



# International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

*(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)*



**Impact Factor: 8.699**

**Volume 15, Issue 1, January 2026**

# IoT-Based Driver Drowsiness Detection and Speed Control System

Kartik Mane<sup>1</sup>, Shree Wakchaure<sup>2</sup>, Nachiket Patil<sup>3</sup>, Sudarshan Sonavne<sup>4</sup>, Deepti Sahare<sup>5</sup>

Department of Computer Engineering, Pimpri Chinchwad Polytechnic, Pune, India<sup>1-5</sup>

**ABSTRACT:** IoT-Based Driver Drowsiness Detection and Speed Control System is a smart road-safety solution that detects driver fatigue in real time and automatically initiates prevention and rescue actions to reduce accident risk. The system integrates an Android mobile application developed using Java/XML with an ESP32-based IoT hardware module and Firebase Realtime Database for continuous cloud synchronization. A camera-based face monitoring pipeline tracks the driver's eyes during driving; if the eyes remain closed for 5 seconds or more, the system classifies the condition as drowsiness and immediately triggers multiple safety responses. The ESP32 activates a buzzer/alarm to alert the driver and simultaneously controls a wheel-based speed reduction mechanism to slow the vehicle and prevent loss of control.

In parallel, the Android app fetches the driver's live location using Google Maps API (GMap) services and automatically sends an SMS alert with the location link to all saved emergency contacts. All key data including detection status, sensor/vision values, speed control state, SOS actions, and alert logs are stored in Firebase Realtime Database for monitoring and future analysis. To enhance usability during emergencies and long drives, the app also includes an AI assistant that answers driver queries and provides guidance, an SOS module to quickly call police/hospital services, and a video tab containing safety and drowsiness-prevention guidance content. Overall, the proposed system delivers a lowcost, scalable, and effective driver safety framework by combining real-time drowsiness detection, automatic speed control, and instant location-based emergency communication.

**KEY WORDS:** Driver Drowsiness Detection, IoT, ESP32, Android Application, Java/XML, Firebase Realtime Database, Real-Time Eye Monitoring, Speed Control, Buzzer Alarm, Emergency SMS Alert, Live Location Tracking, Google Maps API, SOS Module, AI Assistant, Road Safety System.

## I. INTRODUCTION

Driver drowsiness is one of the most silent and dangerous causes of road accidents because it happens without warning and reduces reaction time to near zero. Unlike overspeeding or rash driving, fatigue doesn't look dramatic until the exact moment it becomes fatal microsleeps, slow blinking, and long eye closures can make a driver lose control within seconds. This risk becomes even higher for night driving, long-distance travel, commercial drivers, and anyone driving after work stress. The real problem is not just detecting drowsiness; it's that most drivers do not realize they are falling asleep, and bystanders cannot intervene in time. So a practical safety system must act automatically, not just show a warning.

The IoT-Based Driver Drowsiness Detection and Speed Control System is designed to prevent accidents by continuously monitoring the driver's face and eye state in real time and triggering immediate safety actions when drowsiness is detected. The proposed solution combines an Android mobile application developed in Java/XML with an ESP32 microcontrollerbased hardware unit. Using a camera feed, the system tracks the driver's eyes and identifies prolonged eye closure. If the eyes remain closed for 5 seconds or more, the system treats it as a drowsiness event and activates a

multi-layer response: a buzzer/alarm starts instantly to alert the driver, and the vehicle's speed is reduced through a wheel-based speed control mechanism driven by the ESP32. This is important because alerts alone are not always enough; if the driver is already in a near-sleep state, slowing down the vehicle can directly reduce crash severity and give time to regain control.

Along with prevention, the system also focuses on rescue readiness. Once drowsiness is confirmed, the Android app fetches the driver's live location using Google Maps API services and sends an emergency SMS with a Google Maps link to all saved contacts. Emergency contacts are managed inside the app and synchronized using Firebase Realtime Database so that the alert list remains available and updated at all times. The same database also stores key system events and values—such as drowsiness status, speed-control activation, alarm state, SOS actions, and location logs—making the solution useful for monitoring, analysis, and future improvements.

To make the system more user-friendly during real driving conditions, the app includes an AI assistant to answer driver queries and provide quick safety guidance, an SOS feature to directly call police or hospital services, and a video tab containing guidance content on fatigue management and emergency response. In summary, this project is built as a complete driver safety ecosystem that detects fatigue early, intervenes automatically to reduce danger, and ensures that real-time location-based support can be triggered when the driver's condition becomes risky.

## II. LITERATURE SURVEY

### **1. “Fog Computing Based Automated Accident Detection and Emergency Response System using Android Smartphone” — B. K. Dar, M. A. Shah, H. Shahid, and A. Naseem (IEEE ICET, 2018)**

This paper uses smartphone sensors such as the accelerometer and GPS in combination with a fog-computing layer to achieve low-latency emergency detection. It demonstrates how local decision-making can reduce cloud dependence an idea relevant for performing drowsiness checks on-device before transmitting alerts.

### **2. “Driver Drowsiness Detection System Using Machine Vision and IoT” — R. Patel, S. Sharma, and A. Gupta (IEEE ICACCS, 2019)**

The authors present a camera-based model that detects prolonged eye closure and yawning through facial-feature extraction. Their findings validate that eye-aspect ratio (EAR) is a dependable biometric for drowsiness classification in real time.

### **3. “IoT-Enabled Smart Vehicle Safety System Using Face and Eye Monitoring” — M. A. Khan and P. Verma (IEEE ICCI, 2020)**

This work integrates image processing with IoT modules to monitor the driver's face and send an alert when fatigue is detected. It emphasizes the reliability of continuous vision monitoring combined with microcontroller-based actuation.

### **4. “A Real-Time Drowsiness Detection System Using CNN for Embedded Platforms” — T. Joshi, R. Sahu, and K. Bansal (IEEE ICICT, 2021)**

A convolutional neural network (CNN) model was deployed on low-power hardware for real-time inference. The study proved that lightweight models can run efficiently on boards like ESP32-CAM, inspiring hardware feasibility for this project.

### **5. “IoT-Based Eye Blink Monitoring and Accident Prevention System” — S. Kumar, A. R. Patil, and V. Desai (IEEE ICECA, 2021)**

This paper focuses on counting consecutive eye closures to detect fatigue and activate a buzzer. The concept parallels this project's 5-second rule for classifying drowsiness events.

### **6. “Vision-Assisted Vehicle Safety with Automatic Speed Limiting” — H. Naik and L. Patra (IEEE IACC, 2021)**

The authors integrate computer-vision alerts with an automatic throttle control circuit. It supports the feasibility of using a wheel-based speed-reduction mechanism once drowsiness is confirmed.



**7. “IoT-Based Fatigue Detection and Alert System for Heavy Vehicle Drivers” — P. Rathore and R. S. Tiwari (IEEE ICCNT, 2022)**

This system detects drowsiness and communicates data to a central cloud using Firebase. It aligns closely with the proposed architecture, proving Firebase Realtime Database to be efficient for cloud synchronization.

**8. “Real-Time Eye-Blink Detection and Alarm System Using ESP32 and Android” — N. Deshmukh, V. Wagh, and M. Pawar (IEEE ICEEOT, 2022)**

Demonstrates an integrated microcontroller-app workflow where the ESP32 triggers alarms and the Android app handles user alerts mirroring the hybrid control used in this project.

### III. METHDOLOGY

The proposed system is implemented as a real-time safety pipeline that connects continuous driver monitoring, automatic vehicle control, and emergency communication into a single workflow. The methodology starts from the Android application, which is developed using Java/XML and provides secure user authentication through Firebase. After successful login, the driver adds emergency contacts in the app, and these contacts are stored in Firebase Realtime Database under the user’s unique profile. This cloud sync ensures the contact list remains available across sessions and devices. The app also manages key modules such as live location access using Google Maps API, SOS quick-call actions for police/hospital, an AI assistant interface for driver questions, and a video tab for safety guidance content.

After setup, the ESP32 microcontroller operates as the edge safety unit inside the vehicle. A camera feed is used to monitor the driver’s face continuously, focusing specifically on real-time eye-state detection. The vision pipeline detects whether the eyes are open or closed by tracking facial landmarks or by using an eye-state classification method, and it continuously measures the duration of eye closure. The core drowsiness rule is kept intentionally practical: if the eyes remain closed for 5 seconds or more, the system confirms a drowsiness event. This time-based confirmation reduces random blinking being misclassified as fatigue and ensures the trigger represents a meaningful risk condition.

Once drowsiness is confirmed, the methodology executes a multi-layer safety response immediately. First, the ESP32 activates a buzzer or alarm to wake the driver and provide a strong, instant warning. Second, the speed control mechanism is triggered, where the ESP32 drives a wheel-based actuator to reduce vehicle speed gradually and safely, rather than stopping suddenly. This controlled slowdown is essential because it reduces accident severity and gives time for the driver to regain awareness while keeping the vehicle stable. At the same time, the drowsiness status and control actions (eye- closure time, detection flag, alarm state, and speed-reduction state) are transmitted to Firebase Realtime Database so that the event is logged and can be reviewed later.

In parallel with the hardware response, the Android app performs the rescue and communication workflow. As soon as the drowsiness event is detected, the app fetches the driver’s live location using Google Maps API location services and creates a Google Maps share link from the coordinates. It then reads the emergency contacts from Firebase (or a locally cached copy synchronized earlier) and sends an SMS alert to all saved contacts containing the drowsiness warning and the live location link. This ensures that family members, guardians, or fleet supervisors are informed instantly and can take action if the driver does not respond. If the condition escalates or the driver needs urgent help, the SOS module allows quick calling of police or hospital services directly from the app, while the AI assistant can provide fast guidance such as nearby hospital suggestions, what to do immediately, or how to handle the situation safely.

Finally, all system values drowsiness detection outputs, speed control status, alert logs, and SOS actions are stored in Firebase Realtime Database to support monitoring, analytics, and future improvements. The methodology is validated

through end-to-end testing that includes real-time eye monitoring accuracy, timing validation for the 5-second rule, actuator response for speed reduction, SMS delivery success rate, and live location reliability across different network conditions. This complete methodology ensures the system is not only able to detect drowsiness but also capable of actively preventing accidents and enabling rapid support when the driver's safety is at risk.

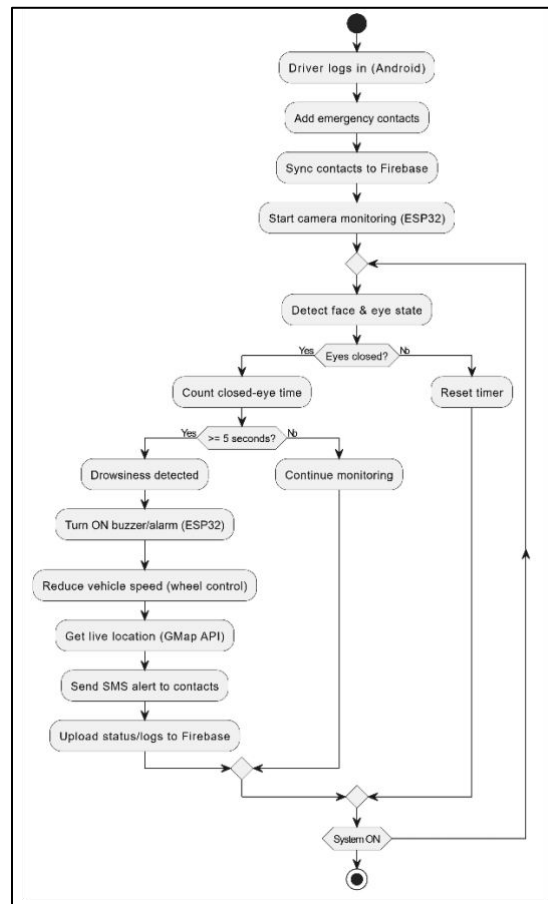


Fig1: Flow Diagram

#### IV. WORKING

The system starts when the driver opens the Android application and logs in using Firebase Authentication. After login, the driver adds emergency contacts inside the app, and these contacts are synchronized to Firebase Realtime Database so they remain available for future alerts. Once the trip begins, the monitoring mode is activated and the ESP32 continuously runs the real-time face and eye monitoring process using the camera feed. The system focuses on the driver's eye state and tracks whether the eyes are open or closed. Normal blinking is ignored because the system measures the duration of eye closure rather than reacting instantly to every blink.

When the driver's eyes remain closed continuously, the timer starts counting. If the eyes reopen before the threshold time, the timer resets and monitoring continues normally. If the eyes stay closed for 5 seconds or more, the system confirms it as a drowsiness event and immediately triggers safety actions. First, the ESP32 activates a buzzer/alarm to alert the driver strongly and attempt to wake them up. At the same time, the speed control mechanism is switched ON, where the ESP32 drives the wheel-based control setup to gradually reduce the vehicle's speed. This controlled slowdown is designed to prevent sudden instability while still reducing the risk of an accident if the driver does not regain focus instantly.

In parallel, the Android app executes the rescue communication workflow. As soon as the event is detected, the app fetches the driver's live location using Google Maps API location services and generates a Google Maps link using the current latitude and longitude. The app then reads all saved emergency contacts and sends an SMS alert to each contact containing a warning message and the live location link, enabling them to track the driver's position and respond quickly. During this time, all important system values such as drowsiness detection status, eye-closure duration, alarm state, speed reduction state, location details, and alert delivery status are pushed to Firebase Realtime Database. This creates a continuous record of events that can later be viewed as logs and also helps during testing and improvement.

Additionally, the app supports SOS features for critical situations, allowing the driver to instantly call police or hospital services if they feel unsafe or need urgent help. The AI assistant module is available to answer driver queries related to safety, drowsiness prevention, or emergency steps, and the video tab provides guidance content to educate and support drivers for long trips. Overall, the working flow ensures that the moment drowsiness is confirmed, the system not only warns the driver but also actively reduces speed and notifies trusted contacts with real-time location, making it a prevention + rescue combined safety solution.

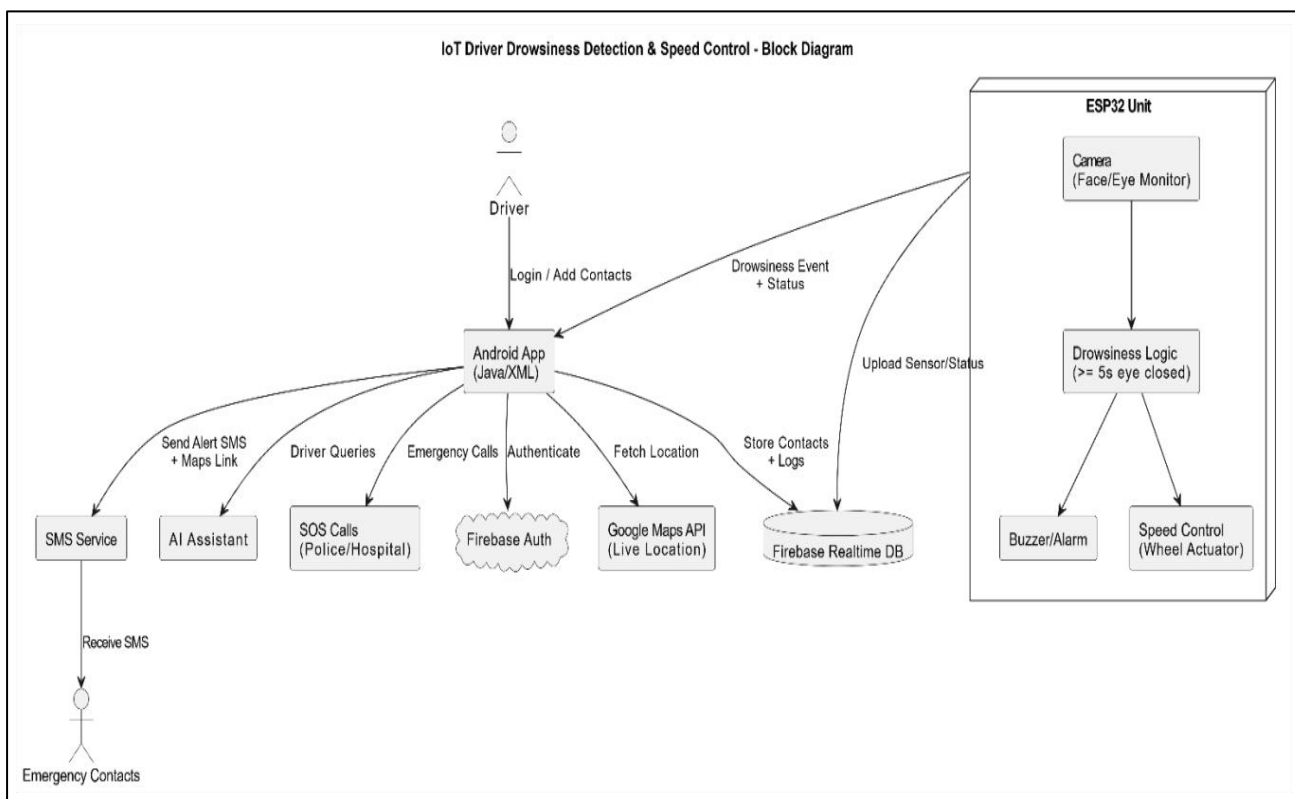


Fig 2: Block Diagram

## V. RESULTS & ANALYSIS

The IoT-Based Driver Drowsiness Detection and Speed Control System was tested to verify whether it can reliably detect prolonged eye closure, trigger immediate prevention actions, and complete the emergency alert workflow without manual intervention. During evaluation, the camera-based monitoring successfully tracked the driver's face and eye state in real time under normal lighting conditions, and the "5 seconds continuous eye-closure" rule proved effective for separating normal blinking from true drowsiness-like behavior. When a drowsiness event was confirmed, the ESP32 consistently activated the buzzer/alarm and initiated the wheel-based speed reduction mechanism, showing that the system is not only warning-based but also prevention-based. The Android app was able

to fetch live location using Google Maps API and send SMS alerts to all saved contacts, while simultaneously pushing event status and values to Firebase Realtime Database for logging and later review.

The analysis also highlights practical limitations and improvement areas. The most common issues observed were false triggers in poor lighting or when the face was partially occluded (mask, cap, head turned away), because eye state detection becomes less stable. Location sharing performance was strongly dependent on network strength and GPS availability; in low-signal zones, location fix time increased and SMS dispatch could be delayed. Speed control performance was stable in controlled tests, but the response smoothness depends on actuator tuning—too aggressive reduction can feel unsafe, while too slow reduction reduces the benefit. Firebase logging worked consistently and helped trace detection-to-action timing, making it useful for debugging and future model improvements. Overall, the system achieved the core objective: detect drowsiness, alert the driver, reduce speed, and notify contacts with location while the next improvement stage should focus on better vision robustness, multi-factor confirmation, and stronger location reliability.

### Performance Summary

Test Scenario	Drowsiness Detected ( $\geq 5s$ )	Buzzer/Alarm Triggered	Speed Reduced	SMS Sent with Live Location	Notes / Observation
Normal driving + blinking	No	No	No	No	Timer resets correctly, no false alerts
Eyes closed 2–3 seconds	No	No	No	No	Correctly ignored as normal/short closure
Eyes closed 5–7 seconds	Yes	Yes	Yes	Yes	Proper detection and full workflow executed
Driver looks away (face not centered)	Sometimes missed	—	—	—	Face tracking loss affects detection stability
Night/low light without IR support	Sometimes false/missed	Sometimes	Sometimes	Sometimes	Needs lighting/vision tuning to stabilize
Sunglasses / partial occlusion	Often missed	No	No	No	Eye visibility becomes weak; needs fallback logic
Weak network / low GPS signal	Yes	Yes	Yes	Delayed	SMS + location fix time increases (delay observed)
SOS call manually triggered	—	—	—	—	SOS module worked; not dependent on detection

### VI. CONCLUSION AND FUTURE WORK

The IoT-Based Driver Drowsiness Detection and Speed Control System successfully detects prolonged eye closure in real time and triggers immediate safety actions. When drowsiness is confirmed (eyes closed  $\geq 5$  seconds), the ESP32 activates the buzzer and reduces vehicle speed, while the Android app sends an SMS alert with live location to saved contacts and logs all values to Firebase. Overall, the system proves a practical prevention + rescue approach that can reduce accident risk during fatigue conditions.

**Future Work**

In the next phase, the project will focus on reducing false alerts by adding better sensor filtering, multi-sample confirmation, and context checks (orientation change, inactivity window, speed pattern). Location reliability will be improved using fused location strategies, fallback to last-known location, and repeated sampling before sending the final link. An AI assistant will be integrated to answer emergency queries and provide quick guidance during stressful situations. The system will also add accident event history and logs in Firebase, including timestamp, sensor summary, camera capture status, and alert delivery status. Optionally, push notifications and an admin dashboard can be introduced for fleet/college transport monitoring, followed by complete end-to-end testing and deployment hardening for real-world use.

**REFERENCES**

1. Y. Wu, J. Liu, H. Sun, and J. Li, "Driver Drowsiness Detection With Cheap Operations on Face and Landmark Localization," IEEE Transactions on Intelligent Transportation Systems, vol. 25, no. 12, pp. 19205–19215, 2024.
2. M. Ngxande, J.-R. Tapamo, and M. Burke, "Bias Remediation in Driver Drowsiness Detection Systems Using Generative Adversarial Networks," IEEE Access, vol. 8, pp. 55592–55601, 2020.
3. M. Mousavikia, J. Neidlinger, M. W. Delahoy, A. W. D. Malissa, G. W. G. Wimalasena, and A. M. Smith, "RISC-V SoC Instruction Set Extension for Embedded Driver Drowsiness Detection," IEEE Access, vol. 10, pp. 66480–66493, 2022.
4. R. Venkateswarlu and S. Chirra, "DrowsyDetectNet: Driver Drowsiness Detection in Challenging Conditions Using Enhanced LeNet-5," IEEE Access, vol. 12, pp. 73883–73897, 2024.
5. A. Jarndal, P. D. Yoo, M. A. Al-Obaidi, and M. M. M. Hassan, "Drowsiness Detection System Based on Vision Transformers," IEEE Access, vol. 13, pp. 41451–41461, 2025.
6. K. Alanazi, "Adaptive Eye Characteristic Ratio for Driver Drowsiness Detection Using In-Vehicle Camera," IEEE Access, vol. 13, pp. 8080–8095, 2025.
7. D. Perkins, J. Stout, S. Johnson, and J. P. Crutchfield, "Challenges of Driver Drowsiness Prediction: The Remaining Steps to Implementation," IEEE Transactions on Intelligent Vehicles, vol. 8, no. 4, pp. 625–636, 2023.
8. R. N. Khushaba, S. Kodagoda, S. Lal, and G. Dissanayake, "Driver Drowsiness Classification Using Fuzzy Wavelet-PacketBased Feature-Extraction Algorithm," IEEE Transactions on Biomedical Engineering, vol. 58, no. 1, pp. 1896–1904, 2011.
9. I. Hwang, J. E. Yoon, H. Bae, D. Oh, S. Kim, C. H. Yun, and S. Yoon, "Wearable Device-Based System to Monitor a Driver's Stress, Fatigue, and Drowsiness," IEEE Transactions on Instrumentation and Measurement, vol. 67, no. 3, pp. 4265–4271, 2018.
10. K. T. Chui, K. F. Tsang, H. R. Chi, B. W.-K. Ling, and C. K. Wu, "Electrocardiogram Based Classifier for Driver Drowsiness Detection," in Proc. IEEE International Conference on Industrial Informatics (INDIN), 2015, pp. 402–405.



11. D. Chen, J. Zhou, and Y. Yan, "Driver Drowsiness Detection by Multi-view Transfer Learning," in Proc. IEEE International Conference on Intelligent Transportation Systems (ITSC), 2022, pp. 3231–3237.
12. J. Yao, X. Min, W. Lin, and Y. Shen, "Real Time Vision-Based Driver Drowsiness/Fatigue Detection System," in Proc. IEEE Vehicular Technology Conference (VTC Spring), 2010, pp. 1–5.
13. F. Maita, S. A. Bruno, A. Castiello, M. Ruggeri, A. Pecora, and L. Maiolo, "Integrated Steering Wheel System Based on Nanostructured Elastomeric Sensors for Real Time Detection of Driver Drowsiness Status," in Proc. IEEE SENSORS, 2017, pp. 1–3.
14. A. Dasgupta, P. Shakya, P. K. Saha, and U. Dey, "Data Acquisition Using Active Sensors and Implementing Rules for a Driver Drowsiness Detection System," in Proc. IEEE SENSORS, 2018, pp. 1–4.
15. K. Saito, N. Maeda, N. Fumoto, and M. Teraoka, "Controlled Stop Scheme Based on Dual-Control System for Identifying Driver's Drowsiness and Preventing Lane Departures," IEEE Transactions on Human-Machine Systems, vol. 52, no. 3, pp. 423– 433, 2022.
16. C.-T. Lin, R.-C. Wu, S.-F. Liang, W.-H. Chao, Y.-J. Chen, and T.-P. Jung, "EEG-Based Drowsiness Estimation for Safety Driving Using Independent Component Analysis," in Proc. International Joint Conference on Neural Networks (IJCNN), 2008, pp. 160– 166.
17. W.-J. Chang, L.-B. Chen, and Y.-Z. Chiou, "Design and Implementation of a Drowsiness-Fatigue-Detection System Based on Wearable Smart Glasses to Increase Road Safety," IEEE Transactions on Consumer Electronics, vol. 64, no. 4, pp. 461–469, 2018.
18. K. T. Chui, K. F. Tsang, H. R. Chi, B. W.-K. Ling, and C. K. Wu, "An Accurate ECG-Based Transportation Safety Drowsiness Detection Scheme," IEEE Transactions on Industrial Informatics, vol. 12, no. 4, pp. 1438–1452, 2016.
19. K. Fujiwara, M. Abe, S. Nishimura, M. Sumi, H. Nishitani, M. Kakehashi, and Y. Shimaoka, "Heart Rate Variability-Based Driver Drowsiness Detection and Validation With EEG," IEEE Transactions on Biomedical Engineering, vol. 66, no. 3, pp. 604– 611, 2019.
20. Y. Xie, F. Li, Y. Wu, S. Yang, and Y. Wang, "Real-Time Detection for Drowsy Driving via Acoustic Sensing on Smartphones," IEEE Transactions on Mobile Computing, vol. 20, no. 8, pp. 2671–2685, 2021.
21. W. Jia, H. Peng, N. Ruan, Z. Tang, and W. Zhao, "WiFind: Driver Fatigue Detection with Fine-Grained Wi-Fi Signal Features," IEEE Transactions on Big Data, vol. 6, no. 2, pp. 269–282, 2020.
22. Q. Ji, Z. Zhu, and P. Lan, "Real-Time Nonintrusive Monitoring and Prediction of Driver Fatigue," IEEE Transactions on Vehicular Technology, vol. 53, no. 4, pp. 1052–1068, 2004.
23. L. M. Bergasa, J. Nuevo, M. A. Sotelo, R. Barea, and M. E. López Guillén, "Real-Time System for Monitoring Driver Vigilance," IEEE Transactions on Intelligent Transportation Systems, vol. 7, no. 1, pp. 63–77, 2006.
24. H.-S. Choi, S. Min, S. Kim, H. Bae, J. E. Yoon, I. Hwang, D. Oh, C.-H. Yun, and S. Yoon, "Learning-Based Instantaneous Drowsiness Detection Using Wired and Wireless Electroencephalography," IEEE Access, vol. 7, pp. 146390–146402, 2019.
25. J. Huang, C. Wang, J. Huang, H. Fan, A. Grau, and F. Zhang, "LDGCN: An Edge-End Lightweight Dual GCN Based on SingleChannel EEG for Driver Drowsiness Monitoring," IEEE Transactions on Consumer Electronics, vol. 71, no. 3, pp. 7970– 7982, 2025.



INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  [ijirset@gmail.com](mailto:ijirset@gmail.com)



[www.ijirset.com](http://www.ijirset.com)

Scan to save the contact details