Raspberry Pi-Based Driver Alertness Monitoring Using Facial Landmark Analysis

1st Abdurahman*
Department of Computer System
Sriwijaya University
Palembang, Indonesia
abdurahman@unsri.ac.id

2nd Ahmad Zarkasi
Department of Computer Science
Sriwijaya University
Palembang, Indonesia
ahmadzarkasi@unsri.ac.id

3rd M. Aren Saputra
Department of Computer Engineering
Sriwijaya University
Palembang, Indonesia
arensptr10@gmail.com

4th Sutarno
Department of Computer System
Sriwijaya University
Palembang, Indonesia
sutarno@unsri.ac.id

5th Aditya P. P. Prasetyo
Department of Computer Engineering
Sriwijaya University
Palembang, Indonesia
aditrecca@gmail.com

6th Sarmayanta Sembiring
Department of Computer Engineering
Sriwijaya University
Palembang, Indonesia
yanta@unsri.ac.id

*Corresponding author: abdurahman@unsri.ac.id
Received: 2026-06-15; **Accepted:** 2026-08-12

***Abstract*—Drowsiness while driving is one of the conditions that can reduce a driver's concentration and alertness; therefore, an early warning system capable of directly monitoring the driver's condition is required. This study aims to design a drowsiness detection system for four-wheeled vehicle drivers based on Raspberry Pi using the Facial Landmark Detection (FLD) method with the Eye Aspect Ratio (EAR) parameter. The system is designed by utilizing a webcam as the input device to capture the driver's facial image, Raspberry Pi as the data processing unit, a buzzer as the warning output device, and a push button as the manual reset button. The detection process is carried out using MediaPipe Face Mesh to obtain landmark points in the eye area. The EAR value is used to determine whether the eyes are open or closed, while the number of eye blinks is used as an indicator to determine the driver's drowsiness condition. Testing was conducted by measuring EAR values at various distances and face positions, as well as by testing the Eye Blink Counter by comparing the number of blinks calculated by the system with manual observations. The test results show that the system is able to detect faces and eye landmarks in most testing conditions with distance variations ranging from 10 to 95 centimeters. In the Eye Blink Counter test conducted 24 times with a duration of 60 seconds, an average error of 2.49 percent was obtained.**

Keywords—drowsiness detection, Raspberry Pi, facial landmark detection, eye aspect ratio

I. INTRODUCTION

Driver safety is one of the important aspects of the transportation system, especially in four-wheeled vehicles used for both short-distance and long-distance travel. One condition that can reduce driving safety is drowsiness. Drowsiness while driving can lead to decreased concentration, reduced alertness, slower responses to road conditions, and an increased risk of traffic accidents [1]. Based on a report by the World Health Organization, road traffic accidents remain a serious problem, causing approximately 1.19 million deaths each year and becoming one of the leading causes of death among people aged 5 to 29 years [2].

In Indonesia, road traffic accident problems are also greatly influenced by driver-related factors. A study conducted on the Batang–Semarang Toll Road showed that driver-related factors were the dominant cause of road traffic accidents, with a percentage of 89.82 percent. Among these factors, drowsiness was one of the main causes, with a percentage of 61.08 percent [3]. These data indicate that driver drowsiness requires serious attention because it can affect the driver's ability to maintain concentration and make decisions while driving.

Several approaches have been used to detect driver drowsiness, such as monitoring vehicle conditions, measuring physiological signals, and visually analyzing the driver's face. Visual-based approaches have become one of the widely used methods because they do not require direct contact with the driver's body [4]. A camera-based system can monitor facial changes, particularly around the eyes, which can provide important visual cues for detecting driver drowsiness. One commonly used parameter for analyzing eye conditions is the Eye Aspect Ratio (EAR), which quantifies the geometric relationship between the vertical and horizontal distances of the eye based on facial landmark points [5].

To implement such visual-based approaches efficiently inside vehicles, current advancements in intelligent transportation systems have increasingly relied on edge computing paradigms to facilitate real-time data processing, thereby reducing latency and minimizing dependencies on cloud storage. For instance, the deployment of cost-effective microcomputers, such as the Raspberry Pi, has proven highly effective for localized edge intelligence in vehicular environments [10]. However, embedding video-based analysis into these systems demands highly efficient computational methods to maintain stability under limited hardware constraints. To address this, current research highlights the necessity of optimizing the computational efficiency of the EAR metric, ensuring that fatigue and drowsiness indicators can be extracted and evaluated accurately within real-time embedded systems without causing significant processing delays [11].

In this study, facial landmark detection is employed to identify key points around the driver's eyes. MediaPipe Face Mesh is used to obtain the facial landmark points, while the

EAR value is calculated from these points to determine whether the driver's eyes are open or closed [5]. In addition, the system calculates the number of eye blinks within a certain observation duration as an indicator to determine the driver's drowsiness condition. Raspberry Pi is used as the processing unit because it has a relatively small size, is easy to implement in embedded systems, and is capable of performing image processing in real time [6].

Several previous studies have explored driver drowsiness detection using various algorithms. Traditional methods such as Haar-Cascade classifiers are computationally light but highly sensitive to lighting changes and partial face occlusions. On the other hand, Deep Learning approaches like Convolutional Neural Networks (CNN) offer high accuracy but require significant computational resources, making them less efficient for real-time edge computing on constrained devices like the Raspberry Pi without external hardware accelerators. Furthermore, many existing systems lack comparative efficiency when dealing with dynamic head movements during driving. To address this gap, this study proposes a lightweight yet robust alternative by implementing MediaPipe Face Mesh. Unlike traditional methods, MediaPipe provides a highly optimized 3D facial landmark topology utilizing CPU-based acceleration, which ensures high-speed real-time processing (above 15 frames per second) directly on a Raspberry Pi 5. This approach bridges the gap between the need for high-accuracy landmark detection and the low computational power typical of embedded vehicular systems.

Based on these problems, this study aims to design a drowsiness detection system for four-wheeled vehicle drivers based on Raspberry Pi using the Facial Landmark Detection (FLD) method with the EAR parameter. The designed system uses a webcam as the input device, Raspberry Pi as the processing unit, a buzzer as the warning output device, and a push button as the manual reset button. With this system, drivers are expected to receive an early warning when indications of drowsiness appear while driving.

II. RESEARCH METHOD

A. System Design

This study was conducted by designing a drowsiness detection system for four-wheeled vehicle drivers based on Raspberry Pi using the FLD method with the EAR parameter. The system consists of a power supply, webcam, push button, Raspberry Pi 5, buzzer, and a drowsiness detection program executed using Python The webcam is used as an input device to capture the driver's facial image in real time, while the push button is used as a manual reset button when the system needs to be returned to its initial condition. The Raspberry Pi 5 functions as the data processing unit to perform face detection, eye landmark detection, EAR calculation, and eye blink counting. The buzzer is used as an output device that will be activated when the driver is indicated to be drowsy. The block diagram of the designed system is shown in Fig. 1.

In the system shown in Fig. 1, the webcam is used to capture the driver's facial image in real time, while the push button functions as a reset button to return the system to its initial condition. The captured image is then sent to the Raspberry Pi 5 for further processing. The processing begins with the detection of facial landmark points using MediaPipe Face Mesh, particularly in the eye area. The obtained landmark points are then used to calculate the Eye Aspect Ratio (EAR) value to identify whether the eyes are open or closed. In addition, the system also counts the number of eye blinks through the Eye Blink Counter feature during a certain observation period. The results of this analysis are used as the basis for determining the driver's level of alertness. When the observed indicators show a drowsy condition according to the predetermined threshold, the buzzer will be activated as a warning alarm.

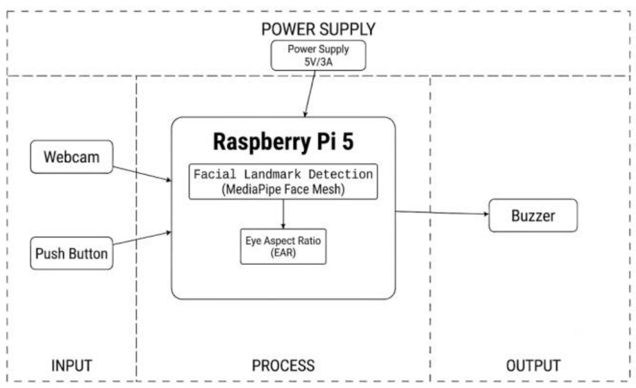

Fig. 1. System Block Diagram

B. Facial Landmark Detection

Facial Landmark Detection is a stage used to identify characteristic points on the driver's face, particularly in the eye area. In this study, landmark detection is performed using MediaPipe Face Mesh, which is capable of generating facial landmark points in real time with good accuracy. The obtained landmarks are then used as references in the EAR calculation process.

MediaPipe Face Mesh provides a dense 3D facial geometry output consisting of 468 landmark points in real-time. The coordinates are initially expressed as normalized values between 0.0 and 1.0. These values are then converted into precise pixel coordinates by multiplying them with the width and height of the captured image frame. To optimize computational load, the system filters these 468 points and strictly extracts only six specific landmark indices for each eye. For instance, the right eye utilizes indices [362, 385, 387, 263, 373, 380], while the left eye utilizes indices [33, 160, 158, 133, 153, 144]. These localized points are mapped using Euclidean distance to calculate the vertical and horizontal dimensions of the eyes, which are critical for the subsequent EAR computation.

C. Eye Aspect Ratio

The EAR is used to determine whether the eyes are open or closed based on the ratio of the vertical and horizontal distances in the eye area. Before calculating the EAR value, the system first calculates the distance between eye landmark points using Euclidean Distance. The distance calculation between two landmark points is shown in (1).

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{1}$$

In (1), d represents the distance between two landmark points, x_1 and x_2 represent the coordinates on the x-axis, while y_1 and y_2 represent the coordinates on the y-axis. After the distances between landmark points are obtained, the EAR value is calculated using six eye landmark points, as shown in (2).

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

In (2), points p_1 and p_4 are used to calculate the horizontal distance of the eye, while points p_2, p_3, p_5, and p_6 are used to calculate the vertical distance of the eye. The EAR value is used as the basis for determining the condition of the driver's eyes. If the EAR value is below the threshold of 0.22, the eyes are categorized as closed. Conversely, if the EAR value is equal to or above 0.22, the eyes are categorized as open [7].

D. *Eye Blink Counter*

The Eye Blink Counter is used to count the number of eye blinks based on changes in the EAR value. The system reads the eye condition in each camera frame. When the EAR value remains below the threshold for several consecutive frames, the system considers the eyes to be in a closed condition. In this study, one blink is counted when the eyes are detected as closed for three consecutive frames and then return to an open condition.

After the eyes reopen, the total blink value increases by one, and the blink counter is reset to detect the next blink. The determination of drowsiness indication is based on the total number of eye blinks within a certain observation duration. In this system, the total blink limit used is 27 blinks within a duration of 60 seconds [8]. If the number of eye blinks reaches or exceeds this limit, the driver is categorized as indicated to be drowsy, and the buzzer will be activated as an early warning. Conversely, if the number of blinks is below this limit, the driver is categorized as being in a normal condition.

E. *Testing Method*

To ensure the robustness and replicability of the proposed system, the experimental validation was conducted under varying practical conditions. The evaluation involved 5 different participants (3 males and 2 females) to account for variations in facial structures, eye shapes, and the use of accessories such as glasses. The testing environments included a stationary vehicle cabin and an indoor driving simulator setup. Furthermore, the experiments were tested under three distinct lighting conditions: bright natural daylight, artificial cabin lighting, and low-light evening environments.

Testing was conducted to determine the system's ability to read the EAR value and count the number of the driver's eye blinks. The EAR test was carried out with variations in the distance between the camera and the driver's face, as well as several face position conditions, such as facing forward, turning to the left, turning to the right, briefly looking downward, and wearing clear or prescription glasses. This test aimed to determine the system's ability to detect faces and eye landmarks under several observation conditions.

$$\text{Error}(\%) = \left|\frac{B_s - B_m}{B_m}\right| \times 100 \quad (3)$$

In (3), (Bs) represents the number of eye blinks counted by the system, while (Bm) represents the number of eye blinks based on manual observation. The (Bm) value is used as reference data because it is obtained through direct observation during the testing process. The error value indicates the difference between the system reading results and the manual observation results. The smaller the error value obtained, the closer the system calculation results are to the reference value.

After the error value for each test is obtained, the average error is calculated to determine the overall error level of the system. The calculation of the average error is shown in (4).

$$\text{Average Error}(\%) = \frac{\sum_{i=1}^{n} E_i}{n} \quad (4)$$

In (4), (E_i) represents the error value obtained from each testing data or trial, while (n) represents the total number of testing data. The average error is used to determine the overall error level of the system from all conducted trials. This value serves as a reference for assessing the accuracy level of the system in counting the number of the driver's eye blinks.

III. RESULTS AND DISCUSSION

This section discusses the implementation and testing results of the Raspberry Pi-based driver drowsiness detection system. Testing was conducted to determine the system's ability to detect eye landmarks using MediaPipe Face Mesh, calculate EAR value, count the number of eye blinks through the Eye Blink Counter, and activate the buzzer when the driver is indicated to be drowsy. The test results are presented in the form of images, tables, and discussions to explain the overall performance of the system.

A. *System Implementation Result*

The implementation results show that the driver drowsiness detection device has been successfully designed in the form of a prototype. The system consists of a webcam, Raspberry Pi 5, buzzer, push button, and power supply. The webcam is used to capture the driver's facial image, the Raspberry Pi 5 is used as the data processing unit, the buzzer is used as a warning device, while the push button is used as a manual reset button. The hardware implementation results of the system are shown in Fig. 2.

Fig. 2. Hardware prototype of the driver drowsiness detection system.

In the system shown in Fig. 2, the webcam is mounted on the top part of the device so that it can directly capture the driver's facial image. The main components of the system are placed inside a casing to make the device neater and easier to use. When the system is operated, the webcam captures the driver's face in real time, and the Raspberry Pi 5 processes the image to detect eye landmarks, calculate the EAR value, and count the number of eye blinks. If the system detects an indication of drowsiness, the buzzer will be activated as an early warning for the driver.

B. Eye Aspect Ratio Testing Result

The EAR test was conducted to determine the system's ability to detect faces, read eye landmarks, and determine eye conditions based on the EAR value. The test was carried out under several driver face conditions, such as facing the camera, wearing glasses, and facing sideways. An EAR value above the threshold of 0.22 indicates an open-eye condition, while an EAR value below the threshold of 0.22 indicates a closed-eye condition. A summary of the EAR testing results is shown in Table I.

Based on Table I, the system is able to detect facial and eye landmarks under most testing conditions. EAR values above the threshold of 0.22 are classified as an open-eye condition, while EAR values below the threshold are classified as a closed-eye condition. The test results show that the system can still detect eye landmarks when the subject faces forward, wears glasses, looks downward, or slightly turns to the side. However, when the face is turned too far sideways, the eye landmarks cannot be detected properly, causing the EAR value to be unobtainable. This condition indicates that the success of EAR detection is influenced by the face position and the visibility of the eye area to the camera

TABLE I. Eye Aspect Ratio Test Results

No	Detection Image	Testing Condition	Result	Eye Condition
1		Face looking forward at close distance with an EAR value of 0.29.	Detected	Open eyes
2		Face looking forward with an EAR value of 0.19.	Detected	Closed eyes
3		Face looking forward at medium distance with an EAR value of 0.25	Detected	Open eyes
4		Female driver facing forward with an EAR value of 0.28.	Detected	Open eyes
5		female driver turned their heads slightly to the side with an EAR value of 0.33.	Detected	Open eyes
6		The female driver turned sideways with an EAR value of 0.20.	Detected	Closed eyes
7		The male driver is wearing glasses and facing slightly to the side with an EAR value of 0.30.	Detected	Open eyes
8		Male driver wearing glasses and looking down with an EAR value of 0.15.	Detected	Closed eyes
9		Male driver wearing glasses and facing forward with an EAR value of 0.28.	Detected	Open eyes
10		The face is tilted too far to the side so that the EAR value cannot be obtained.	Not Detected	EAR unavailable

C. Eye Blink Counter Testing Results

The Eye Blink Counter test was conducted to determine the system's ability to count the number of the driver's eye blinks based on the EAR value. In this test, the number of blinks counted by the system was compared with the number of blinks obtained through manual observation. Each test was conducted for 60 seconds. The results of the Eye Blink Counter test are shown in Table II.

Based on Table II, the Eye Blink Counter test was conducted 24 times, with each test lasting 60 seconds. The test results show that the system is able to count the number of eye blinks properly, as most of the system readings have the same values as the manual observation results. Differences in readings occurred in the 5th, 8th, 18th, 19th, and 24th tests. These differences may be caused by blinks that occurred too quickly, changes in face position, head movements, or eye landmarks that were not detected perfectly in several frames.

Overall, the average error value obtained was 2.49 percent. This value indicates that the system has a relatively low error rate in counting the number of the driver's eye blinks. In addition, the system is also able to determine drowsiness status based on the predetermined blink count threshold. In the 3rd, 9th, 13th, and 24th tests, the system identified a drowsy status because the number of blinks reached or exceeded the threshold of 27 blinks within a duration of 60 seconds.

TABLE II. Eye Blink Counter Test Result

No	Duration (Seconds)	Manual Blink	System Blink	Error (%)	Drowsiness Status
1	60	20	20	0.00	Normal
2	60	17	17	0.00	Normal
3	60	27	27	0.00	Drowsy
4	60	22	22	0.00	Normal
5	60	18	16	11.11	Normal
6	60	24	24	0.00	Normal
7	60	19	19	0.00	Normal
8	60	25	22	12.00	Normal
9	60	27	27	0.00	Drowsy
10	60	25	25	0.00	Normal
11	60	22	22	0.00	Normal
12	60	24	24	0.00	Normal
13	60	27	27	0.00	Drowsy
14	60	25	25	0.00	Normal
15	60	19	19	0.00	Normal
16	60	21	21	0.00	Normal
17	60	22	22	0.00	Normal
18	60	25	20	20.00	Normal
19	60	23	21	8.70	Normal
20	60	25	25	0.00	Normal
21	60	25	25	0.00	Normal
22	60	21	21	0.00	Normal
23	60	19	19	0.00	Normal
24	60	25	27	8.00	Drowsy
Average Error %				2.49	

To objectively evaluate the drowsiness detection performance beyond the average blink-counting error, a quantitative metric evaluation was conducted based on the 24 testing scenarios in Table II. The system correctly identified 3 true drowsy states (True Positives/TP) and 20 normal states (True Negatives/TN). However, there was 1 instance where the system falsely alarmed a normal state as drowsy (False Positive/FP in test 24), and 0 instances of missed drowsiness (False Negatives/FN). Based on a threshold limit of 27 blinks, the system achieved a classification Accuracy of 95.83%. The Precision of the system was recorded at 75.00%, with a Recall (Sensitivity) of 100%, indicating that the system successfully detected all actual drowsy conditions without missing any. Consequently, the F1-Score reached 85.71%. These metrics demonstrate that while the system is highly sensitive to drowsiness, minor false positives may occur due to rapid consecutive blinking or sudden head tilts.

IV. Conclusion

Based on the results of the design and testing that have been carried out, the Raspberry Pi-based drowsiness detection system for four-wheeled vehicle drivers has been successfully designed and implemented using the FLD method with the EAR parameter. This system is able to utilize a webcam to capture the driver's facial image in real time and then process the image using MediaPipe Face Mesh to detect landmark points in the eye area. The EAR value is used to determine whether the eyes are open or closed, while the Eye Blink Counter is used to count the number of eye blinks as the basis for determining indications of drowsiness.

The test results show that the system is able to detect facial and eye landmarks under most testing conditions, such as when the face is facing forward, when glasses are worn, when looking downward, and when slightly turning to the side. However, when the face is turned too far sideways, the system cannot properly detect the eye landmarks, causing the EAR value to be unobtainable. In the Eye Blink Counter test conducted 24 times, with each test lasting 60 seconds, an average error of 2.49 percent was obtained. These results demonstrate that the proposed MediaPipe and EAR-based system on a Raspberry Pi 5 provides a reliable and lightweight edge-computing prototype for early drowsiness detection. However, the current experimental validation reveals certain limitations; the system's performance heavily relies on adequate lighting and standard facial positioning, as extreme head rotations or zero-light environments degrade the landmark detection accuracy. Therefore, while this prototype successfully addresses the need for a low-cost embedded alert system, claims regarding its absolute robustness in dynamic real-world driving require moderation. Future developments should incorporate infrared (IR) cameras for night-time driving and larger-scale participant testing to improve the system's generalization under extreme vehicular conditions.

Acknowledgment

The first author would like to express gratitude to the Faculty of Computer Science, the Strategic Pervasive Computing and Intelligent Embedded Systems Research Group (SPCIES RG), COMNETS RG, and the Embedded Systems, Control System, and Robotic Laboratory of Universitas Sriwijaya for the support provided in the implementation of this research.

References

[1] A. Moradi, S. S. Hashemi Nazari, and K. Rahmani, "Sleepiness and the risk of road traffic accidents: A systematic review and meta-analysis of previous studies," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 65, pp. 620–629, 2019, doi: 10.1016/j.trf.2018.09.013.

[2] World Health Organization, "Road traffic injuries," World Health Organization Fact Sheets, 2023. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries

[3] G. Zhang, K. K. W. Yau, X. Zhang, and Y. Li, "Traffic accidents involving fatigue driving and their extent of casualties," *Accident Analysis & Prevention*, vol. 87, pp. 34–42, 2016, doi: 10.1016/j.aap.2015.10.033.

[4] N. Kamarudin, N. A. Jumadi, N. L. Mun, N. C. Keat, A. H. K. Ching, W. M. H. W. Mahmud, M. Morsin, and F. Mahmud, "Implementation of Haar Cascade Classifier and Eye Aspect Ratio for Driver Drowsiness

Detection Using Raspberry Pi," *Universal Journal of Electrical and Electronic Engineering*, vol. 6, no. 5B, pp. 67–75, 2019, doi: 10.13189/ujeee.2019.061609.

[5] V. U. Maheswari, R. Aluvalu, M. V. V. Prasad Kantipudi, K. K. Chennam, K. Kotecha, and J. R. Saini, "Driver Drowsiness Prediction Based on Multiple Aspects Using Image Processing Techniques," *IEEE Access*, vol. 10, pp. 54980–54990, 2022, doi: 10.1109/ACCESS.2022.3176451.

[6] J. He, W. Choi, Y. Yang, J. Lu, X. Wu, and K. Peng, "Detection of driver drowsiness using wearable devices: A feasibility study of the proximity sensor," *Applied Ergonomics*, vol. 65, pp. 473–480, 2017, doi: 10.1016/j.apergo.2017.02.016.

[7] P. K. Singh, A. Gupta, M. Upadhyay, A. Jain, M. Khari, and P. S. Lamba, "Multimodal Driver Drowsiness Detection From Video Frames," *Journal of Mobile Multimedia*, vol. 19, no. 2, pp. 567–586, 2022, doi: 10.13052/jmm1550-4646.19210.

[8] T. Zhu, C. Zhang, T. Wu, Z. Ouyang, H. Li, X. Na, J. Liang, and W. Li, "Research on a Real-Time Driver Fatigue Detection Algorithm Based on Facial Video Sequences," *Applied Sciences*, vol. 12, no. 4, Art. no. 2224, 2022, doi: 10.3390/app12042224..

[9] C. Lugaresi et al., "MediaPipe: A Framework for Building Perception Pipelines," *arXiv preprint arXiv:1906.08172*, 2019, doi: 10.48550/arXiv.1906.08172.

[10] S. Ahmad, S. Malik, D.-H. Park, and D. Kim, "Design of Lightweight Driver-Assistance System for Safe Driving in Electric Vehicles," *Sensors*, vol. 19, no. 21, Art. no. 4761, 2019, doi: 10.3390/s19214761.

[11] C. B. S. Maior, M. J. das C. Moura, J. M. M. Santana, and I. D. Lins, "Real-time classification for autonomous drowsiness detection using eye aspect ratio," *Expert Systems with Applications*, vol. 158, Art. no. 113505, 2020, doi: 10.1016/j.eswa.2020.113505.