# **AUTO ALERTING AND IDENTIFICATION SYSTEM**

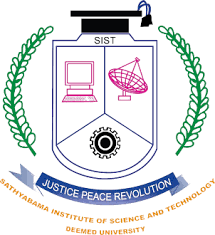
Submitted in partial fulfillment of the requirements for the award of

Bachelor of Engineering Degree in Computer Science and Engineering

By

**SATHI SARATH VENKATA REDDY (Reg.No - 39110906)**

**SHAIK SHEERU ALI MOULALI (Reg.No - 39110932)**

****

## **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**SCHOOL OF COMPUTING**

**SATHYABAMA**

## **INSTITUTE OF SCIENCE AND TECHNOLOGY**

## **(DEEMED TO BE UNIVERSITY)**

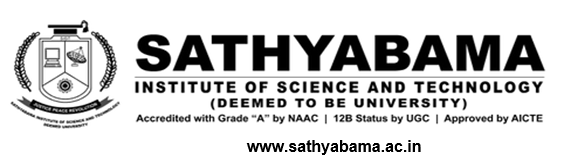
**Accredited with Grade “A” by NAAC | 12B Status by UGC | Approved by AICTE**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,**

**CHENNAI - 600119**

# 

# **APRIL - 2023**

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**BONAFIDE CERTIFICATE**

This is to certify that this Project Report is the bona fide work of  **SATHI SARATH VENKATA REDDY (REG. NO. 39110906) and SHAIK SHEERU ALI MOULALI (REG. NO. 39110932)** who carried out the project entitled “**AUTO ALERTING AND IDENTIFICATION SYSTEM”** under my supervision from January 2023 to April 2023.

**Internal Guide**

**Dr. JOSHILA GRACE L.K., B.E., Ph.D.,**

**Head of the Department**

**Dr. L. LaKSHMANAN, M.E., Ph.D.,**

**Submitted for Viva-voce Examination held on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Internal Examiner External Examiner**

**DECLARATION**

I, **SATHI SARATH VENKATA REDDY (REG. NO. 39110906),** hereby declare that the project report entitled “**AUTO ALERTING AND IDENTIFICATION SYSTEM”** done by me under the guidance of **Dr. JOSHILA GRACE L.K., B.E., Ph.D.**  is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

**DATE:**

## **PLACE: Chennai**

## **SIGNATURE OF THE CANDIDATE**

**ACKNOWLEDGEMENT**

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D.**, **Dean**, School of Computing, **Dr. S. Vigneshwari, M.E., Ph.D. and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of **Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.Joshila Grace L.K., B.E., Ph.D.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project

**ABSTRACT**

To improve safety in road areas, it is essential that vehicles are equipped with reliable safety features that help prevent accidents and minimize loss of life and property. Recent reports have shown a significant increase in accidents in night times due to drivers rushing to reach their destination. As a result, Driver drowsiness detection systems are becoming increasingly popular as a means of reducing the number of accidents caused by driver fatigue. These systems use a combination of sensors and machine learning algorithms to monitor the physical and behavioural characteristics of drivers in real-time, allowing them to detect when a driver is becoming drowsy or fatigued. Sensors used in these systems may include cameras, biometric sensors, and steering sensors, which track various indicators of driver drowsiness, such as eye movements, facial expressions, heart rate, and skin conductance. When the system detects signs of drowsiness, it can alert the driver using audible warnings, visual cues, or vibrations, giving the driver time to take a break and rest before continuing their journey. The potential benefits of driver drowsiness detection systems are significant, as they can help reduce the number of accidents caused by fatigue, improving road safety for all road users.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ABSTRACT** | **v** |
|  | **TABLE OF CONTENTS** | **vi** |
|  | **LIST OF FIGURES** | **viii** |
|  | **LIST OF ABBREVIATIONS** | **x** |
| **1** | **INTRODUCTION**   * 1. Domains Used | **1**  **2** |
| **2** | **LITERATURE SURVEY**  2.1 Inferences From Literature Survey  2.2 Problems In Existing Systems | **6**  **6**  **9** |
| **3** | **REQUIREMENTS ANALYSIS**  3.1 Proposed System  3.2 Drowsiness Detection | **10**  **10**  **10** |
| **4** | **DESCRIPTION OF PROPOSED SYSTEM**  4.1 Selected Methodology  4.2 Required Libraries  4.3 Architecture of Proposed System  4.4 Design of the Proposed System | **13**  **13**  **13**  **16**  **16** |
| **5** | **IMPLEMENTATION DETAILS**  5.1 Haar Cascade Classifiers  5.2 Euclidean Distance  5.3 Eye Aspect Ratio  5.4 Driver Drowsiness Detection Identity Locations | **18**  **18**  **19**  **19**  **20** |
| **6** | **RESULTS AND DISCUSSION** | **22** |
| **7** | **CONCLUSION**  7.1 Future Work  7.2 Research Issues  7.3 Implementation Issues | **24**  **25**  **25**  **28** |
|  | **REFERENCES** | **29** |
|  | **APPENDIX**   1. **SOURCE CODE** 2. **SCREENSHOTS** 3. **RESEARCH PAPER** 4. **PLAGIARISM REPORT** | **31**  **31**  **35**  **37**  **40** |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **TITLE** | **PAGE NO.** |
| 1.1 | Application of Computer Vision | 2 |
| 1.2 | Types of Machine Learning Models | 3 |
| 1.3 | Sensor Technology | 4 |
| 1.4 | Embedded System | 5 |
| 4.1  4.2 | Working Function of Proposed System  System Architecture of Proposed System | 13  16 |
| 4.3 | Flowchart of Proposed System | 16 |
| 5.1  5.2 | Haar Cascade Classifiers Image Processing  Euclidean Distance Formula | 18  19 |
| 5.3  6.1  6.2 | Working Function of Eye Blink Detection  Day Time Detection With/Without Spectacles  Night Time Detection With/Without Spectacles | 20  22  23 |

|  |  |
| --- | --- |
| **ABBREVIATION** | **FULL FORM** |
| FFT | Fast Fourier transforms |
| ICA | Independent component analysis |
| ANN | Artificial neural network |
| SVM | Support vector machines |
| KNN | K-Nearest neighbour |
| PCA | Principal component analysis |
| LDA | Linear discriminant analysis |
| GMM | Gaussian mixture models |
| RBFN | Radial basis function networks |
| EAR | Eye aspect ratio |
| CNN | Convolutional neural network |

**LIST OF ABBREVIATIONS**

**CHAPTER 1**

**INTRODUCTION**

According to the survey, Driver fatigue results in over 50% of road accidents each year. Using technology to detect driver fatigue is an interesting challenge that would help in preventing accidents. In the past various efforts have been reported in the literature on approaches for fatigue detection by automobile drivers. In the last decade alone, many countries have begun to pay great attention to the automobile driver safety problem. Driver drowsiness detection system is an innovative technology that uses various sensors to monitor the driver's behavior and detect signs of drowsiness. The system is designed to improve road safety by alerting the driver when they are becoming too tired to drive safely.

The system works by analyzing the driver's facial expressions, eye movements, and other physiological signals such as heart rate and respiration. The data collected from these sensors is processed using machine learning algorithms to determine the driver's state of alertness. When the system detects that the driver is becoming drowsy, it will issue an alert to warn the driver to take a break or stop driving altogether. The alert can be in the form of an audible alarm, a visual alert, or a vibration on the steering wheel or seat. Driver drowsiness detection systems are becoming increasingly common in modern vehicles, especially in commercial vehicles where long hours of driving can lead to fatigue.

The technology can also be integrated into wearable devices such as smartwatches or glasses, enabling drivers to monitor their alertness levels even when they are not driving. In addition to improving road safety, driver drowsiness detection systems can also help to reduce the number of accidents caused by driver fatigue, which is a significant contributor to road accidents worldwide. The technology is also cost-effective and easy to install, making it accessible to a wide range of vehicles and drivers. However, it is worth noting that driver drowsiness detection systems are not fool proof, and drivers should still take responsibility for their safety by taking regular breaks and getting enough sleep before driving. Nevertheless, the technology represents a significant step forward in improving road safety and reducing the number of accidents caused by driver fatigue.

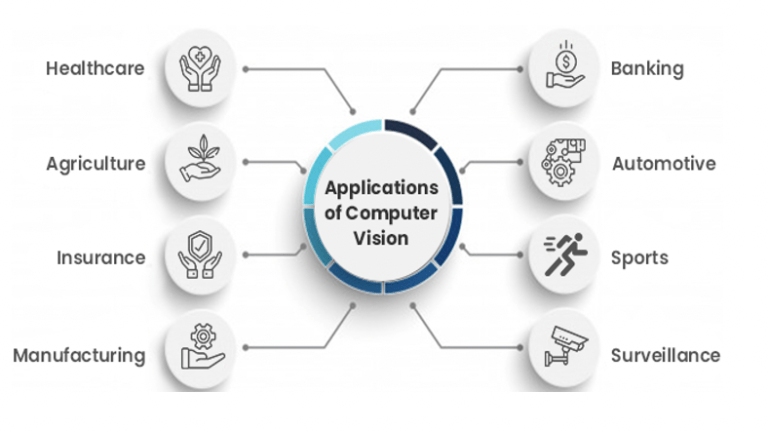
# **DOMAINS USED:**

# In this project, we propose a direct approach that makes use of vision-based techniques to detect drowsiness.

# Fatigue detection is done twice, before and while driving, and physiological tests consist of the facial analysis determined by the head position, blinking duration, and yawning frequency, and is done before and while driving. While driving data is also taken into consideration together with facial analysis to detect. The driver is warned of fatigue before and while driving. Unseen obstructions and high, uncontrollable speeds that exceed the permitted speed limits for a given zone are the major causes of deadly traffic accidents. The most challenging task for the auto industry, traffic regulators, and research and development teams in the automobile industry is reducing crashes and their worst consequences. An audio or visual signal should warn drivers of the restricted area before they begin driving, alerting them to the obstacle in front of the road.

# There are various domains used in computer networks system includes:

***1.1.1. COMPUTER VISION:*** Computer vision is a field of artificial intelligence and computer science that deals with enabling machines to interpret, understand, and analyse digital images and videos. The goal of computer vision is to enable machines to perceive the world in the same way that humans do, by extracting meaning and insights from visual data. Computer vision algorithms typically involve techniques such as image processing, pattern recognition, machine learning, and deep learning.

****

***Fig 1.1: Applications of computer vision***

# The computer vision algorithms used in these applications can identify patterns and features in visual data to make accurate predictions, decisions, and recommendations. Some of the challenges in computer vision include dealing with noise, variations in lighting conditions, occlusion, and dealing with high-dimensional data.

# ***1.1.2.MACHINE LEARNING:*** Machine learning is a subfield of artificial intelligence that involves training algorithms to recognize patterns in data and make predictions or decisions based on those patterns. It is based on the idea that machines can learn from experience and improve their performance over time. Machine learning algorithms can be classified into three types: supervised learning, unsupervised learning, and reinforcement learning.

# 

# ***Fig 1.2: Types of Machine Learning Models***

# Machine learning has many practical applications, including natural language processing, computer vision, recommendation systems, and predictive analytics. Machine learning algorithms require large amounts of data and computational power to train effectively.

# ***1.1.3.SENSOR TECHNOLOGY:*** Sensor technology involves the use of electronic devices to detect changes in the physical environment and convert them into electrical signals. These sensors are used to monitor and measure various parameters such as temperature, pressure, humidity, light, sound, and motion.Sensor technology is used in a wide range of applications, including industrial automation, healthcare, transportation, and smart homes. For example, in healthcare, sensors can be used to monitor a patient's vital signs and alert medical staff if there are any abnormalities.

# There are various types of sensors, including temperature sensors, pressure sensors, accelerometers, gyroscopes, and proximity sensors. These sensors can be integrated into electronic devices such as smartphones, wearables, and home appliances.

****

***Fig 1.3: Sensor Technology***

# The sensors are key components of the Internet of Things (IoT), which enables devices to communicate and share data with each other. Sensor technology is a rapidly advancing field that has many practical applications. The development of smart sensors and the IoT is expected to revolutionize many industries, making our lives more efficient, safer, and comfortable.

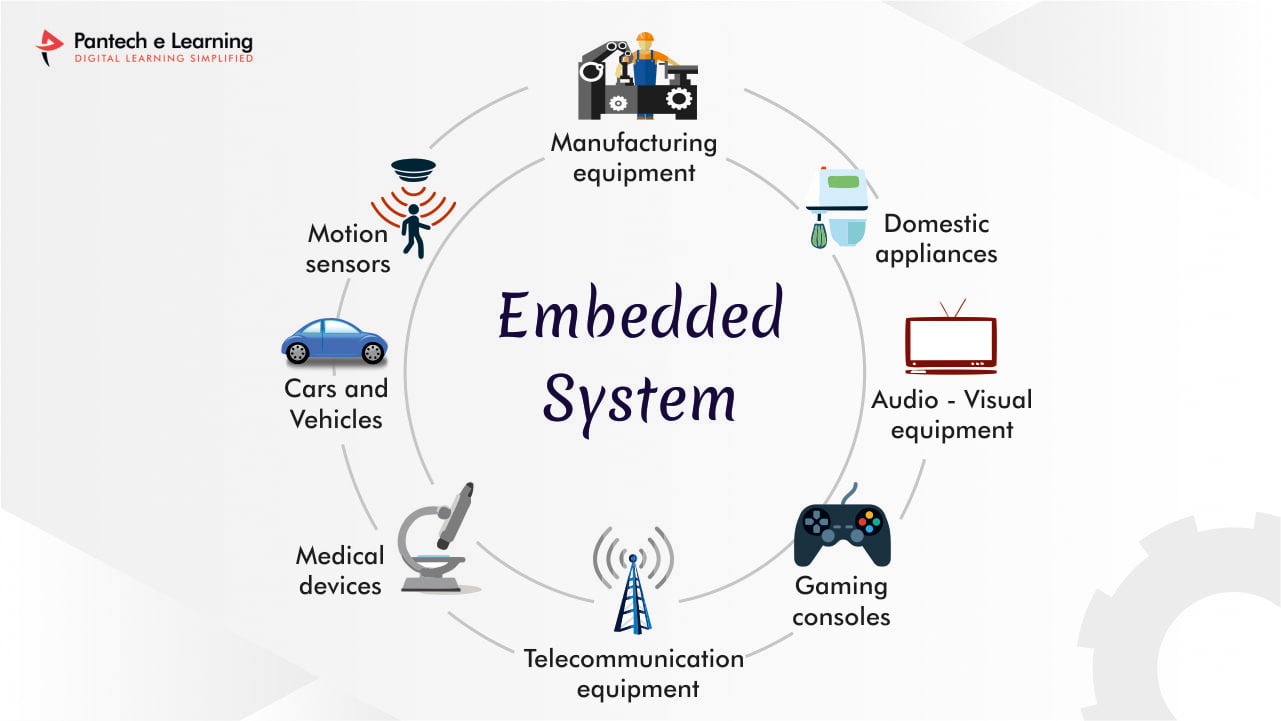
# ***1.1.4.HUMAN FACTORS:*** Human factors is a field of study that focuses on understanding how people interact with technology, equipment, and systems. It is concerned with optimizing the design of products and services to enhance usability, safety, and user experience.Its principles are used in many industries, including aviation, healthcare, transportation, and consumer products. For example, in aviation, human factors research is used to improve cockpit design, pilot training, and air traffic control systems to enhance safety.

# The application of human factors principles can lead to many benefits, including increased productivity, reduced errors, and improved user satisfaction. It can also help to reduce the risk of accidents and injuries. Human factors research is used to understand the causes of drowsiness in drivers and to develop effective interventions to prevent drowsy driving.

# ***1.1.5.SIGNAL PROCESSING:*** Signal processing is a branch of electrical engineering that deals with the manipulation and analysis of signals. Signals can be any type of data that varies with time or space, such as sound, images, or sensor readings. The goal of signal processing is to extract useful information from signals, filter out noise and unwanted components, and enhance or modify the signal for further analysis or communication.

# Signal processing also plays an important role in the development of machine learning and artificial intelligence systems, as it is often used to pre-process and clean data before it is fed into these systems. Signal processing techniques are used to filter, extract, and analyze the data collected from the various sensors and cameras used in the system.

# ***1.1.6.EMBEDDED SYSTEMS:*** Itis a computer system that is designed to perform specific tasks within a larger system or product. Unlike general-purpose computers, which can run a wide range of software applications, embedded systems are typically dedicated to a single function or set of functions. It can be used in a wide range of applications, from consumer electronics and appliances to industrial automation and transportation. They are designed to be low-power and low-cost and often operate in real-time environments, where timing and responsiveness are critical.

****

***Fig 1.4: Embedded System***

# These systems often include sensors, actuators, and other components that allow them to interact with their environment and perform their intended functions.The challenges in designing embedded systems are balancing the need for functionality with the constraints of cost, power consumption, and size.

# **CHAPTER 2**

# **LITERATURE SURVEY**

**2.1** **INFERENCES FROM LITERATURE SURVEY**

# Bandpass channels and thresholding were used by Li and Chung to remove disturbance from low and high-frequency ECG information. It is necessary to eliminate important highlights for grouping after pre-processing in which the yield information is broken up in recurrence space using Fast Fourier Transforms (FFT). Li et al performed comparative examinations on EEG information to decide the drowsiness of a driver. They utilized Independent Component Analysis (ICA) to isolate and limit blended EEG information to specific mental exercises. From the preprocessed information, highlights are separated in the recurrence area.

Sunagawa et al discussed the actualized drowsiness location framework by utilizing EOG information. The first distinguished the eye squinted from the recorded EOG information It removed the eye top development parameters as highlights to be characterized using Support Vector Machines. The fundamental characteristic of physiological estimating systems is that they can decide the reduction in the measure of sharpness early before the genuine drowsiness scene begins. People do not usually get sleepy in a moment, and there is a progressive decline accordingly or action of the different body parts which in the long run lead to drowsiness.

For instance, in the EEG investigation, the adjustment of the flag controller at the alpha range (8 – 12Hz) shows early tiredness. Physical estimating procedures can quantify such changes at the beginning times. The individual can be cautioned, or the best possible well-being measure can be taken before mishaps could happen. The deliberate signs are likewise dependable to distinguish drowsiness as their connection with the driver’s readiness is very precise. They are generally free of external factors, for example, the nature of the street, the sort of vehicle, or the traffic. Subsequently, they have more specific drowsiness location ability than vehicle-based and conduct estimating methods.

Dasgupta et al. used progressive image-sifting items such as picture subtraction, morphologically closed activities, and binarization. Lastly, I checked the number of pixels around the eyes distract to distinguish the eye conclusion. Ramzan et al. removed essential highlights from the fleeting contrast of sequential picture outlines. They utilized them to investigate the standards of eyelid development amid drowsiness.

Kaplan et al. has additionally displayed a non-meddlesome way to deal with drowsiness discovery. They utilized an R.I.R. brightening framework and a high-goal camera to acknowledge a surge of pictures and perform face and eye location. They connected channels on the eyes district and performed flat and vertical projections of the 162806 VOLUME 9, 2021 A. Altameem et al.: Early Identification and Detection of Driver Drowsiness by Hybrid Machine Learning pixel estimations of the identified eye territory. The vertical forecast compares to the eye tallness, which is utilized to assess the PERCLOS.

You et al. performed face and eye recognition. They followed the eye understudies utilizing non-straight Kalman and mean-move following. They likewise performed vertical and flat projections of the pixels around the eyes district. Since the eyeball shading is a lot darker than the encompassing, they determined the pixel esteems in the vertical projection to decide the level of eyelid conclusion. Chowdhury et al. processed the twofold inclination and logarithm picture of the eyes area, acquired arbitrary examples around the locale, and utilized an elliptic shape to speak to the eyes. They, at that point, employed an SVM classifier to choose whether the eyes were shut or not. One of the fundamental elements influencing the execution of PEPERCLOS-based frameworks is the surrounding lighting condition.

Utilizing a webcam could be suitable in the daytime or when there is adequate light to observe the driver’s eyes yet plainly could ineffectively perform when there is restricted lighting condition. Then again, a camera with infrared innovation may function admirably amid the night but perform inadequately in the sunlight since the retinal impressions of infrared cannot be gotten within sight of surrounding daylight reflections. Also, an unimportant examination of eye conclusion may not be sufficient to foresee drowsiness. The driver may not close his eyes all through the sleepy scenes, particularly amid the beginning times. A languid driver ordinarily does not dive to deep rest promptly; instead switches back and forth between falling asleep and opening his eyes. Opening the eyes in such advances can be confused as wakeful if eye conclusion is the main parameter being investigated. Subsequently, lately, a few specialists have been thinking about other facial developments notwithstanding eye conclusion, such as eyebrow raises, yawning, and head or eye position orientation.

U.S. Branch of Transportation pre-processed and shaped vectors utilizing 15 seconds of directing wheel information proposed by Khushaba et al. They additionally executed eye following to record the understudies’ breadth and shaped a vector of eye conclusion information of 15 seconds. At that point, they linked the two vectors and prepared an Artificial Neural Network (ANN) to decide the driver’s condition.

Soares et al. additionally joined to conduct vehicle-based measures. They reasoned that the unwavering quality and precision of the half-and-half technique was altogether higher than those utilizing single sensors. Budak et al. recommends that observing the driver’s head posture and introduction can give enough pieces of information to anticipate the driver’s aim. The driver’s face must be identified first to determine what the driver’s head represents. This is a critical advancement in any conduct procedure that requires a subject’s face to be checked.

Oyini Mbouna et al. proposed face recognition calculation has turned into a reference after which other face location strategies can be fabricated. An endeavor to improve this calculation is finished by preparing three classifiers (concentrated on the even pivot of the driver’s head) to effectively identify forward-looking, left-confronting, and right-confronting faces.

**2.2 PROBLEMS IN EXISTING SYSTEMS**

In this project, we have discussed several ways to detect drivers who are drowsy on the road. Each of these methods is described in length, along with the benefits and drawbacks of using it. According to the comparison study results, none of these procedures is 100 percent accurate. However, physiological parameters-based techniques are more accurate. Wireless sensors on the driver’s body, the driving seat, the seat cover, and the steering wheel can lessen their non-intrusiveness. Combining physiological measurements with vehicular or behavioral measures helps overcome the limitations of individual techniques, resulting in more accurate drowsiness detection. As the combination of ECG and EEG features, hybrid techniques help overcome the problems associated with one-of-a-kind techniques, emphasizing that combining physiological signals improves performance overusing them separately.

These are the best methods for supervised learning. The advantages and disadvantages of these methods, as well as a comparison study, are thoroughly examined. Classifiers differ in their accuracy depending on the scenario, according to research comparing them. Classifiers based on support vector machines (SVM) have the best accuracy and speed in most cases but are unsuitable for massive datasets. While HMM has a lower error rate, both CNN and HMM need more time to train and are more costly than SVM classifiers.

# **CHAPTER 3**

**REQUIREMENT ANALYSIS**

**3.1 PROPOSED SYSTEM**

The driver tiredness detection system in automotive vehicles is the main topic of this study. The driver's attention was observed in a variety of circumstances, including when they were wearing eyeglasses and when it was dark inside the car. The suggested system would continually monitor the driver's retina and every signal that is being watched and relayed to the microcontroller. In less than two seconds, the system can identify the sleepiness situation. When an aberrant situation is found, the driver is made aware of it by alerts, and the parking lights come on to stop the car, reducing accidents caused by drowsy driving.

***3.1.1 TARGET SPECIFICATIONS :***

The goal requirements for the interfaces were set after the project and customer needs for something like the solution. The manner of operation and the system's vehicle compactness are specified by this circumstance. fundamental norms.

The system will be evaluated and designed as follows:

* Reliable
* Low cost
* Low power consumption
* Reliable
* Quick response
* Durability
* Easy adaption in vehicles
* Compact process
* Highly secured

**3.2 DROWSINESS DETECTION**

Driver fatigue detection may actively track a driver's state of consciousness and warn them of any potentially dangerous driving circumstances. This sleepiness detection system's main goal is to provide a method that can lessen the number of accidents caused by exhaustion. Our research begins with the identification of weariness using a camera to identify face traits or images. The core task of the project, eye detection, will be carried out via OpenCV. The 8-megapixel camera input allows for the shooting of live pictures and video. An Arduino Uno R3 board may evaluate the collected frame. Python is used to apply algorithms. The Haar cascade classifier uses a library of pictures with both positive and negative values that can object recognition that seem to be close to the camera. When there is a match, the classifier displays a red border rectangle enclosing the found area. Examine the alarm that will sound if a driver or user becomes weary while keeping in mind the Haar cascades. implementing dib's facial landmark enhancements so that the display screen's sleepy detection may be turned on. witnessing how our upgraded driver tiredness detection system performs. The algorithm oversees identifying driver fatigue. Eye face landmarks have been drawn for both the closed and open eyes. The eye-aspect ratio is then plotted throughout time. The eye aspect ratio stays constant and then rapidly drops to nearly zero before rising once more, indicating a blink.

***3.2.1.DATA COLLECTION:*** The system collects data on the driver's facial features, head movement, and eye movement using a camera installed in the car. The camera should be able to capture clear images of the driver's face, even in low-light conditions.

***3.2.2.DATA PROCESSING:*** The system processes the collected data to extract the relevant features. The features can include the driver's eye closure rate, head position, and facial expression. Machine learning algorithms can be used to identify patterns in the data and classify the driver's state as drowsy or alert.

***3.2.3.ALERT MECHANISM:***The system should be able to alert the driver if signs of drowsiness are detected. The alert mechanism can be a visual or audio signal, such as an alarm or vibration in the driver's seat.

***3.2.4.ADAPTIVE SYSTEM:*** The system can be designed to adapt to the driver's behavior over time. For instance, the system can learn the driver's typical patterns of behavior and use that information to set a baseline for comparison. If the driver deviates significantly from their baseline, the system can assume that they are drowsy and trigger an alert.

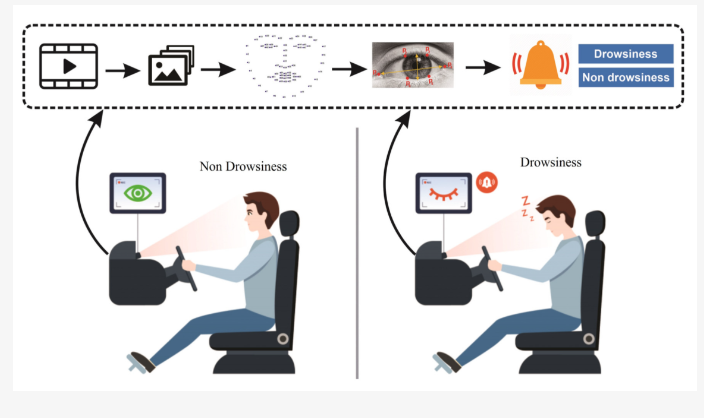
***3.2.5.INTEGRATION WITH OTHER SYSTEMS:*** The drowsiness detection system can be integrated with other safety systems in the car, such as the lane departure warning system or the automatic braking system. This integration can improve the overall safety of the car and reduce the risk of accidents.

***3.2.6.DATA STORAGE:*** The system can store the collected data for future analysis or for research purposes. The data can be used to develop more advanced algorithms for drowsiness detection or to study driver behavior.

# **CHAPTER 4**

**DESCRIPTION OF THE PROPOSED SYSTEM**

**4.1 SELECTED METHODOLOGY**

****

***Fig 4.1: Working function of proposed system***

***4.1.1.IMAGE ACQUISITION:*** An image is taken with a camera. The camera will be positioned in front of the driver to record the frontal face.

***4.1.2.PRE-PROCESSING:*** This module prepares the captured frames. Only grayscale photographs will function in the system. A color image must be pre-processed into a grayscale image for the system to operate.

***4.1.3.FACE DETECTION:*** The face detection function attempts to identify the face of the car driver within every picture it obtains from the frames the frame grabber provides. Using a collection of predetermined Haar cascade samples, this is accomplished.

**4.2. REQUIRED LIBRARIES**

***4.2.1.PYTHON***:

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics developed by Guido van Rossum.

***4.2.2.NUMPY LIBRARY:***

NumPy is a Python library for numerical computations. It provides a convenient way to work with arrays and matrices, which are fundamental data structures for many scientific and engineering applications. NumPy is built on top of low-level numerical libraries such as BLAS and LAPACK, which are written in C or Fortran for efficient numerical computations.

***4.2.3.ARGPARSE LIBRARY:***

Argparse is a Python library for parsing command-line arguments. It provides an easy and intuitive way to define command-line arguments for your Python scripts. Argparse can handle complex argument parsing scenarios and provides a lot of flexibility and customization options.

***4.2.4.IMUTILS LIBRARY:***

Imutils is a Python library for image processing and computer vision tasks. It provides a set of utility functions for working with images, such as resizing, rotating, cropping, and more. Imutils is built on top of the OpenCV library and provides a simplified interface for many common image-processing tasks.

***4.2.5.TIME LIBRARY:***

This module in Python provides functions for working with time-related operations. It includes functions for measuring time, converting time formats, and working with time intervals. such as benchmarking code performance, scheduling tasks, and more.

***4.2.6.DLIB LIBRARY:***

Dlib is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real-world problems. It is designed to be efficient and suitable for use in high-performance computing environments, such as mobile devices and embedded systems.

It provides a variety of machine learning algorithms such as classification, regression, clustering, and anomaly detection. It also includes tools for image processing, data visualization, and data analysis. Some of the algorithms implemented in dlib include:

1. Support Vector Machines (SVM)
2. K-Nearest Neighbor (KNN)
3. Principal Component Analysis (PCA)
4. Linear Discriminant Analysis (LDA)
5. Gaussian Mixture Models (GMM)
6. Ridge Regression
7. Radial Basis Function Networks (RBFN)

***4.2.7.CV2 LIBRARY:***

CV2 is a Python library for computer vision and image processing. It is built on top of the OpenCV library and provides a Python interface for working with images and videos.

CV2 provides a variety of functions for working with images and videos, such as:

1. Reading and writing images and videos
2. Displaying images and videos
3. Basic image manipulation such as resizing, cropping, and flipping
4. Image filtering such as smoothing, sharpening, and edge detection
5. Image transformation such as rotation, scaling, and perspective correction
6. Object detection and recognition using techniques such as Haar cascades and deep learning
7. cv2 is widely used in fields such as robotics, self-driving cars, and augmented reality.

***4.2.8. OS LIBRARY:***

OS is a Python module that provides a way of using operating system-dependent functionality like reading or writing to the file system, setting environment variables, etc. It provides a way to interact with the operating system in a platform-independent manner.

**4.3. ARCHITECTURE OF PROPOSED SYSTEM**

Diagram

Description automatically generated

# ***Fig 4.2: System Architecture of Proposed System***

***4.3.1 INPUT IMAGE:*** The input image for the system is a video stream captured from a camera mounted in the car facing the driver.

***4.3.2.PRE-PROCESSING:*** The preprocessing step involves converting the color image to grayscale and then applying a series of filters to enhance the image quality and remove noise.

***4.3.3 HAAR CASCADE CLASSIFIER ALGORITHM:*** The Haar Cascade Classifier is used for face detection. It works by searching for patterns of pixel intensities that match the profile of a face. Once the face is detected, the system extracts the region of interest (ROI) around the eyes.

***4.3.4 FEATURE EXTRACTION:*** The system extracts features such as eye closure, blink duration, and eyelid movement to determine if the driver is drowsy or not.

***4.3.5 FACE DETECTION:*** The system uses face detection algorithms such as the Viola-Jones algorithm to detect the driver's face. Once the face is detected, the system extracts the region of interest (ROI) around the eyes.

***4.3.6 IRIS DETECTION:***  The system uses iris detection algorithms to track the movement of the iris and measure the diameter of the pupil. This helps in determining if the driver is drowsy or not.

***4.3.7 ALARM:***  An alarm is triggered if the system detects that the driver is drowsy. The alarm can be a sound, vibration, or visual warning.

**4.4 DESIGN OF THE PROPOSED SYSTEM**

A driver drowsiness detection system is an intelligent system that helps drivers avoid accidents caused by drowsiness or fatigue while driving. The system works by detecting certain signs of drowsiness or fatigue, such as yawning, blinking, or nodding, and alerting the driver to take a break or stop driving altogether.

Diagram

Description automatically generated

***Fig 4.3: Flowchart of Proposed System***

***4.4.1.ALGORITHM OF THE PROPOSED SYSTEM:***

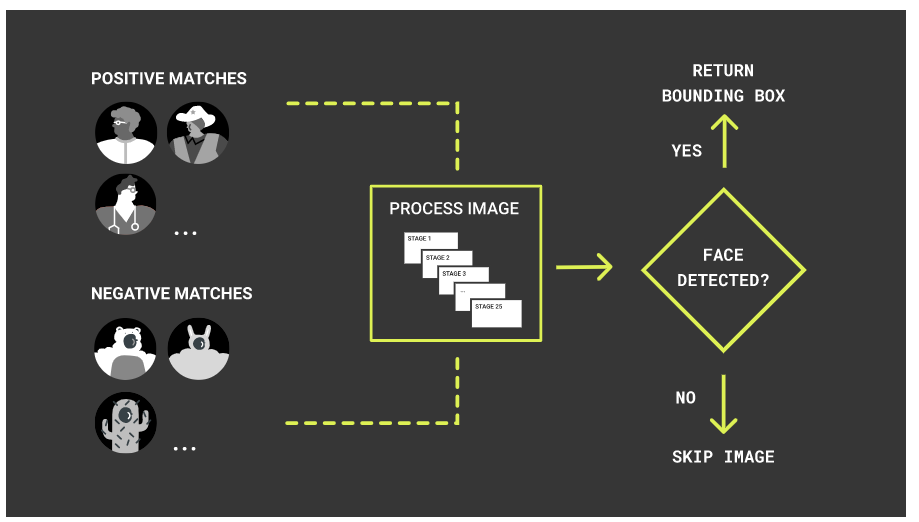
1. Start the system and initialize the camera.
2. Capture a live video feed of the driver's face using the camera.
3. Analyze the video feed to detect facial landmarks, such as the eyes, nose, and mouth. Calculate the position, movement, and size of the driver's eyes using computer vision techniques.
4. Set a threshold value for the driver's state of alertness, below which the system triggers an alarm.
5. Compare the driver's state of alertness to the threshold value.
6. If the driver's state of alertness is below the threshold value, activate the alarm

**CHAPTER 5**

**IMPLEMENTATION DETAILS**

**5.1. HAAR CASCADE CLASSIFIERS:**

The Haar Cascade Classifier algorithm is an object detection technique that uses a set of Haar-like features to identify objects in digital images and videos. The algorithm works by scanning the image at different scales and positions and evaluating the Haar-like features at each window position. These features are computed by comparing the sums of pixel values in rectangular regions of the image. The feature vector is then compared to a set of pre-trained classifiers, which are created using machine learning techniques such as Adaboost and Cascade Classifiers. The algorithm uses a cascading approach to speed up the detection process, with each classifier being more complex and accurate than the previous one. The feature vector is then compared to a set of pre-trained classifiers, which are created using machine learning techniques such as Adaboost and Cascade Classifiers. The algorithm uses a cascading approach to speed up the detection process, with each classifier being more complex and accurate than the previous one.



***Fig 5.1: Haar cascade classifiers image processing***

This approach uses a cascade algorithm that has multiple stages, where the output from one stage acts as additional information for the next stage in the cascade. The different stages are responsible for detecting edges, lines, contrast checks and calculating pixel values in a given image. Larger areas of the image are checked first in the earlier stages, followed by more numerous and smaller area checks in later stages. Fewer stages make the detection faster, while leading to more false positives. The built-in Haar Cascade model for faces was trained with hundreds of images containing faces that are labelled as such and images that do not contain faces labelled differently. That allows the algorithm to distinguish such images after it is being trained.

**5.2. EUCLIDEAN DISTANCE**

The Euclidean distance formula is often used in driver drowsiness detection systems to calculate the distance between different facial landmarks, which helps in analyzing the driver's facial expressions and movements.

1. Facial Landmark Detection: The system uses computer vision algorithms to detect and locate facial landmarks, such as the eyes, nose, and mouth.
2. Landmark Coordinates**:** The system extracts each landmark’s x and y coordinates from the video feed captured by the camera.
3. Distance Calculation: The system calculates the Euclidean distance between two facial landmarks by using the following formula

Graphical user interface, application

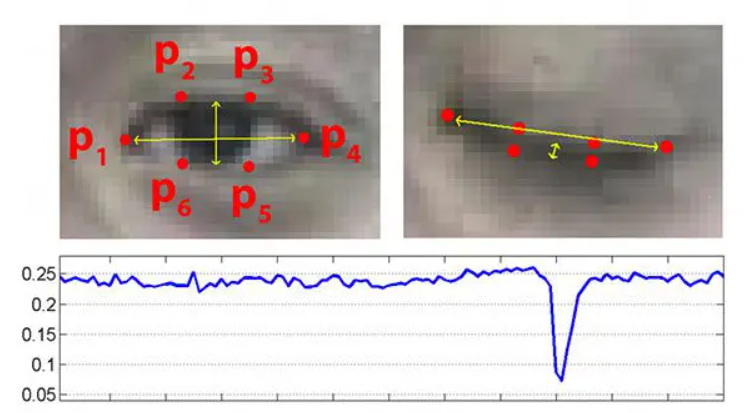
Description automatically generated

***Fig 5.2: Euclidean distance formula.***

1. Here, **(x1, y1)** and **(x2, y2)** are the coordinates of two facial landmarks.

**5.3. EYE ASPECT RATIO**

The system may use the Euclidean distance formula to calculate the eye aspect ratio (EAR), which is the ratio of the horizontal distance between the two eye landmarks to the vertical distance between the same two landmarks. The EAR is an important indicator of the driver's drowsiness level.



***Fig 5.3: Frequency and Duration of Eye Blink.***

Overall, the Euclidean distance formula is a useful tool for driver drowsiness detection systems to measure the distance between different facial landmarks and calculate various metrics, such as the eye aspect ratio, to detect signs of drowsiness and prevent accidents.

**5.4.** **DRIVER DROWSINESS DETECTION IDENTITY LOCATION**

**5.4.1.CAMERA CAPTURING:** The system captures a video stream of the driver's face using a camera, typically mounted on the dashboard or steering column.

**5.4.2.THE FACIAL LANDMARK DETECTION:** The system uses facial landmark detection algorithms to locate and track the position of the driver's eyes and other facial features.

Chart, scatter chart

Description automatically generated

***Fig 5.4: Working function of Eye Blink Detection.***

**5.4.3.THE EYE BLINK DETECTION:** The system then analyzes the movement of the driver's eyelids to detect signs of drowsiness or fatigue. It may use various techniques to identify eye blinks, such as threshold-based methods or machine learning-based methods.

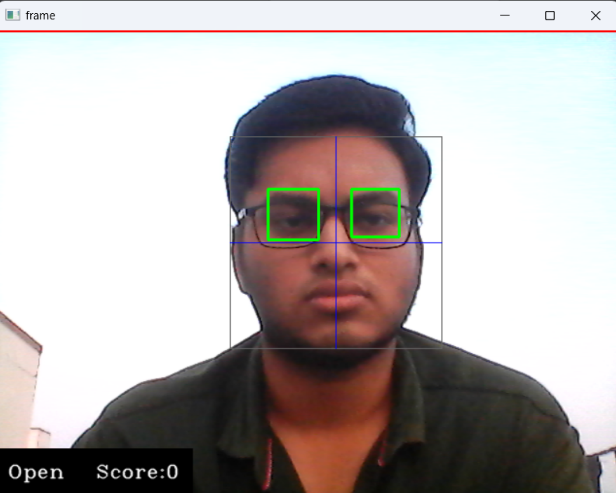
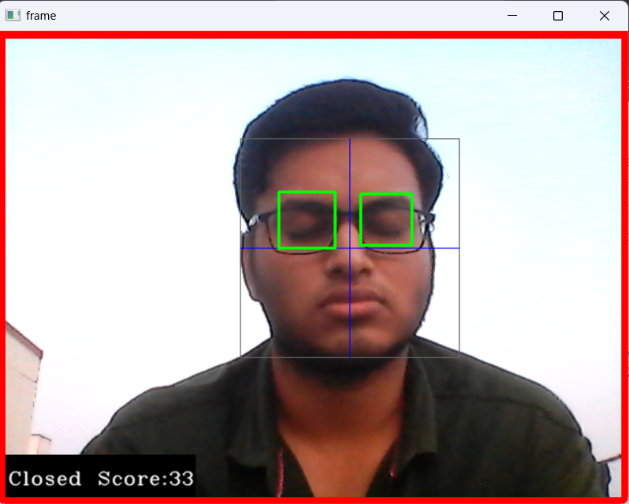
**5.4.4.THE BLINK DURATION AND FREQUENCY:** The system calculates the duration and frequency of eye blink over time and compares them to a predetermined threshold. If the blink duration or frequency exceeds the threshold, the system raises an alert to notify the driver of their drowsiness.

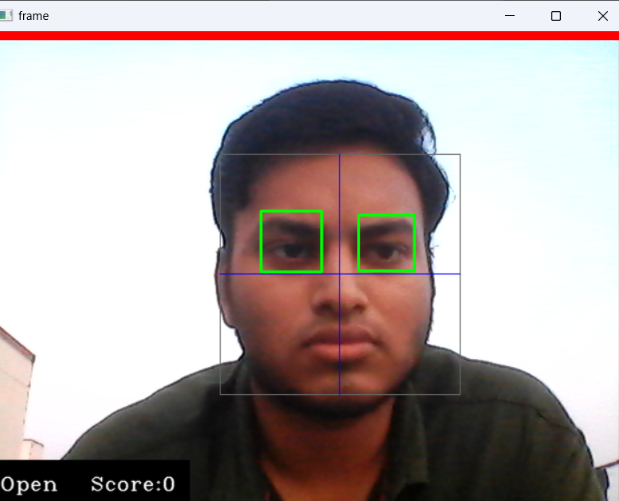
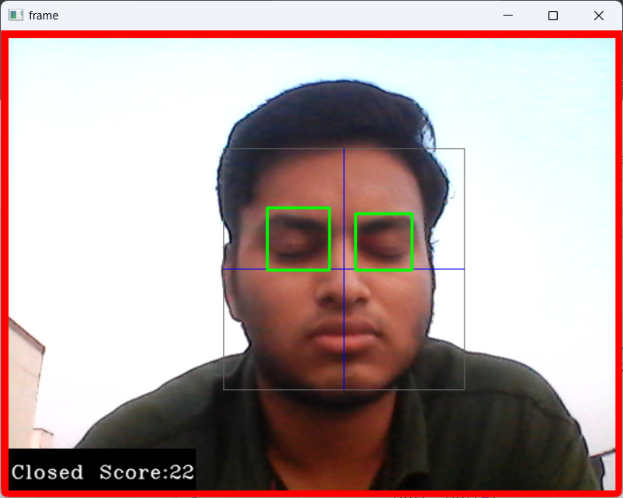
**5.4.5.ALERT GENERATION**: The system may generate an alert in various forms, such as an audible warning, visual warning, or vibration warning, to notify the driver to take a break or stop driving.

# **CHAPTER 6**

# **RESULT AND DISCUSSION**

In our project, we can detect the signs of drowsiness in drivers and alert them to take a break or rest, thereby reducing the risk of accidents due to fatigue. The result of a driver drowsiness detection system is typically evaluated based on its accuracy in detecting drowsiness in real-world driving scenarios.

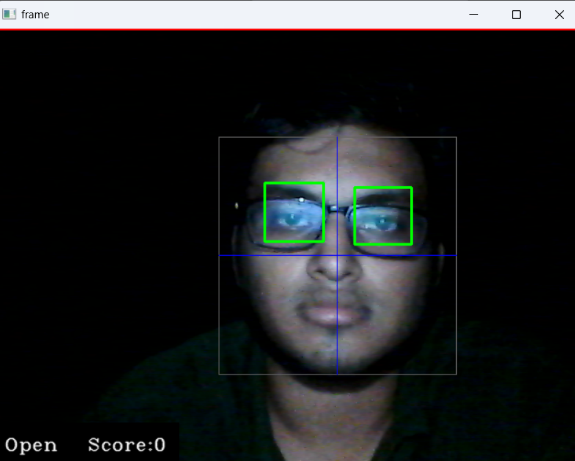
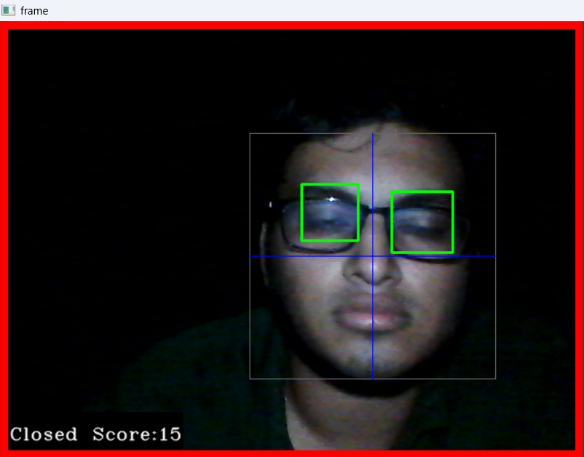
 

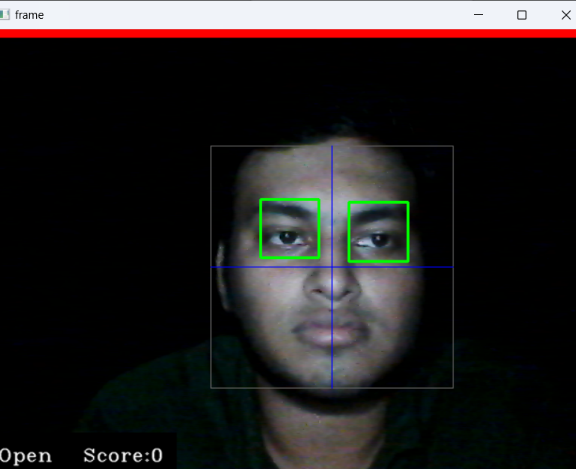
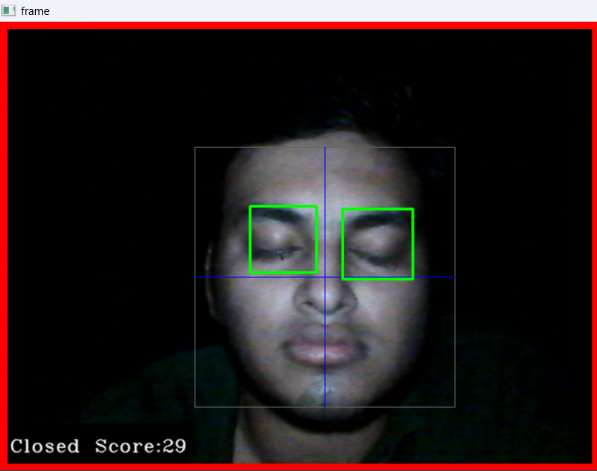
 

***Fig 6.1 : Day time detection with/without spectacles***

The accuracy of a driver drowsiness detection system can be measured in terms of its sensitivity, specificity, and overall accuracy. Sensitivity refers to the ability of the system to correctly identify drivers who are actually drowsy, while specificity refers to the ability to correctly identify drivers who are not drowsy. Overall accuracy refers to the system's ability to correctly classify both drowsy and non-drowsy drivers.

The performance of a driver drowsiness detection system depends on various factors, such as the quality of the data used for training, the selection of appropriate features, and the choice of the machine learning algorithm. Additionally, real-world factors such as changes in lighting conditions, driver posture, and camera angles can also affect the performance of the system.

***Fig 6.2: Night time detection with/without spectacles***

In recent years, several driver drowsiness detection systems have been proposed and evaluated using various datasets and machine learning techniques. Overall, the results of these studies demonstrate the potential of

driver drowsiness detection systems to improve road safety by alerting drivers to take a break or rest when signs of drowsiness are detected. However, further research is needed to develop more robust and reliable systems that can handle real-world driving conditions and variations in driver behavior.

# **CHAPTER 7**

# **CONCLUSION**

Driver drowsiness detection systems are a vital tool in preventing accidents caused by driver fatigue. The conclusion of these systems is that they are highly effective in detecting drowsiness and alerting drivers to take a break or rest before an accident occurs. By monitoring various physiological and behavioural parameters such as eye movements, head position, and facial expressions, these systems can detect when a driver is becoming drowsy and alert them to take action. One of the key benefits of these systems is their ability to operate in real-time, providing immediate feedback to drivers. This enables them to respond quickly to changes in the driver's behaviour and prevent accidents before they happen. Additionally, these systems can be customized to fit different types of vehicles and driving conditions, making them highly versatile and adaptable. Another significant advantage of driver drowsiness detection systems is their ability to integrate with other safety features, such as automatic braking and lane departure warnings. This can help create a more comprehensive safety system that can detect and respond to various road hazards. Driver drowsiness detection systems are also an effective way to reduce the financial and human cost of accidents caused by driver fatigue.

These accidents can lead to property damage, injury, and even loss of life. By preventing these accidents, driver drowsiness detection systems can save lives, reduce healthcare costs, and prevent lost productivity. Drowsiness detection systems are a crucial tool in promoting road safety. They are effective in detecting drowsiness and alerting drivers to take a break or rest before an accident occurs. They are versatile and adaptable, and can be integrated with other safety features to create a comprehensive safety system. Driver drowsiness detection systems can also save lives and prevent financial losses caused by accidents. Overall, the use of driver drowsiness detection systems is a crucial step towards creating a safer and more efficient transportation system for all.

**7.1 FUTURE WORK**

The future of drowsiness detection systems looks promising, with several potential work sources being implemented. One possible direction is the use of machine learning and artificial intelligence algorithms to improve the accuracy of drowsiness detection. By training these systems on large datasets of driver behaviour, they can learn to detect subtle patterns and indicators of drowsiness that may be missed by traditional methods. Another area of development is the integration of biometric sensors that can measure additional physiological parameters such as heart rate variability, skin conductance, and brain activity. By combining these measurements with other indicators of drowsiness, such as eye movements and head position, drowsiness detection systems can become even more accurate and reliable.

The development of more advanced camera and sensor technologies could also improve the performance of drowsiness detection systems. For example, high-resolution cameras with infrared capabilities could enable more accurate detection of eye movements and facial expressions, even in low-light conditions. In addition, the integration of drowsiness detection systems with other advanced driver assistance systems, such as adaptive cruise control and lane keeping assist, could further enhance their effectiveness in preventing accidents caused by driver fatigue. Overall, the future of drowsiness detection systems looks bright, with numerous potential work sources for improving their accuracy and reliability. As technology continues to advance, it is likely that these systems will become even more sophisticated and effective in promoting road safety.

**7.2 RESEARCH ISSUES**

1. **Accuracy**:

Driver drowsiness detection systems is to improve the accuracy of the system. This can be achieved by developing more sophisticated algorithms that can detect even subtle signs of drowsiness.

1. **Data Collection**:

Collecting high-quality data is crucial for the development of a robust driver drowsiness detection system. Researchers need to develop new methods for collecting data that is representative of real-world driving scenarios.

1. **Sensor Selection**:

Researchers need to identify the most appropriate sensors that can detect the physiological and behavioral changes associated with driver drowsiness. These sensors should be non-invasive, comfortable to wear, and easy to use.

1. **Calibration**:

Once the sensors are selected, they need to be calibrated to ensure accurate and consistent measurements. Researchers need to develop new calibration methods that can account for individual differences in physiology and behaviour.

1. **Feature Extraction:**

After collecting the data, researchers need to extract relevant features that can be used to detect driver drowsiness. This requires the development of new algorithms that can automatically extract relevant features from the data.

1. **Machine Learning**:

Researchers can use machine learning algorithms to develop driver drowsiness detection systems. However, these algorithms require large amounts of data to be trained effectively. Therefore, researchers need to develop new methods for collecting and labelling large datasets.

1. **Real-Time Processing:**

Driver drowsiness detection systems need to process data in real-time to provide timely alerts to the driver. Therefore, researchers need to develop new algorithms that can process data quickly and efficiently.

1. **User Interaction**:

Researchers need to consider how drivers will interact with the driver drowsiness detection system. The system needs to be intuitive and easy to use, and it should not distract the driver from the task of driving.

1. **Environmental Factors:**

Environmental factors such as lighting, noise, and temperature can affect the performance of driver drowsiness detection systems. Therefore, researchers need to develop new methods for mitigating the effects of environmental factors.

1. **Integration with Other Systems**:

Driver drowsiness detection systems need to be integrated with other systems such as navigation and entertainment systems. Therefore, researchers need to develop new methods for integrating driver drowsiness detection systems with other systems.

1. **Robustness**:

Driver drowsiness detection systems need to be robust and able to operate under a variety of conditions. Therefore, researchers need to develop new methods for testing and validating the system under different conditions.

1. **Privacy:**

Driver drowsiness detection systems collect sensitive data such as physiological and behavioral data. Therefore, researchers need to develop new methods for protecting the privacy of the driver.

1. **Cost**:

Driver drowsiness detection systems can be expensive to develop and implement. Therefore, researchers need to develop new methods for reducing the cost of the system.

1. **Localization**:

Driver drowsiness detection systems need to be localized for different regions and cultures. Therefore, researchers need to develop new methods for adapting the system to different regions and cultures.

1. **Validation:**

Systems need to be validated to ensure that they are effective at detecting drowsiness to develop new methods for validating the system.

1. **Power Consumption**:

Driver drowsiness detection systems need to have low power consumption to avoid draining the vehicle's battery. Therefore, researchers need to develop new methods for reducing the power consumption of the system.

**7.3 IMPLEMENTATION ISSUES**

1. **Data Collection:**

Collect a representative dataset of images or videos of drivers to train the drowsiness detection model. This data should include images or videos of drivers who are alert as well as those who are drowsy.

1. **Feature Extraction:**

Extracting meaningful features from the collected images or videos that can be used to train the drowsiness detection model. This may involve using techniques such as facial landmark detection, eye tracking, or head pose estimation.

1. **Model Selection:**

Selecting an appropriate machine learning model, such as a convolutional neural network (CNN) or a support vector machine (SVM), to classify the extracted features as indicating drowsiness or alertness.

1. **Model Training:**

Training the selected model on the collected data to learn to accurately classify drowsiness in new images or videos.

1. **Thresholding:**

Setting a threshold value for the output of the trained model that determines whether a driver is classified as drowsy or alert.

1. **Real-time Implementation:**

Implementing the trained model on a device capable of capturing images or videos in real-time, such as a smartphone or a camera embedded in a car.

1. **Integration:**

Integrating the system with other systems in the car, such as an alarm or a notification system, to alert the driver when drowsiness is detected

**REFERENCES: -**

[1] According to G. Li and W.-Y. Chung's study, "Combined EEG-gyroscopet DCS brain-machine interface system for the early treatment of driver fatigue," IEEE Trans. Human-Mach. Syst., vol. 48, no. 1, pp. 50-62, Feb. 2018.

[2] Akash Batra et al., Automated Automobile Speed Regulation with Wireless RF Control, 7(4): April 2018, pp.592-597, ISSN: 2277-9655.

[3] Automobile Speed Control Using Radio Frequency Communication, Volume 13, Issue 1, May 2016, ISSN 2349-9303. K. B. Shanghai and others.

[4] Automated Speed Regulation of Automobile in Limited Places Including RF as well as GSM, by A. Venkatesh and colleagues, Volume 02, No 09, December 2015, p. 875-877.

[5] Bhatti, A., Kamal, M. A., Ali, W., & Ahmed, S. (2020). A review on driver drowsiness detection using smartphone sensors. Computer Communications, 157, 26-39.

[6] "Detection of driver tiredness using a condition-adaptive representation learning framework," S. Park, J. Yu, M. Jeon, and S. Lee. Doi:

10.1109/TITS.2018.2883823, IEEE Trans. Intel. Transp. Syst., vol. 20, no. 11, pages 420–4218, November 2019.

[7] "Driver sleepiness identification based on respiratory signal analysis," F. Guede-Fernández, M. Fernández-Chimeno, J. Ramos-Castro, and M. A. Garca-González. doi:10.1109/ACCESS.2019.2924481, IEEE Access, vol. 7, p. 81826–81388, 2019.

[8] Driver Drowsiness Recognition Using 3D Conditional GAN with Two-Level Attention Bi-LSTM was a paper written by Y. Hu, M. Lu, C. Xie, and X. Lu that was published in IEEE Trans. Circuits Syst. Video Technol., vol. 30, no. 12, Dec. 2020, pp. 4755-4768.

[9] Liu, X., Guo, Z., & Zhang, L. (2020). A review of driver drowsiness detection methods based on vision techniques. IEEE Access, 8, 136366-136380.

[10] R. Ashok Kumar, "A Beacon Based Automated Vehicle Speed Control System for Limited Zone," Volume 7, No. X, October 2018, pp. 1152-1159.

[11] R. Deepa's article "Design of Vehicle Speed Control System Utilizing Wireless Instrument Cluster," Volume 4, Issue 1, January 2015.

[12] Sharma, A., & Gupta, R. (2021). A review on driver drowsiness detection techniques. SN Computer Science, 2(4).

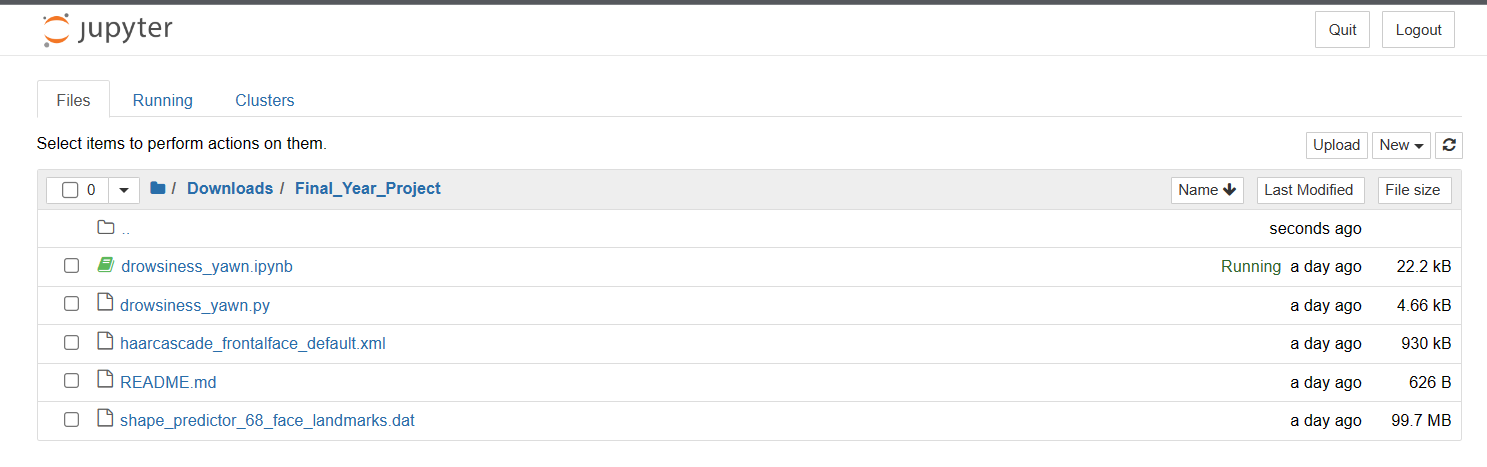
[13] Y. Saito, M. Itoh, and T. Inagaki conducted the study under the title "Driver assistance system with a dual control scheme: Effectiveness of detecting driver fatigue and reducing lane departure accidents." IEEE Trans. Human-Mach. Syst., vol. 46, no. 5, Oct. 2016, pp. 660–671. 10.1109/THMS.2016.2549032 is the ISBN.

**APPENDIX**

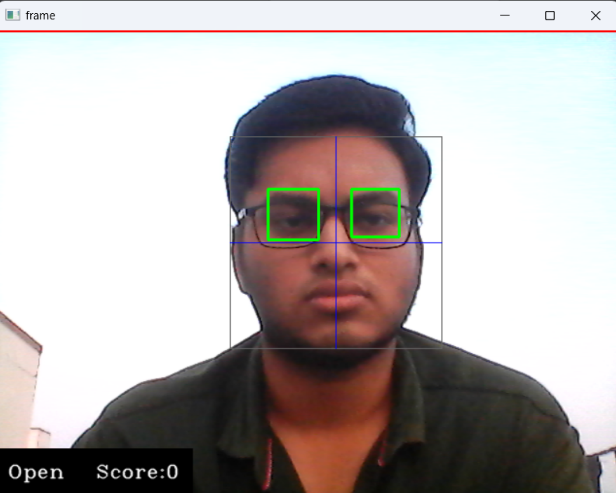
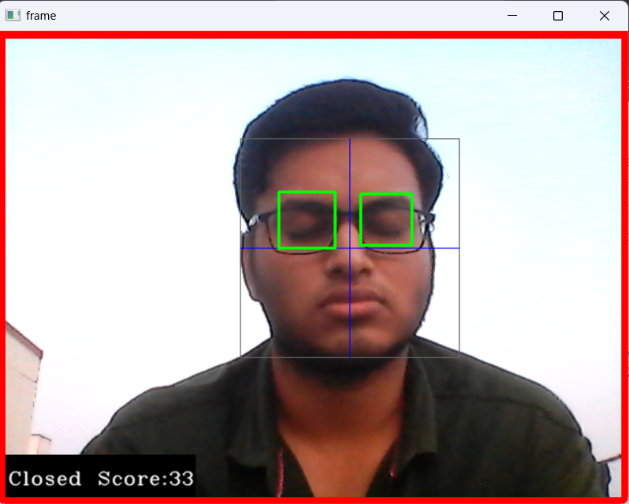
1. **SOURCE CODE**

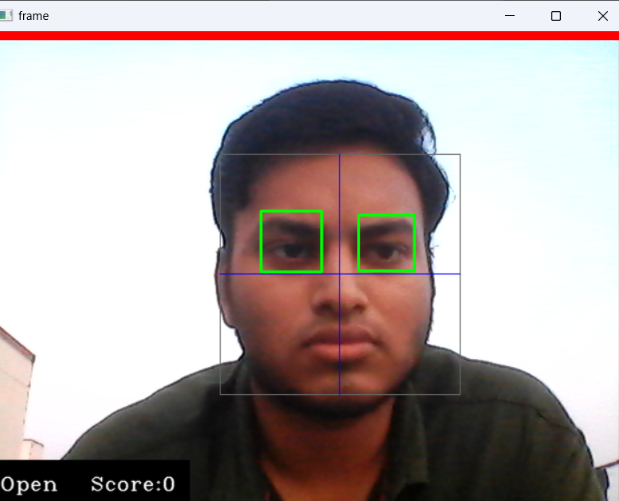
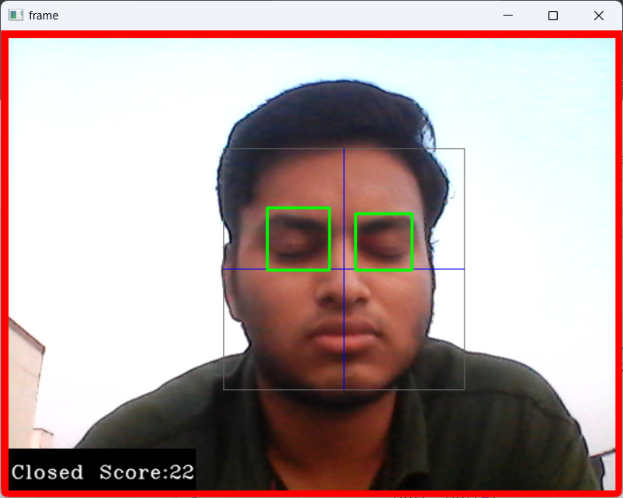
import cv2  
import os  
from keras.models import load\_model  
import numpy as np  
from pygame import mixer  
import time  
  
mixer.init()  
sound = mixer.Sound('alarm.wav')  
  
face = cv2.CascadeClassifier('haar cascade files\haarcascade\_frontalface\_alt.xml')  
leye = cv2.CascadeClassifier('haar cascade files\haarcascade\_lefteye\_2splits.xml')  
reye = cv2.CascadeClassifier('haar cascade files\haarcascade\_righteye\_2splits.xml')  
  
lbl=['Close','Open']  
  
model = load\_model('models/cnncat2.h5')  
path = os.getcwd()  
cap = cv2.VideoCapture(0)  
font = cv2.FONT\_HERSHEY\_COMPLEX\_SMALL  
count=0  
score=0  
thicc=2  
rpred=[99]  
lpred=[99]  
  
while(True):  
    ret, frame = cap.read()  
    height,width = frame.shape[:2]  
  
    gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)  
     
    faces = face.detectMultiScale(gray,minNeighbors=5,scaleFactor=1.1,minSize=(25,25))  
    left\_eye = leye.detectMultiScale(gray)  
    right\_eye =  reye.detectMultiScale(gray)  
  
    cv2.rectangle(frame, (0,height-50) , (200,height) , (0,0,0) , thickness=cv2.FILLED )  
  
    for (x,y,w,h) in faces:  
        cv2.rectangle(frame, (x,y) , (x+w,y+h) , (100,100,100) , 1 )  
        cv2.line(frame,(x+int(w/2),y),(x+int(w/2),y+h),(255,0,0),1)  
        cv2.line(frame,(x,y+int(h/2)),(x+w,y+int(h/2)),(255,0,0),1)  
    for (x,y,w,h) in right\_eye:  
        r\_eye=frame[y:y+h,x:x+w]  
        count=count+1  
        r\_eye = cv2.cvtColor(r\_eye,cv2.COLOR\_BGR2GRAY)  
        cv2.rectangle(frame,(x,y),(x+w,y+h),(0,255,0),2)  
        r\_eye = cv2.resize(r\_eye,(24,24))  
        r\_eye= r\_eye/255  
        r\_eye=  r\_eye.reshape(24,24,-1)  
        r\_eye = np.expand\_dims(r\_eye,axis=0)  
        rpred = model.predict\_classes(r\_eye)  
        if(rpred[0]==1):  
            lbl='Open'  
        if(rpred[0]==0):  
            lbl='Closed'  
        break  
  
    for (x,y,w,h) in left\_eye:  
        l\_eye=frame[y:y+h,x:x+w]  
        count=count+1  
        l\_eye = cv2.cvtColor(l\_eye,cv2.COLOR\_BGR2GRAY)  
        cv2.rectangle(frame,(x,y),(x+w,y+h),(0,255,0),2)  
        l\_eye = cv2.resize(l\_eye,(24,24))  
        l\_eye= l\_eye/255  
        l\_eye=l\_eye.reshape(24,24,-1)  
        l\_eye = np.expand\_dims(l\_eye,axis=0)  
        lpred = model.predict\_classes(l\_eye)  
        if(lpred[0]==1):  
            lbl='Open'    
        if(lpred[0]==0):  
            lbl='Closed'  
        break  
  
    if(rpred[0]==0 and lpred[0]==0):  
        score=score+1  
        cv2.putText(frame,"Closed",(10,height-20), font, 1,(255,255,255),1,cv2.LINE\_AA)  
    # if(rpred[0]==1 or lpred[0]==1):  
    else:  
        score=score-1  
        cv2.putText(frame,"Open",(10,height-20), font, 1,(255,255,255),1,cv2.LINE\_AA)  
  
    if(score<0):  
        score=0    
    cv2.putText(frame,'Score:'+str(score),(100,height-20), font, 1,(255,255,255),1,cv2.LINE\_AA)  
  
    if(score>15):  
    #person is feeling sleepy so we beep the alarm  
        cv2.imwrite(os.path.join(path,'image.jpg'),frame)  
        try:  
            sound.play()  
         
        except:  # isplaying = False  
            pass  
     
        if(thicc<16):  
            thicc= thicc+2  
        else:  
            thicc=thicc-2  
            if(thicc<2):  
                thicc=2  
    cv2.rectangle(frame,(0,0),(width,height),(0,0,255),thicc)  
    cv2.imshow('frame',frame)  
    if cv2.waitKey(1) & 0xFF == ord('q'):  
        break  
cap.release()  
cv2.destroyAllWindows()

**B. SCREEN SHOTS**

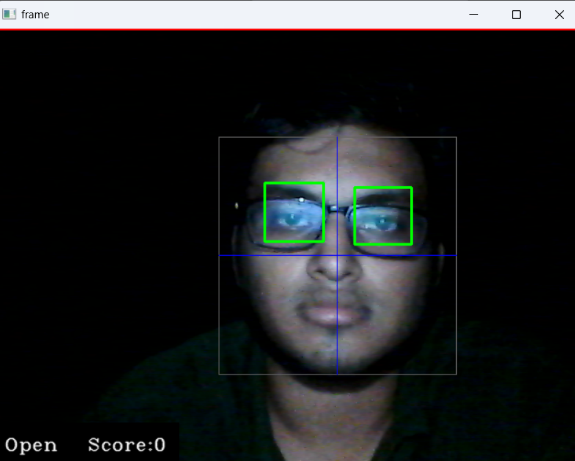
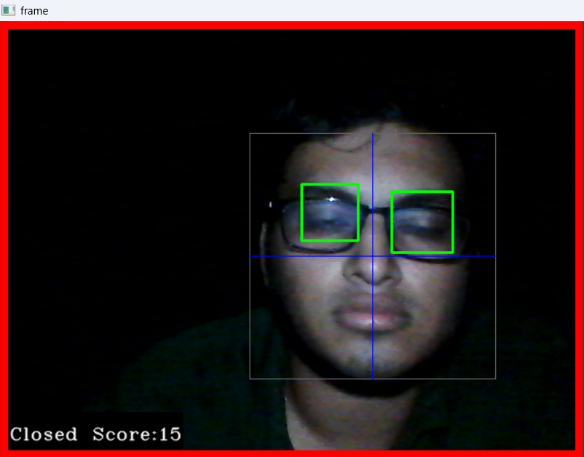


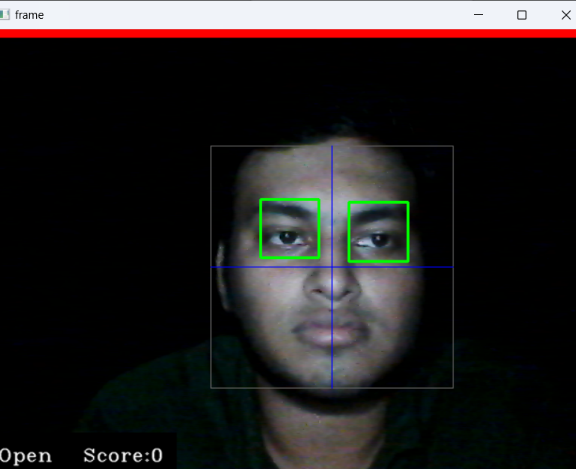
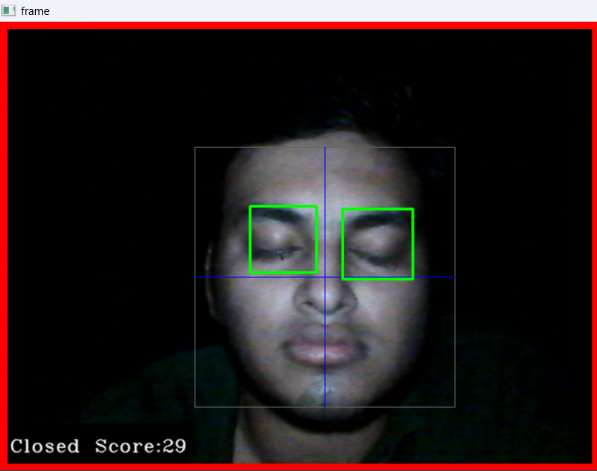
**Fig 1: Required files**

**Fig 2 : Day time detection with/without spectacles**

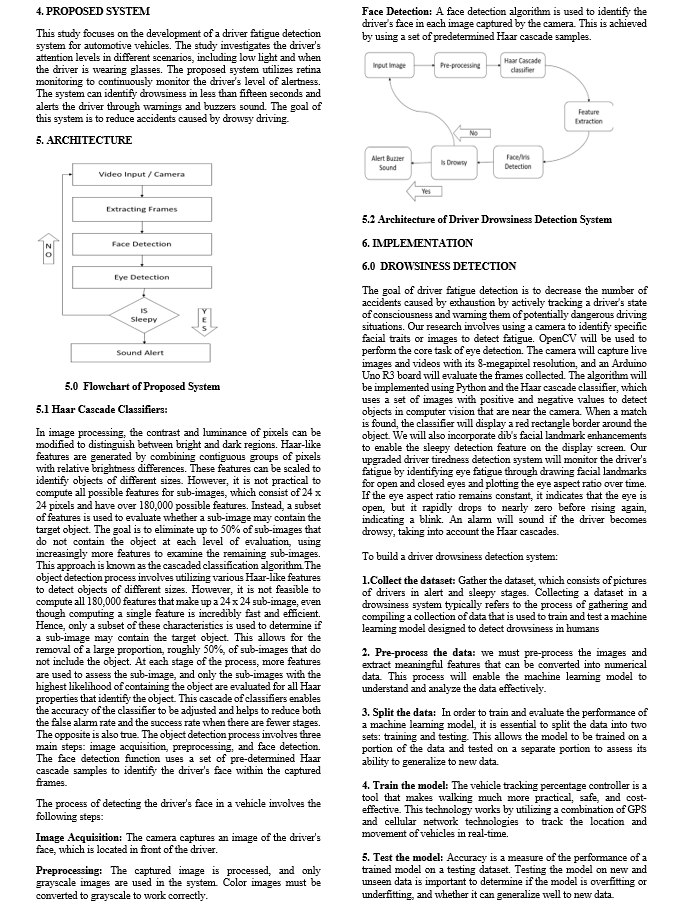
 

**Fig 3 : Night time detection with/without spectacles**

**C. RESEARCH PAPER**







1. **PLAGIARISM REPORT**

