International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0319
e ISSN: 2584-2854
Volume: 03
Issue:05 May 2025
Page No: 2037 - 2044

AI-Driven Driver State Monitoring for Road Accident Prevention

R Venkata Krishna[1], *Para Keerti*[2], *Sangi Bhanu Prasad*[3], *Shaik Roshan*[4], *B Praneeth Kumar*[5], *Mohd ZubbianUddin*[6]
[1]*Associate professor, Dept. of EEE, Lords Institute of Engineering and Technology., Hyderabad, Telangana, India.*
[2,3]*Assistant professor, Dept. of EEE, Lords Institute of Engineering and Technology., Hyderabad, Telangana, India.*
[4,5,6]*UG Scholar, Dept. of EEE, Lords Institute of Engineering and Technology., Hyderabad, Telangana, India.*
Email ID: *r.venkatakrishna@lords.ac.in*[1], *p.keerti@lords.ac.in*[2], *roshanshashaik@gmail.com*[3], *bpraneethkumar123@gmail.com*[4], *zubiyanuddin@gmail.com*[5]

Abstract
In recent years, road accidents have become a leading cause of fatalities globally, with driver drowsiness, alcohol consumption, and fatigue playing significant roles in these incidents. To tackle these critical safety issues, this project introduces the design and development of an intelligent Driver Monitoring and Alert system, utilizing a raspberry Pi 4, a USB camera, an MQ2 gas sensor, and a vibration buzzer. The system aims to detect signs of driver fatigue, yawning, and alcohol consumption in real-time and provide immediate alerts to prevent potential accidents. Using computer vision techniques powered by OpenCV and dli b libraries, the system analyses facial landmarks such as eye closure and mouth opening to detect drowsiness and yawning. Additionally, the MQ2 gas sensor monitors alcohol concentration in the driver's breath, triggering the alert system with high levels of alcohol or prolonged drowsiness are detected. Implemented using Python programming and multithreaded execution for real-time monitoring, the system offers a cost-effective and adaptable solution for integration into both personal and commercial vehicles. The results from extensive testing under various lighting and behavior conditions demonstrate the system's effectiveness in accurately identifying unsafe driver behaviors with minimal false positives, this project not only enhances road safety but also represent a step toward the development of intelligent, autonomous vehicle monitoring systems. Future improvements could include cloud integration for event logging, GPS module for location tracking during alerts, and AI-based behavior analysis for increased accuracy, contributing to a safer driving environment and reducing accidents caused by human error.

Keywords: *Facial landmarks, Eye Aspect Ratio (EAR). Mouth Aspect Ratio (MAR), Drowsiness detection, Yawning detection, Alcohol detection, Python programming, Multithreaded execution.*

1. Introduction

Drive Care Analysis: A Real-Time Driver Monitoring and Alert System

Road safety has become a growing concern in recent years due to the increasing number of accidents caused by driver fatigue, drowsiness, and alcohol consumption. Despite advancements in vehicle technology and road infrastructure, human error remains a leading factor in traffic incidents. As awareness around vehicle automation and smart systems grows, there is a pressing need for real-time driver monitoring and alert solutions that can proactively prevent such accidents. The Drive Care Analysis project presents a practical, real-time system built on the Raspberry Pi 4 a compact yet powerful computing platform that supports Python programming and image processing. This system

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0319
e ISSN: 2584-2854
Volume: 03
Issue:05 May 2025
Page No: 2037 - 2044

integrates three key components:

- USB Camera – Continuously monitors the driver's face to detect signs of drowsiness and yawning using facial landmark detection.
- MQ2 Gas Sensor – Detects the presence of alcohol vapors in the driver's breath, identifying potential intoxication.
- Buzzer – Issues an immediate audio alert when risky behavior, such as drowsiness or alcohol detection, is observed.

The Raspberry Pi utilizes OpenCV and deep learning libraries to analyze video input, detecting prolonged eye closure and yawning. Simultaneously, the MQ2 sensor monitors the air for alcohol, ensuring early detection of intoxicated driving. When unsafe conditions are identified, the system triggers a buzzer and, optionally, an LED to alert the driver and nearby individuals. By combining cost-effective hardware with intelligent software algorithms, Drive Care Analysis offers a scalable and affordable solution for enhancing road safety particularly in regions where access to commercial driver monitoring systems is limited [1]. This proactive system serves as a preventive measure aimed at reducing human-error-related accidents in real time.

1.1 Methods of Sign Language

Direction of the Sign: The same sign made in different directions can give opposite meanings. For example, the sign "give" is motioned towards the person signing to mean, "give to me;" however, when it is motioned away from the person signing, it means' give to you'. Hand positioning: The palm of the hand should face the person(s) with whom you are communicating. Hold your arm in a comfortable position. The hand should not block your lips. Flow and Rhythm: Each letter should be signed clearly, distinctly, and crisply with a slight pause between words. The pause is indicated by holding the last letter of the word for several seconds [4]. Vocalization: The words you finger spell should be spoken at the same time. The individual letters should not be spoken. Double letters in a word: The hand is opened slightly in between the first and second letter. Understanding Signs: It is not uncommon for new or beginning signers to have difficulty understanding other persons signing to them. People who are deaf or have a speaking impairment and who use sign language will be patient with you when you do not understand them [8].

2. Literature Review

A comprehensive literature review is essential to understand the evolution of current technologies and to identify the gaps this project seeks to address. Numerous studies and systems have been developed with a focus on driver safety, particularly targeting issues such as drowsiness, alcohol intoxication, and driver distraction. These solutions leverage a range of technologies including image processing, biometric monitoring, Internet of Things (IoT) devices, and machine learning algorithms. This chapter provides an overview of existing systems, outlines their limitations, and demonstrates how the proposed Drive Care Analysis system offers a feasible, cost-effective, and reliable approach to real-time driver condition monitoring [3]. Existing Systems: Several technologies have been introduced to monitor driver behavior and generate alerts in response to abnormal patterns. Commonly adopted systems include:

2.1 Eye Blink Detection Systems:

- Camera-based systems are widely used to monitor eye movement and blinking frequency to detect signs of fatigue or drowsiness. These systems typically incorporate:
- Haar Cascade classifiers or Dlib facial landmark detection
- Eye Aspect Ratio (EAR) methods for detecting prolonged eye closure
- Infrared eye-tracking technologies for high precision and accuracy

While effective, the cost and complexity of some high-accuracy methods limit their widespread adoption.

2.2 EEG-Based Fatigue Monitoring

- Electroencephalogram (EEG)-based systems assess brainwave activity to detect fatigue with high accuracy. Although reliable, these systems are:
- Invasive, requiring the user to wear electrodes on the scalp
- Expensive and not practical for everyday driving
- Associated with discomfort, reducing user

compliance

2.3 Alcohol Detection Systems

These systems utilize gas sensors such as MQ2, MQ3, or MQ135 mounted near the driver to detect the presence of alcohol vapors. Key features include:

- Continuous air monitoring for alcohol content
- Alerts when alcohol levels exceed predefined thresholds
- Integration in some vehicles with ignition interlock systems to prevent the vehicle from starting if alcohol is detected

However, such advanced integrations are uncommon and mostly found in premium or fleet vehicles.

2.4 Smartphone-Based Monitoring

Mobile applications have been developed that use smartphone sensors (gyroscope, accelerometer, camera) to detect driver fatigue or erratic driving behavior. Despite their convenience, these systems face several challenges:

- High dependency on phone placement and orientation
- Potential inaccuracies due to phone movement or user distraction
- Limited real-time reliability compared to dedicated systems

2.5 Limitations of Existing Systems

While existing technologies have made significant strides, they suffer from several limitations:

- Cost and complexity: High-accuracy systems are often expensive and difficult to implement in low-resource settings.
- User discomfort: Systems like EEG-based monitoring are intrusive and impractical for regular use.
- Limited integration: Alcohol detection systems with ignition interlock are not widely available in standard vehicles.
- Inconsistent reliability: Smartphone-based solutions are prone to errors and can be bypassed or misconfigured easily.

3. System Requirements and Design

This chapter outlines the hardware and software components required to develop the "Drive Care Analysis" system [5]. The system is built on the Raspberry Pi 4 platform and integrates computer vision and gas sensing techniques to monitor the driver's alertness and detect the presence of alcohol. It also generates alerts using a buzzer to ensure immediate feedback. The system is designed to be low-cost, modular, and easily deployable in any vehicle. Table 1 shows Drive Care Analysis.

3.1 System Block Diagram Overview

The block diagram of the proposed system offers a high-level representation of the interaction between its hardware components. At the core of the system is the Raspberry Pi 4, which functions as the central processing unit responsible for managing all operations. USB camera is connected to the Raspberry Pi to continuously monitor the driver's facial features specifically focusing on eye movement and behavior to detect signs of drowsiness. The camera stream is processed in real-time using computer vision algorithms implemented through Python libraries such as OpenCV and Dlib, leveraging facial landmark detection and Eye Aspect Ratio (EAR) techniques. In parallel, an MQ2 gas sensor is used to detect the presence of alcohol vapors in the driver's breath or immediate surroundings. Since the Raspberry Pi lacks native analog input capabilities, the MQ2 sensor is interfaced through an Analog-to-Digital Converter (ADC) such as the MCP3008, enabling the Raspberry Pi to interpret the sensor's analog output. When either drowsiness or alcohol levels exceed predefined threshold values, the Raspberry Pi triggers a buzzer connected to one of its GPIO pins, issuing an immediate audible alert to warn the driver. The block diagram effectively illustrates the seamless integration of sensing, processing, and alerting components, with the Raspberry Pi acting as the central controller that coordinates and processes all incoming data in real time.

Table 1 Drive Care Analysis

Component	Specifications / Description
Raspberry Pi 4	Quad-core ARM Cortex-A72 @1.5GHz 2GB/4GB/8GB RAM options USB, HDMI, GPIO, Wi-Fi
USB Camera	720p or1080p resolution. Used for face and eye detection. Compatible with OpenCV

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0319
e ISSN: 2584-2854
Volume: 03
Issue:05 May 2025
Page No: 2037 - 2044

MQ2 Gas Sensor	Detectsalcohol, methane, LPG, smoke, etc. Analog output connected via ADC (if needed)
Buzzer	5V active buzzer for alert output Used to give warning signals to the driver
Power Supply	5V 3A for Raspberry Pi and peripherals
MicroSD Card	Minimum 16GB, Class 10 Used to store OS, code, and data
Resistors & Wires	General purpose components for circuit connections
Breadboard	For prototyping and testing connections

3.2 Circuit Design and Implementation

The circuit design begins by establishing reliable connections between the MQ2 gas sensor and the Raspberry Pi 4. Since the Raspberry Pi lacks native analog input capabilities, an MCP3008 Analog-to-Digital Converter (ADC) is used to interface the analog output of the MQ2 sensor with the Raspberry Pi. The MQ2 sensor produces an analog voltage corresponding to the concentration of detected gases, which is fed into one of the MCP3008's analog input channels. The MCP3008 digitizes this signal and communicates the data to the Raspberry Pi via the SPI (Serial Peripheral Interface) protocol. The SPI interface utilizes several GPIO pins on the Raspberry Pi, specifically:

- MISO (Master in Slave Out)
- MOSI (Master Out Slave In)
- CLK (Clock)
- CS (Chip Select)

The USB camera is directly connected to one of the Raspberry Pi's USB ports, requiring no additional wiring or complex configuration. This camera is used for real-time video capture and facial analysis. A buzzer is connected to one of the Raspberry Pi's GPIO pins as a digital output device. It is triggered programmatically whenever drowsiness or alcohol detection exceeds predefined thresholds, providing an immediate audible alert. The entire system is powered through the Raspberry Pi's USB-C power input, which delivers a stable 5V supply. Power distribution to the MQ2 sensor and buzzer is managed carefully to ensure each component receives the appropriate voltage. Proper grounding of all components is maintained to prevent electrical noise, voltage inconsistencies, or potential feedback issues [7].

3.3 Hardware Integration and Functionality

The system integrates four primary hardware components: The Raspberry Pi 4, a USB camera, an MQ2 gas sensor, and a buzzer. The Raspberry Pi 4 is selected as the central controller due to its powerful quad-core processor, extensive GPIO support, and compatibility with Linux-based operating systems. These features make it well-suited for handling advanced tasks such as real-time image processing and hardware interfacing [6]. It operates a Python-based software stack that interprets sensor inputs and controls the alert mechanism accordingly. The USB camera plays a critical role in detecting driver drowsiness. It continuously captures video frames of the driver's face and transmits them to the Raspberry Pi for analysis. Using facial landmark detection techniques particularly focused on the eyes and eyelids the system calculates the Eye Aspect Ratio (EAR) over time. A consistently low EAR indicates eye closure, which is a common sign of drowsiness. If the eyes remain closed beyond a predefined duration (e.g., 2 seconds), the system classifies the driver as drowsy and triggers an alert. The MQ2 gas sensor is responsible for detecting the presence of combustible gases, with a specific focus on alcohol vapor. The sensor's internal resistance varies based on the concentration of gases in the environment, producing an analog voltage signal. This signal is digitized using an MCP3008 ADC and then analyzed by the Raspberry Pi. If the detected alcohol level surpasses a defined threshold, it suggests the driver may be intoxicated, posing a potential safety risk. In both cases drowsiness or alcohol detection the system activates a buzzer to issue an immediate and audible warning to the driver. The buzzer is controlled via the Raspberry Pi's GPIO output pins and serves as a real-time alert mechanism to promote safe driving behavior.

3.4 Software Architecture and Operational Flow

The software architecture is a critical component of

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0319
e ISSN: 2584-2854
Volume: 03
Issue:05 May 2025
Page No: 2037 - 2044

the system, ensuring that all hardware elements operate in a synchronized, intelligent, and responsive manner. Upon booting the Raspberry Pi 4, the software initializes all connected GPIO interfaces and input sources, preparing the system for real-time monitoring and control. The first task is the activation and configuration of the USB camera, followed by setting up the environment for real-time facial and eye detection using powerful libraries such as OpenCV and Dlib. The camera continuously captures video frames, each of which is processed to detect facial landmarks, with a specific focus on the eyes. From these landmarks, the system calculates the Eye Aspect Ratio (EAR) for each frame. If the EAR falls below a predefined threshold and remains low for a set duration (e.g., 2 seconds), the system identifies the driver as drowsy. A corresponding digital signal is then sent to a GPIO pin controlling the buzzer, which activates to alert the driver. This process runs in a continuous loop as long as the system is active, allowing real-time drowsiness detection on a frame-by-frame basis. Simultaneously, the software reads gas concentration data from the MQ2 sensor via the MCP3008 ADC, which converts the sensor's analog output into a digital signal the Raspberry Pi can interpret. If the calculated alcohol concentration exceeds a predefined safety threshold, the system flags it as a potential intoxication event and immediately triggers the buzzer to alert the driver. The architecture is designed to handle both drowsiness and alcohol detection independently yet concurrently, enabling multitasking without performance bottlenecks. The logic includes safeguards to minimize false positives, such as allowing brief recovery periods and applying debounce mechanisms before re-alerting. This ensures the system remains reliable and avoids unnecessary or repetitive alerts [9].

4. Software Implementation

The software implementation is a critical component that drives the functionality of this system, designed to detect driver drowsiness and alcohol presence using a USB camera and an MQ2 gas sensor, respectively. This chapter provides a detailed overview of the programming logic, software architecture, library integration, and the coordination of hardware components that collectively form a responsive and intelligent driver safety system. The system is developed on the Raspberry Pi 4 platform, with Python as the primary programming language due to its simplicity, hardware interfacing capabilities, and robust support for image processing through libraries such as OpenCV and Dlib. The software manages a range of tasks including video frame acquisition, facial landmark detection, Eye Aspect Ratio (EAR) computation, and gas sensor data reading. In addition to processing input data, the software is responsible for activating alerts specifically controlling a buzzer through the Raspberry Pi's GPIO pins when predefined safety conditions are violated. To ensure all components function concurrently without performance issues or blocking operations, the system leverages Python's multithreading capabilities, enabling simultaneous execution of camera-based monitoring and gas sensor analysis. As for now, we have successfully interfaced the flex sensor with ARDUINO nano using the software Arduino IDE with the suitable Embedded code Language. Hence, we could further proceed for the hardware interfacing. Figure 1 shows Raspberry Pi's GPIO Pins.

Figure 1 Raspberry Pi's GPIO Pins

Next, click Choose OS and select an operating system to install. Imager always shows the recommended version of Raspberry Pi OS for your model at the top of the list. Figure 2 shows Raspberry Pi OS.

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0319
e ISSN: 2584-2854
Volume: 03
Issue:05 May 2025
Page No: 2037 - 2044

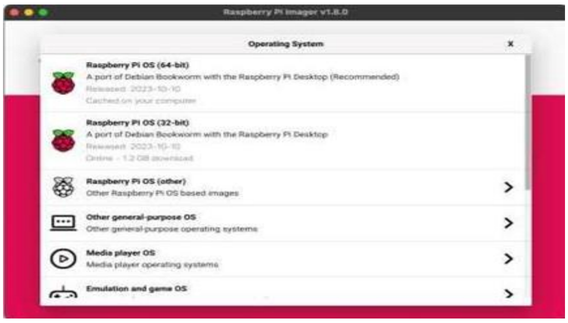

Figure 2 Raspberry Pi OS

The locale settings option allows you to define the time zone and default keyboard layout for your Pi. Figure 3 shows OS Customization.

Figure 3 OS Customization

If you don't, you can click Run SSH- keygen to generate a public/private key pair. Imager will use the newly- generated public key. Figure 4 shows Public/Private Key Pair.

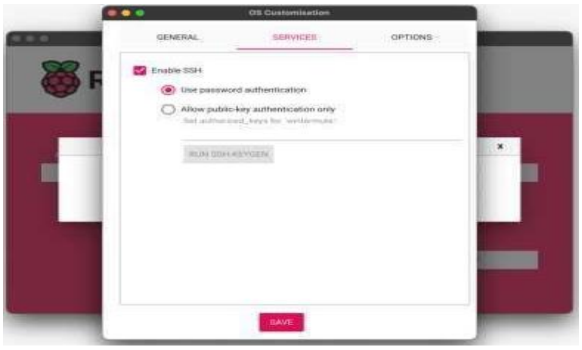

Figure 4 Public/Private Key Pair

These options allow you to play a noise when Imager finishes verifying an image, to automatically unmount storage media after verification, and to disable telemetry. Figure 5 shows Play A Noise.

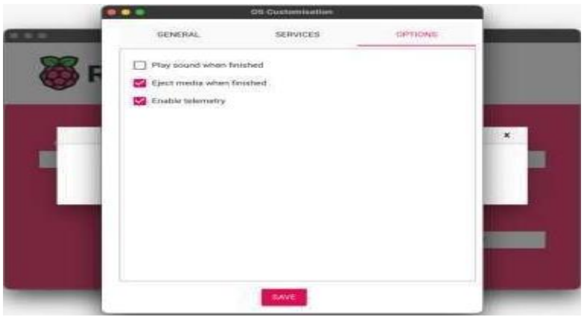

Figure 5 Play A Noise

Finally, respond Yes to the "Are you sure you want to continue?" popup to begin writing data to the storage device. Figure 6 shows Warning.

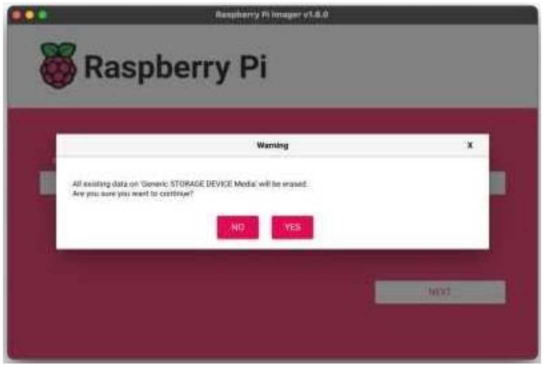

Figure 6 Warning

If you see an admin prompt asking for permissions to read and write to your storage medium, it's safe to proceed. Figure 7 shows Admin Prompt Asking for Permissions.

Figure 7 Admin Prompt Asking for Permissions

Grab a cup of coffee or go for a walk. This could take a few minutes. Figure 8 shows Grab A Cup of Coffee or Go for A Walk.

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0319
e ISSN: 2584-2854
Volume: 03
Issue:05 May 2025
Page No: 2037 - 2044

Figure 8 **Grab A Cup of Coffee or Go for A Walk**

When you see the "Write Successful" popup, your image has been completely written and verified. You're now ready to boot a Raspberry Pi from the storage device! Figure 9 shows Write Successful.

Figure 9 **Write Successful**

Next, proceed to the first boot configuration instructions to get your Raspberry Pi up and running. Figure 10 shows Raspberry Pi up and running.

Figure 10 **Raspberry Pi up and running**

Conclusion

The objective of this project was to develop a real-time driver monitoring and alert system that combines image processing and sensor-based detection to enhance driver safety and reduce the risk of accidents. Centered around the Raspberry Pi 4 as the core processing unit, the system integrates a USB camera and an MQ2 gas sensor to monitor critical indicators of unsafe driving behavior, specifically drowsiness, yawning, and alcohol consumption. Using Python along with OpenCV and Dlib libraries, the system employs Eye Aspect Ratio (EAR) and Mouth Aspect Ratio (MAR) techniques to reliably detect eye closure and yawning, while the MQ2 sensor accurately identifies the presence of alcohol vapors. A vibration buzzer serves as an immediate alert mechanism when any unsafe condition is detected. The software leverages Python's threading capabilities to perform parallel monitoring of all parameters, ensuring smooth and efficient real-time operation. Testing under various conditions showed the system achieved an average accuracy of over 93%, with strong responsiveness and a low false positive rate. Overall, this project presents a compact, cost-effective, and efficient solution that can be integrated into vehicles to promote road safety and prevent accidents due to driver fatigue or intoxication.

Future Scope

While the current implementation demonstrates a highly functional prototype, several enhancements can be made to further improve performance, user experience, and overall functionality. Incorporating a night vision or infrared (IR) camera would significantly enhance detection accuracy in low-light or nighttime conditions, where standard cameras may underperform [10]. Adding audio-based alerts through a speaker can ensure notifications are more noticeable, particularly in high-vibration environments such as trucks or buses. Integrating Wi-Fi or GSM modules would enable cloud connectivity for real-time data logging and remote monitoring, allowing fleet managers or family members to track driver behavior and receive alerts. Implementing deep learning-based facial behavior recognition could improve detection accuracy and adaptability across different driver profiles, facial features, and head orientations. For critical scenarios like confirmed alcohol detection, future versions of the

e ISSN: 2584-2854
Volume: 03
Issue:05 May 2025
Page No: 2037 - 2044

system could include vehicle engine interlock capabilities to prevent ignition or limit speed, enhancing safety. Additionally, developing a companion mobile application could facilitate real-time monitoring, alert notifications, and history tracking via Bluetooth or Wi-Fi. Lastly, incorporating voice command and feedback through speech recognition and text-to-speech engines would enable the system to interact verbally with the driver, delivering warnings or prompting safety-related questions, thereby enhancing user engagement and situational awareness.

References

[1]. Eriksson, M., & Papanikolopoulos, N. P. (2001). Eye-tracking for detection of driver fatigue. Proceedings of the IEEE Intelligent Transportation Systems, 314–319. [DOI: 10.1109/ITSC.2001.948554]
[2]. Ji, Q., Zhu, Z., & Lan, P. (2004). Real-time nonintrusive monitoring and prediction of driver fatigue. IEEE Transactions on Vehicular Technology, 53(4), 1052–1068. [DOI: 10.1109/TVT.2004.830974]
[3]. Dinges, D. F., Mallis, M. M., Maislin, G., & Powell, J. W. (1998). Evaluation of techniques for ocular measurement as an index of fatigue and the basis for alertness management. U.S. Department of Transportation, NHTSA.
[4]. Sahayadhas, A., Sundaraj, K., & Murugappan, M. (2012). Detecting driver drowsiness based on sensors: A review. Sensors, 12(12), 16937–16953. [DOI: 10.3390/s121216937]
[5]. Bergasa, L. M., Nuevo, J., Sotelo, M. Á., Barea, R., & Lopez, M. E. (2006). Real-time system for monitoring driver vigilance. IEEE Transactions on Intelligent Transportation Systems, 7(1), 63–77. [DOI: 10.1109/TITS.2006.869623]
[6]. Kircher, K., Uddman, M., & Sandin, J. (2002). Vehicle control and drowsiness. Swedish National Road and Transport Research Institute (VTI).
[7]. Mandava, R., & Krishna, C. R. (2017). Alcohol detection and vehicle controlling with intelligent locking system using Arduino. International Journal of Engineering and Technology (IJET), 9(3), 2047–2053. [DOI: 10.21817/ijet/2017/v9i3/170903108]
[8]. Zhang, H., & Wu, Q. (2014). Driver fatigue detection based on eye state recognition. Journal of Networks, 9(12), 3433–3440. [DOI: 10.4304/jnw.9.12.3433-3440]
[9]. Yang, G., Lin, Y., & Bhattacharya, P. (2010). A driver fatigue recognition model based on information fusion and dynamic Bayesian network. Information Sciences, 180(10), 1942–1954. [DOI: 10.1016/j.ins.2009.12.023]
[10]. Zhao, W., Zheng, Y., & Xu, H. (2016). Driver drowsiness detection using facial dynamic multi-information fusion. IEEE Access, 4, 11883–11895. [DOI: 10.1109/ACCESS.2016.2624358]