JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Driver Distraction Detection Using Machine Learning

Shreyash Wankhede, Rushikesh Nalgire, Vikrant Salunke, Faique Khan, Prasad Bhosale (Professor)

Computer Science BE Student, Trinity College of Engineering & Research, Pune, India

ABSTRACT

Driver drowsiness and fatigue decrease the vehicle management skills of a driver. The operator driving vehicles at night has become a significant downside today. Driver in a drowsing state is one of the important reasons for an increasing amount of road accidents and death. Hence the drowsiness detection of the driver is considered the most active research field. Many ways are created recently to detect the drowsiness of the driver. Existing methods can be classified into three categories based on physiological measures, performance measures of vehicles and ocular measures.

INTRODUCTION

Recent studies show that the methods using distance calculation between the driver's eyes are open or closed can achieve better reliability and accuracy of driver drowsiness detection compared to other methods. The main idea behind this project is to develop a system which can detect the drowsiness of the driver and issue a timely warning. According to Scroll.in, This new year more than 2000 people were booked for drunk and driving. Between the years 2008 and 2017,76,446 people died in 2,11,405 road accidents nationally wide due to alcohol. This system is implemented to reduce road accidents. The system implemented is used to detect drowsiness and alcohol detection. The embedded system consists of some combination of hardware and software to do a particular function.

1. LITERATURE SURVEY

2.1 Monitoring Driver's Drowsiness Status at Night Based on Computer Vision

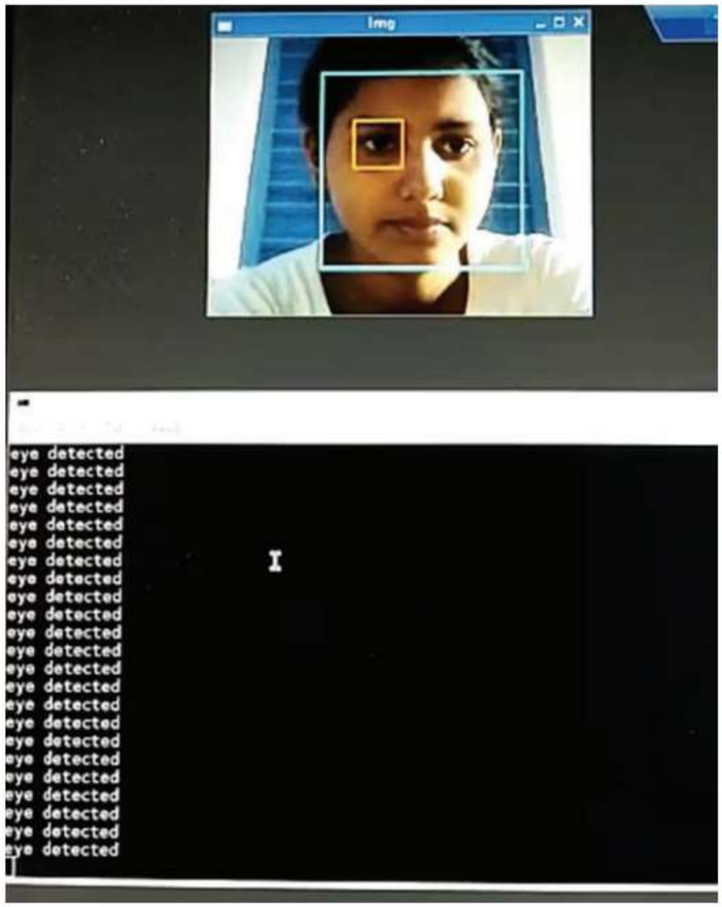
Driver drowsiness and fatigue decrease the vehicle management skills of a driver. The operator driving vehicles at night has become a significant downside today. Driver in a drowsing state is one of the important reasons for the increasing amount of road accidents and death. Hence the drowsiness detection of the driver is considered the most active research field. Many ways are created recently to detect the drowsiness of the driver. Drowsiness is detected based on the values of these parameters. The adaptive thresholding method is used to set the thresholds. Machine learning algorithms were also implemented in an offline manner. The proposed system was tested on the Face Dataset and also tested in real time. The experimental results show that the system is accurate and robust.

Fig. The facial landmark points

A low-cost, real-time driver's drowsiness detection system is developed with acceptable accuracy. This presents a night-based real-time fatigue driving detection system. It makes up for the deficiencies of fatigue driving detection during the night. In this proposed system, real-time video of the driver is recorded using a digital camera. Using some image processing techniques, the face of the driver is detected in each frame of the video. Facial landmarks points on the driver's face are localized using one shape predictor and calculating eye aspect ratio, mouth opening ratio, and yawning frequency subsequently. The enhancement using CLAHE enables us to detect drowsiness in low-lighting conditions as well as at night. This proposed framework is more accurate and robust. Decreased false rate, less computational cost, easy to use and portable compared to other existing methods. It is also possible to implement this proposed system as a mobile app in future.

2.2 Alert System For Driver's Drowsiness Using Image Processing

Accidents occur all over the world cause of being not able to concentrate on the road while driving. The concentration is missed due to driving the car without resting which makes the person drowsy. And this drowsiness becomes the reason for major accidents. This problem is overcome by developing various systems for detecting sleep. The proposed system here uses Raspberry Pi and various sensors like Gas Sensors, Vibration sensors for the detection of the type of drowsiness. The driver is been monitored by placing a camera which captures the vital sign. If the eye is closed for a longer period then the image of the person is sent. The accident is detected using a vibration sensor and the server is notified by sending latitude and longitude. The location of the car is sent by the IOT modem which is embedded in the car. If the driver consumed alcohol it is sensed by the gas sensor and the server is notified by a message. The motor of the car is continuously running and can be stopped or cut off if the server side is notified by the driver that the person is not in a situation to drive the car. By this, the accident rate is reduced and the risk to customer life is decreased.



This paper presents a method to detect the driver's drowsiness problem that will help to avoid road accidents. The main cause of an accident can be due to lack of sleep which leads to drowsiness and the consumption of alcohol. These causes are solved by using sensors to sense the alcohol and a camera which is used to monitor the person continuously. The gas sensor detects the alcohol, and the vibration sensor is fixed in the car that detects the accident and reports it to the server side. The driver actions are stored in the cloud which is controlled by the server side. The location of the driver is also notified to the server side. By this, the accident occurring rate can be lowered and the risk of life will not be high. This system can be used as a prevention method so that the accident may not occur cause of sleep or alcohol. It is always better to take precautions by embedding this type of system in the car for safety.

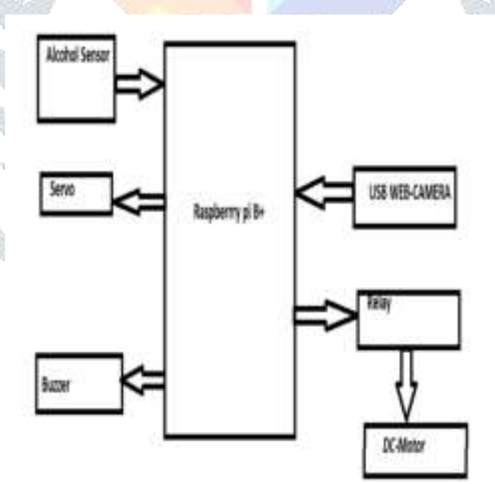
2.3 Comprehensive Drowsiness Level Detection Model Combining Multi-modal Information

This paper presents a drowsiness detection model that is capable of sensing the entire range of stages of drowsiness, from weak to strong. The key assumption underlying our approach is that the sitting posture-related index can indicate weak drowsiness that drivers themselves do not notice. We first determined the sensitivity of the posture index and conventional indices for the stages of drowsiness. Then, we designed a drowsiness detection model combining several indices sensitive to weak drowsiness and strong drowsiness, to cover all drowsiness stages. Subsequently, the model was trained and evaluated on a dataset comprised of data collected from approximately 50 drivers in simulated driving experiments. The results indicated that posture information improved the accuracy of weak drowsiness detection, and our proposed model using the driver's blink and posture information covered all stages of drowsiness (F1-score 53.6%, root mean square error 0.620). Future applications of this model include not only warning systems for dangerously drowsy drivers but also systems which can take action before their drivers become drowsy. Since measuring the information requires no restrictive equipment such as on-body electrodes, the model presented here based on the blink and posture information can be used in several practical applications.

In this paper, we proposed a drowsiness detection model designed to cover all drowsiness levels, from slight to severe. The posture information was particularly useful in conjunction with blink information because the posture index showed higher sensitivity to weak drowsiness than conventional information and was able to compensate for the shortcomings of the blink information. Since blink and posture information can be obtained even while not driving, this knowledge has the potential to contribute to drowsiness detection for occupants during automated driving in addition to manual driving. Future studies will focus on the development of arousing and arousal-maintenance technologies after drowsiness detection. Of note, people in a state of slight drowsiness are likely to be aroused by a relatively weak stimulus. Our achievement of drowsiness detection over a wide range of levels, including slight drowsiness, will enable the provision of interfaces that allow the selection of appropriate stimuli optimized to match the driver's degree of drowsiness and driving conditions.

2.4 Safe Driving By Detecting Lane Discipline and Driver Drowsiness

This paper explores two methodologies for drowsiness detection using EEG signals in a sustained attention-driving task considering pre-event time windows, and focusing on cross-subject zero calibration. Driving accidents are a major cause of injuries and deaths on the road. A considerable portion of those are due to fatigue and drowsiness. Advanced driver assistance systems that could detect mental states which are associated with hazardous situations, such as drowsiness, are of critical importance. EEG signals are used widely for brain-computer interfaces, as well as mental state recognition. However, these systems are still difficult to design due to very low signal-to-noise ratios and cross-subject disparities, requiring individual calibration cycles. To tackle this research domain, here, we explore drowsiness detection based on EEG signals' spatiotemporal image encoding representations in the form of either recurrence plots or grammar angular fields for deep convolutional neural network (CNN) classification. Results comparing both techniques using a public dataset of 27 subjects show a superior balanced accuracy of up to 75.87% for leave-one-out cross-validation, using both techniques, against works in the literature, demonstrating the possibility to pursue cross-subject zero calibration design.



This paper presents a real-time lane detection and driver fatigue or driver drowsiness detection system, which can effectively detect the anomaly while driving. The system uses Hough Transform to detect lanes on the road as well as tacks the eyes to detect if they are open or closed. The Hough transform used is simple and faster for adequate detection of lanes of the road. For eyes detection, first voila jones method is used to detect the face, then image segmentation is done, Otsu thresholding is performed and Canny edge detection is done, the result obtained is then applied with Circle detection Hough Transform, to detect the eyes. This is a foolproof method for detecting the eyes, and the accuracy is very high. Since it uses Otsu thresholding and canny edge detection, the eyes are detected even in low-light conditions. This system will be particularly useful for drivers travelling on long routes, night drivers and people who drink and drive.

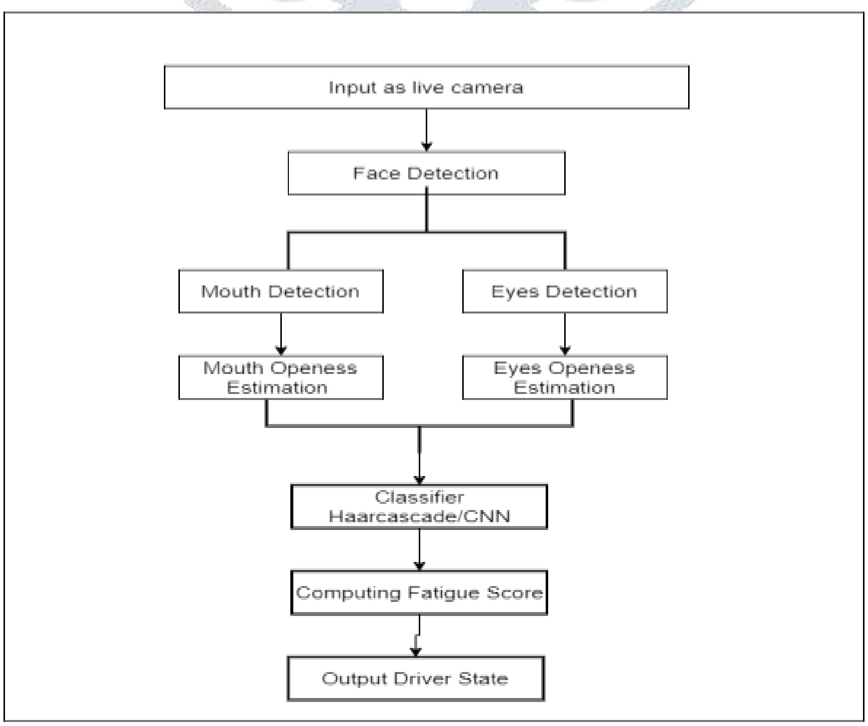
2.5 A Study on Feature Extraction Methods Used to Estimate a Driver's Level of Drowsiness

Recently, in addition to autonomous vehicle technology research and development, machine learning methods have been used to predict a driver's condition and emotions to provide information that will improve road safety. A driver's condition can be estimated not only by basic characteristics such as gender, age, and driving experience, but also by a driver's facial expressions, bio-signals, and driving behaviours. Recent developments in video processing using machine learning have enabled images obtained from cameras to be analysed with high accuracy. Therefore, based on the relationship between facial features and a driver's drowsy state, variables that reflect facial features have been established. In this paper, we proposed a method for extracting detailed features of the eyes, the mouth, and the positions of the head using OpenCV and Dlib library to estimate a driver's level of drowsiness.

To estimate a driver's condition, certain facial features were identified on the video of a male individual driving using a driving simulator. Using OpenCV and Dlib in Python, the features examined were frequency of eye closure, PERCLOS, ECD, frequency of yawning, and head positions. There will undoubtedly be limitations concerning the detection of drivers' behaviours and facial expressions due to factors such as light reflection, the wearing of sunglasses, darkness, and obstructions by drivers' hands. Therefore, a driver's condition can be also measured by biosensors such as EEG and ECG sensors. Furthermore, information obtained by in-vehicle environmental sensors during actual driving would be valuable. Temperature, humidity, and CO_2 sensors can be used to predict a driver's condition. Monitoring travel routes and a driver's behavioural characteristics, such as driving speed, steering, acceleration, and deceleration, is also necessary to estimate a driver's fitness for driving.

4. METHODOLOGY & IMPLEMENTATION

In this section, the proposed scheme to determine driver drowsiness is introduced. There are four steps to decide about the driver's fatigue. First, the face region should be extracted from the captured image. Second, the eye area is found in the face. Third, the mouth should be detected in the face area. These two tasks collaborate to improve the face detection results. Then yawn and eye closure detection tasks are applied to the extracted mouth and eye.
Next, the results are fused and the decision is made on the drowsiness of the driver. Finally, if a drowsiness state is determined, an alert is sent to the driver. Subsequently, the face is tracked in the next captured frames and the procedure is repeated. In the following figure, the architecture of the system is given.



We are using the waterfall model for our project.

1. Requirement gathering and analysis:

In this step of the waterfall, we identify what various requirements are needed for our project such are software and hardware required, database, and interfaces.

1. System Design:

In this system design phase, we design a system which is easily understood by the end user i.e., user friendly.
We design some UML diagrams and data flow diagrams to understand the system flow and system module and sequence of execution.

1. Implementation:

In the implementation phase of our project, we have implemented various modules required to successfully get the expected outcome at the different module levels.
With inputs from system design, the system is first developed in small programs called units, which are integrated into the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.

1. Testing:

The different test cases are performed to test whether the project modules are giving the expected outcome in the assumed time.
All the units developed in the implementation phase are integrated into a system after testing each unit. Post integration the entire system is tested for any faults and failures.

1. Deployment of System:

Once the functional and non-functional testing is done, the product is deployed in the customer environment or released into the market.

1. Maintenance:

There are some issues which come up in the client environment. To fix those issues patches are released. Also, to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.
All these phases are cascaded to each other in which progress is seen as flowing steadily downwards like a waterfall through the phases. The next phase is started only after the defined set of goals are achieved for the previous phase and it is signed off, so the name "Waterfall Model". In this model, phases do not overlap.

5. TESTING & RESULTS

• GUI Testing
Graphical User Interface (GUI) testing is one of the mechanisms by which a user interface is developed System Under some graphical rules. GUI testing includes checking various controls- menus, buttons, icons, dialogue boxes and windows etc. The proposed system is tested for user inputs against different modules, validations are done. GUI is tested for the appearance of different controls; visibility graphs are tested. GUI testing involves the following actions:

1. Check all elements for size, position, width, length and acceptance of characters or numbers. For instance, you must be able to provide inputs to the input fields.
2. Overall functionality related to the performance of the user's graphical interface is checked.
3. Check Error Messages are displayed correctly
4. Check the font, layout details, and style and display warning messages if it is false.
5. check the positioning of GUI elements.

• 8.2.1 Unit Testing

It is the testing of individual software units of the application.it is done after the completion of an individual unit before integration. Unit testing involves the design of test cases that validate that the internal program logic is functioning properly and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. This is a structural testing, that relies on knowledge of its construction and is invasive.

Unit tests perform basic tests at the component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

• 8.2.2 Integration Testing

Integration tests are designed to test integrated software components to determine if they actually, run as one program. Testing is event-driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successful unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

• Testing Strategy

Software testing methods are traditionally divided into white- and black-box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

1. White-box testing

In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases.

2. Black-box testing

Black-box testing treats the software as a 'black box', examining functionality without any knowledge of the internal implementation. The testers are only aware of what the software is supposed to do, not how it does it.

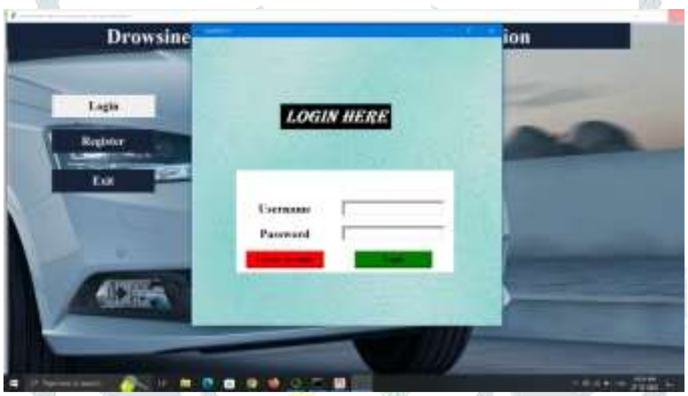
3. Grey-box testing

Grey-box testing involves having knowledge of internal data structures and algorithms for purposes of designing tests while executing those tests at the user, or black-box level. The tester is not required to have full access to the software.

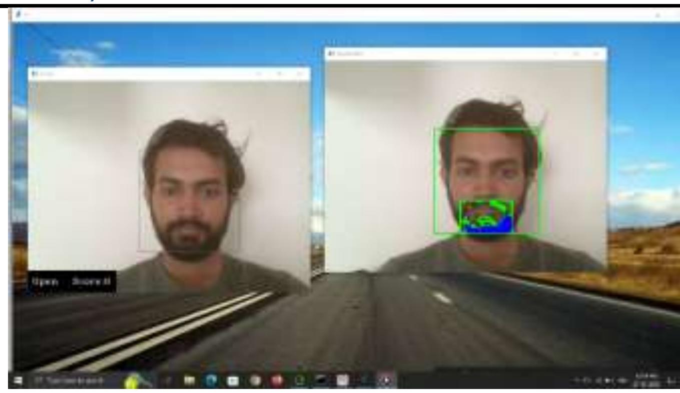
The accuracy test results used different test subjects as samples such as people with regular eyes, small or squinty eyes, and people wearing eyeglasses. The captured images are compared to the actual reading of the prototype. The reading can be normal, which means the eyes of the driver are detected to be open and not drowsy. The other status is drowsy, which means the eyes of the driver are closed for over 2 seconds and denotes that the driver is sleepy. The conducted testing procedure is able to determine the drowsiness of a driver. The sample for different eye size variations is continuously evaluated for the accuracy of the eye detection and drowsiness prevention without resetting the device while matched means drowsiness was detected upon closing the eyelid for more than three seconds.

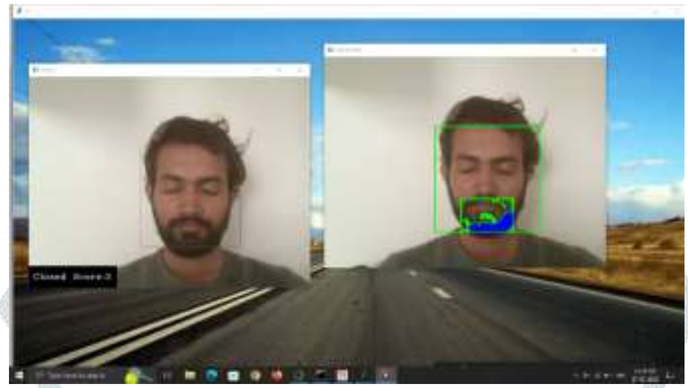

Drowsiness and Autonomous Driving detection
Login
Register
Exit
CZ20 FSE

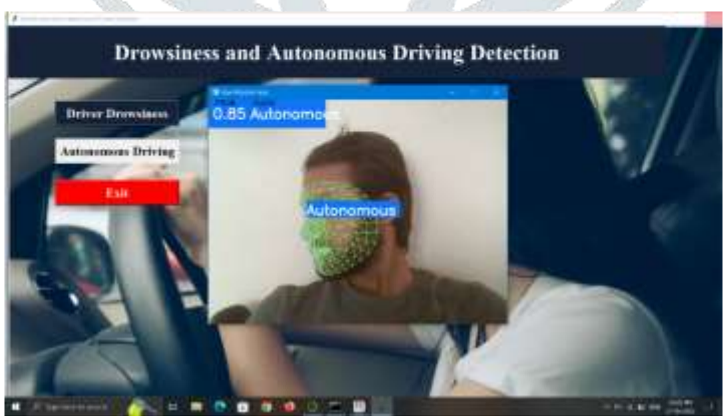

Registration Form
Login
Register
Exit
Male
Female
Register


LOGIN HERE
Login
Register
Exit
Username
Password


Drowsiness and Autonomous Driving Detection
Driver Drowsiness
Autonomous Driving
Exit







7. CONCLUSION

The increasing number of traffic accidents due to a diminished driver's vigilance level has become a serious problem for society. Statistics show that 20 per cent of all traffic accidents are due to drivers with a diminished vigilance level. Furthermore, accidents related to driver hypo-vigilance are more serious than other types of accidents, since sleepy drivers often do not take corrective action before a collision. For this reason, developing systems for monitoring the driver's level of vigilance and alerting the driver, when he is drowsy and not paying adequate attention to the road, is essential to prevent accidents.

It has been developed by integrating hardware and software. The microcontroller working with computer vision helps to be trendy and improve the quality of service along with the module. The presence of every module is reasoned out and placed thus contributing to the best working of the unit. The service allows protecting the drivers and kernel to message the reliable person and protect the driver from accidents. The model also detects the collision of the vehicle.

REFERENCES

[1] Vidhu Valsan A, Paul P Mathai and Ierin Babu, "**Monitoring Driver's Drowsiness Status at Night Based on Computer Vision**," 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS) 978-1-7281-8529-3/20/$31.00 ©2021 IEEE

[2] Ratna Kaavya M, Ramya V and Ramya G Franklin, "**Alert System For Driver's Drowsiness Using Image Processing**," 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)

[3] Mika Sunagawa, Shin-ichi Shikii, Wataru Nakai, Makoto Mochizuki, Koichi Kusukame, and Hiroki Kitajima, "**Comprehensive Drowsiness Level Detection Model Combining Multi-modal Information**," in Proc. Int. Conf. Fatigue Manage. DOI 10.1109/JSEN.2019.2960158, IEEE

[4] Yashika Katyal, Suhas Alur and Shipra Dwivedi, "**Safe Driving By Detecting Lane Discipline and Driver Drowsiness**," 2017 IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)

[5] Kyong Hee Lee, Whui Kim, Hyun Kyun Choi and Byung Tae Jang, "**A Study on Feature Extraction Methods Used to Estimate a Driver's Level of Drowsiness**," ISBN 979-11-88428-02-1 ICACT2019 February 17 ~ 20, 2019

[6] A. Sahayadhas, K. Sundaraj, and M. Murugappan, "Detecting driver drowsiness based on sensors: A review," Sensors, vol. 12, no. 12, pp. 6937–16953, Dec. 2012.

[7] Y. Dong, Z. Hu, K. Uchimura, and N. Murayama, "Driver inattention monitoring system for intelligent vehicles: A review," IEEE Trans. Transp. Syst., vol. 12, no. 2, pp. 596–614, Jun. 2011

[8] C. Bila, F. Sivrikaya, M. A. Khan, and S. Albayrak, "Vehicles of the future:
A survey of research on safety issues," IEEE Trans. Intell. Transp. Syst., vol.18, no. 5, pp. 1046–1065, 2017.