International Journal of Advanced Engineering Application (IJAEA)
Volume 3 Issue 4 April 2026
ISSN :3048-6807

Developing Driver Safety Monitoring System Using Deep Learning and IOT

[1]JVN. Raju, [2]K. Lavanya, [3]G. Uma Maheswari, [4]D. Phanindra, [5]G. Sudheer Kumar, [6]D. Suraj Harsha
[1]Associate Professor, [2,3,4,5,6]Student,
[1,2,3,4,5,6]Department Of Information technology
[1,2,3,4,5,6]Dhanekula Institute of Engineering & Technology, Vijayawada, Andhra Pradesh-India

Abstract

The project proposes a Driver Monitoring and Alert System which aims to improve the safety of the driver by enabling real time detection of drowsiness, distraction and unsafe behaviors of the driver. By making use of a Raspberry Pi camera along with YOLOv8, Dlib and MediaPipe the system can accurately detect driver drowsiness, eye closure, yawning and other unsafe behaviors like usage of phone, smoking and consumption of drink; along with detection of driver emotions. If an unsafe act is detected by the system an alarm is generated using the LCD display, buzzer and voice alert, and with the help of the DC motor the speed of the vehicle is simulated to be slowed down or halted entirely. Moreover, the system also sends real-time alerts with images of the detected action pushed to the driver's mobile via Telegram, so they are remotely monitored. This is a low-cost, AI-IoT integrated solution that exploits the benefits of computer vision, real-time alerts and control system to enhance driver safety and alertness. Keywords: Driver Monitoring System (DMS), road safety, driver fatigue detection, drowsiness detection, distraction detection, YOLOv8, Dlib, MediaPipe, Raspberry Pi, computer vision, real-time monitoring, eye closure detection, yawning detection, mobile usage detection, smoking detection, alcohol detection, emotion recognition, AI-IoT integration, alert system, Telegram notifications, automated vehicle control, DC motor simulation, embedded systems, low-cost safety solution.

Keywords: **Driver Monitoring System (DMS), Deep Learning, Computer Vision, Drowsiness Detection, Distraction Detection, YOLOv8, Internet of Things (IoT), Raspberry Pi, Real-Time Monitoring, OpenCV.**

1. Introduction

In recent years, driver fatigue and distraction are recognized as one of the primary causes of traffic accidents globally, resulting in significant injury, death, and economic loss. Driving for long hours, getting no rest, and performing activities such as the use of mobile phones lead to decreased alertness and slowed reaction time while driving. While the modern vehicle may provide with some safety features such as seatbelt reminder, simple warning system, there are still some conventional safety devices which lack real-time behavior monitoring and active intervention functionality. Constant driver monitoring is thus considered an important step in the effort to minimize accident risks and improve road safety. In this regard, an AI-based driver monitoring unit implemented on a Raspberry Pi is proposed as an inexpensive and effective real-time driver behavior monitoring system.

AI algorithms are used to continually monitor a driver's facial features and behavior via a camera. Object and behavior detection is performed using YOLOv8 to identify risky objects and actions such as cell phone usage, and smoking; and facial landmarks and gesture analysis is performed using Dlib and MediaPipe to determine drivers' drowsiness, yawning, long closure of eyelids, and distraction. In the case that the drivers are exhibiting any risky behavior, the drivers are given warning through LCD, Buzzer, and voice warning to regain drivers' attention. Alert, real-time image, and driving behavior is also sent to Guardian/Fleet manager remotely via Telegram and simulated control commands are sent out as speed decreasing, parking, to provide driver intervention. By combining AI, IoT and automatic control system, the solution is a portable, economic and useful tool in real world application for improving driver attention and road safety.

Problem Statement

Driver fatigue, distraction and unsafe driving habits still pose one of the most challenging safety issues worldwide, even with the advancements in various vehicular safety technologies. Most existing vehicle safety systems tend to be either based on external environment or mechanical factors, instead of real-time observation and analysis of the driver's state. For instance, human actions such as prolonged eye closure, yawning, using of cell phone, smoking, or driving under psychological stress negatively affect the drivers' level of awareness and response times, which cannot be fully resolved with manually detection or conventional warning signals. Most conventional systems do not perform real-time, non-stop driving monitoring, and many do not integrate the diverse behavioral detection methods in to one system or they are not cost effective. Some may only focus on one type of risky behaviors (such as driver drowsiness detection) and do not address the various risky behaviors such as distraction and object usage. Also, a notification mechanism to allow guardians or managers to take action in real time is missing. This means there is still no system available that combines computer vision and artificial intelligence to the existing IoT-based infrastructure to detect multiple risky driving behaviors automatically, inform the driver in real time and alert external stakeholders, with a low cost, small form-factor and portable design.

2. Literature Review

The use of Driver Monitoring Systems has emerged to be a vital area of study with the increasing number of road accidents and the proliferation of intelligent transportation systems. Initial driver safety systems mainly relied on vehicle sensors and rule-based approaches where the focus was primarily on the vehicle state rather than the physiological and psychological state of the driver. These methods lacked real-time behavioral analysis and were not effective enough to detect the complex nature of human actions.

Recent research in driver monitoring has shifted towards vision-based and deep learning based methods. Convolutional Neural Networks (CNN) has become popular for drowsiness detection with its use in measuring and evaluating eye closure, yawn and PERCLOS features. Ramzan et al. Suggested a custom CNN combined with HOG-PCA features to enhance drowsiness detection but it was computationally too expensive for deployment in low-end embedded devices. Similar, Jahan et al. Proposed a 4D CNN model that could detect eye state in real time and with high accuracy but only the eye aspect is considered and not other distracting behaviors.

Many systems utilized facial landmark based techniques and various ML methods for estimating Eye Aspect Ratio (EAR) and Mouth Aspect Ratio (MAR) as well as the head pose for drowsiness detection. It was proven by Albadawi et al. That a combination of visual features with cheap ML models can provide a robust real-time system but external distractions are not considered such as use of phone and smoking. Mobile Net-SSD has been adopted to predict the object of distraction (drowsiness) for detecting it in real-time using edge devices such as Raspberry Pi but limitations included detecting only the eyes opening and closing.

Some researchers attempted to include distraction along with drowsiness monitoring. Hanafi et al. Provided a CNN-based real time system that could detect both drowsiness and a part of the distraction behavior but mainly relied on eye states. Many systems could detect only one aspect of driver behavior or have been too complicated for integration of multiple modules. Also, most methods were computationally too expensive for real-time embedded systems.

Based on the review of literature, a crucial research gap can be identified which is the need for a real-time, low cost and unified driver monitoring platform that can detect drowsiness, distraction and risky behaviors simultaneously suitable for an embedded system. The presented system overcomes this deficiency by combining YOLOv8 for object detection based on risky behavior with Dlib and MediaPipe for analyzing facial landmarks with a real-time alerting system and an IoT notification facility.

4. System Architecture

The Driver Safety Monitoring system adopts a well structured and modular architecture that converts raw driver video data into safety alerts and control actions in real time. Each of the module performs an exclusive task making the system effective and maintainable and scalable.

The Video Acquisition module captures continuous real time video of the driver, using a camera positioned inside the vehicle. These images are transmitted to the Raspberry Pi (Central Processing and control module) which executes the entire computation in its internal processor to facilitate real-time performance and protect driver's privacy.

The intelligent behavior analysis module utilizes artificial intelligence and computer vision techniques. Using YOLOv8, the Risky behaviors such as phone usage, smoking and alcohol presence is detected and using Dlib and Media Pipe the facial landmark points and head pose is estimated to detect driver drowsiness, eye closure, yawning and distraction.

The extracted features are analyzed by the decision making module. A pre defined threshold is used to check if the condition of the driver is unsafe. Once an unsafe behavior is detected the Alert and Actuation module is activated. Visual alerts are given to the driver using a 162 LCD. Buzzer and Audio alerts are given and the speed reduction or the stoppage of the vehicle is simulated using a DC motor.

Simultaneously, the IoT communication module informs the authorized personnel by sending real time alerts along with the captured images to a registered mobile using Telegram. The whole system works continuously, to give real-time driver monitoring and instantaneous alerts enhancement. leading to road safety.

5. Methodology

This paper suggests an end-to-end intelligent driver safety monitoring approach that simultaneously recognizes fatigue, distraction, and dangerous behaviors based on deep learning and IoT. The whole system can be characterized by the processing cascade, in which each stage would improve the detection accuracy by using the processing outcome from the previous stage to get a precise and on time alarm. The system is expected to constantly monitor the driver and activate some safety measures if some unsafe behaviors are detected.

1. Video Stream Acquisition

Video stream V is continuously acquired from the camera in the vehicle. The stream V is related to facial expression, head motion and driver's activities, which are used for behavioral modelling.

2. Frame Extraction and Preprocessing

The input video V is split into frames, denoted by F= {f1, f2,..., fn }. Every frame will first pre-processed by OpenCV (size normalization, noise reduction etc) to improve the accuracy of the detection and lower the computational costs of Raspberry Pi.

5.1 Behavior and Object Detection

In each processing frame, models are employed to detect actions: YOLOv8 is utilized to detect dangerous objects and action including phone, cigarette and alcohol. The Dlib facial landmark detection is used to identify eyes and mouth landmarks in order to count the duration that eyes closed and open mouth wide for yawning.

MediaPipe Face Mesh is used to estimate head pose and face orientation to detect head posture for inattention and distraction.

5.2 Feature Evaluation and Decision Logic

The identified features are sent to a thresholding based decision making unit. Values of eye closure duration, yawning frequency, head pose deviation, unauthorized object in the prohibited region, etc., are compared against a pre-defined safety threshold. If the values go above the threshold, the state of the driver is labeled as unsafe.

5.3 Alert and Safety Action Generation

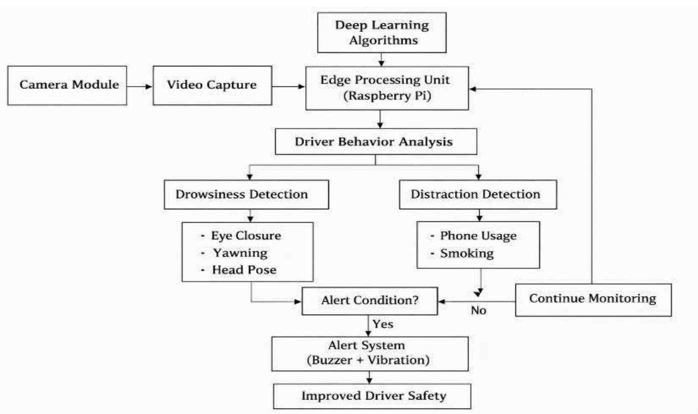
The system responds by triggering several alarming factors as soon as the unsafe act is detected. Warning notifications appear on a 162 LCD and a buzzer as well as an audio module are triggered. The dc motor is used to simulate an automated reduction in the vehicles speed or a halt to create the feeling of automated action.

5.4 IoT-Based Notification and Monitoring

In addition, during system operation, snapshots of evidence images were taken and send notifications to a registered mobile number instantly through Telegram, so guardians/fleet owners could track and monitor remotely, and get notified in case of emergency.

5.5 Continuous Monitoring Loop

The complete framework runs in a continuous loop where there is continuous monitoring of the driver, instant reaction to unsafe driving and road safety is increased by integrating computer vision, deep learning &IOT.



6. Future Scope

1. **Model Enhancement**

The system could be enhanced by utilizing more sophisticated and efficient deep learning models, enabling higher accuracy of detection in realistic driving environments. To make it feasible 24/7, it would need to work efficiently even in the presence of darkness, for which infrared cameras would be ideal.

2. **IOT and Cloud platform connectivity**

In the future it can be connected to IoT and cloud services so that monitoring can be done remotely, and used in fleet management. The transport authority or fleet owner can thus track the driver, and also keep historical data for safety performance evaluation.

3. **Car control system connection**

The system can be connected to the control system of the car, and if there is acute level of distraction or drowsiness detected, then it would reduce the speed of the vehicle or activate emergency measures.

4. **GPS and mobile application facility**

It can have GPS and also a mobile application interface, so that its location could be tracked in real time and emergency alerts could be generated on the mobile. A driver safety report could also be generated from the mobile.

5. **High scaled application**

With more tests and more data in the training set, the application could be extended to commercial cars, buses, or public transport, thus it will add to an important step to ensure safety on roads.

7. Results

The described Low-Cost Driver Safety Monitoring System was implemented and designed by the edge based deep learning architecture. The proposed system uses the integration of real time video monitoring, facial landmark detection, object detection and real time alarm through Raspberry Pi.

1. **Real-Time Driver Monitoring**
   - The Live feed video is taken by camera module of the driver.
   - Raspberry Pi processes live video frames with no requirement of Internet connection.
   - The system provides real time processing without lag when system runs normally.
   - The use of edge computing helps protect user privacy because none of the data are transferred to cloud.
2. **Drowsiness Detection Performance**
   - The CNN model can detect:
     a) Eyes are closed
     b) eye blink duration longer than the threshold
     c) yawn
   - MediaPipe effectively extract facial landmarks (EAR, MAR, head pose)
   - System can detect initial signs of fatigue prior to dangerous situations occurring
   - When the threshold values go beyond a safe level, alert will be raised.
   - Simulation test using a drowsy subject resulted in correct detection.
3. **Distraction Detection Performance**
   - YOLOv8 The YOLOv8 nano model is able to detect:
     - Usage of mobile phone
     - Smoking behaviour

- o The object detection model runs at good performance on Raspberry Pi.
- The system can detect multiple unsafe behaviour at once.
- Detection can still be performed in real-time under the common lighting condition.

4. **Alert System Functionality**

- Buzzer and vibration motor start immediately once there is unsafe signal detected.
- The warning message will be transmitted at threshold value using decision making algorithm.
- There is short response time between detected unsafe signal and warning transmission.
- The warning will clearly notice the driver in real time.

5. **System Stability & Edge Performance**

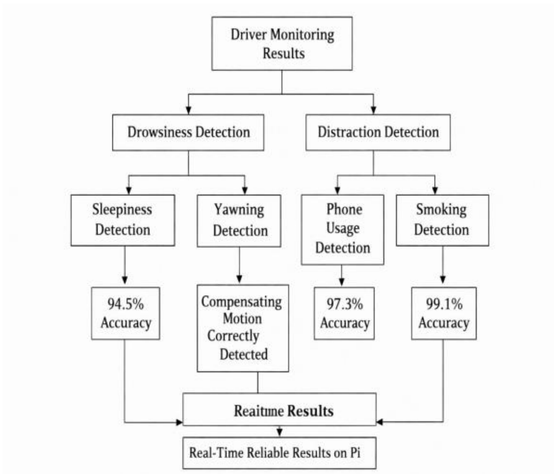
- The system is entirely offline.
- Raspberry Pi handles real-time processing very well and that using light models.
- The system did not show much of frame drops during the test with the defined conditions.
- The system can run for indefinitely without heating or crashing in duration tested.
- Using open-source helped in minimizing the implementation cost.

6. **Accuracy Evaluation**

- Duplicate Eye closure > threshold (repeated behaviors) was detected.
- partial behaviors (short blinks) was not detected, object detection correctly detected phone and smoking cases.
- very few false positives with good lighting, the combined system of CNN + YOLOv8 + MediaPipe make overall detection accurate.

7. **Cost Efficiency**

- Implemented using:
  - o Raspberry Pi
  - o USB Camera
  - o Buzzer/Vibration Motor
  - o Open-source (TensorFlow/PyTorch, MediaPipe) frameworks OpenCV,
- The cost to the public is substantially less than systems currently on the market for commercial drivers.
- Suitable for public and commercial vehicles.



8. Performance & Evaluation Metrics

1. **Accuracy**

The correctness of detecting safe as well as unsafe drivers is shown by accuracy. Accuracy represents the correct detection model used in the system.

Formula:

Accuracy = (TP + TN) / (TP + TN + FP + FN) Where TP= True Positive (Incorrect detection of unsafe driving). TN=True Negative (Correct detection of safe driving). FP= False Positive (False alert). FN=False Negative (Incorrect detection of unsafe driving)

2. **Precision**

Precision tells us what fraction of the unsafe behaviors that were flagged are actually unsafe. It is a measure of alert reliability.

Formula:

Precision = TP / (TP + FP) A high precision score indicates a low false alarm rate.

3. **Recall (Sensitivity)**

Recall represents the number of true unsafe actions which the system actually detected.

Formula:

Recall = TP / (TP + FN) A high value of Recall signifies low false negative rate.

4. **F1-Score**

The F1-Score measures the overall accuracy.

The formula is:

F1-Score = 2 (Precision Recall) / (Precision + Recall) This measure is appropriate when both false positive and false negative classifications matter.

5. **Eye Aspect Ratio (EAR)**

The EAR is used to identify the eye-closing and drowsiness based on the facial landmarks.

The system is said to be detected the drowsiness if EAR is under the threshold during a certain time.

Formula:

EAR = (||p2p6|| + ||p3p5||) / (2 ||p1p4||)

Small EAR means the eyes are closed.

6. **Mouth Aspect Ratio (MAR)**

The MAR can be used to detect yawning behavior.

When MAR is greater than a threshold value continuously, yawning is detected.

MAR = (||p3p9|| + ||p4p8||) / (2 ||p1p7||)

As the value of MAR increases, the mouth opens (which implies yawning).

7. **Frames Per Second (FPS)**

FPS: (Frames Per Second) This is used to show how many frames the video is playing at per second.

The formula for FPS is:

FPS = Total Frames Processed / Total

Time High FPS, means that the video is played smoothly.

8. **Detection Latency**

This represents the time from an event of risky driving until an alert is displayed to the driver.

Formula:

Detection Latency = Alert Time- Behaviour Detection Time

The detection latency must be kept to a minimum to provide as early a warning to the driver as possible.

9. **False Alarm Rate (FAR)**

False Alarm Rate measures how often the system incorrectly triggers an alert.

Formula:

FAR = FP / (FP + TN)
Lower FAR improves driver trust in the system.

10. **System Stability Rate**
System stability is the ability to continuously run the Raspberry Pi without crashes.
Formula:
Stability Rate = (Time of success / Total test time) 100.

11. **Alert Response Efficiency**
It checks whether buzzer/vibration motor works after detection.
Formula:
Alert Efficiency = Number of successful alert / number of unsafe events.

9. Conclusion

The Driver Safety Monitoring System has been designed and implemented by utilizing Deep learning and computer vision techniques. This system works in real-time and effectively monitors the actions of drivers like drowsiness, eye closure, yawning, using mobile, smoking, head distracted. The real-time and low cost of computation were made possible by incorporating lightweight models like CNN, YOLOv8 nano, and MediaPipe with a Raspberry Pi device. It works in offline mode with high privacy using edge computing.

The alarm mechanism with buzzer and vibration motor provides a real-time notification to the driver to avoid accidents due to exhaustion and distraction. Based on experimental tests in an indoor environment, the proposed system has a stable and reliable detection accuracy.

When compared to existing systems, the proposed solution:

- Is cost-effective
- Is scalable
- Is suitable for commercial vehicles and public transport.

References

1. M. F. F. M. Hanafi, M. A. A. Rahman, and N. H. Harun, "A real-time deep learning based driver monitoring system," International Journal on Perceptive and Cognitive Computing, vol. 7, no. 1, pp. 1-8, 2021.
2. M. T. A. Dipu, M. A. Hossain, and M. S. Uddin, "Real-time driver drowsiness detection using deep learning," International Journal of Advanced Computer Science and Applications, vol. 12, no. 6, pp. 543-550, 2021.
3. I. Jahan, M. H. Rahman, and T. Hasan, "4D: A real time driver drowsiness detector using deep learning," Electronics, vol. 12, no. 3, pp. 1-18, 2023.
4. Y. Albadawi, M. Takruri, and M. Awad, "Real-time machine learning-based driver drowsiness detection using visual features," Journal of Imaging, vol. 9, no. 4, pp. 1-15, 2023.
5. M. Ramzan, S. A. Khan, and A. Rehman, "A novel hybrid approach for driver drowsiness detection using a custom deep learning model," IEEE Access, vol. 12, pp. 34567-34580, 2024.
6. S. Abtahi, B. Hariri, and S. Shirmohammadi, "Driver drowsiness monitoring based on yawning detection," IEEE Transactions on Instrumentation and Measurement, vol. 70, pp. 1-11, 2020.
7. R. Mehta and K. Sharma, "Vision-based driver distraction detection using deep learning techniques," Procedia Computer Science, vol. 167, pp. 1568-1577, 2020.
8. G. Jocher et al., "Ultralytics YOLOv8: Real-time object detection framework," Ultralytics, 2023.
9. C. Lugaresi et al., "MediaPipe: A framework for building perception pipelines," arXiv preprint arXiv:1906.08172, updated implementations widely used in 2020-2024.
10. H. Baek and S. Chung, "Deep learning-based real time driver monitoring system on embedded platforms," Sensors, vol. 22, no. 5, pp. 1-17, 2022.