ABSTRACT

Intoxicated driving, sleepiness and reckless driving stands as the pre-eminent precipitant of vehicular accidents and fatalities worldwide. To mitigate such fatigue-related incident, the development of an intelligent system becomes imperative. This project provides a machine-based method for detecting driver's drowsiness automatically, leveraging continuous monitoring of driver through webcam or camera. Utilizing OpenCV, we extract the driver's facial features from the ongoing stream of image frames captured by the camera. The focal point of this model is the Region of Interest (ROI), namely the eye, recognizing its pivotal role in detecting drowsiness. Employing an algorithm, we track and analyze the blink rate of the eye. If the eyes are closed in more than two frames, the system promptly identifies the driver as drowsy and issues a timely alert by means of an alert sound.

Keywords: Drowsiness Detection, Eye Detection, Image Processing, OpenCV, Dlib

Chapter One: INTRODUCTION

1.1 Background of Study

Drowsiness refers to the state of being excessively tired or sleepy, often resulting in reduced alertness, impaired cognitive function, and a propensity to nod or fall asleep involuntarily. In the context of driving, drowsiness can significantly impair a person's ability to stay attentive, react swiftly to hazards, and make decisions there by increasing the risk of accident due to lack of sleep. Driving while drowsy, sleep deprived or exhausted are referred to as drowsy driving, fatigue driving or sleep deprived driving.

The National Highway Traffic Safety Administration (NHTSA) states that 91,000 police-reported crashes resulting in 50,000 injuries and 800 fatalities annually—amounting to approximately 1%–2% of all crashes, injuries, and deaths—involve drowsy driving; however, the contribution of drowsy driving in motor vehicle crashes is difficult to measure. (Brian C. Tefft *et al.,*2021)

Sleepiness can be caused by factors such as irregular work schedule, depressions, grief, and travelling across different time zone. Detecting driver's drowsiness early and issuing alert through an alarm is one of the techniques to reduce the incidence of accidents as frequent accident stem from drivers who are sleep deprived. To mitigate road accident, it is crucial to have a technology that detects driver's drowsiness. This technology shows the role of computer science in advancing the society by delivering beneficial services across multiple domains.

Fig

1.2 Problem Statement

Driver's inattention can stem from fatigue and distraction. Driver's drowsiness is the incremental decline in attention or focus when driving on the road. Similar to driver's drowsiness is drivers distraction which is the prevents someone from concentrating on driving task. Both driver's drowsiness and driver's distraction may have same effect such as decreased driver's performance and accident.

Drowsiness which is a safety concern that remains unaddressed globally due to its innate nature. Drowsiness, in general, is very tedious to measure unlike drugs, which have tests that are easily available. Creating awareness about fatigue related accident can be one of the solution to driver's drowsiness.

1.3 Aim and Objectives

The aim of this project is to detect signs of driver's drowsiness. The detection system comprised three components: the acquisition system, processing system and warning system. The acquisition system captures the video of the driver face and transfers it to the next stage which is the processing system. Processing is conducted in real time and if driver's drowsiness is detected, then the warning system triggers an alarm.

1.4 Methodology

Machine learning (ML) is a branch of Artificial intelligence that provides machine the ability to automatically learn from data and previous experience to identify pattern and make predictions with minimal human intervention. It is an approach for getting result automatically and it can be used with python programming language. This project was implemented using the following technologies:

1.4.1 OpenCV

OpenCV (Open-Source Computer Vision Library) is an open-source computer vision and machine learning software library. OpenCV was built to provide a common vision application and to accelerate the use of machine perception. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-art computer vision and machine learning. These algorithms can be used to detect and recognize faces, identify objects, classify human actions un videos, track camera movement, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch image together to produce high resolution image of an entire scene, find similar image from an image database and remove red eyes from images taken using flash. (Open-CV).

1.4.2 NumPy
NumPy stand for Numerical Python. It is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices.

1.4.3 Imutils
Imutils is a series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, displaying Matplotlib images sorting contours, detecting edges, and much more easier with OpenCV and Python.

1.4.4 SciPy
SciPy stands for Scientific Python. It is a free open-source python library. It provides more utility functions for optimization, stats and signal processing.

1.4.5 Dlib
Dlib stands for Digital Library. It is a popular toolkit for machine learning that is used for computer vision and image processing task, such as face detection, facial landmark detection, object detection, and more. It is written in C++ but has Python buildings, making it easily accessible from Python code.

a

1.4.4 Pygame
Pygame is a game development framework which provides graphical, audio and input functions for creating games and multimedia applications. Pygame has recently had a camera module added to it and with that can be used for basic image processing.

1.5 Significance of the study
This project was carried out to mitigate the high rate of accident caused by drivers drowsiness. The need to develop an intelligent system to detect drivers drowsiness and provide an appropriate response is crucial and also beneficial in the development of science and society in different ways. This project makes some important contribution to computer science by developing sensor technologies for drowsiness detection which fosters innovation in machine learning and computer vision. This project moves the automobile industries forward because drowsiness detection system can advance the development of safety standard and regulation in automobile industries by ensuring that vehicles are designed with drivers drowsiness detection system to protect driver's and passengers. Also, this system would beneficial to the society because it contributes to the well being of the society by saving lives and preventing injuries.

1.6 Scope and limitation

This study focuses on the development of driver’s drowsiness system using machine learning. Data collection involves testing the system on over 100 randomly selected drivers in Lagos State Nigeria.

This study will not cover other factors that may affect driving performance, such as vehicle malfunctions, road condition or distractions. Additionally, the finding will be applicable solely to the participant involved in this study and cannot be generalized to broader population of drivers.

CHAPTER TWO

LITERATURE REVIEW

2.0 OVERVIEW OF DRIVERS DROWSINESS DETECTION

Drivers' drowsiness detection is car safety technology which helps reduce accidents caused by fatigue driving. Various studies have suggested that around 20% of all road accidents are fatigue- related (Brian C. Tefft *et al.,*2021).

There are various methods developed over the years to detect driver drowsiness. These methods can be categorized into physiological-monitoring, vehicle monitoring and behavior al-monitoring.

2.1 Physiological Monitoring

Physiological monitoring involves monitoring the drivers biological signals electrooculography (EOG), electroencephalography (ECG), heartbeat, pulse rate etc., helps in detecting driver drowsiness based on physiological method and they are intrusive measurement due to the direct connection of sensors to the driver. But this may not be good for both drivers and car companies. To date this has not been considered successful. (A. Azman *et al.,* 2021). [9]

2.2. Heart Rate and Electroencephalography (EEG)

2.2.1 Heart rate monitoring: Variation in heart rate can indicate drowsiness. These changes are monitored by sensors attached to the drivers body. (K. T Chui *et al.,* 2021)

2.2.2 Electroencephalography (EEG): This is one of the most accurate method for detecting driver's drowsiness. (K. T Chui *et al.,* 2021)

2.3 Vehicles Based Monitoring:

The steering wheel movement, the accelerator of the vehicle or pattern of vehicle breaks, vehicles speed and deviation in position of lane are monitored continuously in the method which is based on vehicle. If there are any deviation from the values detected, it is considered as drowsiness. (IEEE, 2014)

2.3.1 Steering Patterns and Lane Positioning

Steering Patterns: Abnormal steering behavior such as reduced steering activities indicated drowsiness. Sensors attached to the steering monitors this often and any deviation from the values detected is considered drowsiness. (IEEE, 2014).

Lane Detection: lane detection algorithm tracks byte vehicle's position on the road. If frequent lane drifting is detected, it is detected, it is considered drowsiness. (IEEE, 2014).

2.4 Behavioral Monitoring

In behavioral monitoring, a camera is used together with an image processing technique to analyze the driver's physical state. This method is non- intrusive. (Kumaran C. et al., 2020)

2.4.1 Facial Expression and Eye Tracking

Facial Expression: Facial recognition algorithm can be used to detect visual behaviors like yawning. (Kumaran C. et al., 2020)

Eye Tracking: Drowsiness can be detected by monitoring eye movements such as blinking of eyes and eye closure duration. (Kumaran C. et al., 2020)

2.4.2 Head Position: Drowsiness and be detected by the position and movement of the driver's head such as bending of head, nodding etc. (Kumaran C. et al., 2020)

2.5 Proposed Approach

Almost all drivers have experienced this drowsiness problem while driving. Youngsters and professional drivers are mostly affected by this drowsy driving because of continuous hours of driving without any rest. (Satish K. et al., 2020).

Auto drivers and cab drivers in many cities drive continuously overtime sometimes to complete their targets or to get bonus profit. In order to meet daily expenses, most workers tend to work in night shifts for long time, this can be one of the main reasons for accidents taking place because of drowsy driving. Therefore, a driver drowsiness monitoring system has been developed to mitigate the rate of accident (Satish K. et al., 2020).

The four steps of our driver sleepiness detection system are as follows:

1. Detection Stage: This is the system's startup step. The system must be set up and optimised for the present user and conditions every time it is started. The effective identification of heads is the most important step in this stage. If the driver's head is accurately found, we can extract the characteristics required for the system's setup.

2. Tracking Stage: The system begins the regular tracking (monitoring) stage after the driver's head and eyes have been adequately found and all relevant characteristics have been retrieved. The constant monitoring of the driver's eyes within a dynamically allotted tracking region is a critical step in this stage. More specifically, the system will calculate the size of the tracking region based on the historical data of eye movements to reduce processing time. For example, if the eyes have been travelling horizontally to the left for a number of frames, that tendency is likely to continue in the next frame. As a result, it makes sense to enlarge the tracking area in the predicted direction of the eyes while shrinking. The

system must also determine the status of the eyes at this point. All of these operations must be completed in real time; depending on the processor's capabilities and the present load, it may be necessary to miss a few frames from time to time without affecting algorithmic precision.

3. Warning Stage: The driver's attention must be enhanced if he keeps his eyes closed for an extended amount of time or begins to nod. The crucial step at this stage is to keep a tight check on the driver's eyes. The system must figure out if the eyes are still closed and where they are in relation to previously determined criteria. At this point, we can't afford to miss frames. In practice, eye tracking is done in very much the same way as the tracking stage, with the addition of the following processes: eye velocity and trajectory computation, and threshold monitoring. These extra calculations are needed to increase the system's capacity to assess whether or not the driver is sleepy.

4. Alert Stage: Once the system has decided that the driver looks to be in an aberrant driving condition, it must be proactive in alerting the driver of potential threats. To draw the driver's attention and enhance their awareness

The block diagram of the proposed method for drowsiness monitoring is shown in figure

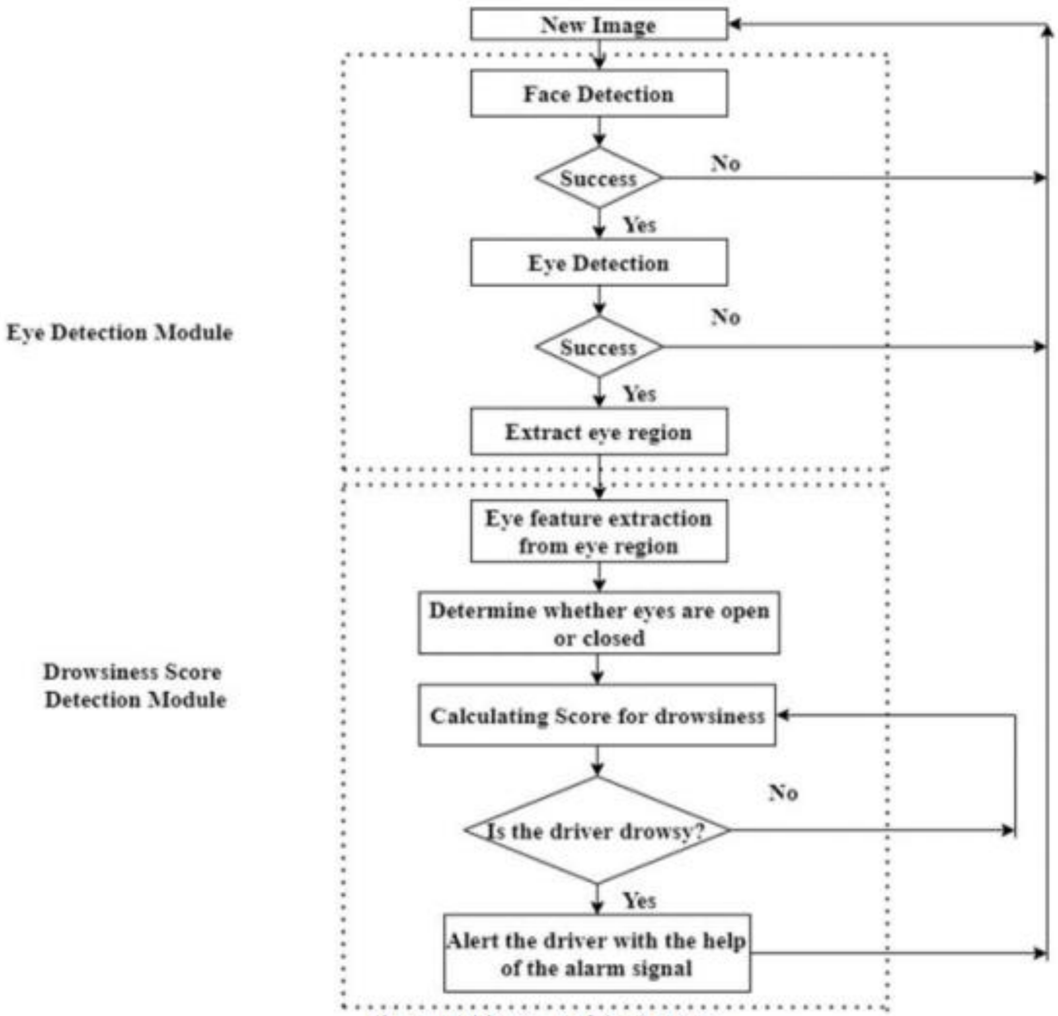

Fig 1. Architecture of the System

CHAPTER THREE

RESERCH METHODOLOGY

3.0 Introduction

Driver drowsiness detection is a vehicle safety feature that helps to save driver's life and mitigate accidents when the driver becomes drowsy. The primary goal is to develop a system that can identify driver drowsiness by continually monitoring the retina of the eye. By deploying an alarm, the driver will be notified if drowsiness is detected.

This chapter outlines the methodology used for developing and evaluating the driver’s drowsiness detection system. This approach combines behavioral monitoring and machine learning technique to achieve high accuracy.

3.1 Dataset Description

The dataset consists of video recording of drivers when drowsy. Facial expression and eye movements are captured by the system under different lightning conditions.

3.1.2 Acquiring Data:

At first, the video is captured and recorded using a laptop webcam and then the frames are released and proposed in a laptop. Then after the completion of extraction of frames, the picture implementation techniques are proposed on the 2-dimensional picture of the recorded video. Now, the required driver information is produced. The volunteers are asked to focus on the laptop webcam and perform activity like continual eye blinking.

3.2 Data Processing

Data processing is crucial because it converts data into machine readable form and evaluation. It is important for data processing to be done correctly as not to negatively affect the end product. Data processing starts with data in it’s raw form and converts it into a more readable format giving it the form and context necessary to be interpreted by the computer.

3.2.1 Facial Landmarks

Facial landmarks are used for localizing and representing salient regions or facial parts of the person’s face, such as: Eyebrows, Eyes, Jaws, Nose, Mouth, etc. In this context of facial landmarks, our vital aim is to detect facial structures on the person’s face using a method called shape prediction. The above method of shape prediction is also achieved using the HOG + Linear SVM Algorithm. The facial landmark detector of the dlib library which is pre-trained inside the dlib library of python for detecting landmarks, is used to estimate the location of 68 points or (x, y) coordinates which map to the facial structures. These indexes of 68 coordinates or points can be easily visualized on the image below:

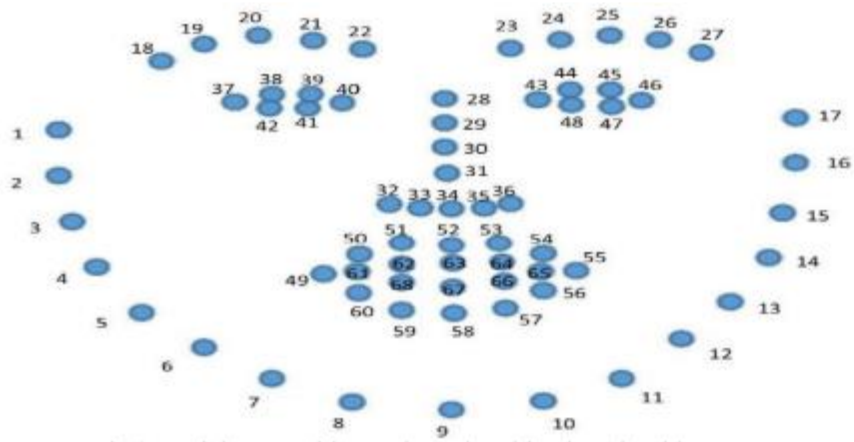

Fig 2. Facial Recognition and Landmarking by Algorithm

Some of the prerequisites for this algorithm are :

- Installation of dlib and opencv library
- Downloading the dlib shape predictor. It is a file with the .dat extension

3.2.2 Feature Extraction

After marking the points on face, the drowsiness features are calculated as given below.

Eye Aspect Ratio (EAR): EAR measures the openness of the eye. EAR consists of distance from left to right corner of the eye, by calculating the EAR ratio for both left and right eye then it takes the average EAR of both the eyes. From the boundary points on the eyes, the EAR is evaluated as inverse ratio of width of eyes to the height of eyes.

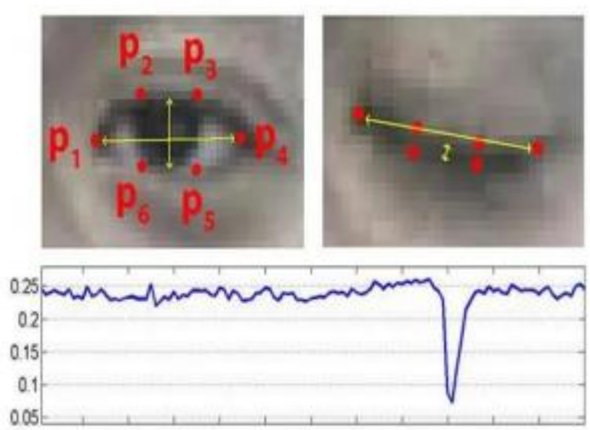

Fig 3: Eye Aspect Ratio

The mathematical formula for EAR is given as,

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

Fig 3: Formula for EAR

where i is the point marked as i in facial landmarking (i − j) gives the distance between the points i and j. When the eyes are open EAR is max and when the eyes are closed, the EAR is approximated to zero. So, the decreasing value of EAR indicates the closing of eyes and identifies the drowsy behavior of the driver.

3.3 Model Selection and Training

Once the data has been prepared and features selected or engineered, the next step is to select a suitable machine learning algorithm to train the model. This involves splitting the data into training and testing sets and using the training set to fit the model.

3.4 Implementation Process Algorithm

In this section, an algorithm is developed for video capture, feature extraction, and drowsiness detection. This algorithm captures videos of the driver from the camera , and use the eye aspect ratio (EAR) model to predict whether the eye is opened or not. A cohesive system is achieved here by integrating and hardware and software components such as the camera.

3. 5 Choice of Tools used

Hardware	• Web Camera • Laptop
Programming Language	• Python
Python Libraries	• Dlib • OpenCV • SciPy • Imutils • Pygame
System Requirement	• 8.0 GB RAM •]Intel or AMD x86–64 processor
Operating System	• Windows 11 • Windows 10 (version 20H2 or higher) • Windows Server 2019 • Windows Server 2022

Table 2. Hardware and Software Requirements

3.6 Validation of the Prediction

The validation process ensures the system's predictions are accurate. This includes real world testing which tests the system with drivers to evaluate its performance and effectiveness. User feedback are being gathered based on the system's performance and effectiveness.

3.6.1 Real-world Testing

Testing the system with drivers in actual driving conditions to evaluate its real-world performance.

3.6.2 User Feedback

Gathering feedback from users on the system's comfort, intrusiveness, and overall effectiveness, and making necessary adjustments based on this input.

3.7 Proposed Pipeline of the Research.

The proposed research pipeline outlines the steps from data collection to system deployment.

3.7.1 Data Collection

Recording video data from diverse drivers under various conditions and annotating the data for training and validation.

3.7.2 Data Preprocessing
Extracting frames, detecting faces, and extracting features from the video data. Normalizing the features to prepare the dataset for model training.

3.7.3 Model Development
Selecting, training, and validating machine learning models. Using ensemble methods to enhance accuracy and robustness.

3.7.4 System Implementation
Developing algorithms for video capture, preprocessing, feature extraction, and drowsiness detection. Integrating software modules with hardware components.

3.7.5 Testing and Validation

Conducting real-world tests to evaluate the system’s performance. Using evaluation metrics and user feedback to refine and improve the system.

3.7.6 Deployment
Deploying the system in vehicles and monitoring its performance over time. Continuously updating the models and algorithms based on new data and user feedback.

CHAPTER FOUR

RESULT AND DISCUSSION

Feeding real world image into the proposed model is important to test the effectiveness of the model. The webcam of the laptop is turned on and the face of the driver is recorded and captured. The captured video is then transferred to the processing block to detect drowsiness. If drowsiness is detected, then an alarm will be triggered to alert the driver. In this proposed model, a performance accuracy of 90-95% was achieved by using Machine Learning and Eye Aspect Ratio (EAR). This indicates that the model is effective and provides accurate results. The result of driver's drowsiness detection is shown in the figure below:

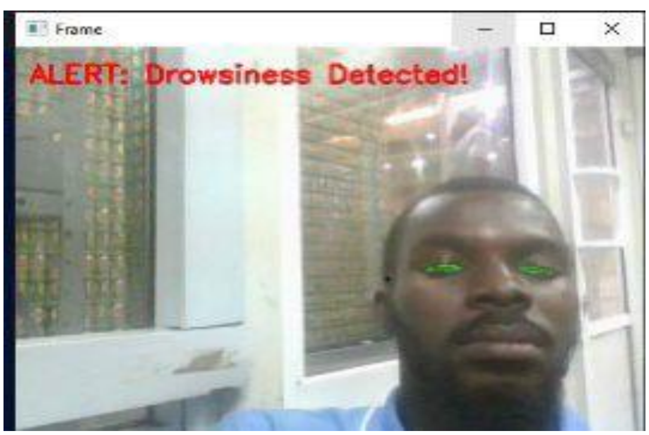

Fig. 4(i) Drowsy identification based on closed eyes

Various drowsiness situations are shown in Fig. 4. Figure 4(i) shows the example of drowsiness alert due to eye closing.

CHAPTER FIVE

RESULT AND DISCUSSION

5.1 CONCLUSION

This project has been developed efficiently to detect driver’s drowsiness using Machine Learning algorithm and visual behavior features such as Eye Aspect Ratio and alerting driver using an alarm which effectively awakes the driver in real time to prevent road accidents. The model was trained for Drivers Drowsiness using OpenCV, NumPy, Pygame and used Eye Aspect

Ratio (EAR) algorithm. By integrating OpenCV, video of the driver was captured through laptop webcam to detect drowsiness. The developed system worked perfectly with given data. The model was able to achieve 90-95% accuracy. Additionally, this model is also useful in other drowsiness detection application such as, Offices, Factories etc.

5.2 RECOMMENDATION

To further enhance the effectiveness and applicability of the driver drowsiness detection system, several recommendations are proposed:

Enhance Data Processing: Improve technique for handling diverse lighting conditions and facial abstractions to ensure consistent feature extraction and detection accuracy.

Integrate Multi-Sensor Data: Combine data from sensors like heart rate monitors and steering patterns sensors to efficiently detect driver's drowsiness.

Upgrade Camera: Utilizing higher resolution camara to improve quality of the captured video and increase accuracy of detection system.

5.3 FUTURE WORK

Driver sleepiness is a major concern in today's culture, as the drowning problem is causing an increase in accidents on a regular basis. With the help of Neural Networks and other real-time sensor devices, it will be deployed in the future. Current drivers drowsiness detection system can be improved so that more accuracy may be achieved:

1. Develop adaptive algorithm that can learn diverse drivers behavior and preference there by improving system performance
2. Ensure compatibility with different vehicle models.
3. Validate system robustness by conducting extensive field trials in a wider range of real-world driving conditions, including various geographical locations and weather scenarios.

4. Ensure conformity with established industrial standard and regulatory protocols to support system deployment in commercial vehicles.