ABSTRACT

Sleep disorders and various common acute directly or indirectly affect the quality and quantity of one‘s sleep or otherwise cause excessive daytime fatigue. About 29600 Norwegian accident-involved drivers received a questionnaire about the last accident reported to their insurance company. About 9200 drivers (31%) returned the questionnaire. The questionnaire contained questions about sleep or fatigue as contributing factors to the accident. In addition, the drivers reported whether or not they had fallen asleep some time whilst driving, and what the consequences had been. Sleep or drowsiness was a contributing factor in 3.9% of all accidents, as reported by drivers who were at fault for the accident. This factor was strongly over-represented in night-time accidents (18.6%), in running-off-the-road accidents (8.3%), accidents after driving more than 150 km on one trip (8.1%), and personal injury accidents (7.3%).Drivers‘ lack of awareness of important precursors of falling asleep—like highway hypnosis, driving without awareness, and similar phenomena—as well as a reluctance to discontinue driving despite feeling tired, are pointed out as likely contributors to sleep-related accidents.

CHAPTER 1

1 INTRODUCTION

The majority of traffic accidents are caused by Driver drowsiness and tiredness and impaired driving, as well as working settings, lack of sleep, and time constraints. These two situations have an impact on the vehicle's capacity to be controlled. To detect tiredness in drivers, some techniques are utilized , such as measuring driver operation or physiological aspects of the driver, such as vehicle movement.Everyone , whether a passenger, driver, or pedestrian, would have noticed several posters along the roadway that serve important roles. This essential road equipment serves as route directions, alerts, and traffic controllers. Indicators require complete attention, respect, and an acceptable motorist response as traffic control devices. We came across some road signs that were from a long time ago. The original road markings were milestones, which marked distance or direction. In the Middle Ages, multidirectional signage at intersections became common, providing directions to cities and towns. With the increase in motorized traffic and the associated increased demand on the road, some countries have adopted pictorial signs and standardised their signs to make international travel easier, particularly in areas where language barriers exist. It's used to increase road safety by displaying appropriate caution, regulatory, and informational signs.The majority of them use symbols instead of words and are widely recognized and accepted around the world. Most states have adopted these signs, which were developed primarily in Europe. In India, the Motor Vehicle Act of 1988 established the standard Road Signs in Schedule I, which describes the shape and sizes of these road signs in detail.

1.1 OVER VIEW OF THE PROJECT

Categorizes Of Conventional Signs

The Convention on Road Signs and Signals held on 8th November 1968 lays down the classes of Road Signs, which were broadly categorized into:

a) Mandatory signs

b) Cautionary signs

c) Informatory signs

The colour of a sign might also help you figure out what it's for. Mandatory instructions are given in blue circles, such as "Compulsory Turn Left" and so on. Information signs are made up of blue rectangles. Red is the colour of all triangle signs. There are few exceptions to the shape and color rules, to give certain sign greater prominence.

Mandatory Signs

For traffic using a specific part of the route, these signs are required. These indicators indicated what should be done rather than what should be avoided. Mandatory road signs are frequently circular and have a red border. They have a bluish hue to some of them. The letters "STOP" and "GIVE WAY" are respectively octagonal and triangular. Violations of these guidelines are punishable by hefty fines and penalties. Breaking these guidelines, moreover, could result in serious accidents.

1.2.2 Cautionary Signs

These signs are intended to alert drivers of potential hazards or situations on the road ahead. For his own safety, the motorist should follow these instructions. Despite the fact that disobeying these road signs does not result in legal action, they are extremely significant because ignoring them could result in serious accidents. Cautionary signs have a red border and are triangular in design.

1.2.3 Informatory signs

These signs are intended to offer road users with information such as directions, destinations, and roadside amenities. Following informative road signs allows a driver to save time and arrive at their destination without getting lost. These signs are usually blue in colour and serve as aids to the driver. A direction arrow and a distance facility from the sign may be included on the sign.

1.2 NEED FOR THE PROJECT

The recent incident we discussed earlier brings to light the fact that distracted driving is a major cause of deadly road accidents. This problem is further compounded in developing countries like Bangladesh, where population growth is outpacing motorization and drivers are inefficient. Even the number of driving hours is so great that it is impossible to drive safely in that amount of time. While the driver's constant attention to respond to rapidly changing events on the road is essential for safety, slight diversion always leads to intractable problems. Drowsiness accelerates the driver's distraction caused by sleep deprivation or continuous driving for long periods of time.Distracted driving can occur for a variety of reasons, but the variables we've just covered are the most common. If we want to see a future with fewer road accidents, we must address the issue of distracted driving. Detecting tiredness or drunkenness in a driver while driving might notify alert people to carry over. While driving, a distracted driver must be brought under supervision. The best method to get rid of distractions is to identify them first.

1.3 OBJECTIVE OF THE PROJECT

Apart from being tired behind the vehicle, being drunk behind wheel is also dangerous. The rise in the number of drunk drivers is also generating tragic incidents. According to the same source, there were 10497 fatal incidents involving drunk drivers with BAC levels of 0.8g/dL (800ppm) or higher. The accusation of intoxicated driving is equally widespread in Bangladesh, despite the fact that no statistical survey has been conducted to date. Distraction when driving due to tiredness, falling asleep, or alcohol intoxication, whether in Bangladesh or abroad, creates an unfillable hole for which nothing can compensate. Result of abnormal and then forgetting about it will simply increase the number.

1.4 SCOPE OF THE PROJECT

There is no mechanism in place in the current driving system to identify and monitor the driver's physical status while driving. Detecting a driver's physical state while driving is difficult in our current system unless he is made visible to some agents at all times. Only driving the vehicle is supported by our traditional driving system.No additional technology is included to detect other issues, no matter how important it is in detecting bad driving. What they can hardly do is place a CCTV camera in the vehicle, which will record everything that happens with the driver and be checked later. However, if an accident occurs, CCTV will be of little value; it will not be able to prevent the accident. Several methods to provide safety while driving like collision avoidance using IR sensors, fuel detection systems, lane change assistance, and adaptive light control system. The primary downside is that the IR sensors used to determine the presence of other vehicles, pedestrians, and some other objects to avoid crashes and accidents. In case of reliability ultrasonic sensors are better than IR sensors and the maximum range of an ultrasonic sensor is about 20 meters while for the IR sensor it is only between 1-5 meters and also depends on the type of IR sensor which is used the operating range will also vary. Other method speed limit recognition using Beacon technology. This beacon work along with Bluetooth enabled in mobile phones to provide notification of speed limits whenever the vehicle comes into the region where beacon tags are placed. The only disadvantage is that the Beacons can be used along with smart phones with Bluetooth enabled if there is no connection or due to poor connection with mobile phones there may be a lack in the transformation of information to the drivers. And the other disadvantage is the cost, it is not cost-effective and battery friendly. The system should be able to deal with traffic and road signs in a wide range of weather and illumination variant environments such as different seasons, different weather condition e.g. sunny, foggy, rainy and snowy conditions.

Different potential difficulties are depicted in one section of this chapter. Using the system in different countries can make the problem even worse. Different countries use different colours and different pictograms. The system should also be adaptive, which means it should allow continuous learning otherwise the training should be repeated for every country. To deal with all these constraints, road sign recognition should be provided with a large number of sign examples to allow the system to respond correctly when a traffic sign is encountered.

CHAPTER 2

2.1 LITERATURE SURVEY

Dilipkumar Borikaar's[1] ., goal is to lessen the number of people killed in traffic accidents. It use smartphone sensors such as the accelerometer to determine whether the emergency situation is an accident. If the system determines that the situation constitutes an accident, it notifies the appropriate service providers, which may include public administration such as a police station and healthcare services such as hospitals and ambulances. This system satisfies the needs of accident victims in locating and alerting the appropriate authorities, as well as the needs of service organisations in tracking the emergency situation and mobilising aid. The technique helps victims of road traffic accidents have a better chance of surviving.

Sheela.s et al [2]., have successfully identified the Tiredness System, which is based on the driver's eye closure and can distinguish between normal eye twitch and drowsiness, as well as identify drowsiness while driving. The proposed approach will aid in the prevention of drowsy driving-related injuries. OpenCV was used to detect and recognize and eyes using a Haar cascade classifier, and then a CNN model was used to forecast the state. When the eyes are closed for a lengthy period of time, a warning signal is given. Continuous eye closures are used to determine the driver's level of awareness . This detecting system can be turned into hardware with advanced functionality for future work.

M.Chikezic et al [4]., have successfully built a sleepiness detection system that analyses the driver's state in real-time and sends a warning to a messaging app as an alarm and a remote notification. Because existing procedures based on psychological and vehicle-based methods are intrusive and unreliable, a behavioral-based approach was chosen. A hardware prototype was created to check the driver's tiredness and transmit an IoT notification. The Raspberry Pi3 model B module, a USB webcam camera used for real-time data capture, and a stream of video on which face detection is performed using OpenCV are the system's key components. Dlib's pre-trained facial landmark detector was utilised to localise face landmarks, and the localised facial landmark points were then used to compute EAR. If the estimated EAR value exceeds the preset threshold, the eyes remain still open and there is no change in the system's state. Similarly, if the EAR value falls below a certain threshold, the system sends a voice message to the driver, as well as a warning message featuring an image of the asleep, to a telegram account driver who functions as the organization's admin/regulatory body. The technology was tested and found to be 90 percent accurate.

Sasikumar Periyasamy et al [4]., to create a real-time sleepiness detection system that can identify a driver's fatigue and propose nearby stop spots if the person feels weak. The majority of existing sleepiness detection systems employ heart rate as a detecting metric. The driving heart rate rises

85.56 beats per minute during the day to 89.85.6 beats per minute at night. When the driver is detected to be tired, however, the heart rate drops to 81.5 9.2 beats per minute. These heart rate variations may not be as accurate in detecting tiredness while driving. The proposed detection method, on the other hand, uses the EAR value to determine the level of drowsiness. The facial points of the face are constantly checked and the EAR values are updated using webcams and external cameras. An alarm is sounded and a navigation option to a nearby lodging establishment is presented if the EAR value falls below a specified threshold (0.25) for a particular number of frames (48 frames). If the driver is detected as being drowsy too frequently, an email is sent to the manager/owner. Long-distance drivers, particularly truck/cab drivers, may find this approach useful. Features such as dynamic mapping and voice integration can also be incorporated to improve the system's usability and interactivity.

Kirti Mahajan et al [5]., have identified the high number of HOS regulation infractions and holding a licence without a required educational level demonstrates the authorities' lack of zeal in enforcing the laws. Drivers are discovered to be involved in occurrences of falling asleep at the wheel as a result of infractions of HOS guidelines. The study's findings highlight the detrimental effects of payment incentives on driver safety and tiredness management. This research reveals that the remuneration structure for freight employers and enterprises is primarily based on production (22 percent receive pay with a task or trip completion and majority is interested in earning incentives). According to the majority of long-haul truck drivers, non-driving jobs such as loading/unloading or waiting in lines are not counted for wages.

K. Fagerberg et al [6]., developed a method on Driver Drowsiness and Alcohol Intoxication Detection. They purpose to develop a drowsiness detection system. In this work, images are processed using image processing techniques for identifying driver's current state. Driver's drowsiness is analyzed by his/her facial expression and head movement. This system manages utilizing data gained for the image which is in binary form to locate the face. Detection of alcohol consumption is done with the help of sensors. The number of road accidents might then be avoided if an alert is sent to a driver that is deemed drowsy. The drowsiness measure based on camera give an appreciated contribution.

C.Shembekar [16]., have successfully identified a growing number of car crashes caused by a reduced driver's awareness level has turned into a major societal problem. According to statistics, 20 percent of all car collisions occur as a direct result of drivers who are not paying attention. Furthermore, calamities linked to driver hypo-vigilance are more reliable than other types of accidents, because drowsy drivers frequently fail to perform the proper manoeuvres before to a collision. Currently, frameworks for monitoring a driver's level of alertness and prompting the motorist when he is tired and not presenting a suitable impression of the road are necessary to maintain a vital path away from disasters. It also makes use of alcohol and beat disclosure to examine the individual. Faces, as the most fundamental form of human communication, have long been a criterion in computer vision. Face area, outward appearance data extraction, and demeanour depiction are the three degrees of assignments in modified attestation (or assessment) of extraction is the main concern in these assignments for segment based outward appearance confirmation from a picture movement. Under various enlightenments, face headings, and outward looks, it connects territory, ID, and following facial part communities. SVM Classifier is currently being used to investigate the fatigue problem and obtain diverse results.

Rateb Jabbar et al [10]., have successfully enhance the Drowsiness detection system based on CNN-based Machine Learning . The major goal is to create a lightweight system that can be implemented in embedded systems while yet obtaining good performance. The system was able to recognise face landmarks in photographs collected on a mobile device and feed them to a CNN-based trained Deep Learning model to detect drowsydriving behaviour. The accomplishment in this case was the creation of a deep learning model that is tiny in size but has a high level of accuracy. For all categories where the maximum size of the model did not exceed 75KB, the model described here achieved an average of 83.33 percent accuracy.

M.S Satyanarayana, et al [15]., The goal of Driver Drowsiness Detection was to enable a driver to remain alert while driving, with the ultimate goal of reducing vehicle accidents caused by sluggishness. When the driver's fatigue level reaches one hundred, an uproarious equipped for being heard alerted will startle him and transmit a warning to the expert community. The loud equipment for being heard advised to wake the driver up before he or she falls asleep behind the wheel. Which boosts shopper trust and profitability for companies like Uber, Ola taxis, Rivigo, Meru cab, and others.This device should be made mandatory in every car, and it should not be removed by the driver or mechanic, as it will assist to save thousands of lives. Also, the system will aid in the understanding of driver behaviour, which will aid in the selection of drivers depending on task.

C.Jacobe de Naurois, et al [11]., have identified the Different ANNs were utilised in this work to either detect drowsiness or forecast when a driver's state would become impaired. Eyelid closure, gaze and head movements, and driving duration were all utilised in the top models (those with the highest rates of successful detection or prediction). The model's performance in terms of prediction is quite promising, since it can anticipate when the driver's status will deteriorate to within 5 minutes. Furthermore, characterising tiredness as a continuum can lead to more precise detection systems that go beyond just identifying whether a motorist is awake or asleep. Adding temporality to the model with recurrent neural networks or dynamic neural networks, or adding other features like context information (traffic, type of road, weather etc.) could improve performance in the future.These things can have an impact on the driver's mood. Due to the difficulty of recording eyelid and head movements in a real car, the focus should be on improving a model utilising just driving performance, driving behaviour (based on data provided by car sensors), and physiological measurements. Finally, to validate these models, a bigger and more realistic dataset (many more participants with a wider variety of ages, for example) recorded in real-world, on-road settings (at different times of the day, for example)would be required.

H.Iwamoto, et al [9]., have successfully complete the new drowsiness detection approach based on LSTM-AE was proposed in this work. The proposed method attained an AUC of 0.88, a sensitivity of 81 percent, and a specificity of 91 percent using the driving simulator, which were all greater than methods that used HRV features. This suggests that using raw RRI data rather than HRV processing for detecting driver drowsiness may be more appropriate.The experimental environment was a driving simulator, thus there are certain limitations to this study. Because the participants were not at risk of a traffic accident, they were unlikely to take tiredness seriously while driving. Furthermore, the participants were young because they were chosen from Kyoto University students. Because HRV is age-dependent (Shaffer and Ginsberg, 2017), elderly individuals will be needed for validation. More experimental data will be collected in the future to improve sleepiness detection performance, and the system in development will be evaluated in a real-world driving scenario.

A.K.Biswal et al [3].,This paper study and presents a reliable method for detecting driver tiredness and a collision impact (severity) system in the current era. In most cases, this strategy integrates two separate systems into a single integrated system. However, existing systems rely on psychological or vehicle-based approaches to detect driver drowsiness, and the severity of the collision is also recorded independently, but such a technique is highly intrusive and completely turns on the physical environment. As a result, the suggested method is used to develop a nonintrusive technique for comparing the degree of a driver's tiredness to the severity of an accident caused by braking or a mishap.The Raspberry Pi3 model B module and the Pi camera module are the essential components of this system, and they are utilised for persistent recording of face landmarks that are localised using facial landmark points, and then to calculate EAR. If the computed EAR value exceeds the threshold range, the eyes remain open and the condition of the system remains unchanged. Similarly, if the EAR value falls outside of the threshold range, the system sends an urgent alarm to the authority (owner) via speech speaker and warning e-mail for further driver support. Furthermore, the intensity of the collision (impact) is measured using sensors and a GPS module to accurately trace the position of the accident and alert the nearest medical treatment centre to serve emergency diagnosis.

CHAPTER 3

WORKING MODULE

3.1 DROWSENESS ARCHITECTURE DIAGRAM

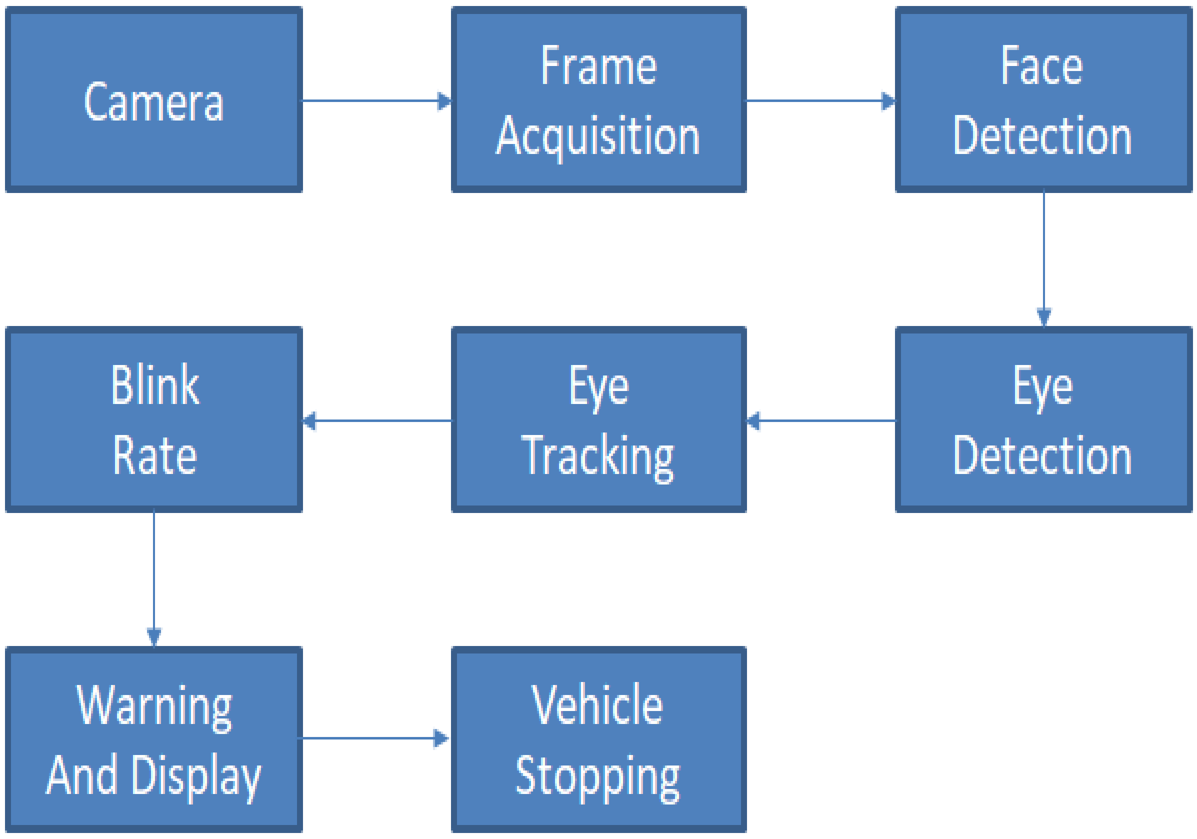
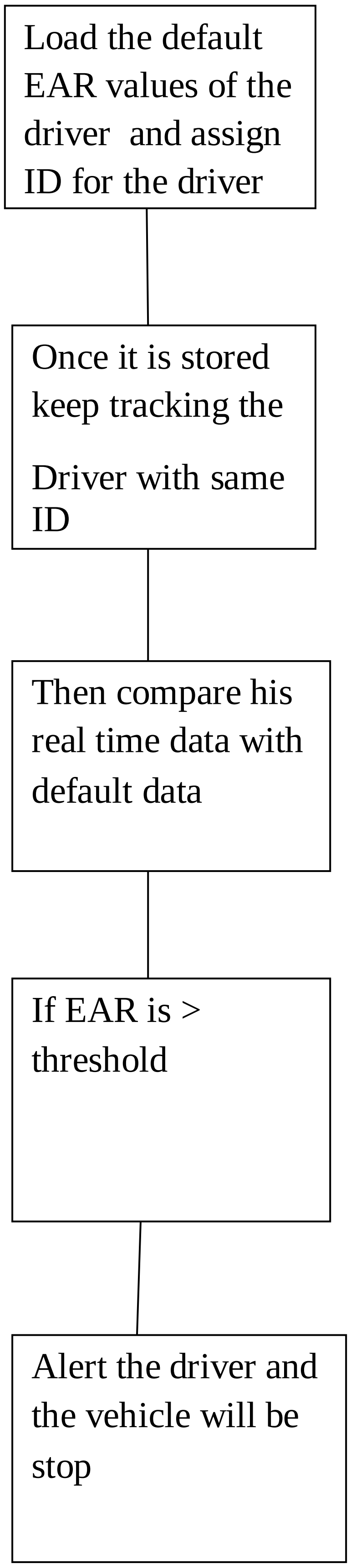


Figure 3.1Drowsiness architecture diagram

Fig 3.1 Depicts the working functionalities of drowsiness detection

3.2 EAR COMPARISON FLOW

Figure 3.2 EAR Comparison Flow


Load the default EAR values of the driver and assign ID for the driver
Once it is stored keep tracking the Driver with same ID
Then compare his real time data with default data
If EAR is > threshold
Alert the driver and the vehicle will be stop

As mentioned in the above algorithm the EAR comparison is very important as it needs to be done at every stage and same to be used for prediction.Once the EAR comparison is done, the driver data will be stored Continously in the database. This will be used for the data analytics where same will be used in order to check driver behaviour based on different parameters.

- How Many hours he is driving continuously
- At which thresh hold he is feeling drowsy
- How much time he is required further to drive
- Health Condition
- How Fast he is driving
- Resting Time

All these parameters will be keeping on tracked to check the behaviour of the driver which will really helps the system to understand and make sure that he will be alerted at right time to avoid accidents etc.

This system can be implemented effectively to understand the clear behaviour of drivers which will helps vehicle owners to deeply think while sending particular driver to an exclusive task .So, if this system gets implemented surely will give value proposition to vehicle owners especially those who are running fleets and network of busses.

3.3 METHODOLOGY

The main aim is to detect drowsiness of driver; it can be done in different ways like detecting facial expression of the driver and measuring Eye Aspect Ratio (EAR). Blinking pattern is different for each and every individual. The pattern gets varied in terms of squeezing degree of eye, blink duration and speed of closing and opening the eye. The proposed method involved with the following methodologies such as Haar Cascade Classifiers, Shape Predictor_68_facial landmark detection, Eye Aspect Ratio (EAR).

3.3.1 Haar Cascade Classifiers

In Haar Cascade Classifiers, a lot of similar and dissimilar images are trained in order to detect fatigue of the driver. OpenCV is a learning-based method, packed with a detector as well as a trainer. For training, a separate database is maintained for face and eye with several positive and negative images having eye closed and opened conditions and different set facial images.

3.3.2 Shape predictor

In order to predict the face and eye region in the live video stream, shape predictor is used. It shows the sleepiness which is measured by calculating the eye aspect ratio (Euclidean distance between the eyes are calculated), the arguments are passed to the predefined dataset and facial landmark detection is carried out. For every video sequence, the eye landmarks are located. The aspect ratio between width and height of the eye is calibrated. The EAR is mostly stable when an eye is open and is getting close to zero while the eye is not in open state. If the person Viewing the camera continuously, the Eye Aspect Ratio (EAR) is found to be normal and it reaches low value when he/she closing the eye for a longer time. When the lower value is reached, then drowsiness is detected.

$$EAR = \frac{\| p2 - p6 \| + \| p3 - p5 \|}{2 \| p1 - p4 \|}$$

Where p1,…,p6 are the two-dimensional landmark location, The EAR is mostly stable when an eye is open and is getting close to zero while the eye is not in open state. If the person viewing the camera continuously, the Eye Aspect Ratio (EAR) is found to be normal and it reaches low value when he/she closing the eye for a longer time. When the lower value is reached, then drowsiness is detected.

CHAPTER 4

REQUIREMENT SPECIFICATION

4.1 HARDWARE REQUIREMENT

1) ATMEGA328MICROCONTROLLER
2) LIQUID CRYSTAL DISPLAY
3) MOTOR DRIVERL298N
4) BUZZER
5) 12V/1A TRANSFORMER
6) 12V DC GEAR MOTOR
7) DATATRANSFER CABLE

MOTOR

Figure 4.1.1 Motor

For the machine-driven action like gyration, motors are used.Here we have cast-off 12-volt DC motor. The voltage variation gives the change in requisite speed.

LCD DISPLAY

The liquid crystal displays cast-off to display the outcomes of the sensors interfaced to upsurge the safety while driving. It is better than LED and other display devices in several ways. LEDs have a variable set of use cases for customers, they'll be usually found in smart phones, televisions, pc monitors and instrument panels. It also depletes much less power than LED and gas display.

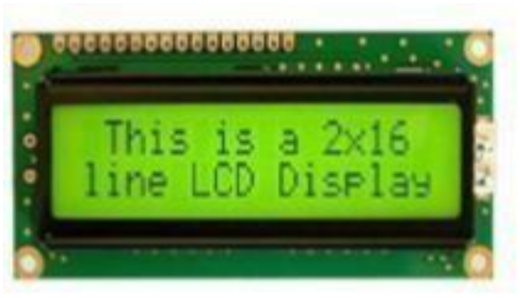


Figure 4.1.2 LCD DISPLAY

ATMEGA328 MICROCONTROLLER

The ATmega328 is one kind of single-chip microcontroller formed with Atmel within the megaAVR family. The architecture of this Arduino Uno is a customized Harvard architecture with 8 bit RISC processor core. Other boards of Arduino Uno include Arduino Pro Mini, Arduino Nano, Arduino Due, Arduino Mega, and Arduino Leonardo. Arduino Uno ATmega328

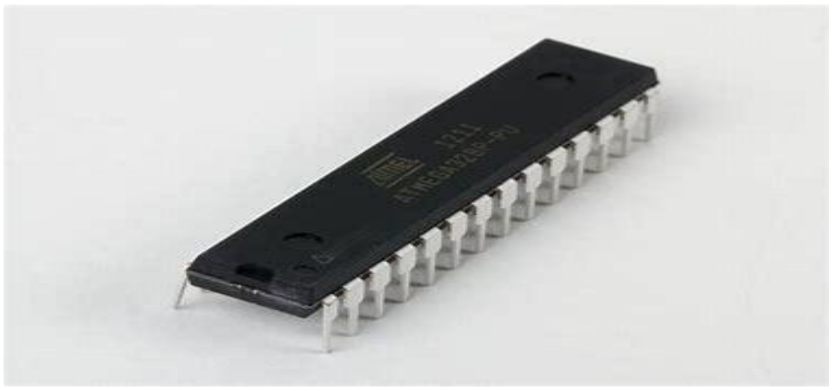


Figure 4.1.3 ATMEGA328 MICROCONTROLLER

MOTOR DRIVER L298N

L298N module is a high voltage, high current dual full-bridge motor driver module for controlling DC motor and stepper motor. It can control both the speed and rotation direction of two DC motors. This module consists of an L298 dual-channel H-Bridge motor driver IC.

Figure 4.1.4 MOTOR DRIVER L298N

ARDUINO UNO

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino. The board is equipped with sets of digital and analog input/output(I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. It is similar to the Arduino Nano and Leonardo. The hardware reference design is distributed under a Creative Commons Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available.

The word "uno" means "one" in Italian and was chosen to mark the initial release of Arduino Software. The Uno board is the first in a series of USB- based Arduino boards; it and version 1.0 of the Arduino IDE were the reference versions of Arduino, which have now evolved to newer releases. The ATmega328 on the board comes preprogrammed with a boot loader that allows uploading new code to it without the use of an external hardware programmer.

While the Uno communicates using the original STK500 protocol, it differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it uses the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

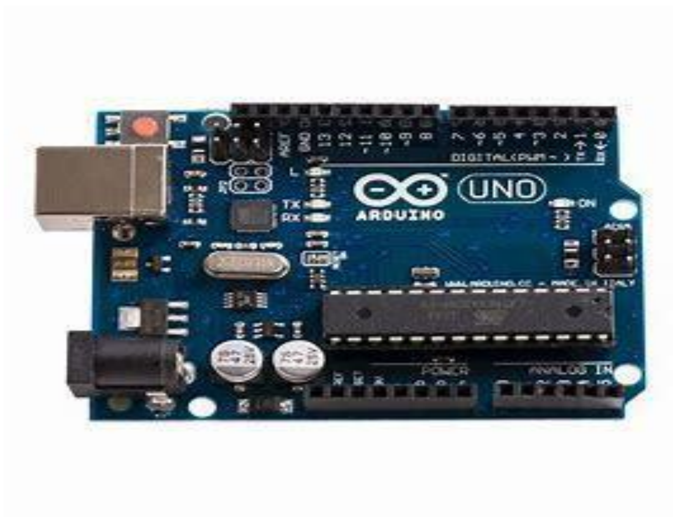


Figure 4.1.5 ARDUINO UNO

Arduino uno Technical Specifications

Table 4.1 Arduino Specification

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6- 20V

Table 4.2 REQUIREMENTS

Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

4.2 SOFTWARE REQUIREMENT

1) OPENCV
2) PYTHON LANGUAGE
3) Arduino IDE

Python

Python is a general-purpose programming language created by Guido van Rossum that has quickly gained popularity due to its simplicity and readability of code. It allows the programmer to communicate his thoughts in fewer lines of code while maintaining readability. Python is slower than other languages like C/C++.

But another important feature of Python is that it can be easily extended with C/C++. This feature helps the user to write computationally intensive codes in C/C++ and create a Python wrapper for it so that it can be used as Python modules. This gives two advantages: First, the code is as fast as original C/C++ code - since it is the actual C++ code working in background and Second, it is very easy to code in Python. This is how OpenCV-Python work; it is a Python wrapper around original C++ implementation. The support ofNumpy makes the task easier. Numpy is a highly optimized library for numerical operations. It gives a MATLAB-style syntax. All the OpenCV array structures are converted to-and-from Numpy arrays. Operations available in Numpy, can be combined with OpenCV. Besides that, several other libraries like SciPy, Matplotlib which supports Numpy can be used with this. Hence Python-OpenCV binding is an appropriate tool for fast prototyping of computer vision problems**.**

OpenCV

OpenCV (Open Source Computer Vision Library) is an Application Programming Interface (API) developed by Intel which can be used for many image processing and computer vision applications. OpenCVofficially launched in 1999 and the project was initially an Intel Research initiative to advance CPU-intensive applications. OpenCV library is a collection of algorithms and C/C++ functions and a fewclasses that implement some Image processing and computer vision algorithms. There is active development on interfaces for C, C++, Python, Ruby, Matlab and other languages. OpenCV was designed for computational efficiency and with a strong focus on real time applications. OpenCV is written in optimised C and can take advantage of multicore processors. OpenCV contains over 500 function that spanmany areas in vision, including factory product inspection, medical imaging, security, user interface, camera calibration, stereo vision androbotics. The principles behind the creation of the library is to aidcommercial uses of computer vision in human computer interface,robotics, monitoring, biometrics and security by providing a free and open infrastructure Where the distributed efforts of the vision community can be consolidated and performance optimized. OpenCV support for vision is extensively including routine support for input, display, and storage of movies and single images One of the OpenCV goals is to provide a simple-to-use computer vision infrastructure that helps people build fairly sophisticated vision applications quickly. There areseveral goals of OpenCV in outset which are following:

By giving not only open but also optimised code for basic vision infrastructure, advanced vision research can be advanced. Disseminate vision information by providing a standard architecture for developers to build on, resulting in more legible and transferrablecode.

Commercial applications based on advanced vision were made free by making portable, performance-optimized code available under a licence that did not require commercial programmes to be open or free. OpenCV is a collection of image processing and computer visionapplications.

The functions are optimized for Intel architecture processors and are particularly effective with MMX technology. The OpenCV Library is a way of establishing an open source vision community thatwill make better use of up-to-date opportunities to apply computer vision in growing PC environment and mobile platform. The library is open and has platform independent interface and supplied with whole C sources. OpenCV was designed to be portable. It was originally written to compile across Borland C++, Microsoft Visual Studio C++,and the Intel compilers. C and C++ code had to be fairly standard in order to make cross-platform support easier. OpenCV library is multi platform, and runs on both Windows and Linux Operating System.

OpenCV is quickly gaining popularity for developing real-time computer vision applications. Some examples of applications include face recognizers, object recognizers, and motion trackers, just to name a few. The library has especially gained popularity in the computer vision research community. It allows researchers to get demos or research projects up and running quickly, and take advantage of the large collection of algorithms that are already available. The use of term of computer vision and image processing is commonly interleaved. In contrary, there is a gap between computer vision and image processing. Image processing is of low- level processing of still or video image, while computer vision is high-level processing of still or video image. OpenCV is specifically designed to an advent to computer vision development. OpenCV aimed at providing the basic tools needed to solve computer vision problem. MLL is the machine learning library, which includes many statistical classifiers and clustering tools.

High GUI contains I/O routines and functions for storing and loading video and images and CXCORE contains the basic data structures and content. The above figure does not include CVAUX, which contains both defunct areas (embedded HMM face recognition) and experimental algorithms

4.2.1 FEATURES OF Embedded C

Embedded C is one of the most popular and most commonly used Programming Languages in the development of Embedded Systems. Embedded C is perhaps the most popular languages among Embedded Programmers for programming Embedded Systems. There are many popular programming languages like Assembly, BASIC, C++, Python etc. that are often used for developing Embedded Systems but Embedded C remains popular due to its efficiency, less development time and portability.

Before digging in to the basics of Embedded C Program, we will first take a look at what an Embedded System is and the importance of Programming Language in Embedded Systems.

Embedded Systems consists of both Hardware and Software. If we consider a simple Embedded System, the main Hardware Module is the Processor. The Processor is the heart of the Embedded System and it can be anything like a Microprocessor, Microcontroller, DSP, CPLD (Complex Programmable Logic Device) or an FPGA (Field Programmable Gated Array).

All these devices have one thing in common: they are programmable i.e., we can write a program (which is the software part of the Embedded System) to define how the device actually works.

Embedded Software or Program allow Hardware to monitor external events (Inputs / Sensors) and control external devices (Outputs) accordingly. During this process, the program for an Embedded System may have to directly manipulate the internal architecture of the Embedded Hardware (usually the processor) such as Timers, Serial Communications Interface, Interrupt Handling, and I/O Ports etc.

From the above statement, it is clear that the Software part of an Embedded System is equally important as the Hardware part. There is no point in having advanced Hardware Components with poorly written programs (Software).

There are many programming languages that are used for Embedded Systems like Assembly (low-level Programming Language), C, C++, JAVA (high-level programming languages), Visual Basic, JAVA Script (Application level Programming Languages), etc.

In the process of making a better embedded system, the programming of the system plays a vital role and hence, the selection of the Programming Language is very important.

In every embedded system based projects, Embedded C programming plays a key role to make the microcontroller run & perform the preferred actions. At present, we normally utilize several electronic devices like mobile phones, washing machines, security systems, refrigerators, digital cameras, etc. The controlling of these embedded devices can be done with the help of an embedded Cprogram.

CHAPTER 5

IMPLEMENTATION

5.1 SAMPLE CODE

```
#include<LiquidCrystal.>

  LiquidCrystal  lcd(13,  12,
  11, 10, 9, 8);
#define python Serial
#define splash splash1
#define buz 6
String IncomingData ="";
  float ti;
  void setup()
  {
  python.begin(115200);
  LcDSet();
  splash(0,"Initializing");
  splash(1, "Python");
  pinMode(buz,OUTPUT);
  digitalWrite(buz,LOW);
  delay(3000);
   lcd.clear();
  python.println("Ready");
}
void LcDSet() {
 lcd.begin(16, 2);
splash(0, "Drowsiness");
splash(1, "TX");
delay(3000); lcd.clear();
  }
void loop()
  {
    //  ti += 0.5;
 while (python.available())
    {
      IncomingData=python.readString();
 delayMicroseconds(5);
    }
  If
  (IncomingData.length()>0) {
if (IncomingData == "run") {
     splash(1, "Run");
python.println("Running");}
 if (IncomingData == "alert") {
   splash(1, "alert");
python.println("alert");
      }
IncomingData = "";
    }
   delay(200);
  }
  // SPLASH
void splash1( int row, String txt)
```

```
{
  int curs = (17 - txt.length()) / 2;
  lcd.setCursor(0, row);
  lcd.print("----------------");

  lcd.setCursor(curs, row);
  lcd.print(txt);
  delay(300);
}
```

5.1.1 PYTHON PROGRAM

```
# import the necessary packages
from imutils.video import VideoStream
from imutils import face_utils
import numpy as np
import argparse
import imutils
import time
import dlib
import cv2
import serial
import winsound

arduinoData = serial.Serial('COM4', 115200)

# define two constants, one for the eye aspect ratio to indicate
# blink and then a second constant for the number of consecutive
# frames the eye must be below the threshold for to set off the
EYE_AR_THRESH = 0.38
EYE_AR_CONSEC_FRAMES =4
DaTa_r=""
def getSeriAl():
    global DaTa_r
    if(arduinoData.in_waiting >0):
        line = arduinoData.readline()
        readdat=line.decode()
        print(readdat)
        if(readdat == "Ready\r\n"):
            print("device online")
            DaTa_r="run"
def euclidean_dist(ptA, ptB):

        # compute and return the euclidean distance between the two

        # points

        return np.linalg.norm(ptA - ptB)

def eye_aspect_ratio(eye):
        # compute the euclidean distances between the two sets of
        # vertical eye landmarks (x, y)-coordinates
        A = euclidean_dist(eye[1], eye[5])
        B = euclidean_dist(eye[2], eye[4])
        # compute the euclidean distance between the horizontal
        # eye landmark (x, y)-coordinates
        C = euclidean_dist(eye[0], eye[3])
        # compute the eye aspect ratio
        ear = (A + B) / (2.0 * C)
        # return the eye aspect ratio
        return ear
# initialize the frame counter
COUNTER = 0
# load OpenCV's Haar cascade for face detection (which is faster than
# dlib's built-in HOG detector, but less accurate), then create the
# facial landmark predictor
print("[INFO] loading facial landmark predictor...")
detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
predictor = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")
```

```
# grab the indexes of the facial landmarks for the left and
# right eye, respectively
(lStart, lEnd) = face_utils.FACIAL_LANDMARKS_IDXS["left_eye"]
(rStart, rEnd) = face_utils.FACIAL_LANDMARKS_IDXS["right_eye"]
# start the video stream thread
print("[INFO] starting video stream thread...")
vs = VideoStream(src=0).start()
# vs = VideoStream(usePiCamera=True).start()
print("[INFO] Please Wait for hardware booting...")
time.sleep(1.0)
found=set()
# loop over frames from the video stream
while True:
	# grab the frame from the threaded video file stream, resize
	# it, and convert it to grayscale
	# channels)
	getSeriAl()
	frame = vs.read()
	frame = imutils.resize(frame, width=450)
	gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

```
# detect faces in the grayscale frame
  rects = detector.detectMultiScale(gray, scaleFactor=1.1,
          minNeighbors=5, minSize=(30, 30),
          flags=cv2.CASCADE_SCALE_IMAGE)
# loop over the face detections
  for (x, y, w, h) in rects:
          # construct a dlib rectangle object from the Haar cascade
          # bounding box
          rect = dlib.rectangle(int(x), int(y), int(x + w),
                  int(y + h))
         # determine the facial landmarks for the face region, then
          # convert the facial landmark (x, y)-coordinates to a NumPy
          # array
          shape = predictor(gray, rect)
          shape = face_utils.shape_to_np(shape)
        # extract the left and right eye coordinates, then use the
          # coordinates to compute the eye aspect ratio for both eyes
          leftEye = shape[lStart:lEnd]
          rightEye = shape[rStart:rEnd]
          leftEAR = eye_aspect_ratio(leftEye)
```

```
      rightEAR = eye_aspect_ratio(rightEye)
     # average the eye aspect ratio together for both eyes
      ear = (leftEAR + rightEAR) / 2.0
# compute the convex hull for the left and right eye, then
      # visualize each of the eyes
      leftEyeHull = cv2.convexHull(leftEye)
      rightEyeHull = cv2.convexHull(rightEye)
      cv2.drawContours(frame, [leftEyeHull], -1, (0, 255, 0), 1)
      cv2.drawContours(frame, [rightEyeHull], -1, (0, 255, 0), 1)
  # check to see if the eye aspect ratio is below the blink
      # threshold, and if so, increment the blink frame counter
      if ear < EYE_AR_THRESH:
             #print('detected')
       if(COUNTER>EYE_AR_CONSEC_FRAMES):
                    if ("alert" not in found and DaTa_r=="run"):
                           print("alert")
                           arduinoData.write(('alert').encode())
                           time.sleep(0.3)
                           found.clear()
                           found.add("alert")
```

```
cv2.putText(frame, "DROWSINESS ALERT!", (10, 30),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
            elif(COUNTER>2):
                  winsound.PlaySound("*", winsound.SND_ALIAS)
                  COUNTER+=1
            else:
                  COUNTER+=1


      # otherwise, the eye aspect ratio is not below the blink
      # threshold, so reset the counter and alert
      else:
            if ("run" not in found and DaTa_r=="run") :
                  print("run")
                  arduinoData.write(('run').encode())
                  time.sleep(0.3)
                  found.clear()
                  found.add("run")
            COUNTER = 0
            #print("running")
            # draw the computed eye aspect ratio on the frame to help
              # with debugging and setting the correct eye aspect ratio
              # thresholds and frame counters
              cv2.putText(frame, "VAL: {:.3f}".format(ear), (300, 30),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
   # show the frame
        cv2.imshow("Frame", frame)
        key = cv2.waitKey(1) & 0xFF
  # if the ESC key was pressed, break from the loop
        if key == 27:
              break
# do a bit of cleanup
cv2.destroyAllWindows()
vs.stop()
```

5.2 SAMPLE SCREENSHOTS

- The suggested system is still in the research stage, but it has been implemented into a few higher-end cars to test the proposed system's and algorithm's correctness. The Arduino IDE USB camera is loaded with Python, as well as OpenCV, With embedded system . The detection programme can recognise a picture of the driver taken with the camera, with a green rectangle indicating the driver's face and a red rectangle indicating the open eye region.

- In another part of the software, the person's eyes are closed, and the system detects this closed eyes red rectangle frame. If the closed eyes frame detects more than 4 frames in a program, a warning alarm is generated via a buzzer and stop the vehicle.

- The Arduino uno board was used to build the prototype. The Arduino IDE receives the input from the camera and runs these algorithms on it to identify tiredness in the user. By using processor boards with increased processing capacities, the system can be made more powerful and speedy.

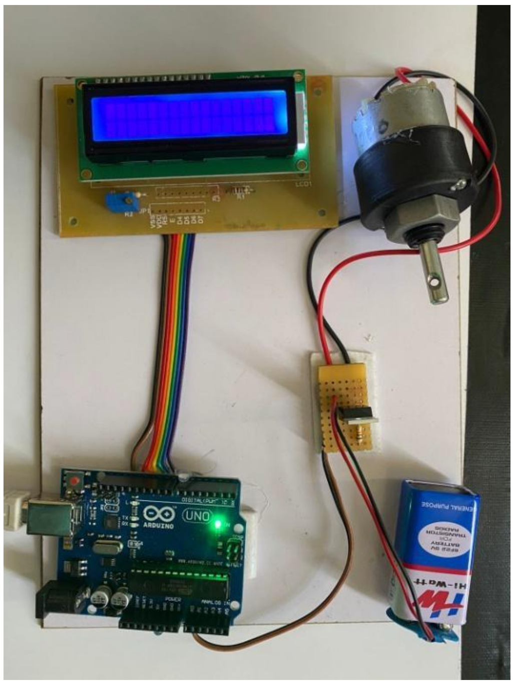


Figure 5.2.1 Circuit

Fig 1 shows the circuit in which Arduino is connected withThe LCD Display and motor driver and Gear motor the circuit is ready to work.

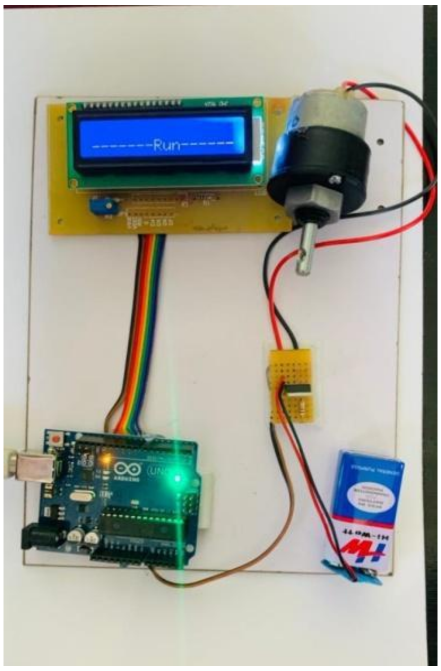


Figure 5. 2 .2Output message on LCD display

Fig 1 shows the real-time output of the system on the LCD display

CHAPTER 6

CONCLUSION

we have presented the application ofcomputer vision with embedded systems and targeted for reducing road accidents due to driver drowsiness . Monitoring and detecting the driver’s behavior to ensure road safety is important because road accidents take place. Hence it is important to capture driver behavior which will control the accidents due to rash driving under the influence of drowsiness . The proposed system deals with detection the Drowsiness using web cam and accordingly precautions are taken. The system works well even in case of drivers wearing spectacles and under low light conditions also. Development of software algorithm is completed which is partially tested and found successfully working.

FUTURE ENHANCEMENT

This model has great scalability and many additional parameters like blink rate, yawning, state of the car, etc can be used for enhanced accuracy of drowsiness. As part of the future scope, we plan to further work on the project by adding a sensor to track the heart rate of the driver in order to prevent accidents caused due to sudden heart attacks or strokes of the driver. Same model and techniques can be used for various other uses like introducing this model under many streaming brands such as Netflix can detect when the user is asleep and stop the video accordingly. It can also be used in applications that require the users to be focused

REFERENCES

1. Adamson, S., & Enright, S. Alcohol Gas Detector “Breathalyzer”. Mitsubayashi, Kohji, et al. "Biochemical gas- sensor (biosniffer) for breath analysis after drinking." SICE 2004 Annual Conference. Vol.1.IEEE,2004.

2. Anjali K U, Athiramol K Thampi, Athira Vijayaraman, Franiya Francis M, Jeffy James N, Bindhu K Rajan “ Real-Time Nonintrusive Monitoring and Detection of Eye Blinking in View of Accident Prevention Due to Drowsiness” 2016 International Conference on Circuit ,Power and Computing Technologies[ICCPCT].

3. A.K. Biswal, D. Singh, B. Pattanayak, D. Samanta, M.-H. Yang, IoT-Based Smart Alert System for Drowsy Driver Detection, Wirel. Commun. Mob. Comput. 2021 (2021) 1–13. https://doi.org/10.1155/2021/6627217.

4. U. Chikezie, N. Nwazor, A DROWSINESS DETECTION SYSTEM USING COMPUTER VISION AND IoT, IARJSET.

5. K. Fagerberg.Vehicle-based detection of inattentive driving for integration in an adaptive lane departure warning system Drowsiness detection,M.S. thesis, KTH Signals Sensors and Systems, Stockholm, Sweden, 2004.

6. G. Federico, G. Petrelli, In-Vehicle Drowsiness Detection, (2022).

7. Grace; Richard, et al. "A drowsy driver detection system for heavy vehicles."Digital Avionics Systems Conference, 1998. Proceedings, 17th DASC. The AIAA/IEEE/SAE. Vol. 2. IEEE, pp.50-70, 1998.

8. D. M., IOT based Drowsiness Detection System for Road Safety, Int. J. Psychosoc. Rehabil. 24 (2020) 7290–7296. https://doi.org/10.37200/IJPR/V24I5/PR2020761.

9. H. Iwamoto, K. Hori, K. Fujiwara, M. Kano,Real-driving-implementable drowsy driving detection method using heart rate variability based on long short-term memory and autoencoder, IFAC-PapersOnLine. 54 (2021)526-531.ttps://doi.org/10.1016/j.ifacol.2021.10.310.

10. R.Jabbar, M. Shinoy, M. Kharbeche, K. Al-Khalifa,M. Krichen, and K. Barkaoui, “Driver drowsiness detectionmodel using convolutional neural networks techniques for android application,” in 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), IEEE, 2020.

11. C. Jacobé de Naurois, C. Bourdin, A. Stratulat,E. Diaz, J.-L. Vercher, Detection and prediction of driver drowsiness using artificial neural network models, Accid. Anal. Prev. 126 (2017). https://doi.org/10.1016/j.aap.2017.11.038

12. R. Kepner, EFFICIENCY OF THE EMERGENCY ALERT SYSTEM, (2022).

13. Killoran, A., Canning, U., Doyle, N., & Sheppard, L, “Review of effectiveness of laws limiting blood alcohol concentration levels to reduce alcohol-related road injuries and deaths” Final Report. London: Centre for Public Health Excellence (NICE), 2010.

14. Murata; Apsua; Yasutaka Hiramatsu. "Evaluation of drowsiness by HRV measures-basic study for drowsy driver detection." Proceedings of 4th International Workshop on Computational Intelligence & Applications. Hiroshima: Hiroshima University, 2008, pp. 38-45.

15. M.S. Satyanarayana, P. Aruna, P. Guruprasad, Continuous Monitoring and Identification of Driver Drowsiness Alert System, Glob. Transitions Proc. 2 (2021). https://doi.org/10.1016/j.gltp.2021.01.017.

16. C. Shembekar, IoT based Alcohol and Driver Drowsiness Detection And Prevention System, Int. J. Eng. Res. V9(2020). https://doi.org/10.17577/IJERTV9IS090375.

17 .Viola, P., Jones, M. Rapid object detection using a boosted cascade of simple features. Computer Vision and Pattern Recognition, IEEE Computer Society Conference.2001;I:I

18. E. Vural, M. Cetin, A. Ercil, G. Littlewort, M. Bartlett, and J. Movellan, “Drowsy driver detection through facial movement analysis,” International Workshop on HumanComputer Interaction, vol. 4796, 2007.

19. M. Wankhede, Vehicle Accident Detection and Emergency Alert System, Int. J. Res. Appl. Sci. Eng. Technol. 7 (2019) 2617–2619. https://doi.org/10.22214/ijraset.2019.5432 .

20. Zutao Zhang; Jiashu Zhang, “A New Real-Time Eye Tracking for Driver Fatigue Detection,” Proc.2006 6th International Conference on ITS Telecommunications, pp.8-11, 2006.