JSS MAHAVIDYAPEETHA
JSS SCIENCE AND TECHNOLOGY UNIVERSITY
JSS Technical Institutions Campus, Mysuru – 570006

"DRIVER MONITORING SYSTEM"

Mini project report submitted in partial fulfillment of curriculum prescribed for the Artificial Intelligence (20CS540) course for the award of the degree of

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING

by

ABHISHEK MB - 01JCE21CS003
SINCHANA S - 01JCE21CS101
ANAGHA KP - 01JCE21CS010
KIRTANA KIRAN - 01JST21CS061
BHOOMIKA KS - 01JCE21CS019
LAXMI R PATIL - 02JST22UCS403

Under the Guidance of

Dr. M P Pushpalatha
Professor,
Dept.of CS & E,
SJCE, JSS STU Mysore

ACKNOWLEDGEMENT

We are profoundly grateful to Dr M P PUSHPALATHA for her expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion.

ABHISHEK M B
SINCHANA S
KIRTANA KIRAN
BHOOMIKA K S
ANAGHA K P
LAXMI R PATIL

ABSTRACT

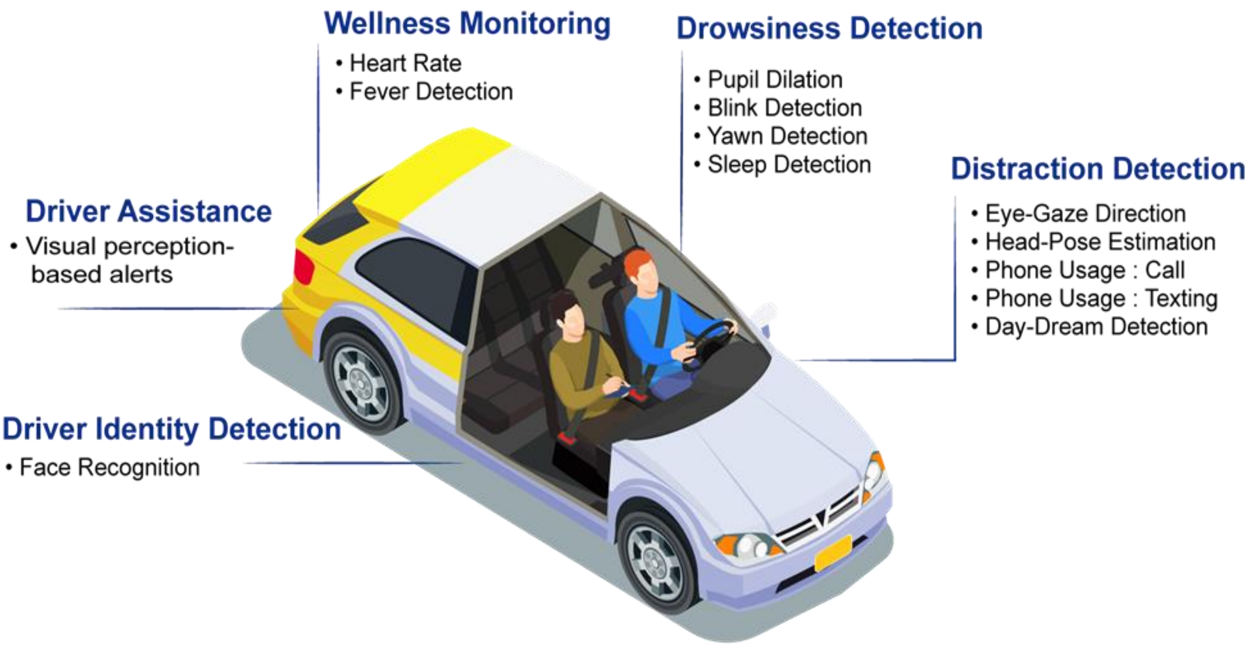
This document is a review report on the research conducted and the project made in the field of computer engineering to develop a system for driver drowsiness detection to prevent accidents from happening because of driver fatigue and sleepiness. The report proposed the results and solutions on the limited implementation of the various techniques that are introduced in the project. Whereas the implementation of the project give the real world idea of how the system works and what changes can be done in order to improve the utility of the overall system. Furthermore, the paper states the overview of the observations made by the authors in order to help further optimization in the mentioned field to achieve the utility at a better efficiency for a safer road

TABLE OF CONTENTS

1.INTRODUCTION

In an era marked by rapid technological advancements, the integration of artificial intelligence has transformed various aspects of our lives. One particularly crucial application of AI technology is the development of Driver Monitoring Systems (DMS), which are at the forefront of enhancing road safety and revolutionizing the way we interact with vehicles. This AI-driven project aims to create an innovative Driver Monitoring System that leverages the power of artificial intelligence to address critical issues related to driver attention, fatigue, and distraction.

DMS intent to provide information about driver and their behaviour. One of the biggest challenges in developing driver monitoring systems is the complexity of human behavior. Drivers can exhibit a wide range of behaviors, and the system must be able to distinguish between normal driving behaviors and potentially dangerous ones. This requires using advanced COMPUTER VISION and MACHINE LEARNING algorithms to detect and accurately classify the driver's behavior in real-time. By solving these challenges, driver monitoring systems can enable safe and reliable autonomous driving.



1.1 PURPOSE

The purpose of a Driver Monitoring System (DMS) for an AI project is multi-faceted and encompasses a range of crucial objectives aimed at improving road safety, driver well-being, and the overall driving experience. Here are some of the key purposes of implementing a DMS in an AI project:

1. Enhance Road Safety: The primary purpose of a DMS is to improve road safety. By continuously monitoring the driver's behavior, the system can detect signs of distraction, drowsiness, sleepy, active, and issue timely alerts to mitigate potential accidents. This proactive approach can significantly reduce the risk of accidents caused by human error, which is a leading factor in road accidents.
2. Prevent Accidents: The DMS's ability to detect driver inattention or fatigue allows it to intervene before an accident occurs. This early warning system can help prevent collisions, especially in situations where a driver's delayed reaction could lead to disaster.
3. Monitor Driver Fatigue: Fatigue is a major cause of accidents, particularly in long-distance or monotonous driving scenarios. The DMS helps identify signs of driver fatigue, such as drooping eyelids or erratic driving behavior, and prompts the driver to take a break, ensuring they stay alert and rested during their journey.
4. Improve Driver Well-Being: Beyond safety, DMS can contribute to the well-being of drivers by encouraging healthy driving habits. By monitoring and alerting drivers about their inattentiveness or drowsiness, the system promotes better driving practices and discourages risky behaviors.
5. Customized Alerts: The DMS can be personalized to provide alerts that suit the driver's specific needs. For example, it can alert a driver to take a break if it detects drowsiness, provide navigation assistance, or even offer suggestions for driving style improvement.

6. Data Insights and Analytics: The data collected by the DMS can be analyzed to gain valuable insights into driver behavior and patterns. This data can be used by fleet operators, insurance companies, and other stakeholders to assess and improve driver performance and safety standards.

8. Insurance Premium Reduction: Some insurance companies offer discounts to drivers who have DMS installed in their vehicles. The purpose here is to encourage the adoption of safer driving practices and to reward policyholders for their commitment to road safety.

9. Vehicle Security: DMS can also play a role in vehicle security by detecting unauthorized access to the vehicle and notifying the owner or authorities.

1.1.1 HUMAN PSYCHOLOGY WITH CURRENT TECHNOLOGY

Our world-leading Driver Monitoring System (DMS) technology is underpinned by a deep understanding of human behaviour. Kyle Wilson PhD explains the important role Human Factors plays in developing and enhancing our life-saving technology.

The two most recognised types of DMS are those that detect whether drivers' hands are on the steering wheel and those that use cameras to measure driver attention and state. The former tends to use torque sensors or capacitive sensors to infer whether there are hands on the wheel, while the latter use cameras to track head and eye movements that are associated with unsafe driver states.

Understanding human psychology and integrating it with current technology is of paramount importance.

1. Attention and Distraction:

- Psychology: Humans can easily become distracted while driving due to a range of factors, including smartphones, passengers, or external stimuli.
- Technology: DMS utilizes computer vision and machine learning algorithms to track the driver's eye movements, head orientation, and facial expressions. By understanding these patterns, the system can detect moments of distraction and provide real-time feedback or warnings to refocus the driver's attention on the road.

2. Alertness and Fatigue:
- Psychology: Human alertness tends to decrease with time, especially during long journeys or late hours. Fatigue can lead to slower reaction times and impaired decision-making.
- Technology: DMS employs sensors like infrared cameras to monitor for signs of drowsiness, such as drooping eyelids and erratic steering. When indicators of fatigue are detected, the system can issue auditory or visual alerts to keep the driver awake and attentive.

3. Emotional State:
- Psychology: A driver's emotional state can significantly impact their driving behavior. Stress, anger, or anxiety can lead to aggressive driving, while a relaxed state can result in safer driving.
- Technology: Advanced DMS can assess emotional states through facial expression analysis and biometric data. By recognizing emotional changes, the system can adapt its alerts or offer interventions to help the driver manage their emotions and maintain safe driving habits.

4. Habitual Behavior:
- Psychology: Drivers often exhibit habitual behaviors, such as aggressive driving, tailgating, or speeding, which may put them at risk.
- Technology: DMS can track and analyze habitual behavior, generating insights to promote safer driving practices. It can provide feedback and coaching to encourage drivers to adopt safer habits over time.

5. Cognitive Load:

- Psychology: Drivers have limited cognitive capacity, and tasks like texting or using a smartphone can increase cognitive load, diverting attention from the road.
- Technology: DMS can gauge cognitive load by monitoring interactions with in-car technology. It can restrict certain tasks or provide warnings when cognitive load is excessively high, ensuring that the driver remains focused on driving.

6. Privacy and Trust:

- Psychology: Drivers may have concerns about the privacy implications of constant monitoring by DMS.
- Technology: To address these concerns, DMS technology must be transparent in terms of data collection and privacy. It should allow drivers to opt-in or opt-out and assure them that their data is used solely for safety purposes, respecting their trust and privacy.

7. User Experience:

- Psychology: A positive user experience is crucial for driver acceptance and adoption of DMS technology.
- Technology: DMS should provide a user-friendly experience with non-intrusive alerts and a clear interface. It should aim to support and assist the driver, enhancing their safety and overall driving experience.

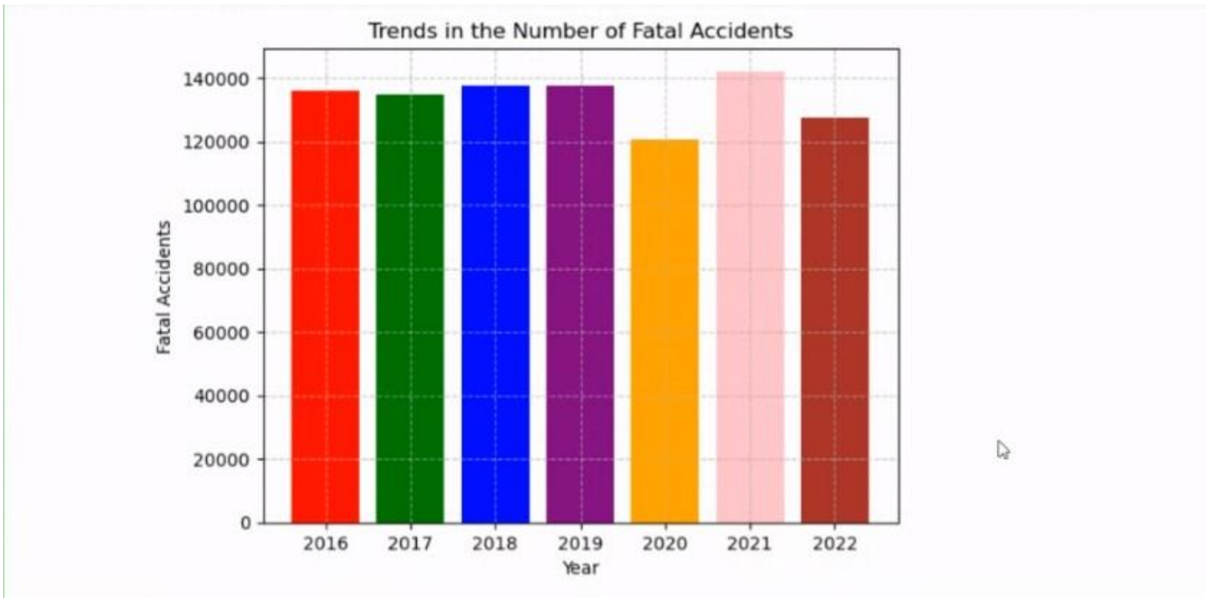
1.1.2 FACTS AND STATISTICS:

Road safety is a significant public health issue, and a cause of injuries and fatalities. According to a report by the Ministry of Road Transport and Highways Transport Research Wing, road accidents claimed 1,53,972 lives and harmed 3,84,448 people in 2021. Unfortunately, the age range that is most severely hit by road accidents is 18 to 45 years old, which accounts for almost 67 percent of all accidental deaths.

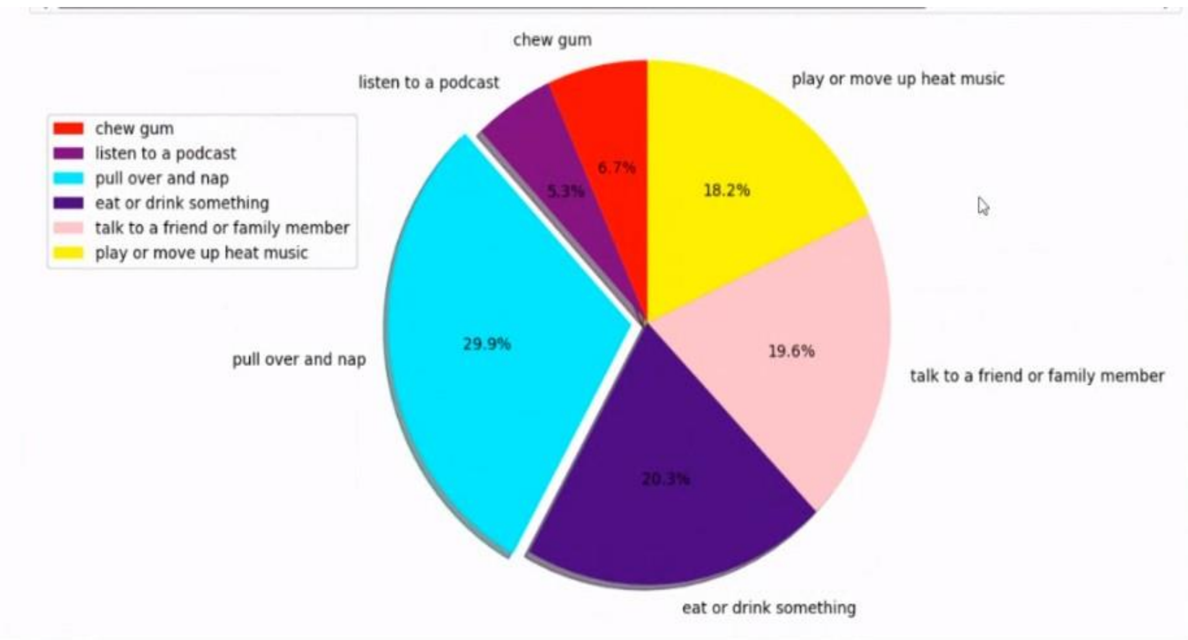
A crucial problem that causes numerous car accidents annually is driver fatigue. Due to the incapacity of a driver to halt or swerve to prevent or minimise the impact, accidents caused by driver sleepiness are much more inclined to result in fatalities or severe accidents. Fatigue lowers attentiveness, alertness, and concentration, which impairs the accomplishment of tasks requiring attention such as driving.

Most drivers understand the dangers of drinking and driving and texting and driving, but many people underestimate the dangers of drowsy driving. Each year, drowsy driving accounts for about 100,000 crashes, 71,000 injuries and 1,550 fatalities, according to the National Safety Council (NSC).

From 2013 to 2017, more than 4,000 people died due to drowsy driving. Drowsy driving is so dangerous because it mirrors so many sypmtoms of drunk driving: blurred vision, slowed reaction time, and poor decision making. From 2006-2016, the NHTSA reported more than 10,000 deaths from drunk driving collisions and impaired drivers. While it is more common in drivers who travel long distances, drowsiness or sleepiness can happen to anyone. Even the most experienced drivers can be susceptible and miss the warning signs that it's no longer safe to be behind the wheel of a motor vehicle. If your vision begins to blur, you nod off, or lose control of the car, you should pull over immediately and sleep.100,000 police-reported crashes and over 1,500 deaths are the results of drowsy driving each year. (NHTSA) More than 40% of drivers admitted they have fallen asleep behind the wheel. (AAA)The cost of drowsy-driving crashes at about 13% of the total $836 billion in societal costs of traffic crashes. (NHTSA) An estimated 1,550 deaths, 71,000 injuries, and $12.5 billion in monetary losses are due to drowsy drivers. (NHTSA) In 2017, 50,000 people were injured in drowsy driving accidents and 795 of those were killed because of drowsy driving. (NHTSA)800 fatalities occurred in 2015 as a result of drivers feeling fatigued behind the wheel. (FHWA)From 2009 to 2013, 72,000 police-reported motor vehicle accidents involving drowsy drivers. (NHTSA)



The most effective ways at keeping a person awake if they fall asleep while driving(based on the survey conducted on 1500 people)



1.2 PROJECT SCOPE:

A driver monitoring system project scope would typically involve designing and implementing a system to track and analyze driver behavior for safety and performance.

Facial Recognition: Implementing technology to recognize the driver's face for identification.

Eye Tracking: Monitoring eye movement to detect drowsiness or distraction.

Head Movement Analysis: Analyzing head gestures to identify signs of inattention or fatigue.

Heart Rate Monitoring: Integrating a system to measure the driver's heart rate for stress detection.

Real-time Alerts: Implementing alerts for unsafe behavior, such as sending notifications for drowsy driving.

Data Logging: Storing and managing data for future analysis and reporting.

Integration with Vehicle Systems: Connecting the monitoring system with the vehicle's technology for seamless operation.

User Interface: Developing a user-friendly interface for both drivers and system administrators.

Compliance with Regulations: Ensuring the system aligns with relevant safety and privacy regulations.

Testing and Validation: Conducting thorough testing to ensure the accuracy and reliability of the monitoring system.

Scalability: Designing the system to be scalable for integration with different vehicle models and types.

Documentation: Providing comprehensive documentation for users and future developers.

Privacy Considerations: Addressing privacy concerns and implementing measures to protect driver data.

Maintenance and Support: Establishing plans for ongoing system maintenance and user support.

1.3 PROBLEM DEFINITION:

A Driver Monitoring System (DMS) project involves creating a technology that monitors and analyzes a driver's behavior to enhance safety. It typically includes features like facial recognition, eye tracking, object detection, gaze detection and other sensors to detect signs of drowsiness or distraction. The project's definition outlines the system's goals, functionality, and specifications, aiming to improve road safety by alerting or intervening when potential risks are identified. The focus is on developing a reliable and efficient solution to mitigate the impact of driver-related issues on road accidents.

2. LITERATURE SURVEY

2.1 SYSTEM REVIEW

The Driver Monitoring System (DMS) employing Euclidean distance as a key metric is designed to enhance driver safety through precise analysis of facial features and movements. The system excels in accurate feature extraction, particularly in determining facial landmarks and eye positioning. Euclidean distance calculations are efficiently performed in real-time, allowing the system to promptly assess and monitor driver behavior. The incorporation of Euclidean distance metrics in alert threshold setting enables a balanced approach, minimizing both false positives and negatives. The system showcases adaptability through individual calibration and the ability to adjust calculations based on varying conditions. Its effectiveness in detecting drowsiness and distraction, evidenced through Euclidean distance analyses, underscores its potential to improve overall driver safety. Seamless integration with other vehicle systems and a user-friendly interface further contribute to the system's robustness, making it a valuable tool for enhancing driver monitoring capabilities.

A system review for a Driver Monitoring System (DMS) using Euclidean distance would focus on the application of this mathematical measure in the system's functionality. Euclidean distance is often used in computer vision and machine learning for comparing the similarity between feature vectors. Here's a structured review based on this approach:

1. Feature Extraction:

   - Facial Landmarks: Evaluate how well the DMS extracts facial landmarks for each driver. The Euclidean distance between corresponding landmarks can be used to measure facial expressions and movements.

   - Eye Positioning: Assess the accuracy of determining eye positions using Euclidean distance. A precise measurement is essential for monitoring gaze direction and detecting signs of distraction or drowsiness.

2. Euclidean Distance Calculation:

   - Algorithm Accuracy: Evaluate the algorithm used to calculate Euclidean distance. It should provide accurate distance measurements between relevant facial features.

   - Real-time Processing: Assess the system's efficiency in calculating Euclidean distances in real-time to ensure timely monitoring and response.

3. Threshold Setting:

   - Alert Thresholds: Review how the system sets and adjusts alert thresholds based on Euclidean distances. It should strike a balance between sensitivity and specificity to minimize false positives and negatives.

4.Driver Behavior Analysis:

   -Drowsiness Detection: Evaluate how effectively the system uses Euclidean distance to detect signs of drowsiness, such as slow eye movements or drooping eyelids.

-Distraction Detection: Assess the system's ability to identify signs of distraction by analyzing facial features and head movements using Euclidean distance.

5. Adaptability and Calibration:

- Individual Calibration: Check if the DMS allows for individual calibration based on the unique facial characteristics of each driver. Euclidean distance calculations may need to be adjusted for different individuals.

- Adaptation to Conditions: Evaluate how well the system adapts Euclidean distance calculations to varying lighting conditions and driver positions.

6. Alert Mechanisms:

-Euclidean Distance Threshold Alerts: Analyze how the system communicates alerts to the driver when certain Euclidean distance thresholds are exceeded. Consider the effectiveness of these alerts in capturing the driver's attention without causing unnecessary disruptions.

7. Integration and Compatibility:

- Compatibility with Other Systems: Ensure that the DMS, relying on Euclidean distance, can integrate seamlessly with other vehicle systems and platforms, such as telematics or fleet management systems.

8. Accuracy and Validation:

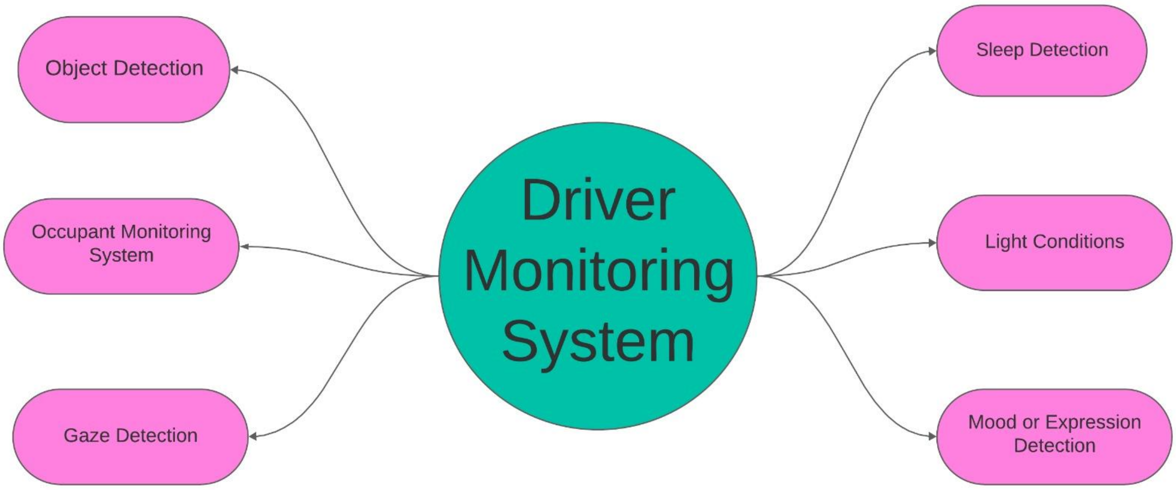
- Validation Studies: Look for validation studies or tests conducted to assess the accuracy of the Euclidean distance-based DMS. Consider real-world scenarios and diverse driving conditions.

9. Usability and User Interface:

- Display of Euclidean Metrics: Evaluate how the Euclidean distance metrics are presented in the user interface, whether for real-time monitoring or post-analysis. The information should be clear and easily interpretable.

10. Security and Privacy:

-Data Encryption: Ensure that Euclidean distance calculations and the associated data are treated with appropriate encryption to safeguard sensitive information.



2.2 DRIVER MONITORING SYSTEM VALIDATION:

The validation of a DMS must take into account quite a large set of technical and human factors. While technical issues can be evaluated rather objectively and straightforwardly, evaluation of human factors is more complex. Moreover, the diversity of factors that impact DMS performance and validation could lead to an extensive landscape of test scenarios. Taken together, the items listed below give an overview of factors to be taken into account. Though the list is not exhaustive, it aims to provide a better understanding of the challenges and requirements related to the validation of DMS

The driver-capabilities pyramid A driver's ability to handle a specific driving situation depends both on their driving skills and their availability/alertness. The monitoring of the driver's availability/alertness is quite complex for two reasons:

- the driver's availability/alertness depends on several first-order factors – see the graph hereunder – such as their behavior and cognitive states
- these factors depend in turn on other, second-order factors (see below) such as lifestyle, road conditions, and experience. In order to support

applications that enhance safety, comfort, and user experience, the DMS has to provide information about the first-order factors. Indeed, by providing these applications with information on the underlying states that impact the driver's availability and their alertness, they are able to work in a relevant way. From a validation standpoint, the evaluation protocols and the related ground truths must make it possible to distinguish among the first-order factors. Evaluation protocols also require appropriate control of some of the second-order factors. The following graph gives an overview of the factors that can impact driver's capabilities.

Physiological and cognitive states Physiological and cognitive states cannot be simulated in a relevant way by a human being. The evaluation protocol thus has to induce the state in question in a natural way. Physiological and cognitive states are not binary, but progressive. For instance, before a driver falls asleep, they will be in a state somewhere between fully awake and fully drowsy/falling asleep. It is thus quite challenging to define key performance indicators (e.g., sensitivity and specificity) for the ability to specifically detect such states. There is no fully objective single ground truth for each state, and there probably never will be. Thus, no single ground truth will ever meet all the requirements for validating a particular state. For instance, the table below compares four ground truths for wakefulness and drowsiness, based on the following requirements:

• Objective data: the ground truth is based on fully objective data. (In some cases, the ground truth data has to be scored in a subjective way, so the data is almost but not fully objective).

• Direct: the ground truth measures the state directly. Behavioral data serve as indirect measurements of the state.

• Continuous: the ground truth is collected continuously.

• Non-disturbing: the collection of the ground truth does not disturb the driver's tasks or behavior

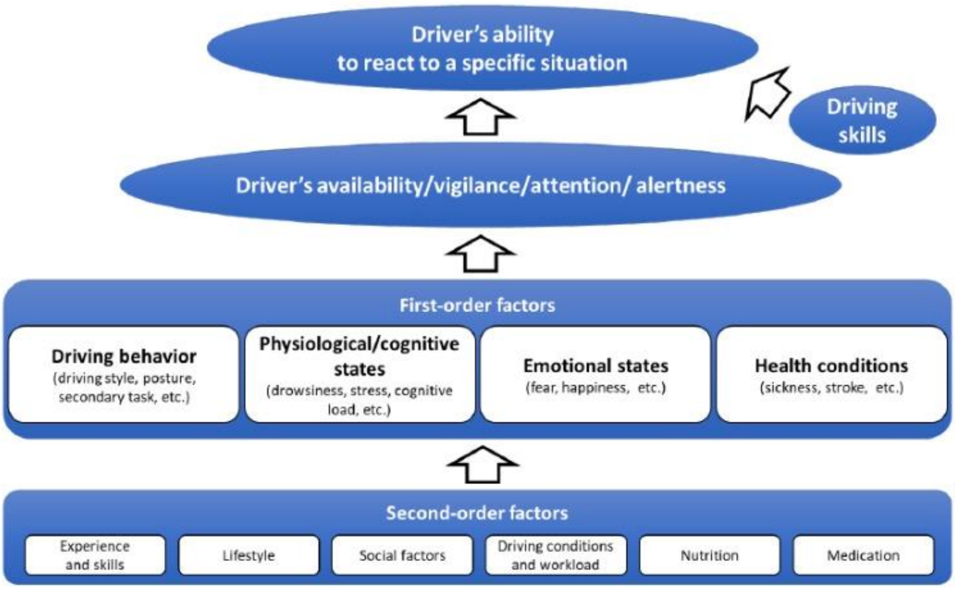
Emotional states The challenges created by emotional states are similar to those created by physiological and cognitive states.

Driver diversity These factors include size, ethnicity, gender, and age; whether the driver is using eyeglasses, has piercings, is wearing make-up, or has a face mask on; and what clothes they are wearing and how their hair is styled. The robustness of a DMS when it comes to driver diversity is a key evaluation criterion for the validation of some DMS features. In these cases, the validation protocol has to ensure a sufficient level of driver diversity in the test scenarios.

Driving conditions Ideally, a DMS should work in all driving conditions. Driving conditions include factors such as luminosity, the quality of any road markings present, weather conditions, traffic conditions, and how much dust there is in the immediate environment or on the sensor. Because driving conditions may affect the sensing modality – the quality and availability of raw data and features – as well as driver behavior, the evaluation protocol has to ensure the validation of the DMS under all relevant conditions.

Sensors and data Depending on the application, one or more sensors and data items are used, such as the driver camera, the angle of the steering wheel, and a seat-pressure sensor. In some cases, one sensor can be used for several applications, while a particular application can be supported by various sensing techniques that are used alone or in combination. Moreover, sensors and data specifications may vary from one DMS supplier to another. Together, these factors lead to the creation of several sensor architectures and data specifications. The evaluation protocol must thus be designed for each sensor architecture and each set of data specifications.

Features and applications A DMS may support several use cases related to safety, comfort, and user experience. Each application has its own requirements in terms of DMS features such as the direction of the driver's gaze, the detection of objects, the detection of drowsiness, and identification of the driver, as well as the specification of each feature. The evaluation protocol must thus evaluate each feature required by the application.



2.3 TECHNOLOGY STACKS:

Object Detection:

Technology: Image sensors, cameras, LiDAR, radar.

Software: Object detection algorithms (e.g., YOLO, SSD, Faster R-CNN).

Integration: Microcontrollers or processors for real-time analysis, communication protocols (e.g., CAN bus).

Occupant Monitoring System (OMS):

Technology: Weight sensors, pressure sensors, cameras, depth sensors.

Software: Machine learning algorithms for occupant detection and recognition.

Integration: Microcontrollers for processing sensor data, communication protocols.

Gaze Detection:

Technology: Infrared cameras, eye-tracking technology.

Software: Gaze estimation algorithms, deep learning models.

Integration: Microcontrollers for real-time analysis, integration with driver monitoring system.

Mood or Expression Detection:

Technology: Cameras, facial recognition technology.

Software: Facial expression analysis algorithms, deep learning models.

Integration: Microcontrollers for real-time analysis, integration with driver monitoring system.

Light Condition Detection:

Technology: Light sensors, ambient light sensors.

Software: Algorithms for analyzing light conditions.

Integration: Microcontrollers for processing sensor data, communication with vehicle systems.

Sleep Detection:

Technology: Cameras, infrared sensors, steering wheel sensors.

Software: Sleep detection algorithms, machine learning models.

Integration: Microcontrollers for real-time analysis, integration with driver monitoring system.

2.4 ALGORITHMS AND FLOWCHARTS

Sleep Detection:

Face Detection

1. Initialize the camera.
2. Load the pre-trained face detection model (e.g., Dlib's frontal face detector).
3. Capture video frames from the camera.
4. Convert each frame to grayscale for efficient processing.
5. Use the face detection model to detect faces in the grayscale frame.
6. For each detected face:

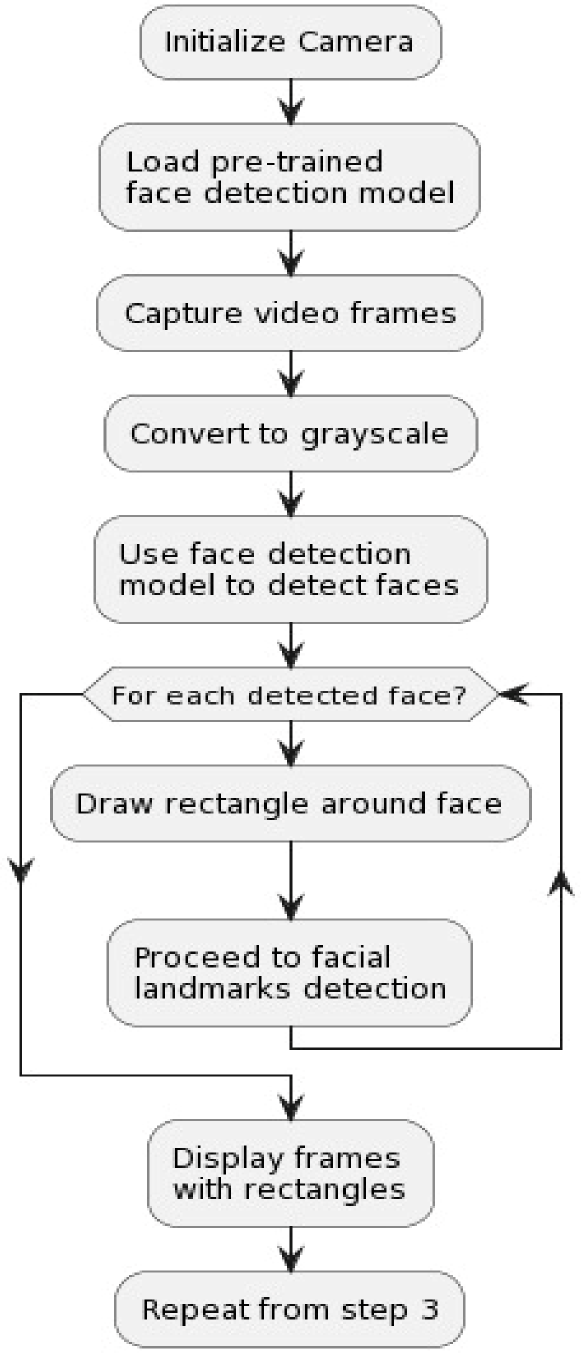
7. Draw a rectangle around the face on the original color frame.

8. Proceed to facial landmarks detection.

9. Display the frames with rectangles around detected faces.

10. Repeat from step 3.

End Algorithm



Facial Landmarks Detection

1. Initialize the facial landmark predictor (e.g., Dlib's shape predictor).

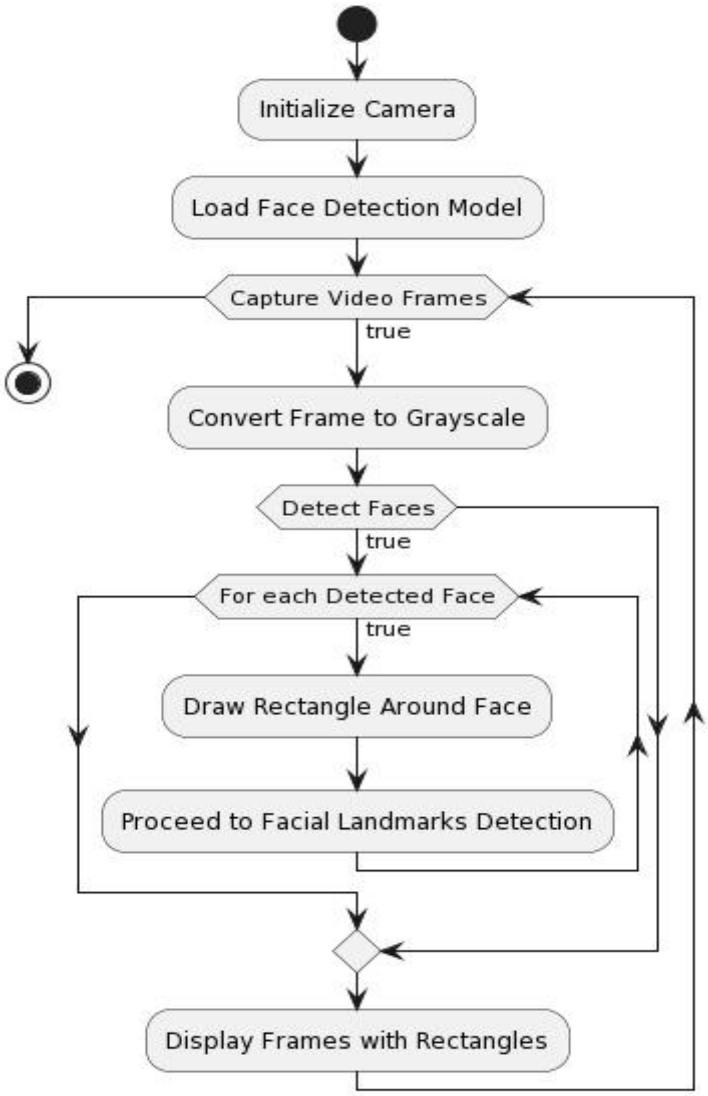
2. For each detected face:

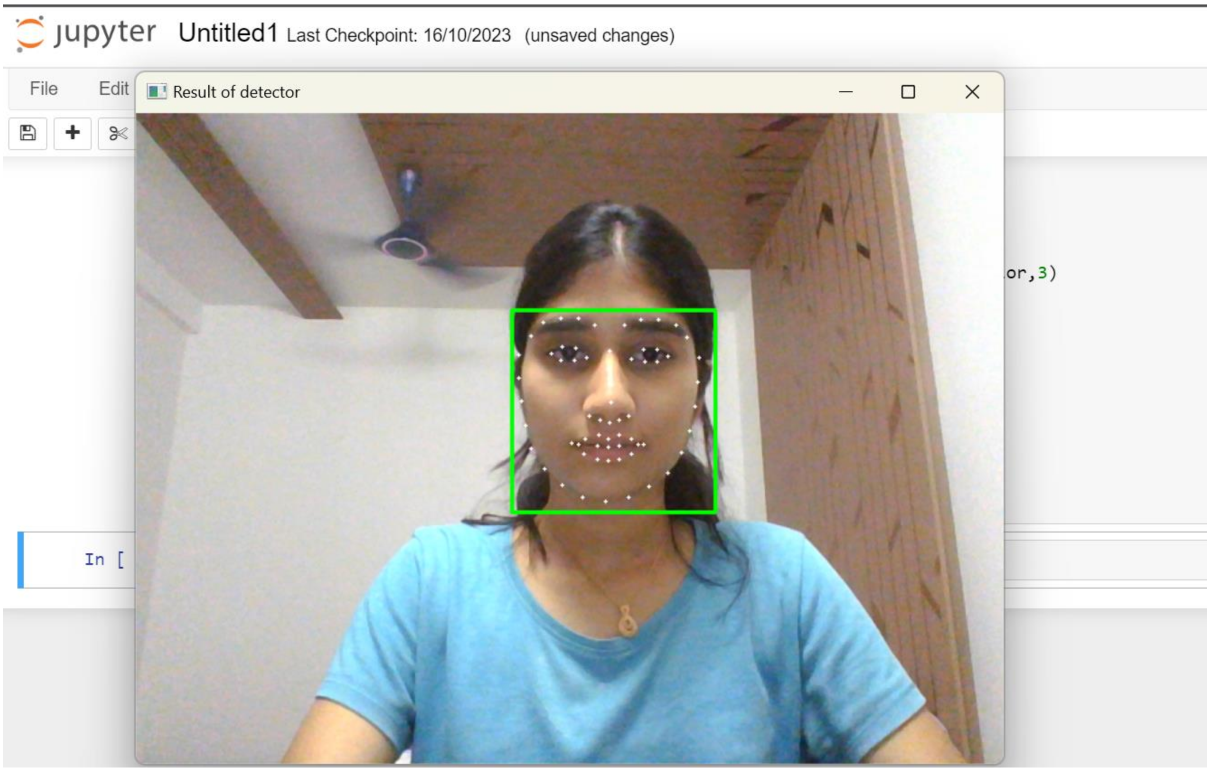
3. Use the facial landmark predictor to identify key points on the face (landmarks).

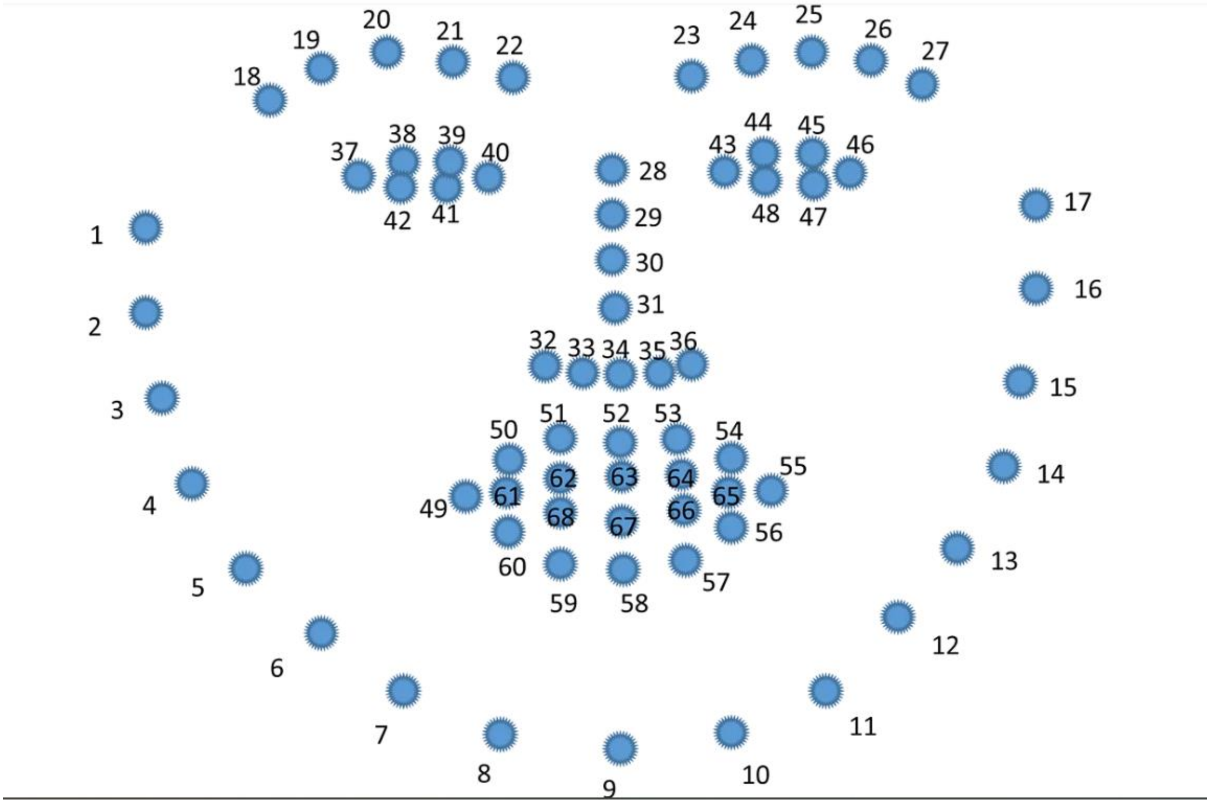
4. Convert the landmarks to NumPy arrays for further processing.

5. Extract specific landmarks for eye blinking analysis (e.g., points around the eyes).

6. End Algorithm




jupyter
Untitled1 Last Checkpoint: 16/10/2023 (unsaved changes)
File
Edit
Result of detector
In [
or,3)



Blink Detection

1. Define a function to calculate the Euclidean distance between two points.

2. Define a function to determine if a blink has occurred based on the ratio of distances between specific facial landmarks.

3. For each frame:

4. Call the blink detection function for both eyes using the relevant landmarks.

5. Based on the detected blinks:

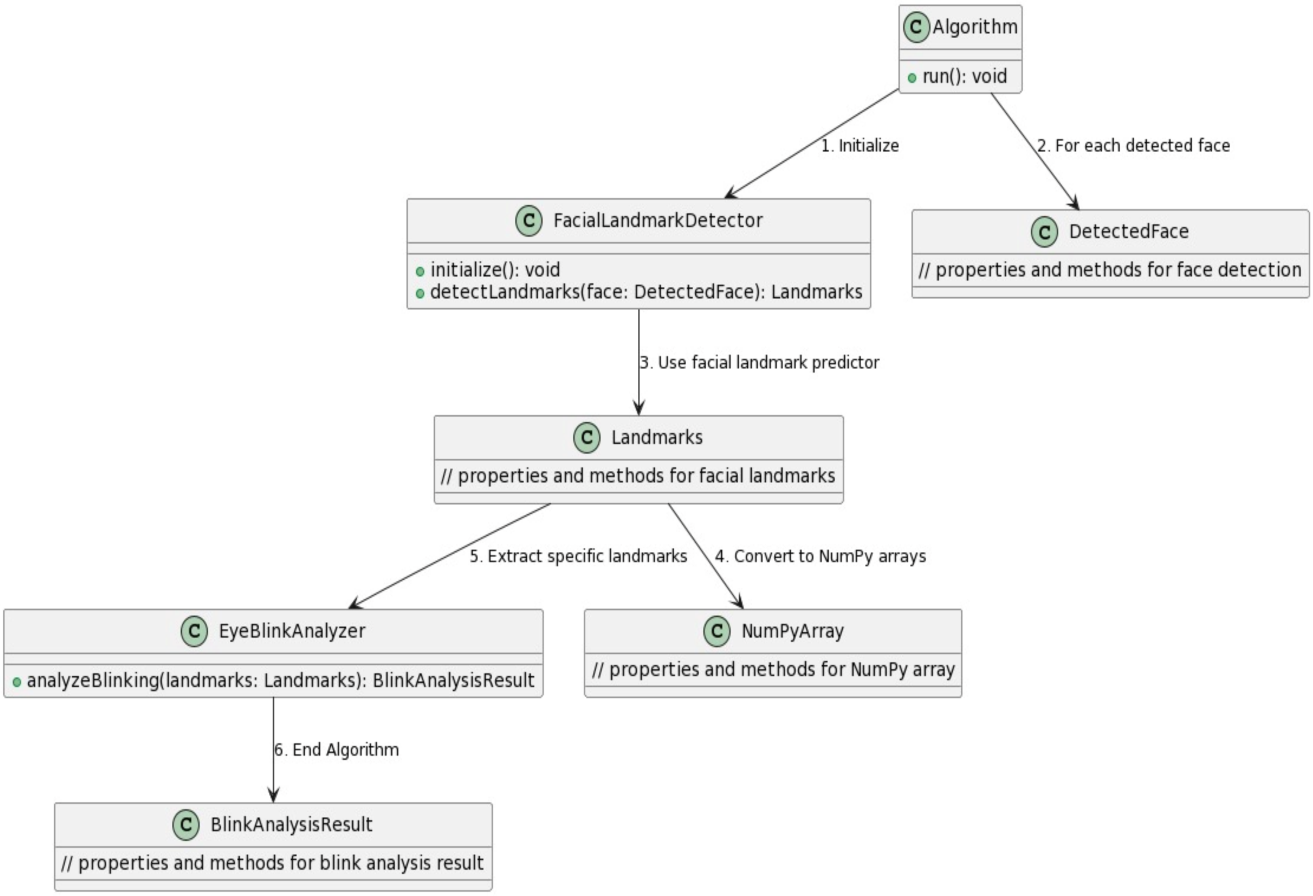
6. Update counters for sleep, drowsy, and active states.

7. If a specific state persists for a predefined threshold:

8. Set the status message and color accordingly (sleeping, drowsy, or active).

9. Display the status message on the frame.

10. End Algorithm



Euclidean distance formula:

$\mathrm{Sum}(D) = w(x_4, y_4) - w(x_2, y_2) - w(x_3, y_3) + w(x_1, y_1)$

$ƛ = P_{A+C} - P_B$
$P_{A+C} = w(0,1) + w(0,4) - w(0,3)$
$P_B = w(0,3) - w(0,1)$
$ƛ = w(0,4) + 2*w(0,1) - 2*w(0,3)$

Euclidean Distance Calculation for Two Points

Given two points P(x1, y1) and Q(x2, y2):

1. Initialize:

   - Set x1, y1, x2, y2 as the coordinates of the two points.

2. Calculate Differences:

   - Compute Δx = x2 - x1 and Δy = y2 - y1.

3. Square and Sum:

   - Compute (Δx)^2 and (Δy)^2.

   - Sum these squared values: s = (Δx)^2 + (Δy)^2.

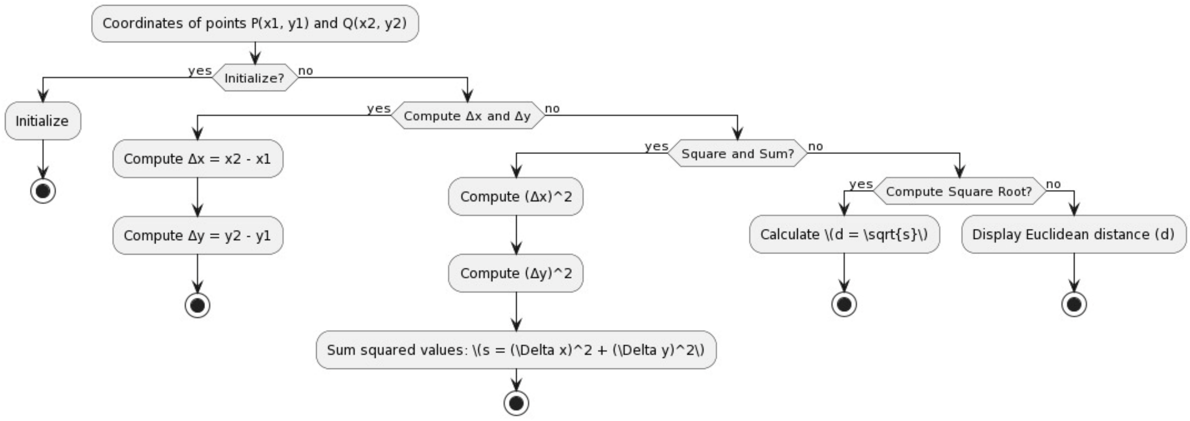
4. Compute Square Root:

   - Calculate the square root of s: d = √s.

5. Result:

   - The value of d is the Euclidean distance between points P and Q.

```
function euclidean_distance(x1, y1, x2, y2):
delta_x = x2 - x1
delta_y = y2 - y1
squared_distance = (delta_x)^2 + (delta_y)^2
distance = square_root(squared_distance)
return distance
```

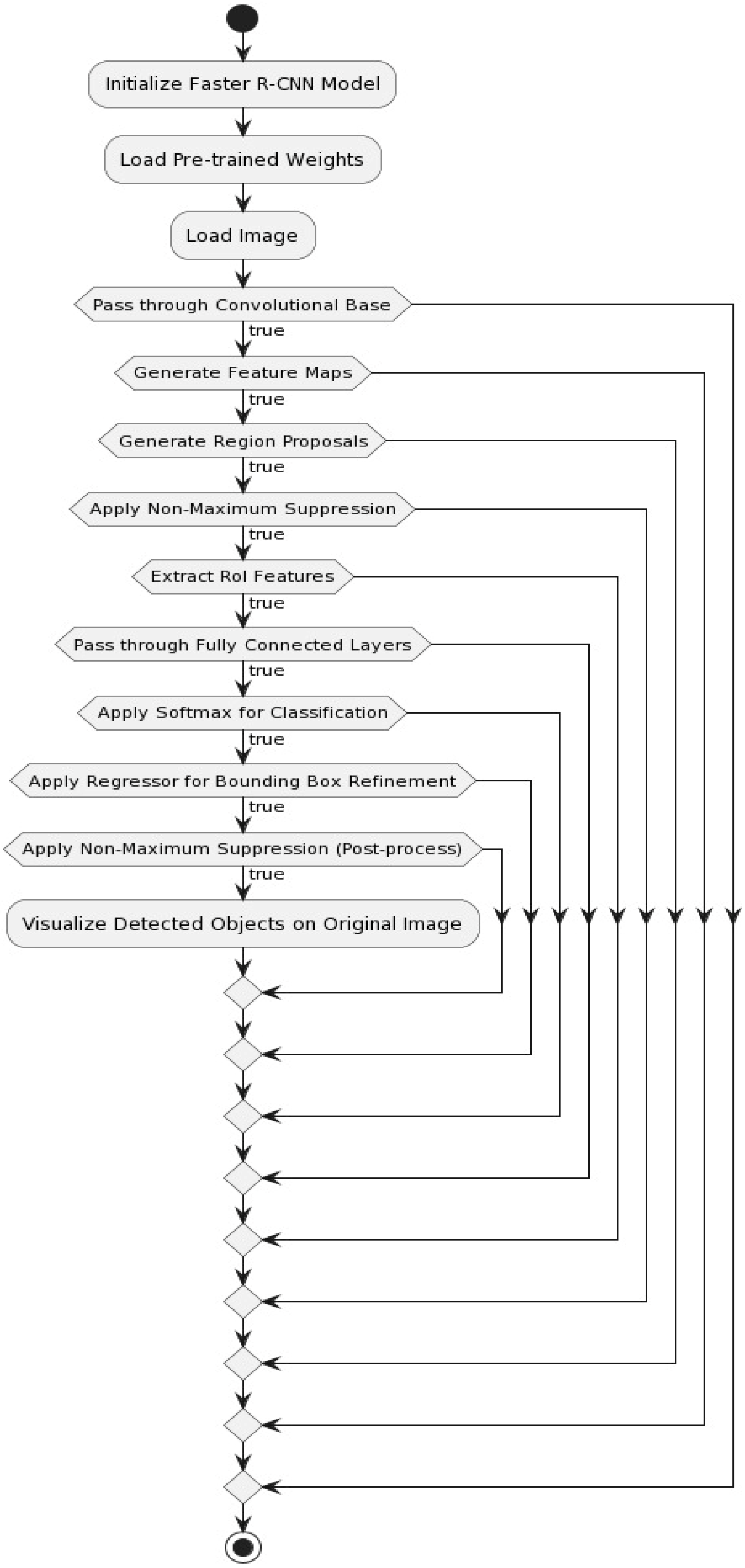


Object Detection with Faster R-CNN:

1. Initialize the Faster R-CNN model with a pre-trained convolutional base (e.g., ResNet).
2. Load the pre-trained weights of the Faster R-CNN model.
3. Initialize the region proposal network (RPN) and the region of interest (RoI) pooling layers.
4. Load the image to be analyzed.
5. Preprocess the image by normalizing pixel values and resizing to the input size expected by the model.
6. Pass the preprocessed image through the convolutional base to obtain feature maps.
7. Use the RPN to generate region proposals (candidate object bounding boxes) based on the feature maps.
8. Apply non-maximum suppression to filter out redundant region proposals.
9. Use RoI pooling to extract fixed-size features from each region proposal.
10. Pass the RoI features through the fully connected layers of the Faster R-CNN model.
11. Apply softmax to classify objects and regressor to refine bounding box coordinates.
12. Post-process the predictions by applying non-maximum suppression on the final bounding boxes.

13. Visualize the detected objects on the original image.

14.End Algorithm



Occupant Monitoring in a Car

1. Initialize the vehicle's camera or sensor system for monitoring the interior.

2. Continuously capture video frames or sensor data from the vehicle.

3. For each frame or data sample:

4. Perform face detection to identify individuals within the vehicle.

5. Track the detected faces across consecutive frames to maintain their identity.

6. Use facial landmarks detection to analyze facial expressions and gaze direction.

7. Employ body pose estimation to understand the posture and positions of occupants.

8. Check for the presence of seat belts by analyzing the visual data or using seat belt sensors.

9. Determine the number of occupants in the vehicle based on the detected faces.

10. Analyze the behavior of occupants, such as whether they are talking, using devices, or appear drowsy.

11. If a child seat is detected, apply specific safety measures (e.g., deactivating the front passenger airbag).

12. Integrate data from various sensors, such as weight sensors in seats, to refine occupant information.

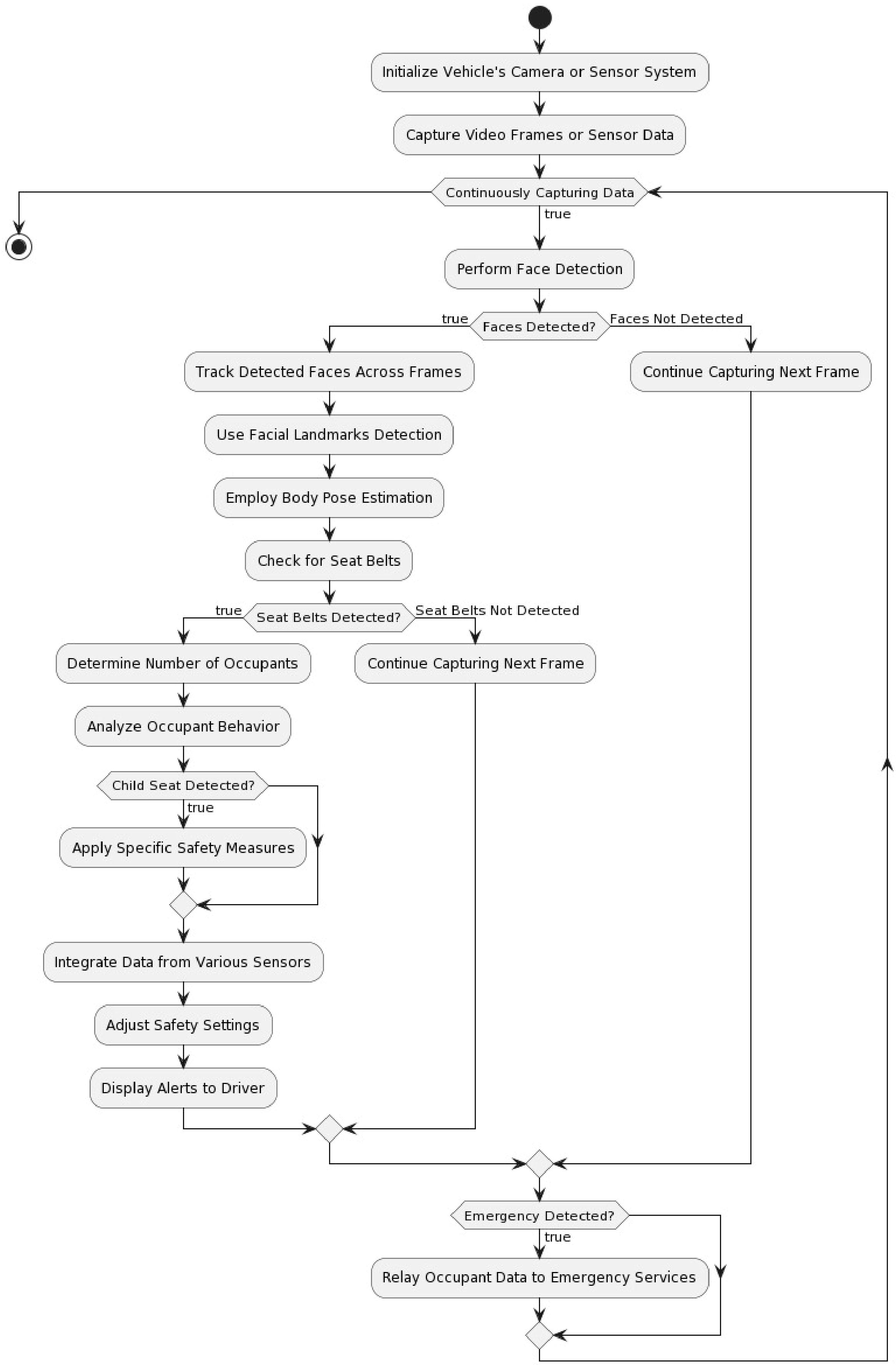
13. Adjust safety settings, such as airbag deployment force and timing, based on occupant characteristics.

14. Display alerts or warnings to the driver if unsafe behaviors are detected.

15. In case of an emergency (e.g., a crash):

16. Relay occupant data to emergency services for better response coordination.

17. End Algorithm


Initialize Vehicle's Camera or Sensor System
Capture Video Frames or Sensor Data
Continuously Capturing Data
true
Perform Face Detection
true
Faces Detected?
Faces Not Detected
Track Detected Faces Across Frames
Continue Capturing Next Frame
Use Facial Landmarks Detection
Employ Body Pose Estimation
Check for Seat Belts
true
Seat Belts Detected?
Seat Belts Not Detected
Determine Number of Occupants
Continue Capturing Next Frame
Analyze Occupant Behavior
Child Seat Detected?
true
Apply Specific Safety Measures
Integrate Data from Various Sensors
Adjust Safety Settings
Display Alerts to Driver
Emergency Detected?
true
Relay Occupant Data to Emergency Services

Mood or Expression Detection

1. Initialize a mood/expression detection model or algorithm.

2. Acquire video frames containing the face of the individual.

3. For each frame:

   4. Detect the face in the image using a face detection algorithm.

   5. Extract facial landmarks or regions of interest (ROIs) corresponding to key facial features.

   6. Preprocess the face or facial landmarks (e.g., resizing, normalization) for input to the mood/expression detection model.

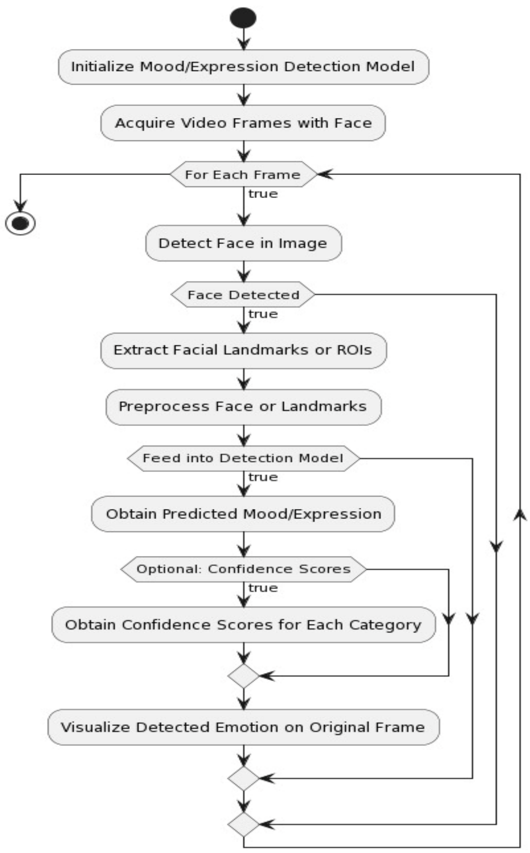
   7. Feed the preprocessed data into the mood/expression detection model.

   8. Obtain the predicted mood or expression category from the model.

   9. Optionally, obtain confidence scores for each predicted emotion category.

   10. Visualize the detected emotion category on the original frame or use it for further analysis.

11. End Algorithm



3. REQUIREMENT ANALYSIS

3.1 SOFTWARE REQUIREMENTS ANALYSIS

1. Coding language: Python version 3.8.Python is the basic of the program that we wrote. It is utilizes many of the Python libraries.

2.Libraries

-OpenCv: use to get the video stream from the webcam etc.

-Dlib: This program is used to find the frontal human face and estimate.

-Imutils: Convention functions written for OpenCv

-Numpy: Pre-requisite for dlib

3.Operating System: Windows

4. Backend : Python

3.2 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

FUNCTIONAL REQUIREMENTS

When designing a Driver Monitoring System (DMS), functional requirements are essential to ensure that the system performs its intended tasks effectively. Here are specific functional requirements for each module you mentioned:

- Sleep Detection: The system should monitor the driver's eye movements, including blink rate and duration. It should detect signs of drowsiness or fatigue based on eye closure patterns. Real-time monitoring of head position and posture to identify signs of drowsiness. Integration with vehicle data to detect erratic driving behavior.
- Lighting Condition: Adjust the monitoring algorithm based on varying lighting conditions, including day and night. Use sensors or cameras with low-light capabilities to ensure accurate monitoring in low-light

environments. Adapt to sudden changes in ambient lighting to maintain effectiveness.

- Mood or Expression Detection: Utilize facial recognition technology to identify facial expressions. Implement an algorithm to analyze facial features and determine the driver's mood. Integrate with the vehicle's internal systems to correlate mood data with driving conditions.
- Gaze Detection: Track the driver's gaze direction and focus on the road. Detect distractions by monitoring the driver's gaze away from the road. Provide real-time alerts or warnings if the driver's gaze is not on the road when necessary.
- Occupant Monitoring System: Identify and differentiate between the driver and other occupants in the vehicle. Monitor the occupants for safety, ensuring they are in proper positions and using seat belts.Integrate with airbag systems to optimize deployment based on occupant positions.
- Object Detection: Use sensors and cameras to detect objects in the vehicle's path. Provide real-time alerts for potential collisions with objects and obstacles.Integrate with the vehicle's collision avoidance system to enhance safety.

NON-FUNCTIONAL REQUIREMENTS:

Non-functional requirements are crucial aspects of a system that describe how it performs its functions rather than what functions it performs. In the context of a Driver Monitoring System (DMS), several non-functional requirements are essential for the system to meet its objectives effectively. Here are some key non-functional requirements for a DMS:

1.Performance:

Response Time: Specify the maximum acceptable delay for the system to respond to events, such as detecting driver distraction or drowsiness.

Throughput: Define the number of video frames or data points the system should be able to process per unit of time.

2.Reliability:

Availability: Specify the minimum required uptime for the DMS to ensure continuous monitoring and alerting.

Fault Tolerance: Describe the system's ability to continue operating and providing alerts in the presence of hardware or software failures.

3.Scalability:

Capacity Planning: Outline how the system can handle an increase in the number of monitored vehicles or cameras.

Load Balancing: Ensure that resources are distributed efficiently to handle varying workloads.

4.Security:

Data Encryption: Specify the encryption standards used to protect sensitive data transmitted or stored by the system.

Access Control: Define who has access to different parts of the system and what actions they are allowed to perform.

5.Usability:

User Interface Design: Specify guidelines for designing a user-friendly interface for system administrators or users.

Accessibility: Ensure that the DMS can be easily used by individuals with diverse abilities and needs.

6.Maintainability:

Modularity: Design the system in a way that allows for easy updates and modifications without affecting the entire system.

Documentation: Provide comprehensive documentation for system components, APIs, and any custom configurations.

7.Interoperability:

Integration with Other Systems: Specify how the DMS should integrate with other in-vehicle systems or external platforms.

Compatibility: Ensure compatibility with different camera hardware, software versions, and communication protocols.

8.Regulatory Compliance:

Adherence to Standards: Ensure that the DMS complies with relevant safety standards, privacy regulations, and automotive industry requirements.

Certification: Obtain necessary certifications based on applicable regulations and standards.

9.Performance Monitoring:

Logging and Auditing: Implement comprehensive logging of system activities for troubleshooting and auditing purposes.

Analytics: Incorporate tools for monitoring and analyzing system performance over time.

10.Resource Utilization:

Memory Usage: Define acceptable levels of memory consumption for the DMS.

CPU Utilization: Specify the maximum allowable CPU usage under normal and peak conditions.

11.Data Privacy:

Anonymization: Implement mechanisms to anonymize or pseudonymize data, especially if it involves personally identifiable information (PII).

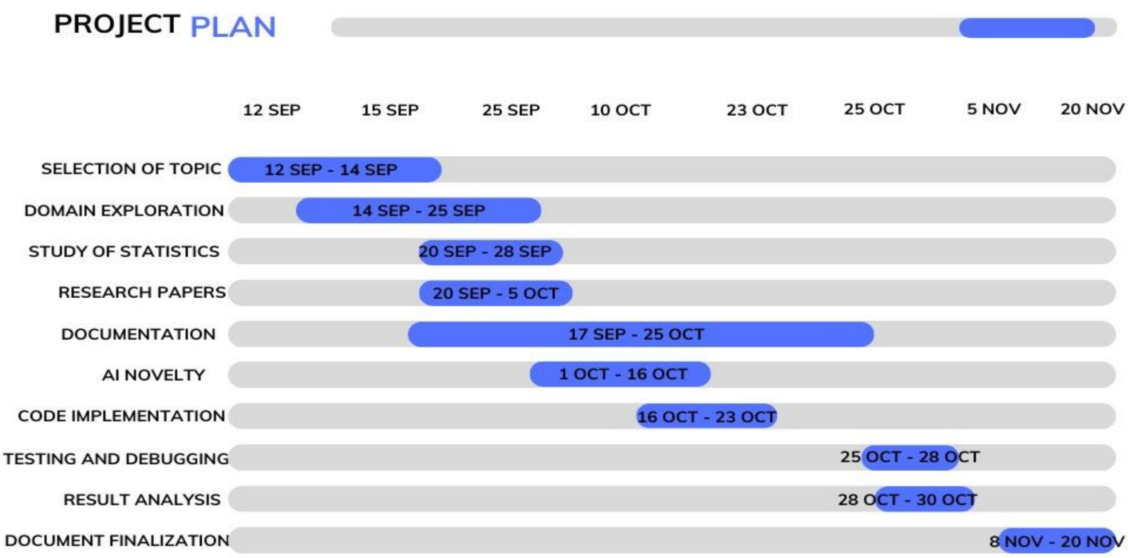
Data Retention: Define policies for how long data should be stored and under what conditions it should be purged.

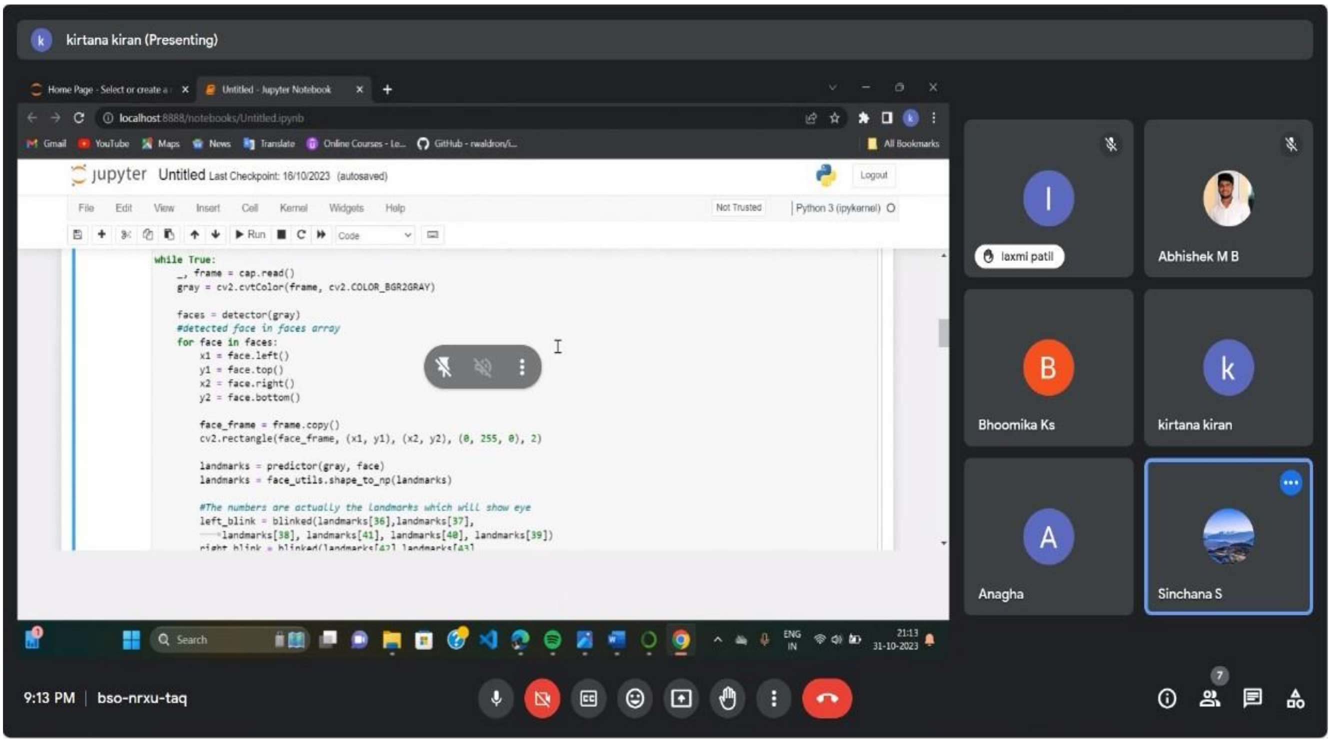
These non-functional requirements are essential for ensuring the effectiveness, reliability, and compliance of a Driver Monitoring System in real-world applications. They provide a framework for evaluating and measuring the system's performance and capabilities beyond its core functionalities.

4. PROJECT DETAILS

4.1 PROJECT PLANNING

Project planning for a Driver Monitoring System (DMS) involves systematically orchestrating the various components essential for the successful development and deployment of the system. The process commences with a clear definition of project objectives and scope, delineating intended outcomes and constraints. Identifying stakeholders, ranging from project managers to end-users, ensures comprehensive engagement throughout the development lifecycle. A thorough needs assessment examines legal requirements, safety standards, and user expectations, laying the groundwork for technology selection. This involves researching and choosing suitable hardware and software technologies aligned with project goals. Adherence to legal and ethical considerations, particularly regarding privacy and data protection, is paramount. Budgeting encompasses hardware, software, personnel, and maintenance costs, while timeline planning establishes realistic milestones. Rigorous risk assessment helps preempt and manage potential challenges. Detailed documentation, including user manuals and system architecture documents, accompanies user-friendly interface design. Development unfolds in phases with robust testing, leading to comprehensive training programs for users and administrators. Deployment is carefully orchestrated to minimize disruptions, followed by ongoing maintenance and monitoring to ensure sustained performance and alignment with project objectives.


PROJECT PLAN
12 SEP
15 SEP
25 SEP
10 OCT
23 OCT
25 OCT
5 NOV
20 NOV
SELECTION OF TOPIC
12 SEP - 14 SEP
DOMAIN EXPLORATION
14 SEP - 25 SEP
STUDY OF STATISTICS
20 SEP - 28 SEP
RESEARCH PAPERS
20 SEP - 5 OCT
DOCUMENTATION
17 SEP - 25 OCT
AI NOVELTY
1 OCT - 16 OCT
CODE IMPLEMENTATION
16 OCT - 23 OCT
TESTING AND DEBUGGING
25 OCT - 28 OCT
RESULT ANALYSIS
28 OCT - 30 OCT
DOCUMENT FINALIZATION
8 NOV - 20 NOV


kirtana kiran (Presenting)
jupyter Untitled Last Checkpoint: 16/10/2023 (autosaved)
while True:
_, frame = cap.read()
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
faces = detector(gray)
#detected face in faces array
for face in faces:
x1 = face.left()
y1 = face.top()
x2 = face.right()
y2 = face.bottom()
face_frame = frame.copy()
cv2.rectangle(face_frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
landmarks = predictor(gray, face)
landmarks = face_utils.shape_to_np(landmarks)
#The numbers are actually the landmarks which will show eye
laxmi patil
Abhishek M B
Bhoomika Ks
kirtana kiran
Anagha
Sinchana S
9:13 PM | bso-nrxu-taq

4.2 IMPLEMENTATION

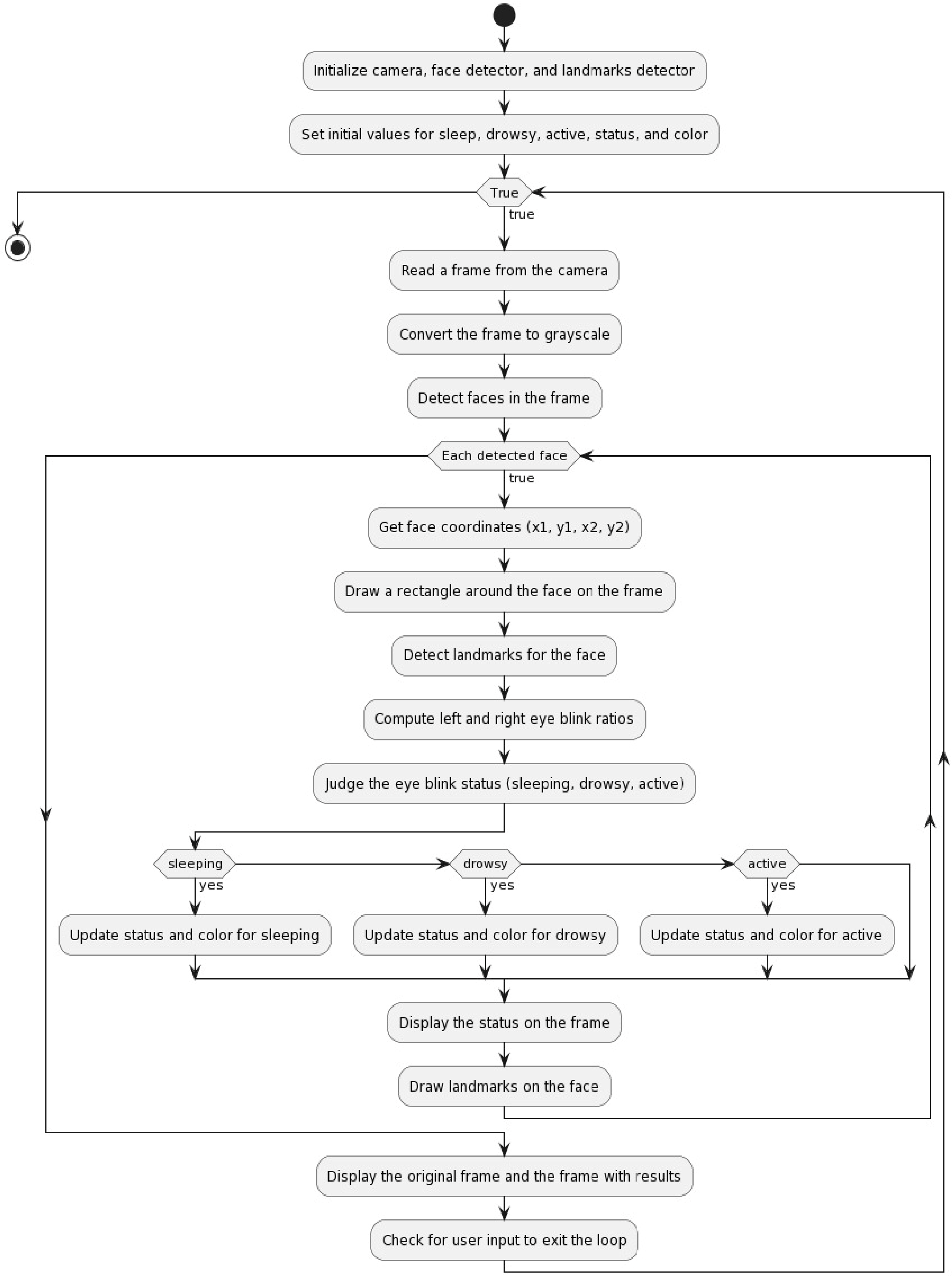
```
#Importing OpenCV Library for basic image processing functions
import cv2
# Numpy for array related functions
import numpy as np
# Dlib for deep learning based Modules and face landmark detection
import dlib
#face_utils for basic operations of conversion
from imutils import face_utils
#Initializing the camera and taking the instance
cap = cv2.VideoCapture(0)
#Initializing the face detector and landmark detector
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")
#status marking for current state
sleep = 0
drowsy = 0
active = 0
status=""
color=(0,0,0)

def compute(ptA,ptB):
	dist = np.linalg.norm(ptA - ptB)
	return dist
def blinked(a,b,c,d,e,f):
	up = compute(b,d) + compute(c,e)
	down = compute(a,f)
	ratio = up/(2.0*down)
#Checking if it is blinked
```

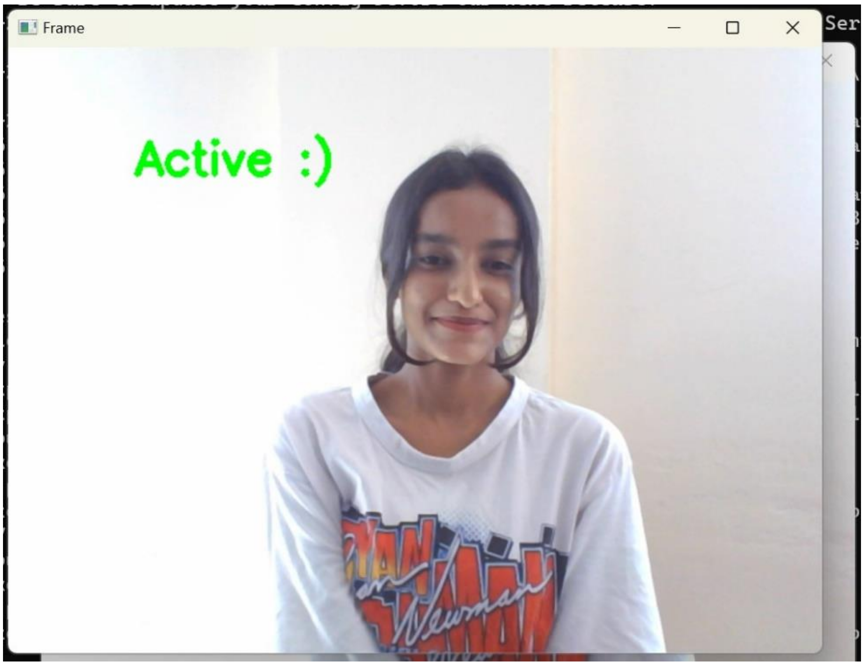
```
        if(ratio>0.25):
                return 2
        elif(ratio>0.21 and ratio<=0.25):
                return 1
        else:
                return 0
while True:
    _, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
     faces = detector(gray)
    #detected face in faces array
    for face in faces:
        x1 = face.left()
        y1 = face.top()
        x2 = face.right()
        y2 = face.bottom()
    face_frame = frame.copy()
        cv2.rectangle(face_frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
        landmarks = predictor(gray, face)
        landmarks = face_utils.shape_to_np(landmarks)

        #The numbers are actually the landmarks which will show eye
        left_blink = blinked(landmarks[36],landmarks[37],
          landmarks[38], landmarks[41], landmarks[40], landmarks[39])
        right_blink = blinked(landmarks[42],landmarks[43],
          landmarks[44], landmarks[47], landmarks[46], landmarks[45])
       #Now judge what to do for the eye blinks
        if(left_blink==0 or right_blink==0):
          sleep+=1
```

```
        drowsy=0
        active=0
        if(sleep>6):
                status="SLEEPING !!!"
                color = (255,0,0)
        elif(left_blink==1 or right_blink==1):
        sleep=0
        active=0
        drowsy+=1
        if(drowsy>6):
                status="Drowsy !"
                color = (0,0,255)
        else:
        drowsy=0
        sleep=0
        active+=1
        if(active>6):
                status="Active :)"
                color = (0,255,0)
        cv2.putText(frame, status, (100,100), cv2.FONT_HERSHEY_SIMPLEX,
1.2, color,3)
            for n in range(0, 68):
        (x,y) = landmarks[n]
        cv2.circle(face_frame, (x, y), 1, (255, 255, 255), -1)
      cv2.imshow("Frame", frame)
   cv2.imshow("Result of detector", face_frame)
   key = cv2.waitKey(1)
   if key == 27:
  break
```

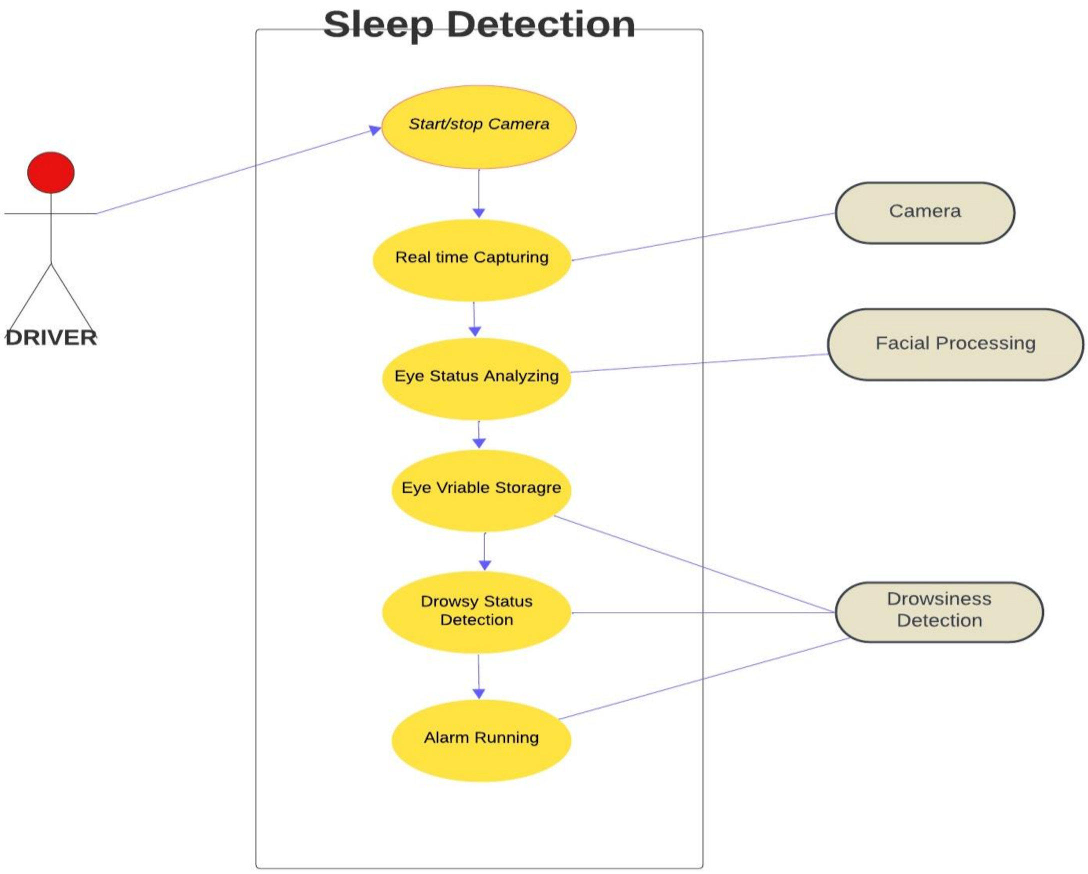

Initialize camera, face detector, and landmarks detector
Set initial values for sleep, drowsy, active, status, and color
True
true
Read a frame from the camera
Convert the frame to grayscale
Detect faces in the frame
Each detected face
true
Get face coordinates (x1, y1, x2, y2)
Draw a rectangle around the face on the frame
Detect landmarks for the face
Compute left and right eye blink ratios
Judge the eye blink status (sleeping, drowsy, active)
sleeping
yes
drowsy
yes
active
yes
Update status and color for sleeping
Update status and color for drowsy
Update status and color for active
Display the status on the frame
Draw landmarks on the face
Display the original frame and the frame with results
Check for user input to exit the loop

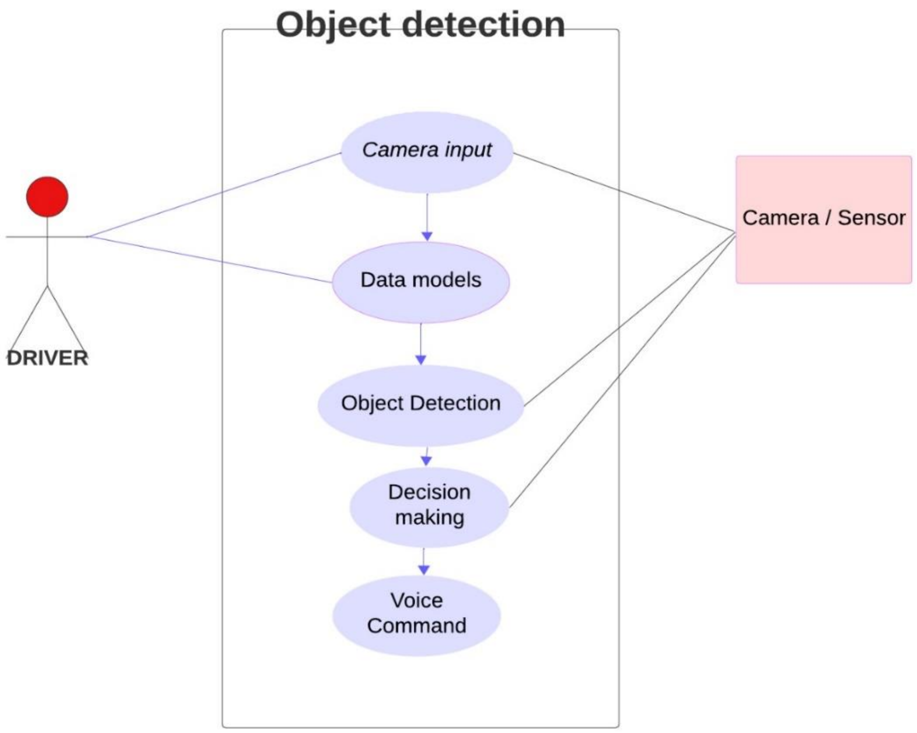
4.3 TESTING:

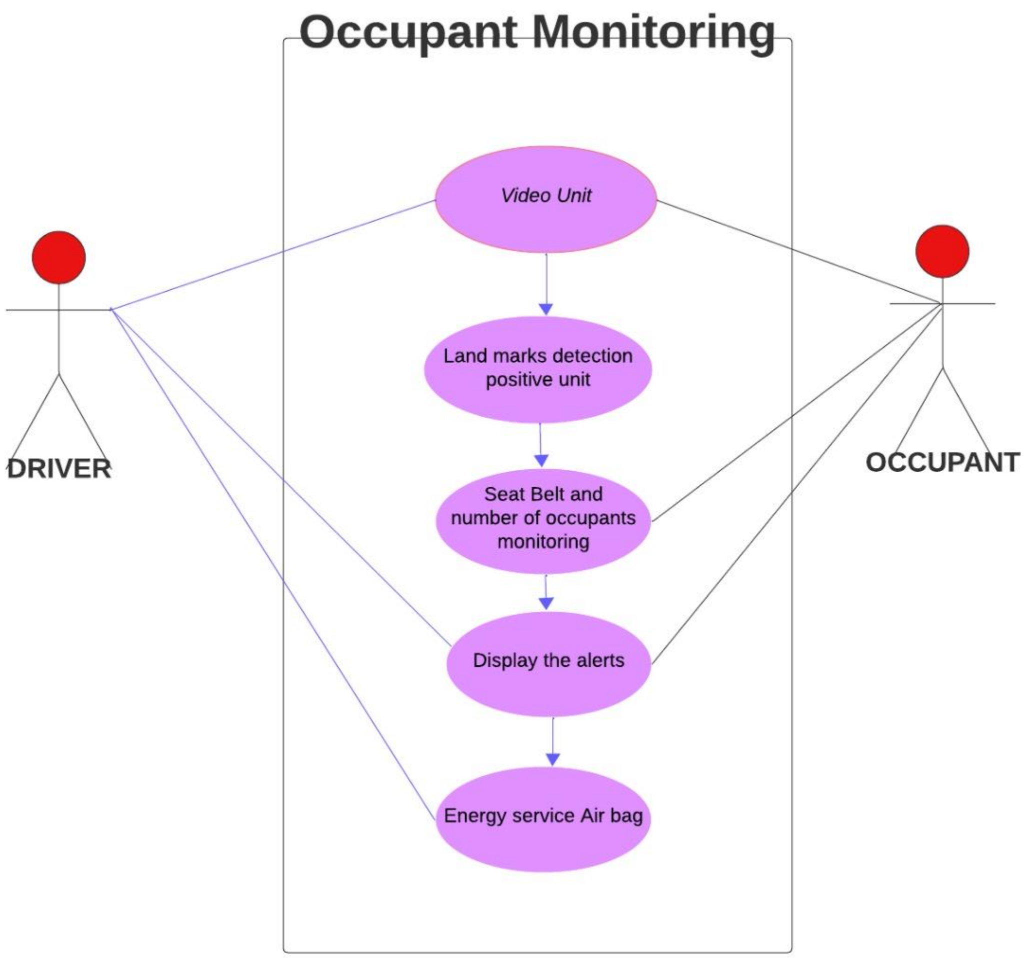


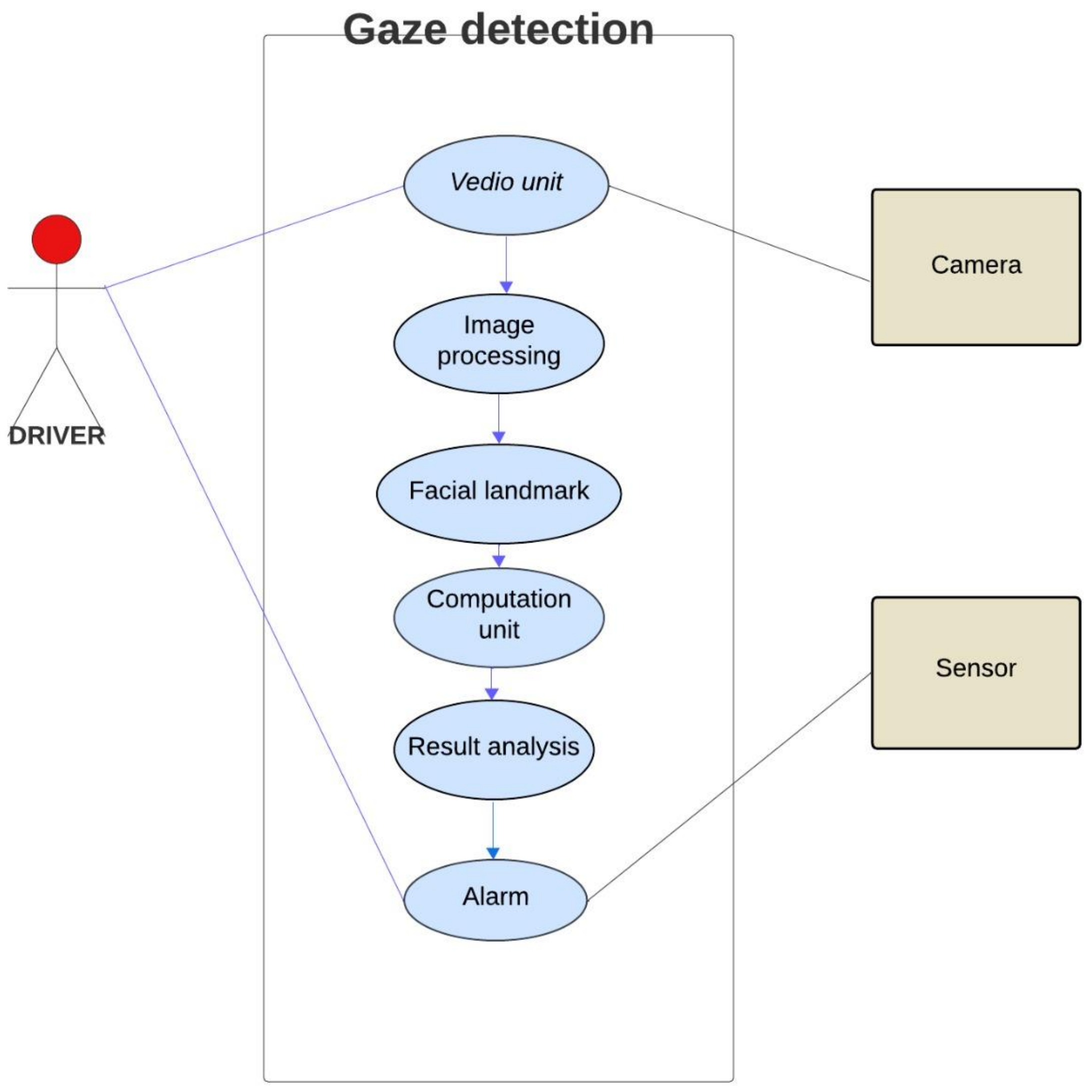
5. SYSTEM DESIGN

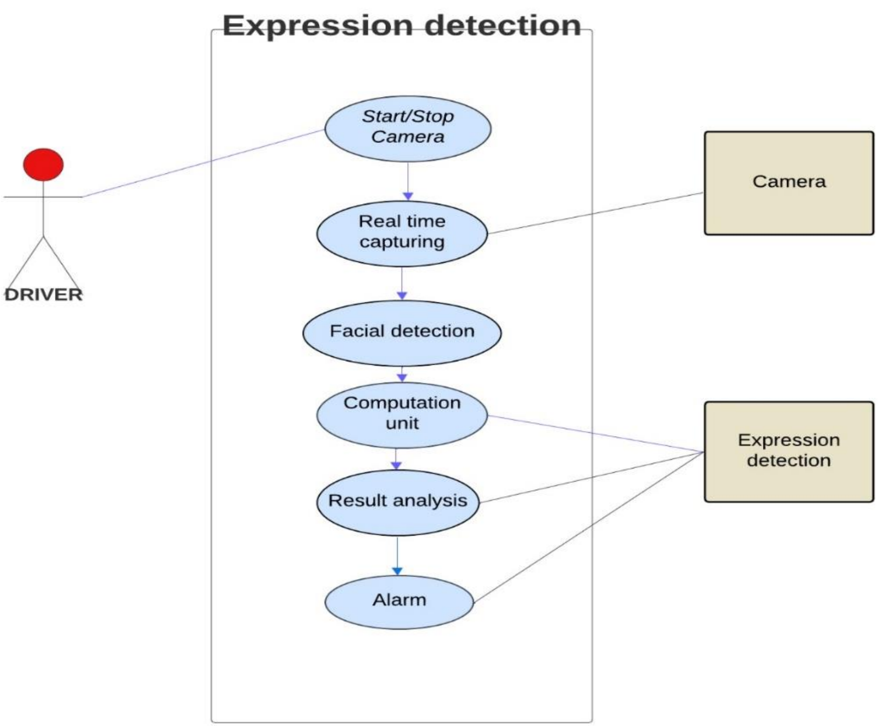
5.1 USE CASE DIAGRAMS

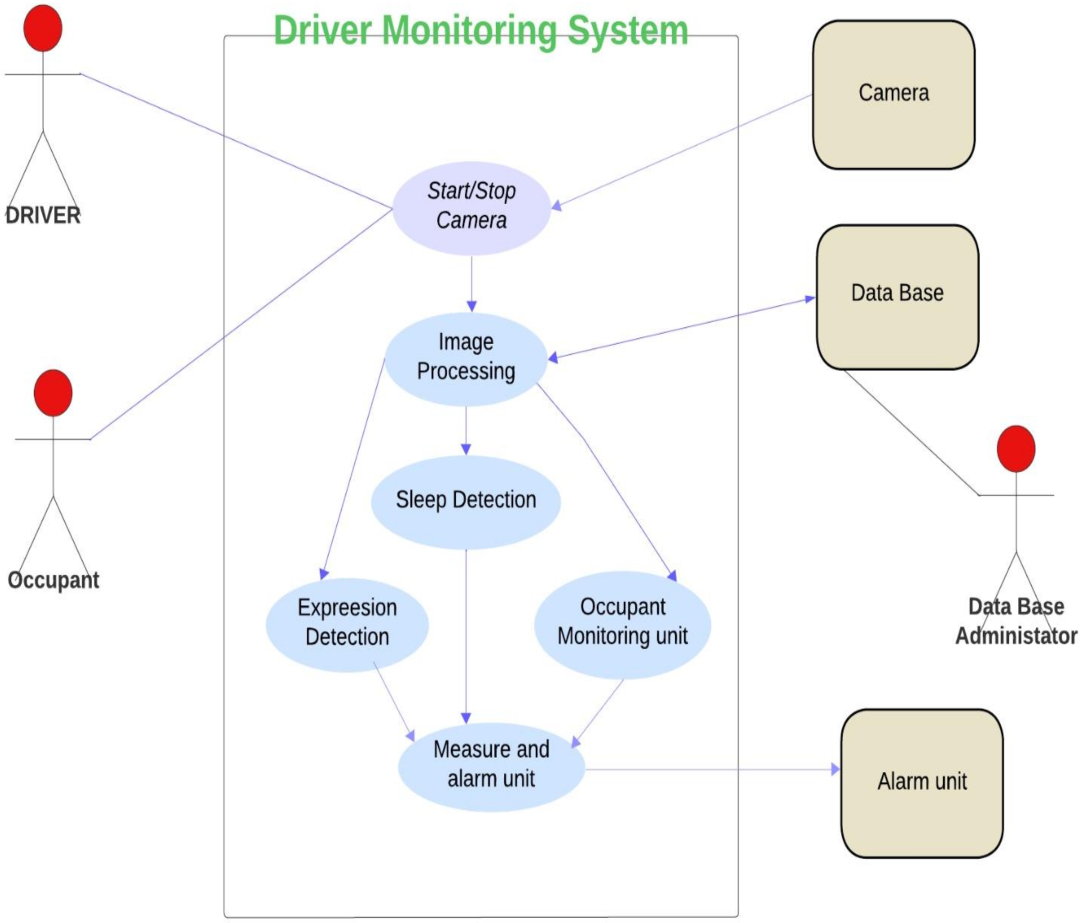



Object detection
Camera input
Data models
Object Detection
Decision making
Voice Command
DRIVER
Camera / Sensor

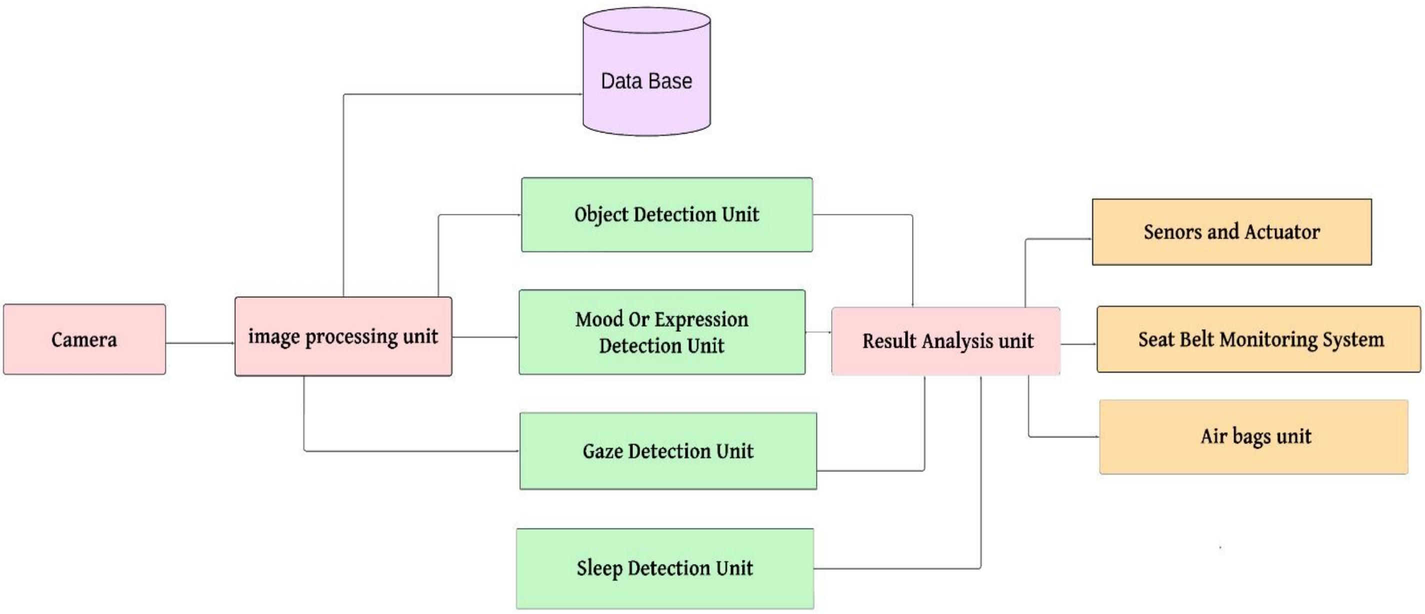

Occupant Monitoring
Video Unit
Land marks detection positive unit
Seat Belt and number of occupants monitoring
Display the alerts
Energy service Air bag
DRIVER
OCCUPANT


Gaze detection
DRIVER
Vedio unit
Image processing
Facial landmark
Computation unit
Result analysis
Alarm
Camera
Sensor


Expression detection
DRIVER
Start/Stop Camera
Real time capturing
Facial detection
Computation unit
Result analysis
Alarm
Camera
Expression detection



5.2 DATA FLOW DIAGRAMS



6. FUTURE SCOPE, CONCLUSION & REFERENCES

FUTURE SCOPE :

The future scope for Driver Monitoring Systems (DMS) is likely to expand as technology continues to advance and as concerns about road safety, driver fatigue, and distracted driving persist. Here are some potential future developments and applications for DMS:

- Advanced Sensing Technologies: Integration of more advanced sensors, such as infrared cameras and 3D facial recognition, to improve accuracy in monitoring driver behavior.
- Artificial Intelligence and Machine Learning:Enhanced AI algorithms and machine learning models for more accurate detection of driver fatigue, distraction, and emotional states. This could lead to real-time adaptive systems that respond to the driver's needs.
- Biometric Monitoring: Integration of biometric sensors to monitor vital signs such as heart rate, blood pressure, and even stress levels. This could provide a more comprehensive understanding of the driver's physical and mental state.
- Autonomous Vehicle Integration: Collaboration with autonomous vehicle systems to ensure a smooth transition between manual and automated driving modes. DMS could play a crucial role in ensuring the driver is ready to take control when needed.
- In-Cabin Monitoring for Passenger Safety:Extending DMS capabilities to monitor the well-being of passengers, especially in the context of autonomous vehicles where traditional driver roles may evolve.
- Customized Alerts and Feedback: Personalized feedback and alerts based on individual driving patterns, preferences, and health conditions, promoting safer driving habits.

- Regulatory Compliance: Increasing adoption and standardization of DMS as a safety feature, potentially leading to regulatory requirements for certain types of vehicles.
- Integration with Fleet Management: DMS could be integrated into fleet management systems to enhance overall safety, driver training, and operational efficiency.
- Cybersecurity Measures:Implementing robust cybersecurity measures to protect the data collected by DMS, ensuring the privacy and security of drivers.
- Research in Human-Machine Interaction: Continued research in human-machine interaction to improve the overall user experience and reduce potential distractions caused by DMS alerts and feedback.As technology continues to evolve, the scope for DMS is likely to expand, leading to more sophisticated and integrated systems aimed at making road travel safer for everyone.

CONCLUSION:

In conclusion, Driver Monitoring Systems (DMS) represent a critical and evolving technology with significant implications for road safety, driver well-being, and the future of transportation. As the automotive industry continues to embrace innovation, the role of DMS is poised to expand and become increasingly sophisticated.

The integration of advanced sensing technologies, artificial intelligence, and biometric monitoring positions DMS as a pivotal tool in addressing issues such as driver fatigue, distraction, and overall situational awareness. The potential for real-time adaptive systems and personalized feedback underscores the transformative impact that DMS can have on individual driving habits and, consequently, road safety.

Moreover, the future scope of DMS extends beyond individual drivers to include broader applications in autonomous vehicles, fleet management, and regulatory compliance. The collaboration between DMS and autonomous vehicle systems is particularly noteworthy, as it emphasizes the seamless transition between manual and automated driving modes, ensuring that drivers remain engaged and ready to take control when necessary.

As DMS becomes more prevalent, it is crucial to address challenges related to cybersecurity and privacy to safeguard the data collected from drivers. Additionally, ongoing research in human-machine interaction will play a key role in refining DMS functionalities, minimizing potential distractions, and optimizing the overall user experience.

In essence, the future of Driver Monitoring Systems is marked by continuous innovation, increased integration with emerging technologies, and a commitment to creating safer and more efficient roadways. The journey ahead involves not only advancing the technological capabilities of DMS but also fostering a holistic approach that prioritizes both the safety and well-being of drivers and passengers.

REFERENCES:

1.https://www.researchgate.net/publication/336878674_DRIVER_DROWSINESS_DETECTION_SYSTEM

2.https://www.labellerr.com/blog/hands-on-with-driver-activity-monitoring-system-for-autonomous-driving/

3.https://www.researchgate.net/publication/344907383
_Fatigue_Detection_among_Operators_in_Industry_Based_on_Euclidean_Distance_Computation_Using_Python_Software

4.https://www.tataelxsi.com/product/artificial-intelligence/aidms

5.https://link.springer.com/chapter/10.1007/978-981-99-5971-6_10

6.https://pdfs.semanticscholar.org/4bb0/799fbbef37af2a83939e92d40e5ab8db8a8c.pdf

7.https://iopscience.iop.org/article/10.1088/1742-6596/1813/1/012036/pdf

8.https://synthesis.ai/applications/driver-monitoring/

9.COMPUTATIONALLY EFFICIENT FACE DETECTION; B. SCHLKOPF-A. BLAKE, S. ROMDHANI, AND P. TORR.

10.https://ieeexplore.ieee.org/abstract/document/7752938