene and lighting conditions. The use of Dlib's optimized facial landmark detection model contributed to maintaining responsive and reliable tracking performance across all test cases.

5.3 Key Observations
- The system was effective in accurately detecting eye closure events based on the EAR thresholding logic.
- The false positive rate remained low, Th positi, particularly in cases of natural

blinking, due to the use of a frame counter mechanism.

- A minor drop in detection reliability was observed under low-light or heavily backlit conditions, suggesting a potential area for enhancement via lighting compensation or infrared camera support in future iterations

Conclusion

This paper presents the design and implementation of a real-time, vision-based drowsiness detection system utilizing Eye Aspect Ratio (EAR) and facial landmark tracking. The system effectively differentiates between natural blinking and prolonged eye closure by analyzing continuous frames and applying a threshold-based decision logic. Its lightweight architecture enables deployment on consumer-grade hardware without the need for specialized sensors, making it a cost-effective and non-intrusive safety solution. Given its accuracy, responsiveness, and ease of integration, the system is well-suited for use in driver monitoring, industrial safety, and other high-risk operational environments where alertness is critical. Future enhancements may include multimodal sensing (e.g., yawning detection, head pose estimation) and optimization for low-light scenarios to improve robustness and usability.

Future Work

While the current system demonstrates robust performance in real-time drowsiness detection using eye aspect ratio and facial landmarks, several enhancements can be pursued to further improve its reliability and applicability:

- Yawning Detection: Integrating mouth aspect ratio analysis to detect yawns can improve the overall accuracy and robustness of fatigue detection [15].
- Deep Learning Integration: Employing convolutional neural networks (CNNs) could enhance the system's ability to handle head pose variations, occlusions (e.g., glasses, hand movements), and partial facial visibility.
- Embedded Deployment: Porting the system to low-cost embedded platforms such as Raspberry Pi or NVIDIA Jetson Nano could facilitate real-world deployment in vehicles or workplace safety systems.
- Vehicle Telemetry Fusion: Combining the drowsiness detection system with vehicle telemetry data (e.g., steering patterns, lane drift, speed anomalies) could provide a comprehensive and intelligent driver assistance system.

References

[1]. K. Satish, A. Lalitesh, K. Bhargavi, M. Sishir Prem, and T. Anjali, (2020) Driver Drowsiness Detection," in Proc. IEEE Int. Conf. Trends in Electronics and Informatics (ICOEI), pp. 1120–1125.

[2]. S. Abtahi, B. Hariri, and L. Shirmohammadi, "YawDD: A yawning detection dataset," in Proc. ACM Multimedia Systems Conf. (MMSys), 2014, pp. 24–28.

[3]. Stancin, I.; Cifrek, M.; Jovic, A. A Review of EEG Signal Features and Their Application in Driver Drowsiness Detection Systems. Sensors 2021, 21, 3786. https://doi.org/10.3390/s21113786

[4]. T. Soukupová and J. Čech, "Real-Time Eye Blink Detection Using Facial Landmarks," in Proc. 21st Computer Vision Winter Workshop (CVWW), Rimske Toplice, Slovenia, 2016, pp. 1–8.

[5]. I. Kiran Kumar, V. Agarwal, and M. S. Reddy, "Image Recognition Based Driver Drowsiness Detection Using Python," in Proc. IEEE Int. Conf. Electrical and Automation Research (ICEARS), 2022, pp. 1–5.

[6]. R. N. Gandhi, P. Ambhorkar, A. Datir, P. Kale, and S. Pundkar, "Driver Drowsiness Detection System using Embedded System," Int. J. Computer Science and Information Technology Research, vol. 10, no. 2, pp. 103–107, 2022.

[7]. B. Alshaqaqi, A. S. Baquhaizel, M. E. Ouis, M. Boumehed, A. Ouamri, and M. Keche, “Driver Drowsiness Detection System,” in Proc. IEEE Int. Workshop on Systems, Signal Processing and Their Applications (WoSSPA), 2013, pp. 229–234.

[8]. A. Ritesh, N. Jagatia, and P. Deshmukh, “Driver Drowsiness Detection Using Machine Learning,” in Proc. IEEE Int. Conf. Advances in Smart Automation and Communication (ICAST), 2023, pp. 1–5.

[9]. L. Yang, H. Yang, H. Wei, Z. Hu, and C. Lv, “Video-Based Driver Drowsiness Detection With Optimised Utilization of Key Facial Features,” IEEE Trans. Intell. Transp. Syst., early access, July 2024, doi: 10.1109/TITS.2024.3411444

[10]. W. Deng and R. Wu, “Real-Time Driver-Drowsiness Detection System Using Facial Features,” IEEE Access, vol. 7, pp. 118081–118089, 2019, doi: 10.1109/ACCESS.2019.2936337.

[11]. [11]. H. K. Dua, S. Goel, and V. Sharma, “Drowsiness Detection and Alert System,” in Proc. IEEE Int. Conf. Advances in Computing, Communication Control and Networking (ICACCCN), 2018, pp. 1162–1166.

[12]. D. E. King, “Dlib-ml: A machine learning toolkit,” Journal of Machine Learning Research, vol. 10, pp. 1755–1758, 2009. [Online]. Available: http://dlib.net

[13]. Safarov F, Akhmedov F, Abdusalomov AB, Nasimov R, Cho YI. Real-Time Deep Learning-Based Drowsiness Detection: Leveraging Computer-Vision and Eye-Blink Analyses for Enhanced Road Safety. Sensors.,2023,23(14):6459. https://doi.org/10.3390/s23146459

[14]. Chaabene S, Bouaziz B, Boudaya A, Hökelmann A, Ammar A, Chaari L. Convolutional Neural Network for Drowsiness Detection Using EEG Signals. Sensors. 2021; 21(5):1734. https://doi.org/10.3390/s21051734.

[15]. Fu, Biying, et al. "A survey on drowsiness detection–modern applications and methods." IEEE Transactions on Intelligent Vehicles ,2024