IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.141186

INTELLIGENT DRIVER MONITORING FOR CAR SAFE JOURNEY

Mrs. Chaithra B V[1], Devaraju J[2], Hanish B N[3], Karthik[4], Mithun K G[5]

Assistant Professor, ECE, East West Institute of Technology, Bengaluru, India[1]

Student, ECE, East West Institute of Technology, Bengaluru, India[2]

Student, ECE, East West Institute of Technology, Bengaluru, India[3]

Student, ECE, East West Institute of Technology, Bengaluru, India[4]

Student, ECE, East West Institute of Technology, Bengaluru, India[5]

Abstract: Driver drowsiness is a major cause of road accidents, resulting in severe injuries and fatalities. This project proposes a Smart Driver Drowsiness Detection System using Raspberry Pi and advanced Object Detection (ADAS & CAOA) algorithms to monitor driver alertness in real time. The system employs Eye Aspect Ratio (EAR) and Mouth Aspect Ratio (MAR) to analyse eye closure, yawning, and head movements through continuous camera monitoring. When fatigue is detected, it triggers alerts, activates hazard lights, or sends SOS messages. Using Virtual Network Computing (VNC) for remote monitoring, the system enhances road safety with an intelligent, scalable, and cost-effective solution.

Keywords: Eye Aspect Ratio (EAR), Virtual Network Computing (VNC), Mouth Aspect Ratio (MAR), Object Detection (ADAS & CAOA).

I. INTRODUCTION

Road safety continues to be a major global concern, with driver drowsiness and fatigue contributing to a significant number of accidents every year. Unlike alcohol or drug impairment, drowsiness develops gradually and often goes unnoticed until it becomes dangerous. Fatigue affects attention, slows reaction time, reduces decision-making ability, and can lead to micro-sleep episodes lasting only a few seconds—yet long enough to cause serious crashes. Visible signs such as prolonged eye closure, frequent blinking, yawning, and head nodding are early indicators of drowsiness. Traditional prevention methods like taking rest breaks or relying on the driver's self-awareness are often unreliable, and many modern vehicle safety systems focus on external hazards rather than monitoring the driver's internal state. This makes it essential to develop an intelligent solution that can detect fatigue early and intervene before an accident occurs.

To address this need, the Smart Driver Drowsiness Detection System uses Raspberry Pi and advanced object-detection algorithms to monitor the driver in real time. A dashboard-mounted camera continuously records facial features, while computer-vision techniques using OpenCV and machine-learning models such as TensorFlow analyze key indicators like PERCLOS, Eye Aspect Ratio (EAR), Mouth Aspect Ratio (MAR), yawning frequency, and head movement. Based on these inputs, the system determines whether the driver is alert or fatigued. When drowsiness is detected, it issues alerts such as buzzer alarms, voice warnings, or seat vibrations. If the driver does not respond, the system escalates by activating hazard lights, slowing the vehicle, or sending SOS messages. Built on Raspberry Pi, the solution is affordable, portable, and compatible with both new and existing vehicles, with VNC support for remote monitoring.
Understanding the causes of fatigue underscores the importance of such a system. Drowsiness can result from lack of sleep, physical exhaustion, night driving, illness, or medication. Human alertness naturally decreases at night due to circadian rhythms, and physiological factors like reduced adrenaline also contribute to sleepiness. By analyzing these behavioural cues, the Smart Driver Drowsiness Detection System provides a proactive, AI-driven approach that helps prevent accidents and enhances road safety.

II. METHODOLOGY

The methodology combines real-time video acquisition, image pre-processing, YOLO detection, feature extraction, temporal analysis, and actuation. A sliding-window based temporal buffer computes statistics like PERCLOS and

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.141186

average EAR. The decision module fuses cues (eye closure duration, yawning events, head pitch) to declare drowsiness.

- Camera captures at 30 FPS (or configured 15–20 FPS for performance).
  Optional
- Pre-processing: resize, histogram equalization, CLAHE to normalize lighting.
  YOLO model inference to detect face/eyes/mouth and bounding boxes.
- Facial landmark or ROI-based fine-grained eye patches processing to compute EAR.
- Temporal analysis: maintain buffer of recent EAR values, compute PERCLOS over 60-second window.
- Decision: if EAR below threshold for consecutive frames or PERCLOS exceeds limit or repeated yawns detected → trigger alerts.

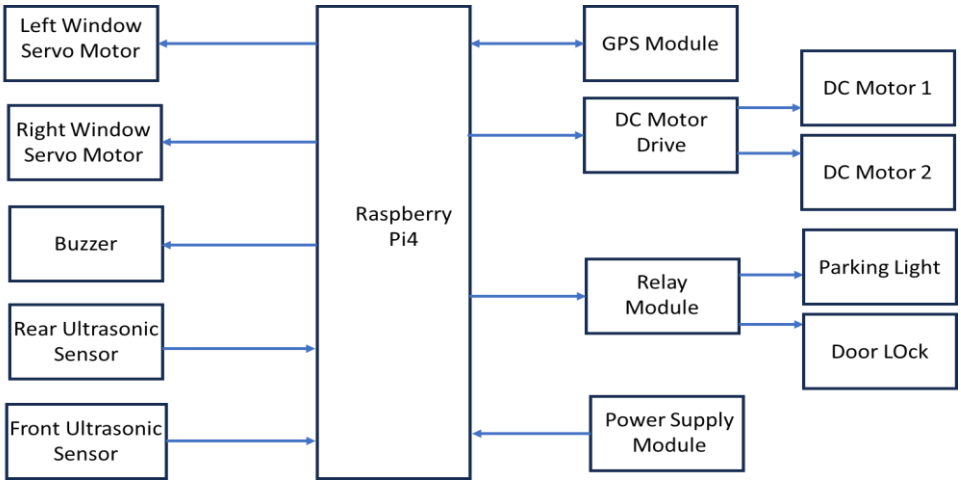

Fig: 1 Block diagram of Intelligent Driver Monitoring For Car Safe Journey

- The Raspberry Pi 4 connects to peripherals as follows (example pin mapping)
- **Servo motor(s):** Connected to GPIO18 (BCM) for PWM control, powered by an external 5V source with a common ground. The PWM signal adjusts the servo position for camera or steering control functions. Ensure stable 5V supply for reliable movement.
- **Relay module:** IN pins connected to GPIO23 and GPIO24, VCC to 5V, and GND common with Raspberry Pi. Relays switch 12V lines for lights or door locks safely. Provides electrical isolation for high-voltage control.
- **Buzzer:** Controlled through GPIO17 using an NPN transistor driver circuit. The buzzer operates on 5V with a common ground, and a base resistor is added for protection. A flyback diode is used to prevent voltage spikes.
- **L298N driver:** IN1, IN2, and ENA are connected to GPIO22, GPIO27, and a suitable GPIO pin. The motor supply (7–12V) is isolated from logic power but shares a common ground. Enables bidirectional control of DC motors for braking or motion.
- **Power and Grounding:** All components share a common ground for signal stability. Use separate regulated supplies for high-current devices. Logic-level MOSFETs or opto-isolators can be added for added safety and automotive robustness.

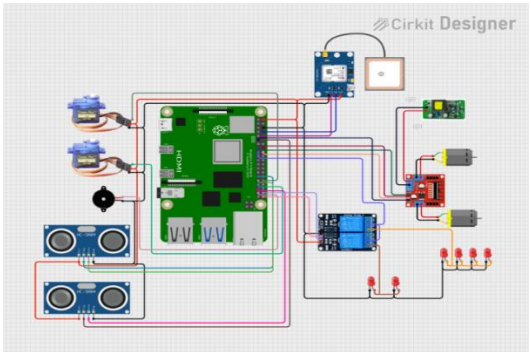

Fig: 2 Circuit Diagram

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.141186

III. IMPLEMENTATIONS

3.1 HARDWARE INTEGRATION

Assemble Raspberry Pi 4 in a safe enclosure and mount the USB 1080p camera on the dashboard or steering column to capture the driver's face. Use external regulated 5V and motor power supplies for servos and L298N motors. Wire the relay module to the vehicle's parking light circuit through proper fuses and relays; DO NOT directly tap into automotive wiring without understanding vehicle electrical systems. Use opto-isolated relay modules for production use.

3.2 SOFTWARE STACK AND ENVIRONMENT

Install Raspberry Pi OS, Python 3, OpenCV, and the required deep learning frameworks. For YOLO, either train using PyTorch on a desktop GPU and export to ONNX/TFLite for Pi inference or use a lightweight training pipeline. Use virtualenv to manage dependencies and leverage multiprocessing to keep capture and inference on separate threads to maintain throughput.

3.3 YOLO TRAINING PIPELINE

1. Data Collection: Record videos of drivers in varied lighting, with/without glasses, and annotate frames for 'open_eye', 'closed_eye', 'yawn', 'face'.
2. Annotation: Use tools like LabelImg or Roboflow; export YOLO-format labels.
3. Augmentation: Apply brightness, contrast, blur, and rotation augmentations to simulate real-world variation.
4. Train: Use a GPU machine to train (e.g., YOLOv5 small or tiny) with transfer learning. Monitor metrics (mAP, loss) and validate.
5. Export: Convert best weights to ONNX, then to TFLite/Edge TPU if using accelerator, or keep as PyTorch for optimized runtime.

3.3 FLOW OF PROGRAM

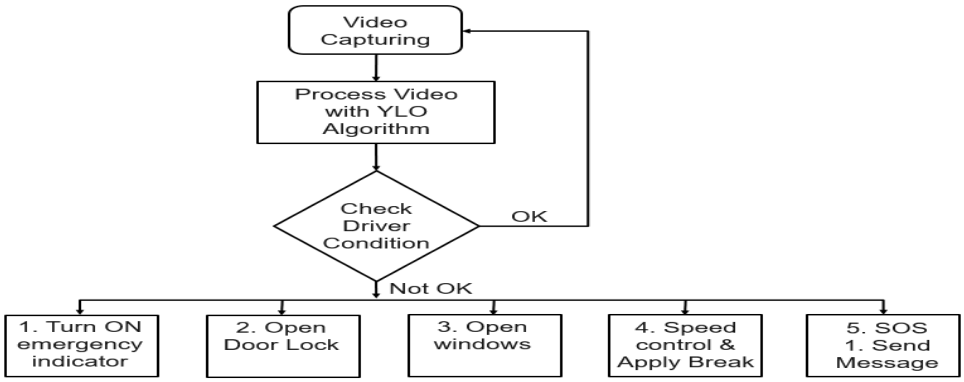

Fig: 3 Program floe of Intelligent Driver Monitoring For Car Safe Journey

The system begins by continuously capturing video from the camera and processing this video stream using the YOLO algorithm to detect the driver's behaviour and overall condition. After analyzing each frame, the system evaluates whether the driver is in a safe and normal state. If the driver's condition is detected as normal, the system simply continues capturing and processing video in a continuous loop. However, if the driver's condition is found to be abnormal—such as in cases of drowsiness, inattention, or medical distress—the system initiates a sequence of corrective safety actions. These actions include turning on the emergency indicator to alert surrounding vehicles, automatically unlocking the doors for emergency access, opening the windows for ventilation, controlling the vehicle's speed and applying the brakes to bring the car to a safe stop, and sending an SOS message to predefined contacts or emergency services. This ensures that the system not only detects unsafe driver conditions but also responds immediately with multiple safety measures to prevent accidents and enable timely assistance.

3.4 ALGORITHM FOR PROGRAM FLOW

- The system initializes the Raspberry Pi camera, facial-landmark detector, and sets threshold values for Eye Aspect Ratio and alert duration.
- The program enters a continuous while-loop where each frame from the camera is captured for processing.
- The system detects the driver's face in the frame and extracts the eye landmarks required for EAR calculation.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025
DOI: 10.17148/IJARCCE.2025.141186

- The Eye Aspect Ratio is computed, and the program checks whether the EAR remains below the threshold for a fixed number of consecutive frames.
- If prolonged low EAR is detected, the system triggers an alert and continues monitoring the eyes during the 5–10 second alert period.
- If the driver does not respond and the eyes remain closed for the entire alert duration, the system sends the command to safely stop the car from motion.
- If the driver opens the eyes before the alert duration ends, the alert is cancelled and the loop continues normal monitoring.

3.4.1 ALGORITHM FOR EYE ASPECT RATIO (EAR)

The system begins by initializing the Raspberry Pi camera, loading the facial-landmark detector, and setting the EAR threshold, consecutive-frame limit, and alert duration. Once this setup is complete, the program enters a continuous while-loop where each incoming frame is captured and processed. For every frame, the face is detected and eye landmarks are extracted so that the Eye Aspect Ratio can be calculated. When the EAR remains below the threshold for the required number of consecutive frames, the system triggers an alert and starts counting the alert duration. If this alert period completes 5–10 seconds without any improvement in the EAR, the system issues the command to safely stop the car from motion. This ensures that prolonged eye closure indicating drowsiness is detected accurately and handled promptly.

However, if the driver opens the eyes before the alert duration ends, the alert is cancelled and the system continues normal monitoring.

3.4.2 ALGORITHM FOR MOUTH ASPECT RATIO (MAR)

The Raspberry Pi first loads the camera stream, initializes the facial-landmark model, and sets the MAR threshold, required consecutive frames, and alert duration. After initialization, the system enters a continuous while-loop where every captured frame is examined for mouth landmark detection. The Mouth Aspect Ratio is calculated for each frame and compared with the yawning threshold. If the MAR stays above this defined threshold for the required number of consecutive frames, the system activates an alert and starts a 5–10 second countdown. If the driver continues yawning throughout this entire alert duration, the system stops the car from motion. This helps identify continuous yawning, which is a strong indicator of fatigue and reduced driver alertness.

If the MAR returns to normal before the timer finishes, the alert is cancelled and the system resumes normal monitoring.

3.4.3 ALGORITHM FOR OBJECT DETECTION

The Raspberry Pi initializes the camera, loads the object-detection model, and sets the danger-zone and alert duration parameters. Once the system starts running, it enters a continuous while-loop where each frame is processed for obstacle detection. If any dangerous object—such as a pedestrian, another vehicle, or any obstruction—is detected inside the predefined danger zone, the system immediately triggers an alert and begins the 5–10 second countdown. If the object remains present within the danger zone for the entire alert duration, the system commands the vehicle to stop from motion. This mechanism ensures real-time hazard response, preventing collisions caused by obstacles in the driving path.

If the detected object moves away or disappears before the alert duration ends, the alert is cancelled and the system returns to its normal monitoring state.

3.4.4 ALGORITHM FOR HEAD POSTURE (LEFT/RIGHT)

The system begins by initializing the Raspberry Pi camera, loading the face-landmark detector, and setting the head-yaw threshold along with the alert duration. After this initialization, the program enters a continuous while-loop where each incoming frame is analysed for head-pose estimation. Using facial landmarks, the system computes the yaw angle and checks whether the driver's head is turned left or right beyond the defined limit. If the head remains turned in either direction for longer than the permitted duration, the system activates an alert and starts the 5–10 second timer. If the head remains turned throughout the entire alert period, the system stops the car from motion.

This prevents driver distraction by ensuring the driver keeps their focus straight on the road.

If the driver returns the head to the forward direction before the alert period ends, the alert is cancelled and the system continues normal monitoring.

IV. RESULTS

The proposed Intelligent Driver Monitoring System demonstrated reliable performance in identifying driver fatigue, distraction, and surrounding hazards. As discussed above, the model accurately recognized normal alert behaviour

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.141186

without generating false alarms, while prolonged eye closure, yawning, and reduced facial alertness were detected promptly through EAR-based analysis and facial feature monitoring. These events triggered timely alerts and corrective actions, confirming the system's ability to respond effectively to early signs of drowsiness. Distraction cues such as head turns were also classified with consistency, helping reorient driver attention. Additionally, the ultrasonic sensor accurately identified obstacles ahead, enabling automatic stopping to prevent collisions. Overall, the integration of computer vision and multi-sensor fusion resulted in a responsive, real-time safety mechanism with strong potential for reducing fatigue-related and distraction-induced accidents.

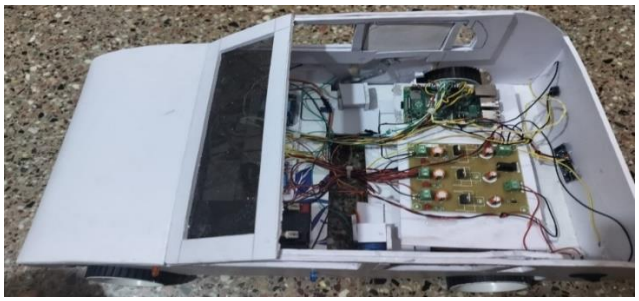

Fig:4 Prototype of Model

In the Fig 5 shows the live camera feed capturing the driver with the eyes detected in an open and alert state. The system correctly identifies normal eye activity, indicating that the driver is attentive and no drowsiness alert is triggered.

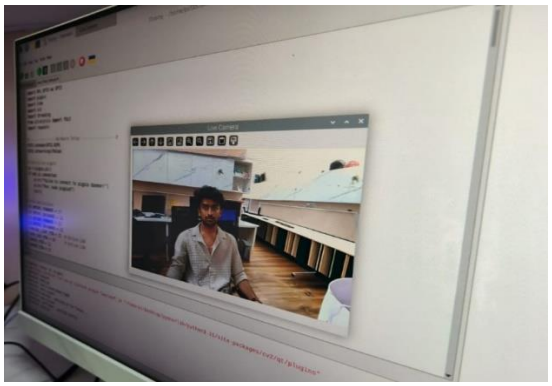

Fig: 5 Driver's eyes detected in an open state

In the Fig 6 shows the prototype vehicle moving while the driver-monitoring system detects the driver's eyes in an open and alert state, as seen in the earlier image. The real-time video processing ensures that the vehicle continues motion only when the driver's condition is normal and safe.

Fig: 6 Vehicle in motion

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.141186

In the Fig7 shows the system identifying the driver's eyes in a closed state through the live camera feed, indicating a possible drowsiness condition. The detection module actively analyses the video input and flags the closed-eye status as a potential safety risk.

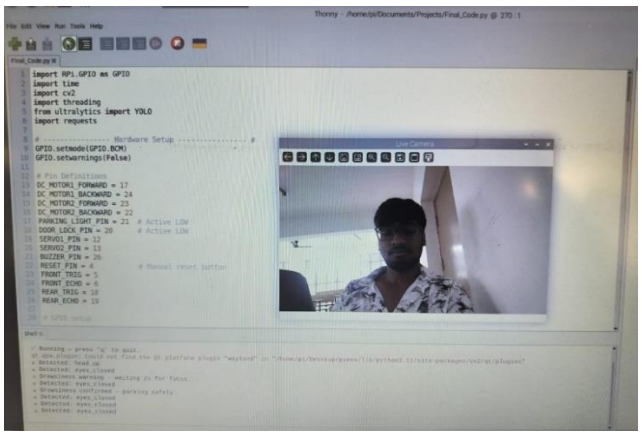

Fig: 7 Eye-closed state detected.

In the Fig 8 provides the corresponding terminal output confirming the closed-eye detection shown in Figure 7. The system logs repeated "eyes_closed" and "drowsiness warning" messages, verifying that the driver's eye-closure has been accurately recognized and the safety actions are being processed in continuation with the detection seen earlier.

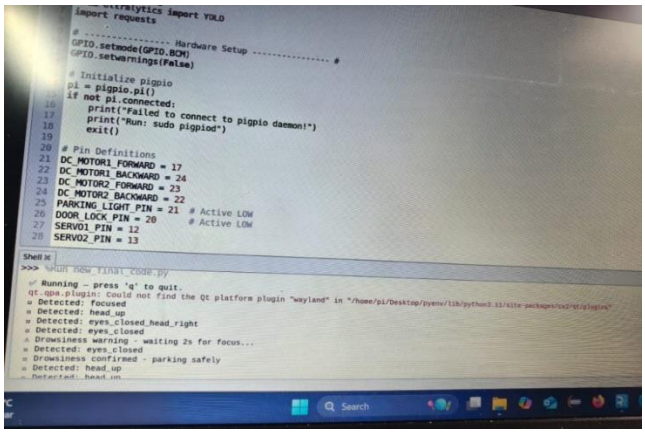

Fig: 8 Eye-closed output confirmed

In the Fig9 shows the system identifying a yawning state through the live camera feed, indicating that the driver may be experiencing drowsiness or fatigue. The YOLO-based detection module processes the video in real time and flags the wide mouth opening as a potential warning condition.

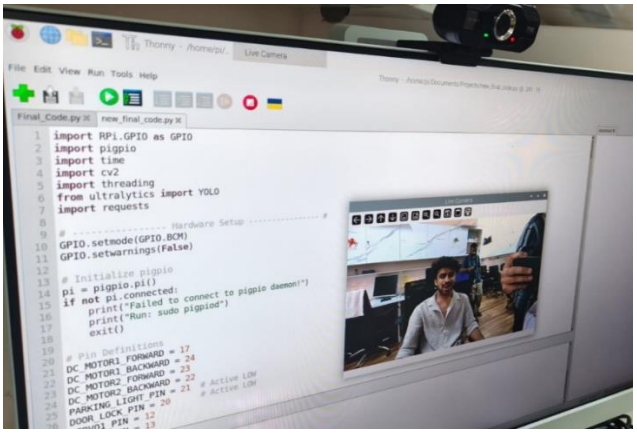

Fig: 9 Yawning state detected

DOI: 10.17148/IJARCCE.2025.141186

In the Fig10 illustrates the system's automatic response to the yawning detection shown in Figure 9, where the prototype vehicle is brought to a safe stop. The emergency indicator turns on and the door unlocks, confirming that the safety protocols have been activated to protect the driver.

Fig: 10 Vehicle stopped for safety

V. CONCLUSION

The Intelligent Driver Monitoring System using Raspberry Pi offers a practical solution for improving road safety and reducing accidents. It integrates drowsiness detection through image processing, alcohol sensing, obstacle detection with ultrasonic sensors, and GPS tracking to monitor both the driver and surroundings. Techniques such as eye aspect ratio analysis enable accurate fatigue detection under different lighting conditions. The system delivers timely alerts through buzzers and notifications, helping prevent accidents caused by fatigue, intoxication, or inattentiveness. With future advancements in AI, IoT, and cloud connectivity, the system can become even more adaptive and autonomous, contributing to safer and smarter transportation.

REFERENCES

[1]. **Choi, Y., et al. (2020).** "Autonomous Vehicle Overtaking: Modeling and an Optimal Trajectory Generation Scheme"-IEEE

[2]. **Kumar, A., & Meena, A. (2019).** "Driver Fatigue Detection System Using Machine Learning and Computer Vision." *International Journal of Transportation Engineering*, 45(1), 45-58.

[3]. **Raspberry Pi Foundation (2021).** "Getting Started with Raspberry Pi for AI and Machine Learning." *Official Raspberry Pi Documentation.*

[4]. **OpenCV (2022).** "Open Source Computer Vision Library." *Official OpenCV Documentation.*

[5]. **TensorFlow (2022).** "TensorFlow: Machine Learning Framework." *Official TensorFlow Website.*

[6]. **Rajendran, P., & Venkatesh, S. (2018).** "Driver Drowsiness Detection and Prevention: A Survey." *International Journal of Vehicular Technology*, 34(4), 45-53.

[7]. **Garcia-Rial, F., et al. (2021).** "Automotive Radar for Detecting Driver Drowsiness Through Vital Sign Monitoring." *IEEE Transactions on Vehicular Technology*, 70(9), 8721-8733.

[8]. **Lee, B. G., & Chung, W. Y. (2022).** "A Smartphone-Based Driver Safety Monitoring System Using Data Fusion." *Sensors*, 22(3), 1044-1058.

[9]. **World Health Organization (2023).** "Global Status Report on Road Safety." *WHO Technical Report Series*. Geneva, Switzerland.

[10]. **Arefnezhad, S., et al. (2022).** "Deep Learning Approaches for Driver Drowsiness Detection: A Comprehensive Review." *IEEE Access*, 10, 39867-39890.

