A Project Report on

**DRIVER MONITORING AND ALERTING SYSTEM**

Submitted in partial fulfillment of the requirements for the award of the degree of

**Bachelor of Technology**

In

**Information Technology**

Submitted by

**Bipin Kumar Soni (16P31A1203)**

**M. Siri Chandana (16P31A1224)**

**Appasani Lavanya (16P31A1201)**

**B. K. Srikar (16P31A1202)**

Under the esteemed supervision of

Mr. Arava Mohan, M.Tech,

Assistant Professor



**Department of Information Technology**

**ADITYA COLLEGE OF ENGINEERING AND TECHNOLOGY**

**(Approved by AICTE, New Delhi & Affiliated to JNTUK, Kakinada)**

**Surampalem, East Godavari District**

**Andhra Pradesh - 533 437**

**2016-2020**

**ADITYA COLLEGE OF ENGINEERING AND TECHNOLOGY**

**(Approved by AICTE, New Delhi & Affiliated to JNTUK, Kakinada)**

**Department of Information Technology**



**CERTIFICATE**

This is to certify that the project work entitled, " **Driver Monitoring And Alerting System**", is a bonafide work carried out by **Bipin Kumar Soni (16P31A1203)**, **M. Siri Chandana (16P31A124)**, **Appasani Lavanya (16P31A1201)**,**B. K. Srikar (16P31A1202)**, in partial fulfilment of the requirements for the award of degree of **BACHELOR** OF TECHNOLOGY in **INFORMATION TECHNOLOGY** from **Aditya college of engineering and Technology**, during the academic year **2016-2020**.

|  |  |  |
| --- | --- | --- |
| SUPERVISOR  **Mr.Arava Mohan,M.Tech,**  **Assistant Professor** |  | HEAD OF THE DEPARTMENT  **Mr.R.V.V.N.BheemaRao,M.Tech,**  **Associate Professor** |
|  |  |  |

**EXTERNAL EXAMINER**

**DECLARATION**

I hereby declare that this project entitled "**Driver Monitoring And Alerting System**" has been undertaken by me and this work has been submitted to **ADITYA COLLEGE OF ENGINEERING & TECHNOLOGY** affiliated to JNTUK, Kakinada, in partial fulfilment of the requirements for the award of the degree of  **Bachelor of Technology** in **Information Technology.**

I further declare that this project work has not been submitted in full or part for the award of any degree of this or in any other educational institutions.

**Bipin Kumar Soni (16P31A1203)**

**M. Siri Chandana (16P31A1224)**

**Appasani Lavanya (16P31A1201)**

**B. K. Srikar (16P31A1202)**

**ACKNOWLEDGEMENTS**

It is with immense pleasure that I would like to express my indebted gratitude to **Project Supervisor,** **Mr.ARAVA MOHAN, M.Tech,** who has guided me a lot and encouraged me in every step of project work, his valuable moral support and guidance throughout the project helped me greater extent.

I wish to express my sincere thanks to Head of the Department, **IT, Mr.R.V.V.N.BheemaRao, M.Tech,** for his valuable guidance given to me throughout the period of the project work.

I feel elated to thank **Vice Principal**, **Dr.A.Rama krishna**, of **Aditya College Of**

**Engineering and Technology** for his cooperation and helping completion of my project and throughout my course.

I feel elated to thank **Principal**, **Dr.T.K.Rama Krishna Rao,** of **Aditya College Of**

**Engineering and Technology** for his cooperation and helping completion of my project and throughout my course.

I wish to express my sincere thanks to all my **Faculty** **members, Lab Programmers** for their valuable guidance given to me throughout the period of the project.

I avail this opportunity to express my deep sense and heart full thanks to the **Management** of **Aditya College Of Engineering and Technology** for providing a great support for me in completing my project by arranging the trainees, and facilities needed to complete my project and for giving me the opportunity of doing the project.

**Bipin Kumar Soni (16P31A1203)**

**Siri Chandana (16P31A1224)**

**Appasani Lavanya (16P31A1201)**

**B. K. Srikar (16P31A1202)**

### CONTENTS

**CHAPTER PAGE NO**

ABSTRACT i

LIST OF FIGURES ii

LIST OF TABLES v

1. **INTRODUCTION** 2
   1. Project Overview 2
   2. Existing System 3
   3. Proposed System 3
2. **LITERATURE SURVEY** 7
3. **REQUIREMENTS** 9
   1. Functional Requirements 9
   2. Non Functional Requirements 10
   3. Hardware Requirements 10
   4. Software Requirements 10
4. **POWER SUPPLY** 56
   1. Description 57
5. **SYSTEM DESIGN** 70
   1. UML diagrams 70
6. **IMPLEMENTATION** 79
   1. Technology Used 79
   2. Sample Code 90
7. **TESTING** 107
   1. Introduction 107
8. **RESULTS** 110

**CONCLUSIONS AND FUTURE ENHANCEMENT** 118

**REFERENCES** 1

## LIST OF FIGURES

### S.NO NAME OF FIGURE PAGE NO

1. Figure 1.1 System Design 4
2. Figure 1.2 Block Diagram 5
3. Figure 1.3 Isometric view of Trolley 5
4. Figure 3.1 Schematic diagram 12
5. Figure 3.2 Address location for 1\*16 line LCD 15
6. Figure 3.3 Electrical block diagram 16
7. Figure 3.4 Power supply for LCD 16
8. Figure 3.5 Pin diagram of 1\*16 lines LCD 17
9. Figure 3.6 Character details in LCD 21
10. Figure 3.7 Flow chart of LCD 22
11. Figure 3.8 Time division multiple access 36
12. Figure 3.9 Cellular network 37
13. Figure 3.10 GSM network 38
14. Figure 3.11 Modem stands for modular-demdulator 40
15. Figure 3.12 GSM modem 42
16. Figure 3.13 Establishing connection between PC and GSM 43

modem

1. Figure 3.14 Arduino Nano 48
2. Figure 3.15 Arduino Nano pinout 50 19 Figure 4.1 Power supply 55 20 Figure 4.2 Circuit diagram 55

21 Figure 4.3 Transformer Symbol 56

|  |  |  |  |
| --- | --- | --- | --- |
| 22 | Figure 4.4 | Transformer | 56 |
| 23 | Figure 4.5 | Basic principle | 57 |
| 24 | Figure 4.6 | Basic transformer | 57 |
| 25 | Figure 4.7 | Step-down transformer | 59 |
| 26 | Figure 4.8 | Step-up transformer | 61 |
| 27 | Figure 4.9 | Diode symbol | 62 |
| 28 | Figure 4.10 | Half wave rectifier | 63 |
| 29 | Figure 4.11 | Half wave rectification | 63 |
| 30 | Figure 4.12 | Full wave rectification | 64 |
| 31 | Figure 4.13 | Capacitor filter | 65 |
| 32 | Figure 4.14 | Centered tapped full-wave rectifier with a capacitor filter | 66 |
| 33 | Figure 4.15 | Regulator | 67 |
| 34 | Figure 5.1 | Representation of Actor | 71 |
| 35 | Figure 5.2 | Representation of Use case | 71 |
| 36 | Figure 5.3 | Representation of Relationship | 71 |
| 37 | Figure 6.1 | Embedded system example | 79 |
| 38 | Figure 6.2 | Circuit diagram | 88 |
| 39 | Figure 7.1 | Unit testing of LCD display | 106 |
| 40 | Figure8.1 | GSM Initializing | 109 |
| 41 | Figure 8.2 | Starting process of smart trolley | 109 |
| 42 | Figure 8.3 | Give a missed call to store number | 110 |
| 43 | Figure 8.4 | Display number | 110 |
| 44 | Figure 8.5 | Sending message | 111 |
| 45 | Figure 8.6 | Starting shopping shows items and total | 111 |
| 46 | Figure 8.7 | Shows no.of items and total | 112 |
| 47 | Figure 8.8 | Adding product details | 112 |

|  |  |  |  |
| --- | --- | --- | --- |
| 48 | Figure 8.9 | Removing product details | 113 |
| 49 | Figure 8.10 | Final bill | 113 |
| 50 | Figure 8.11 | Get a message to mobile | 114 |
| 51 | Figure 8.12 | Thank you | 114 |
| 52 | Figure 8.13 | Restart and continue | 115 |

## DRIVER MONITORING AND ALERTING SYSTEM

**ABSTRACT:**

A new approach towards automobile safety and security with autonomous region primarily based automatic automotive system is projected during this conception. We have a tendency to propose 3 distinct however closely connected ideas viz. a Drowsy Driver Detection system with based conception. In recent time's automobile fatigue connected crashes have very enlarged. so as to attenuate these problems, we've incorporated driver alert system by watching each the driver's eyes still as sensing still because the driver state of affairs based primarily based native setting recognition based AI system is projected.

The main principle of the proposed system is using Open CV(Open Source of Computer Vision) library which is based on real time facial images analysis for warning the driver of drowsiness or in attention to prevent the traffic accidents. The facial images of driver are taken by a camera which is installed. An algorithm and an inference are proposed to determine the level of fatigue by measuring the eye blinking duration and face detection to track the eyes, and warn the driver accordingly. If the eyes are found closed for 5 to 8 consecutive frames, The system draws the conclusion that the driver is falling asleep and issues a warning signal. This system is also able to detect when the eyes cannot be found.

**Chapter 1**

**INTRODUCTION**

#### 1.INTRODUCTION

The most important thing in the world is our lives. Monitoring an eye while driving is the best development we can have presently so far someone’s valuable life is concerned. Drowsy driving is one of the risky case in driving. Here, in our project we are about to develop a system which can give alert to drivers if they fall asleep and shows some signs of fatigue or drowsiness. “Drowsy driving was responsible for 72,000 crashes, 44,000 injuries, and 800 deaths in 2013. However, these numbers are underestimated and up to 6,000 fatal crashes each year may be caused by drowsy drivers”

A new approach towards automobile safety and security with autonomous region primarily based automatic system is projected during this conception. In recent time's automobile fatigue connected crashes have very enlarged. So as to overcome these problems, we've incorporated driver monitoring and alerting system by watching each the driver's eyes still as sensing still because the driver state of affairs primarily based on native setting recognition based AI system is projected.



**Figure 1.1** Bus crash due to driver fatigue



**Figure 1.2** Reason behind accidents

**1.1 EXISTING SYSTEM**

According to a survey in 2015, It was found that 37% of the accidents are caused due to drowsy driving. Nothing is bigger than a life. A risky driving may lead to an accident, which may leads to lose of valuable life, which directly affect one's family and also destruction of high valued assets. The present situation of the drivers is not trust worthy. There is no system to monitor the behavior and performance of the person who is driving a car, bus, heavy vehicle etc.

* + 1. **Problems** **in the Existing System**
  + Loss of valuable life, which ultimately affects the family.
  + Destruction of vehicles.
  + Loss of company assests.
  + Traffic congestion.
  1. **PROPOSED SYSTEM**

The main principle of the proposed system is using Open CV(Open Source of Computer Vision) library which is based on real time facial images analysis for warning the driver of drowsiness or in attention to prevent the traffic accidents. The facial images of driver are taken by a camera which is installed. An Euclidean algorithm and an inference are proposed to determine the level of fatigue by measuring the eye blinking duration and face detection to track the eyes, and warn the driver accordingly. If the eyes are found closed for 5 to 8 consecutive frames, The system draws the conclusion that the driver is falling asleep and issues a warning signal. This system is also able to detect when the eyes cannot be found.

**CHAPTER 2**

**REQUIREMENT ANALYSIS**

**2. REQUIREMENT ANALYSIS**

**2.1 System Requirements Specification**

This project is based on Open CV. The hardware and software specifications of the system are mentioned below.

**2.1.1 Hardware Requirements**

RaspberryPi

 Micro Controller

Camera

**Raspberry Pi :** The Raspberry Pi Foundation works to put the power of computing and digital making into the hands of people all over the world. It does this by providing low-cost, high-performance computers that people use to learn, solve problems, and have fun. It provides outreach and education to help more people access computing and digital making it develops free resources to help people learn about computing and making things with computers and also trains educators who can guide other people to learn.

**A Micro Controller:** Microcontrollers are designed to perform specific tasks. Specific means applications where the relationship of input and output is defined. Depending on the input, some processing needs to be done and output is delivered. For example, keyboards, mouse, washing machine, digital cam, pen driver, remote, microwave, cars, telephone, mobiles, watches,etc. Since the applications are very specific, they need small resources like RAM, ROM, I/O ports etc and hence can be embedded on a single chip.This in turn reduces the size and the cost.

**Camera :**The Raspberry Pi Camera Module is a custom designed add-on for Raspberry Pi. It attaches to Raspberry Pi by way of one of the two small sockets on the board upper surface. This interface uses the dedicated CSI interface, which was designed especially for interface to camera

**Alarm** :It is basically a small homemade alarm controlled by a sensor .There is also the opportunity to take a photo through the webcam when the sensor detects movement. When the sensor detects movement, through GPIO and play an alarm sound.

**Relay :**A relay is an electrically operated switch. These switches can be extremely useful for a variety of Raspberry Pi projects (think turning on a light ). Watch the video below to learn how to drive a relay using your Raspberry Pi..

**2.1.2 Software Requirements**

 **Raspbian(O.S)**:The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python.

 **Python**: Python is a wonderful and powerful programming language that's easy to use (easy to read and write) and with Raspberry Pi lets you connect your project to the real world.

 **Open CV(Module):** The most commonly used library for Computer vision is Open CV. It is an open source free to use cross-platform library from Intel, meaning it could work on every operating system like windows, mac or Linux.

 **Numpy :**It is a library for the Python, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim with contributions from several other developers. In 2005, Travis created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-software and has many contributors.

 **Imutils :** A series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, displaying Matplotlib images, sorting contours, detecting edges, and much more easier with OpenCV and both Python 2.7 and Python 3.

 **Dlib :** Dlib is a general purpose cross-platform software library written in the programming language python. Its design is heavily influenced by ideas from design by contract and component-based software engineering. Thus it is, first and foremost, a set of independent software components

 **Time :** Time module.

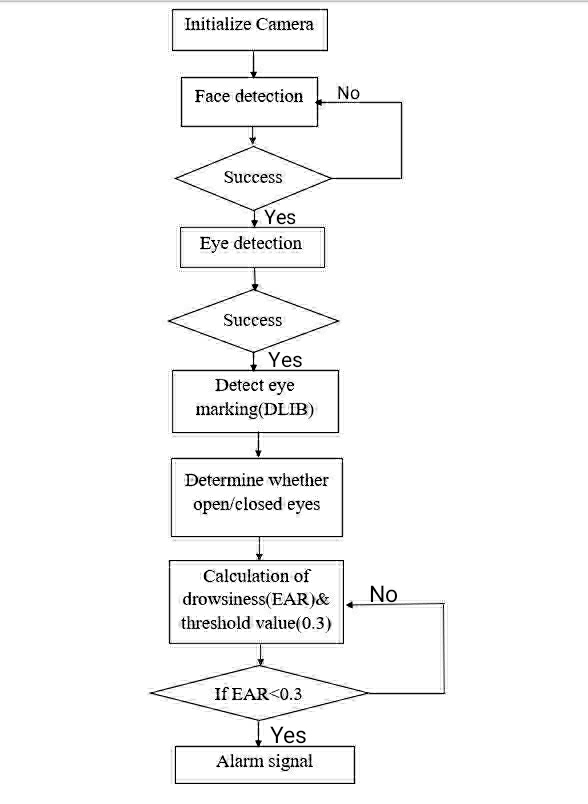
**2.1.3 Working Platform**

Operating Systems : Raspberry Pi

**2.2 Software Requirement Specification**

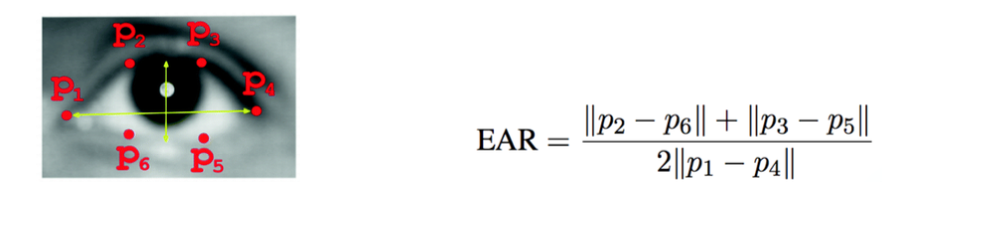
**2.2.1 Product Perspective**

The product is an open source. Cross platform support: Offers operating support for most of the known and commercial operating systems. It is considered that the user does have the basic knowledge of operating the internet and to have access to it. The administrator is expected to be familiar with the interface of the tech support system.



**EAR(EYE ASPECT RATIO)**

**ALOGORITHM CALCULATION**



**Figure .2.1**. EAR Algorithm Calculation

**2.2.2 User Classes and Characteristics**

**Complete:-**

Complete requirements specification is precisely define all the real world situations that will be encountered and the capability‟s responses to them. It will not include situations that will not be encountered or capability features.

**Consistent:-**

System functions and performance level is compatible and with required quality features (reliability, safety, security, etc.) It will not contradict the utility of the system. For example, the only aircraft that is totally safe is one that cannot be started, contains no fuel or other liquids, and is securely tied down.

**Modifiable:-**

Related concerns are grouped together and unrelated concerns are separated.

#### Traceable:-

#### Each requirement stated within the SRS document is uniquely identified to achieve traceability. Uniqueness is facilitated by the use of a consistent and logical scheme for assigning identification to each specification statement within the requirements document.

**2.2.3 Design and Implementation Constraints**

The interest in equipping vehicles with driver drowsiness detection systems has been motivated by alarming statistics, such as the 2013

World Health Organization report stating that 1.24 million people die on the road every year approximately 6% of all the accidents are caused by drivers driving in a drowsy state and most of the accidents of this type result in fatalities.

Some of the limitations: Accuracy, Robustness, Lighting Conditions, Camera Motion, Relative Positioning of Device, Driver Cooperation.

**2.2.4 Assumptions and Dependencies**

**Assumptions:**

 Sleep-deprived or fatigued (six hours of sleep or less triples your risk)

 Driving longer than 100 miles or two hours without proper rest breaks

 Driving alone — having a companion can help you stay alert

 Driving through the night, mid- afternoon or when you would normally be asleep

 Taking sedating medications (e.g., antidepressants, cold tablets, antihistamines)

 Suffering from insomnia or poor quality sleep

 Drinking even small amounts of alcohol

**Dependencies:**

 Difficulty focusing, frequent blinking or heavy eye lids

 Yawning repeatedly or rubbing your eyes

 Trouble keeping your head up

 Drifting from your lane, tailgating or hitting a shoulder rumble strip

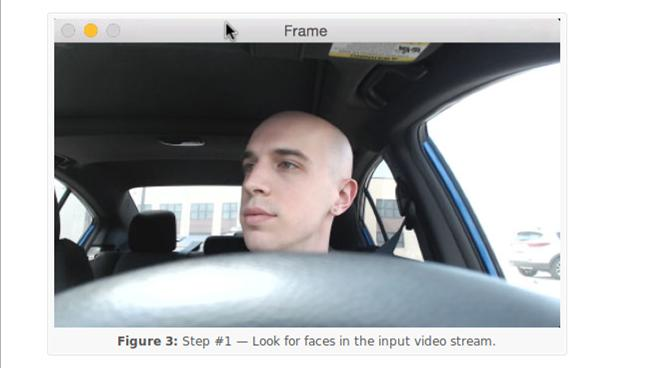
 Feeling restless and irritable

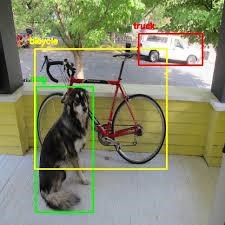
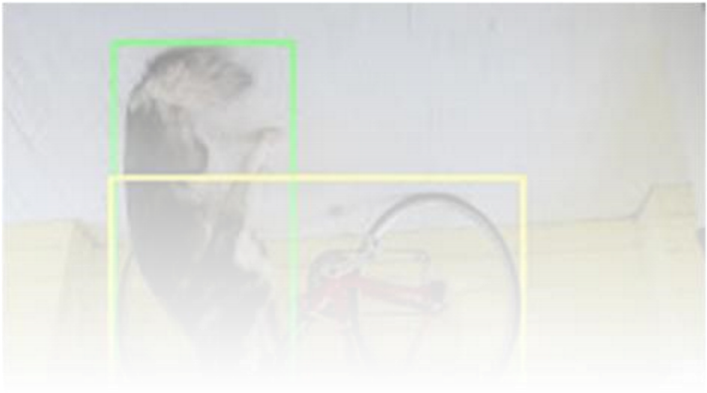
**2.2.5 External Interface Requirements**

**2.2.5.1 User Interfaces**

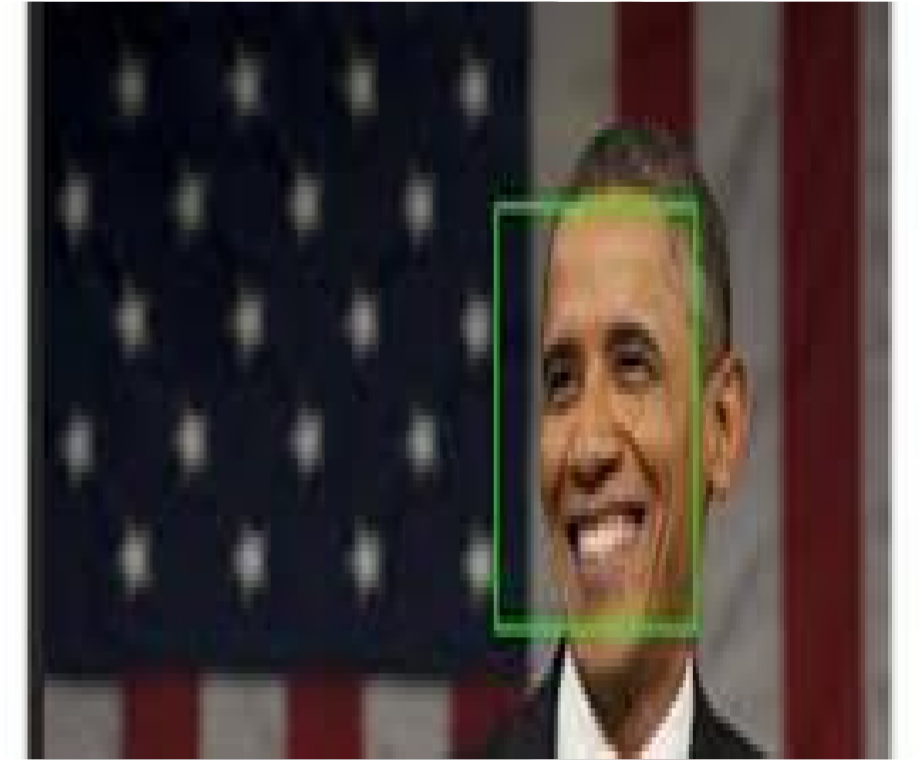
The System requires the raspberry Pi and a webcam as an interface for the user. The camera will detect the face and its features and base on the algorithm, it will be buzzering the alarm.

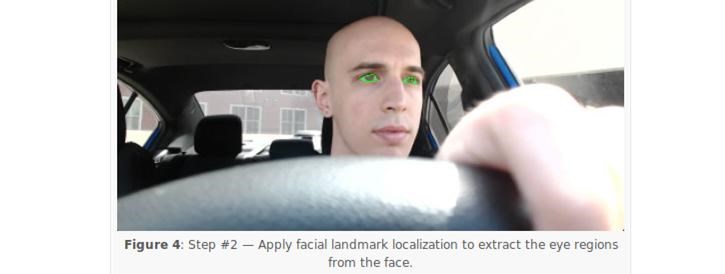
The following images will showcase the user interface and its internal implementation.



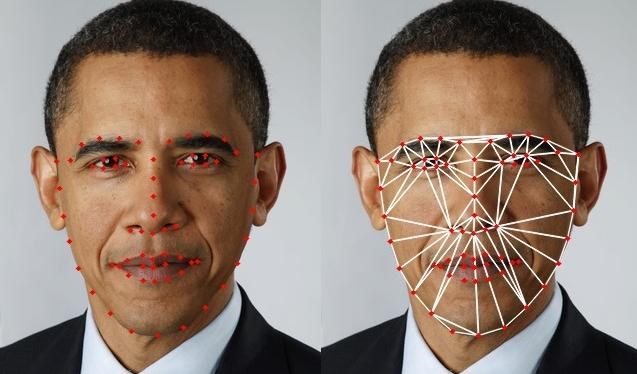
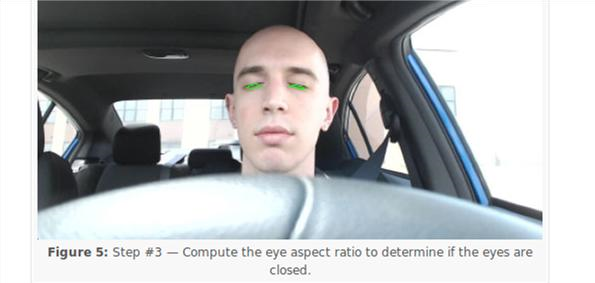


**Figure 2.1** Showing module 1





**Figure 2.2** Showing module 2



**Figure 2.3** showing module 3



**Figure 2.4** Showing module 4

**2.2.5.2 Hardware Interfaces**

 Micro Chip (8 GB)

 Web Camera

 Relay

 Alarm/Vibrator

**2.2.5.3 Software Interfaces**

 Python

 OpenCV(Module)

 Numpy

 Imutlis

 Dlib

 Time

**2.2.6 Other Non-Functional Requirements**

**2.2.6.1 Performance Requirements**

The software should have high performance and low failure rates. RaspberryPi should have all recent updates installed, and have their security not compromised by viruses. Machines must have firewalls installed and active virus scanning software in usage. Machines should solely be used for operation of the software, in order to maximize performance and security. Hardware components like, Camera should be ranging with high quality to ultra hd quality for better vision on driver.

**2.2.6.2 Security Requirements**

The data that is recognized by the system should be stored in any devices, as it compromises the data privacy of the drivers. RaspberryPi should have all recent updates installed, and have their security not compromised by viruses.

**2.2.6.3 Software Quality Attributes**

Flexibility, reusability, robustness, and maintainability of the system should be maximized, in order to detect any kind of face and its facial features.

**CHAPTER 3**

**SYSTEM DESIGN**

**3. SYSTEM DESIGN**

SYSTEM DESIGN phase follows analysis phase. Design is maintaining record of proof design divisions and providing a blueprint for the implementation phase. Design is the bridge between system analysis and system implementation.

System design is transition from a user oriented, document oriented to programmers. The design is a solution, a “how to” approach to the creation of a new system. This is composed of several steps. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Design goes through logical and physical stages of development; a logical design review the present physical system, prepare input and an output specification, detail the implementation plan, and prepares a logical design walkthrough.

**3.1 Design Methodology**

The design process for software system has two levels:

1. System design or Top level design

2. Detailed design or Logical design.

**3.1.1 System Design**

In the system design the focus is on deciding which modules are needed for the system, the specification of these modules and how these modules should be interconnected.

**3.1.2 Detailed Design**

In detailed design the interconnection of the modules or how the specification of the modules can be satisfied is decided. Some properties for a software system design are

1. Verifiability

2. Completeness

3. Consistency

4. Traceability

5. Simplicity / Understandability

**3.2 UML DIAGRAMS**

Unified Modeling Language is the language used to visualize, specify, construct and document any component of software engineering. The Unified Modeling Language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows:

**User Model View**

i. This view represents the system from the user‟s perspective.

ii. The analysis representation describes a usage scenario from the end users perspective.

**Structural model view**

i. In this model the data and functionality are arrived from inside the system.

ii. This model view models the static structures.

**Behavioral Model View**

It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

**Implementation Model View**

In this the structural and behavioral as parts of the system are represented as they are to be built.

**Environmental Model View**

In this structural and behavioral aspects of the environment in which the system is to be implemented are represented. UML is specifically constructed through two different domains they are

i. UML analysis modeling, which focuses on the user model and structural model views of the system.

ii. UML design modeling which focuses on the behavioral modeling, implementation modeling and environmental model views.

Every complex system is best approached through a small set of nearly independent views of a model; no single viewer is sufficient. Every model may be expressed at different levels of fidelity. The best models are connected to reality.

The UML includes diagrams such as:

**Static Diagrams include**

Class diagram

**Dynamic diagrams includes**

Use case diagram

Sequence diagram

State chart diagram

**Class diagram**

A class diagram shows a set of classes, interfaces,and collaborations and their relationships. These diagrams are the most common diagram found in modeling object-oriented systems. Class diagrams address the static design view of a system. Class diagrams that include active classes address the static process view of a system.

**Use case diagram**

A use case diagram shows a set of use cases and actors (a special kind of class) and their relationships. Use case diagrams address the static use case view of a system. These diagrams are especially important in organizing and modeling the behaviors of a system.

**Sequence diagram**

A sequence diagrams an interaction diagram that emphasizes the time ordering of messages. A sequence diagram shows a set of objects and the messages sent and received by those objects. Sequence diagrams are useful design tools because the provide a dynamic view of the system behavior which can be difficult to extract from static diagrams or specifications.

**State chart diagram**

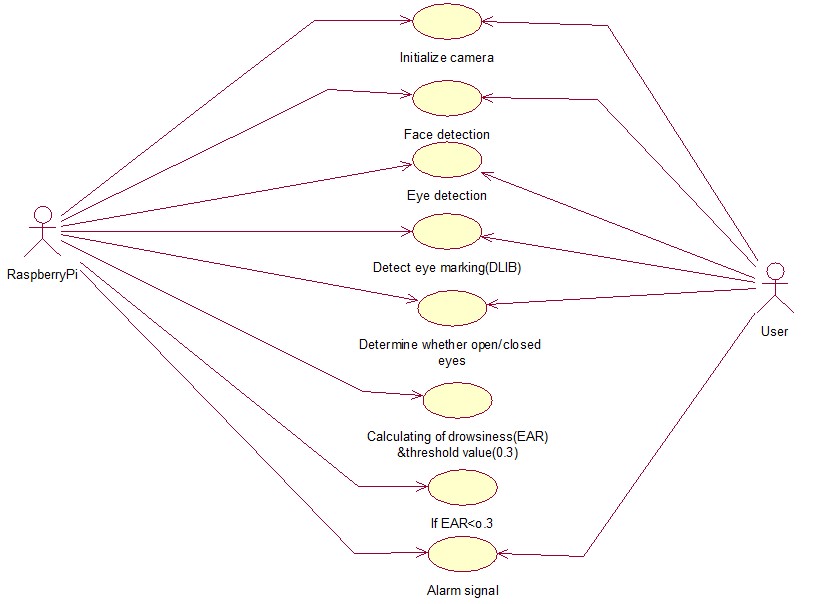
A state chart diagram shows a state machine, consisting of states, transitions, events, and activities. State chart diagrams address the dynamic view of a system. They are especially important in modeling the behavior of an interface, class, or collaboration and emphasize the event- ordered behavior of an object, which is especially useful in modeling reactive systems.

**3.2.1 Use Case Diagram**

Use case diagrams are created to visualize the relationships between actors and use cases. A use case is a pattern of behavior the system exhibits. Each use case is a sequence of related transactions performed by an actor and the system. A use case diagram is a collection of various use cases and different actors. Use case represents the verb form of a system .Use case is pictorially represented as elliptical form where the action performed by it is written inside the use case. For some systems use case has different types of relationships like includes and extends.

**Actor :**Actor in a use case diagram is any entity that performs a role in one given system. This could be a person, organization or an external system and usually drawn like skeleton.

**UseCase :** A use case represents a function or an action within the system. It is drawn as an oval and named with the function.



**Figure 3.1** Use case diagram

**3.3.2 Class Diagram**

A Class diagrams describe the static structure of a system, or how it is structured rather than how it behaves. Class diagram gives an overview of a system by showing its classes and the relationships among them. UML class is a rectangle divided into:class name, attributes, and operations. Our class diagram has three kinds of relationships.

**Association**-A relationship between instances of the two classes. There is an association between two classes if an instance of one class must know about the other in order to perform its work.

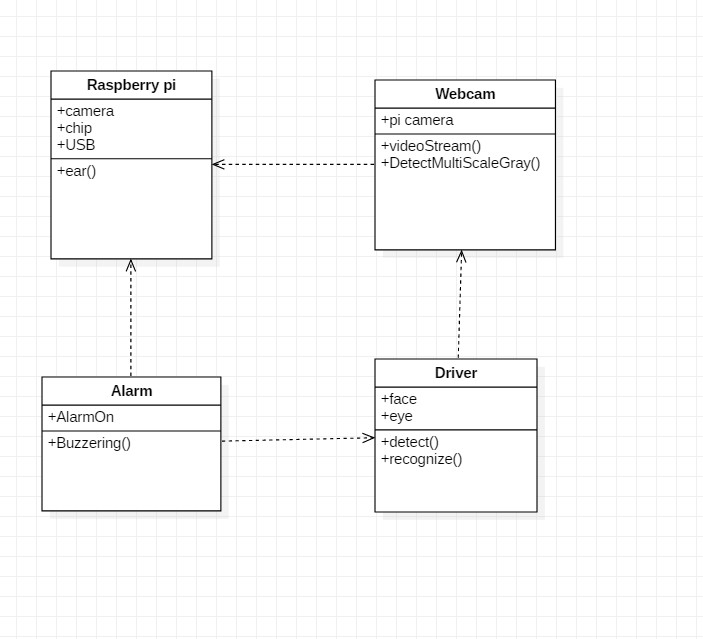
**Aggregation** - An association in which one class belongs to a collection. An aggregation has a diamond end pointing to the part containing the whole. In our diagram, Order has a collection of Order Details.

**Generalization** -An inheritance link indicating one class is a super class of the other.

Class diagrams are widely used to describe the types of objects in a system and their relationships. Class diagrams model class structure and contents using design elements such as classes, packages and objects. Class diagrams describe three different perspectives when designing a system, conceptual, specification, and implementation. These perspectives become evident as the diagram is created and help solidify the design. This example is only meant as an introduction to the UML and class diagrams. If you would like to learn more see the Resources page for more detailed resources on UML.

Classes are composed of three things: a name, attributes, and operations. Below is an example of a class.

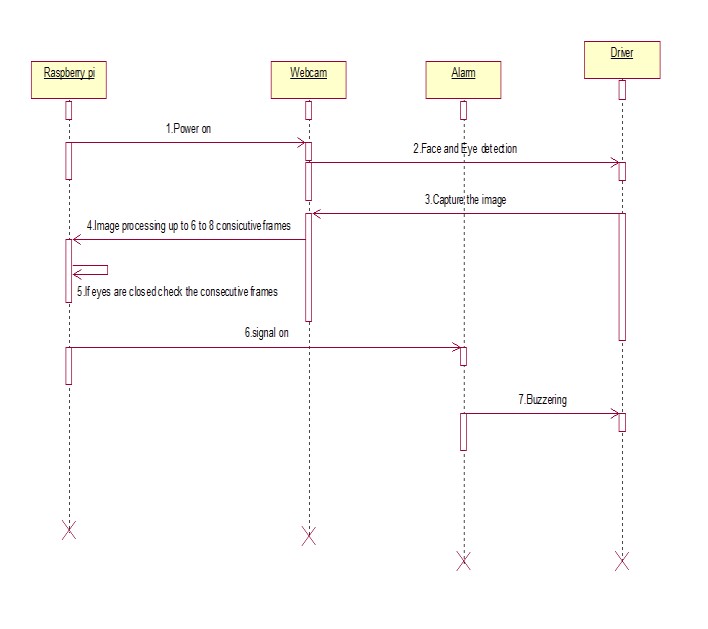
Class diagrams are the mainstay of object-oriented analysis and design. UML class diagrams show the classes of the system, their interrelationships (including inheritance, aggregation, and association), and the operations and attributes of the classes. Class diagrams are used for a wide variety of purposes, including both conceptual/domain modeling and detailed design modeling



**Fig 3.2** Class Diagram

**3.3.3 Sequence Diagram**

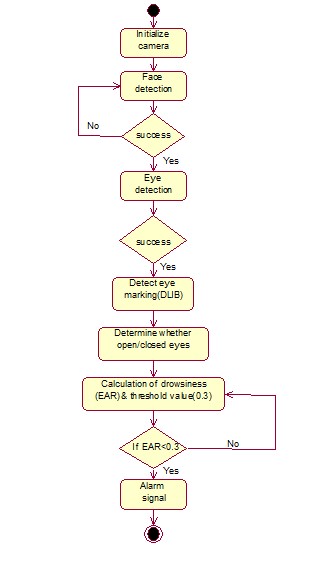
A type of interaction diagram, a sequence diagram shows the actors of the object participating in an interaction and the events they generate arranged in a time sequence. Often a sequence diagram shows the events that results from a particular instance of a particular instance of a use case but a sequence diagram can also exist in a more generic form. The vertical dimension in a sequence diagram represents time, with time preceding down the page the horizontal dimension represents different actors.



**Figure 3.3** Sequence Diagram

**3.3.4 Activity diagram**

Activity diagramis UML behaviour diagram which shows flow of control or object flow with emphasis on the sequence and conditions of the flow. The actions coordinated by activity models can be initiated because other actions finish executing, because objects and data become available, or because some events external to the flow occurs.



**Figure 3.4** Activity diagram

**CHAPTER 4**

**MODULES**

**4.MODULES**

**4.1 Module Description**

It includes 4 basic modules. They are:

1. Face Detection

2. Eyes Detection

3. Face Marking

4. Alarm

**4.1.1 Face Detection**

Face trackingsystem must be robust to head movement, rotation, pose variation and illumination changes. To achieve this goal we propose a method to use face detection and object tracking systems simultaneously which is based on machine learning approach for visual object detection. In this module, a classifier called haarcascade is used to detect the object i.e. face. The driver‟s face will act as an object and in front of the camera which will monitor if any object is residing or not. If the camera is not able to detect any object, it will the message „not detected‟, as soon as it detect the face it will further look for the eye detection.

**4.1.2 Eyes Detection**

Eye detectionis difficult task due to many factors such as lighting condition, expression, facial shadowing, etc. After the position of face has been obtained, locating the eye can be done with better accuracy. If eye could not be detected we can assume that the driver don‟t look at forward. So this situation can be categorized in distraction state and must alarms the driver. It also applicable for nose and mouth. In this module, the camera will detect the eyes. If eyes not found, it will show the message not detected‟.

**4.1.3 Image Processing**

**Image processing** method is used for drowsiness detection. The system should detect the state of the driver as early as possible and the false detection rate should be reduced. The driver inattentiveness is detected by continuously monitoring the driver face. When the system fails to detect the face then it decides that the driver is in attentive. The system uses a video camera to continuously capture the face of the driver. The video is converted to frames and certain pre- processing is done. In this module, Dlib is used for detecting the facial landmarks or points on calculation of which verifies the eye aspect ratios. If the eye aspect ratio is less than the specified threshold value, it will generate the signal.

**4.1.4 Alarm**

In this module, the detection of drowsy eyes will lead to generate the alarm as the signal.

**CHAPTER 5**

**SYSTEM IMPLEMENTATION**

**5.SYSTEM IMPLEMENTATION**

**5.1 Selected Software**

The following are the software components:

• Raspbian

• Python

• opencv

**5.1.1 RASPBIAN:**

Raspbianis a Computer operating system for RaspberryPI. There are several versions of Raspbian including Raspbian Stretch and Raspbian Jessie. Since 2015 it has been officially provided by the RaspberryPI Foundation as the primary operating system for the family of Raspberry Pi Single board computers. Raspbian was created by Mike Thompson and Peter Green as an independent project. The initial build was completed in June 2012. The operating system is still under active development. Raspbian is highly optimized for the Raspberry Pi line's low-performance ARM CPUs.

**5.1.2 PYTHON:**

Pythonis an interpreter, high-level, general purpose programming language. Created by [Gui](https://en.wikipedia.org/wiki/Guido_van_Rossum)do [van Rossum a](https://en.wikipedia.org/wiki/Guido_van_Rossum)nd first released in 1991, Python has a design philosophy that emphasizes code readability, notably using indentation. It provides constructs that enable clear programming on both small and large scales. In July 2018, Van Rossum stepped down as the leader in the language community.

**5.1.3 Open CV:**

Open CV (Open source computer vision) is a [library of programming functions main](https://en.wikipedia.org/wiki/Library_(computing))ly aimed at real-time [computer vision](https://en.wikipedia.org/wiki/Computer_vision)[.[1] O](https://en.wikipedia.org/wiki/OpenCV#cite_note-1)riginally developed b[y I](https://en.wikipedia.org/wiki/Intel_Corporation)[ntel,[2] it](https://en.wikipedia.org/wiki/OpenCV#cite_note-2) was later supported b[y Willow Garage](https://en.wikipedia.org/wiki/Willow_Garage) then Itseez (which was later acquired by Intel[[3]).](https://en.wikipedia.org/wiki/OpenCV#cite_note-3) The library is [cross-platform a](https://en.wikipedia.org/wiki/Cross-platform)nd free for use under the [open-source. O](https://en.wikipedia.org/wiki/Open-source_software)penCV supports the [dee](https://en.wikipedia.org/wiki/Deep_learning)p [learning.](https://en.wikipedia.org/wiki/Deep_learning)

**5.1.4 NUMPY:**

NumPy is a library for the Python, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high- level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim with contributions from several other developers. In 2005, Travis created NumPy by incorporating features of the competing Num array into Numeric, with extensive modifications. NumPy is open-software and has many contributors.

**5.1.5 SCIPY:**

SciPy builds on the NumPy array object and is part of the NumPy stack which includes tools like pandas and an expanding set of scientific computing libraries. This NumPy stack has similar users to other applications such as MATLAB, andScilab. The NumPy stack is also sometimes referred to as the SciPy stack.

**5.1.6 DLIB:**

Dlib is a general purpose cross-platform software library written in the programming language python. Its design is heavily influenced by ideas from design by contract and component-based software engineering. Thus it is, first and foremost, a set of independent software component

**5.2 Sample Code:**

# USAGE

# python pi\_detect\_drowsiness.py --cascade haarcascade\_frontalface\_default.xml --shapepredictor shape\_predictor\_68\_face\_landmarks.dat

# python pi\_detect\_drowsiness.py --cascade haarcascade\_frontalface\_default.xml -shape-predictor shape\_predictor\_68\_face\_landmarks.dat -alarm 1

# import the necessary packages from imutils.video import VideoStream from imutils import face\_utils import numpy as np import argparse import imutils import time import dlib import cv2 import RPi.GPIO as GPIO GPIO.setmode(GPIO.BCM)

def euclidean\_dist(ptA, ptB):

# compute and return the euclidean distance between the two

# points return np.linalg.norm(ptA - ptB)

def eye\_aspect\_ratio(eye):

# compute the euclidean distances between the two sets of

# vertical eye landmarks (x, y)-coordinates

A= euclidean\_dist(eye[1], eye[5])

A= euclidean\_dist(eye[2], eye[4])

# compute the euclidean distance between the horizontal

# eye landmark (x, y)-coordinates

A= euclidean\_dist(eye[0], eye[3])

# compute the eye aspect ratio ear = (A + B) / (2.0 \* C)

# return the eye aspect ratio

return ear

# construct the argument parse and parse the arguments ap = argparse.ArgumentParser() ap.add\_argument("-c", "--cascade", required=True, help = "path to where the face cascade resides")

ap.add\_argument("-p","-shape-predictor", required=True, help="path to facial landmark predictor")

ap.add\_argument("-a","-alarm", type=int, default=0,

help="boolean used to indicate if TraffHat should be used")

args = vars(ap.parse\_args())

# check to see if we are using GPIO/TrafficHat as an alarm if args["alarm"] > 0: from gpiozero import TrafficHat th = TrafficHat()

print("[INFO] using TrafficHat alarm.")

# define two constants, one for the eye aspect ratio to indicate

# blink and then a second constant for the number of consecutive

# frames the eye must be below the threshold for to set off the

# alarm

EYE\_AR\_THRESH = 0.35

EYE\_AR\_CONSEC\_FRAMES = 16

# initialize the frame counter as well as a boolean used to

# indicate if the alarm is going off

COUNTER = 0

ALARM\_ON = False

# load OpenCV's Haar cascade for face detection (which is faster than

# dlib's built-in HOG detector, but less accurate), then create the

# facial landmark predictor print("[INFO] loading facial landmark predictor.")

detector = cv2.CascadeClassifier(args["cascade"]) predictor = dlib.shape\_predictor(args["shape\_predictor"])

# grab the indexes of the facial landmarks for the left and # right eye, respectively

(lStart,lEnd)= face\_utils.FACIAL\_LANDMARKS\_IDXS["left\_eye"] (rStart,rEnd)= face\_utils.FACIAL\_LANDMARKS\_IDXS["right\_eye"]

# start the video stream thread print("[INFO] starting video stream thread...") vs = VideoStream(src=0).start()

# vs=VideoStream(usePiCamera=True).start() time.sleep(1.0)

# loop over frames from the video stream

while True:

# grab the frame from the threaded video file stream, resize

# it, and convert it to grayscale

#channels) frame = vs.read() frame = imutils.resize(frame, width=450) gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

# detect faces in the grayscale frame

rects = detector.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30),

flags=cv2.CASCADE\_SCALE\_IMAGE)

# loop over the face detections for (x, y, w, h) in rects:

# construct a dlib rectangle object from the Haar cascade

# bounding box

rect = dlib.rectangle(int(x), int(y), int(x + w), int(y + h))

# determine the facial landmarks for the face region, then

# convert the facial landmark (x, y)coordinates to a NumPy

# array shape = predictor(gray, rect) shape = face\_utils.shape\_to\_np(shape)

# extract the left and right eye coordinates, then use the

# coordinates to compute the eye aspect ratio for both eyes leftEye = shape[lStart:lEnd] rightEye = shape[rStart:rEnd] leftEAR = eye\_aspect\_ratio(leftEye) rightEAR = eye\_aspect\_ratio(rightEye)

# average the eye aspect ratio together for both eyes ear = (leftEAR + rightEAR) / 2.0

# compute the convex hull for the left and right eye, then

# visualize each of the eyes leftEyeHull = cv2.convexHull(leftEye) rightEyeHull = cv2.convexHull(rightEye) cv2.drawContours(frame, [leftEyeHull], -1, (0, 255, 0), 1)

cv2.drawContours(frame, [rightEyeHull], -1, (0, 255, 0), 1)

# check to see if the eye aspect ratio is below the blink

# threshold, and if so, increment the blink frame counter

if ear < EYE\_AR\_THRESH:

COUNTER += 1

# if the eyes were closed for a sufficient number of

# frames, then sound the alarm

if COUNTER >= EYE\_AR\_CONSEC\_FRAMES:

GPIO.setmode(GPIO.BCM)

GPIO.setup(17,GPIO.OUT) GPIO.output(17,GPIO.LOW)

print("detected")

# if the alarm is not on, turn it on if not ALARM\_ON: ALARM\_ON = True

# check to see if the TrafficHat buzzer should

# be sounded if args["alarm"] > 0: th.buzzer.blink(0.1, 0.1,10, background=True)

# draw an alarm on the frame

cv2.putText(frame, "DROWSINESS ALERT!", (10, 30),

cv2.FONT\_HERSHEY\_SIMPLEX, 0.7, (0, 0, 255), 2)

# otherwise, the eye aspect ratio is not below the blink

# threshold, so reset the counter and alarm

else:

COUNTER = 0

GPIO.setmode(GPIO.BCM)

GPIO.setup(17,GPIO.OUT) GPIO.output(17,GPIO.HIGH) print(" Not detected") GPIO.cleanup()

ALARM\_ON = False

# draw the computed eye aspect ratio on the frame to help

# with debugging and setting the correct eye aspect ratio

# thresholds and frame counters cv2.putText(frame, "EAR: {:.3f}".format(ear), (300, 30),

cv2.FONT\_HERSHEY\_SIMPLEX, 0.7, (0, 0, 255), 2)

# show the frame cv2.imshow("Frame", frame)

key = cv2.waitKey(1) & 0xFF

# if the `q` key was pressed, break from the loop if key == ord("q"):

break

# do a bit of cleanup cv2.destroyAllWindows() vs.stop()

**CHAPTER 6**

**TEST CASES**

**6.TEST CASES**

**6.1 Software Testing**

In any software development, testing is a process to show the correctness of program and it needs the design specifications. Testing is needed to prove correctness completeness, to improve the quality of the software and to provide the maintenance aid. Some testing standards are therefore necessary to ensure completeness of testing, improve the quality of software and reduce the testing costs and to reduce study needs and operation time.

**6.2 Goals of Testing**

The following are goals of testing…

1. Testing is a process of executing a program with the intent of finding error.

2. A good test case is the one that has a high probability of finding an as at undiscovered error.

3. A successful test is one that uncovers an as at undiscovered error.

**6.3 Testing Methodology**

**6.3.1 Black box testing**

Black Box Testing is the testing process in which tester can perform testing on an application without having any internal structural knowledge of application. Usually Test Engineers are involved in the black box testing.

**6.3.2. White box testing**

White Box Testing is the testing process in which tester can perform testing on an application with having internal structural knowledge. Usually the developers are involved in the white box testing.

**6.3.3. Gray box testing**

Gray Box Testing is the process in which the combination of black box and white box techniques is use.

**6.4 Levels of Testing**

**6.4.1. Unit testing**

Individual components are tested to ensure that they operate correctly. Each component is tested independently without other system components.

**6.4.2. System testing**

The sub-systems are integrated to makeup the entire system. The testing process is concerned with finding errors, which result from un-anticipated interactions between subsystem components.

**6.4.3. Integration testing**

Sometimes global data structures can represent the problems to uncover errors that are associated with interfacing the objective is to make unit test modules and built a program structure that has been detected by design.

**6.4.4. Acceptance testing**

This is the final stage in the testing process before the system is accepted for operational use. Acceptance testing may reveal errors and omissions in the system requirements definition because real data exercises the system in different ways from the test data.

**6.4.5. Regression testing**

Regression testing is actually that helps to ensure changes that don‟t introduce unintended behavior as additional errors. Regression testing may be conducted manually by executing a subset of all test cases or using automated capture play back tools.

**6.5 Unit Test Cases**

**Test case for Functional requirements:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TC ID** | **Input** | **Descripti on** | **Expected**  **Results** | **Pass/ Fail** |
| TC01 | Driver Facing the camera | This Will check  whether  the face is being detected or not | Green Box around the face will be appeared | Pass |
| TC02 | Driver Facing  left to frame | This Will check whether  the face is being detected  or not | Green Box around the face will be appeared | Pass |
| TC03 | Driver Facing center to frame | This Will check whether  the face is being detected  or not | Green Box around the face will be appeared | Pass |
| TC04 | Ambie  nt Lighti ng | The parameter of lighting and surroundi ngs will be considere  d | Face detection will be taken place | Pass |
| TC05 | Driver Facing right to frame | This Will check whether  the face is being detected  or not | Green Box around the face will be appeared | Pass |
| TC06 | Sleepy eyes | Checks weather the eyes are closed or not | Drowsiness  alert will be shown on screen | Pass |
| TC07 | Low  Lighti ng | The parameter of lighting and surroundi ngs will be considere d | Face detection may not be taken place | Fail |
| TC08 | Differe nt  Postur es | Face will not be facing the camera | No face detection | Fail |

**TC ID Input Description Expected Results**

**Table 6.1** Test cases

**CHAPTER 7**

**SCREENSHOTS**

**7. SCREENSHOTS**

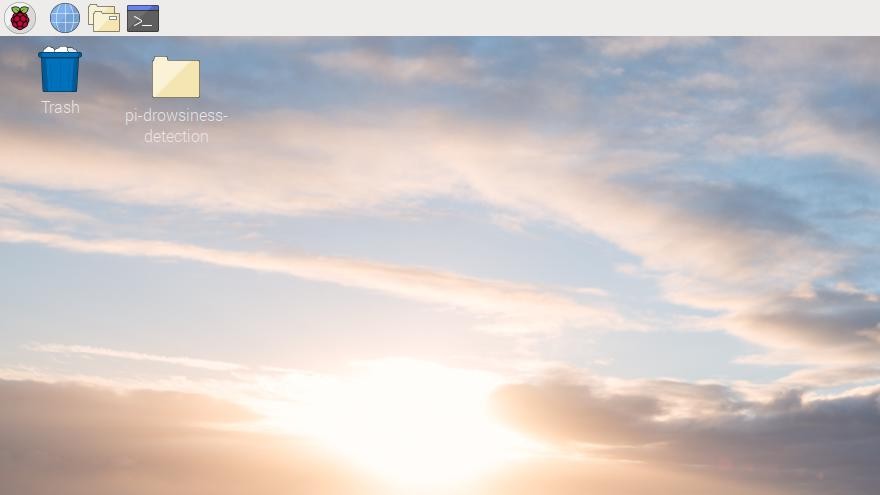
**Modules:**

 **Module 1 (Object detection)**

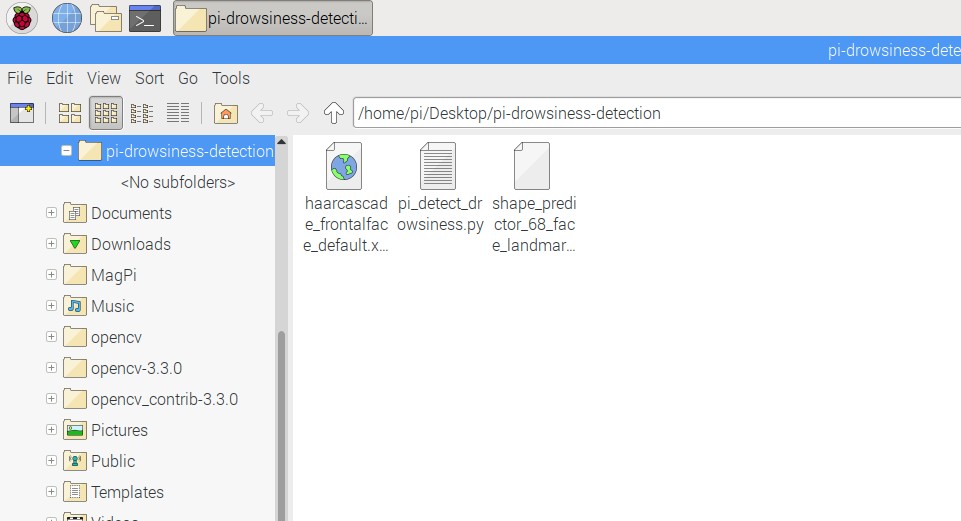
 **Module 2 ( Eyes, nose, mouth detection)**

 **Module 3**( **Image processing)**

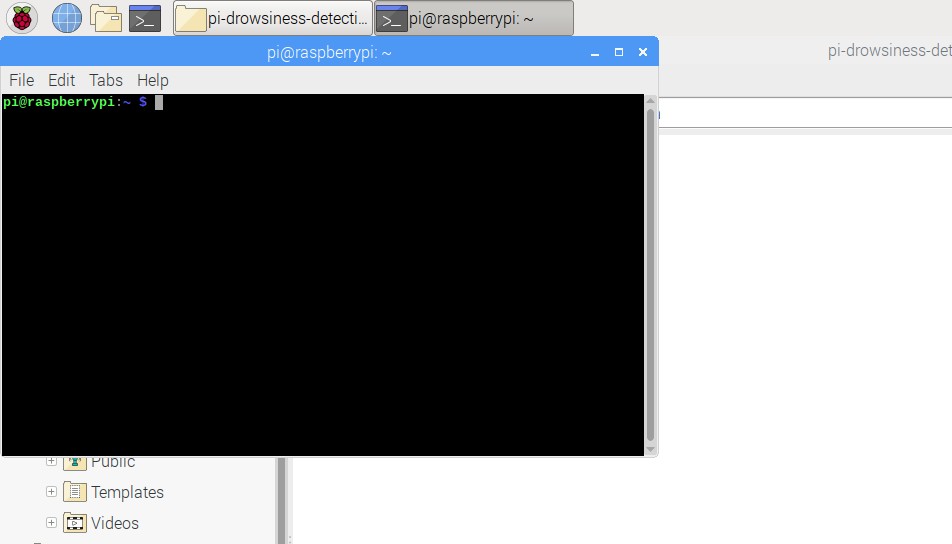
 **Module 4 ( Alarming)**



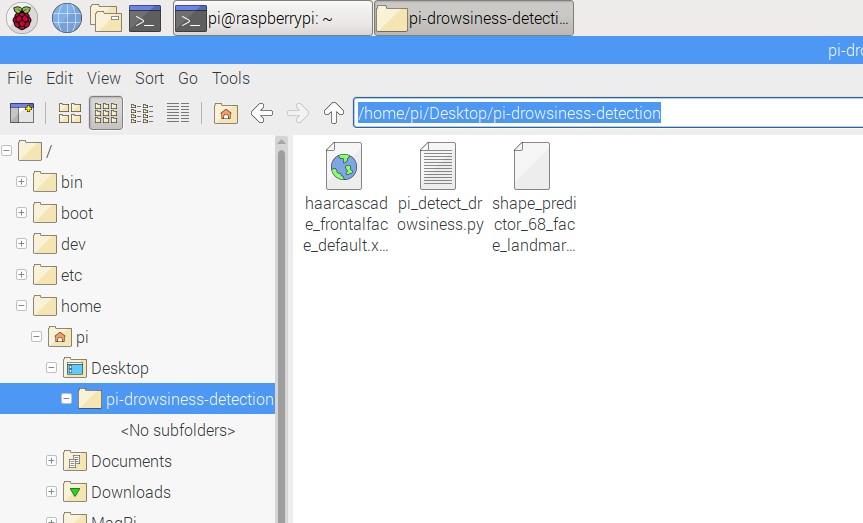
**Figure 7.1** Home Page



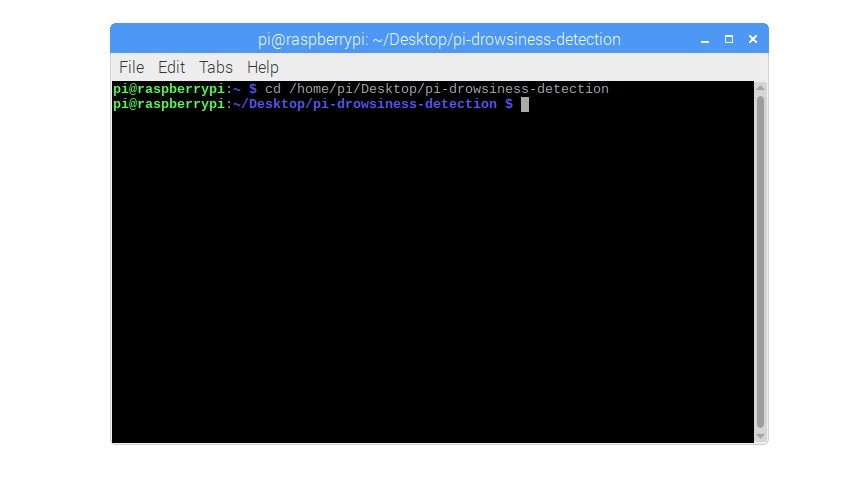
**Figure 7.2** sub page



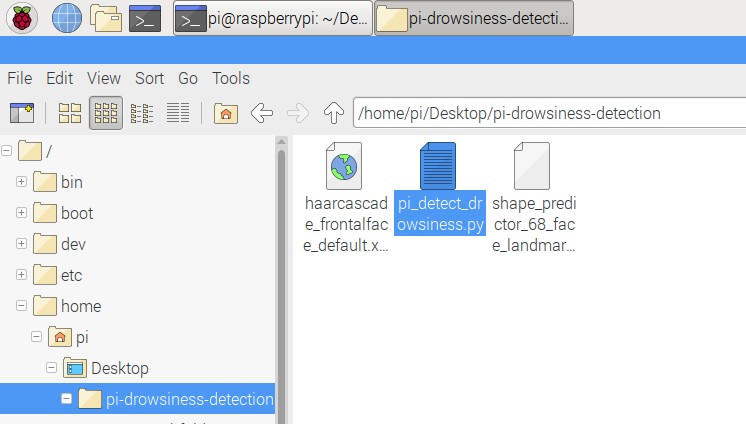
**Figure 7.3** command page



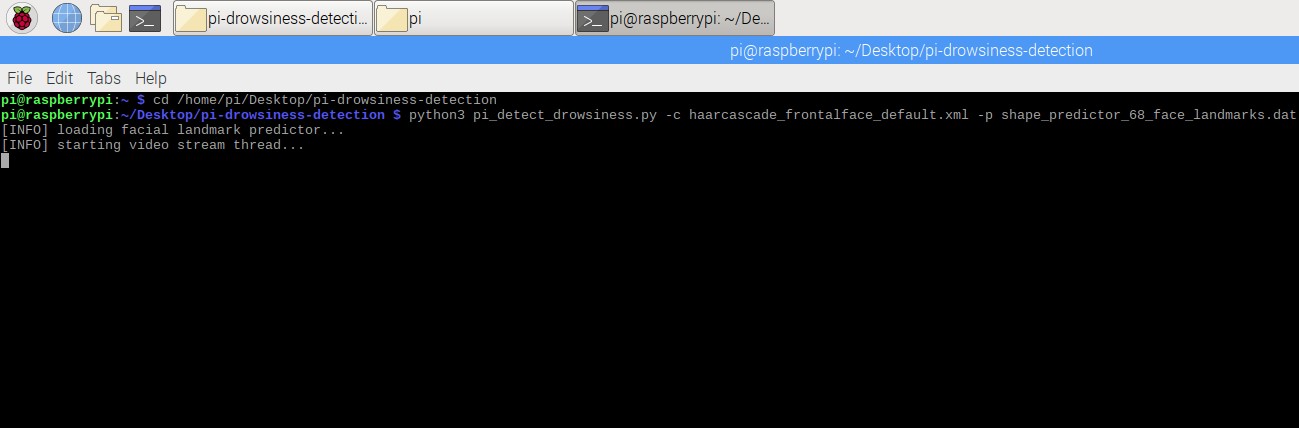
**Figure 7.4 Copy file page**



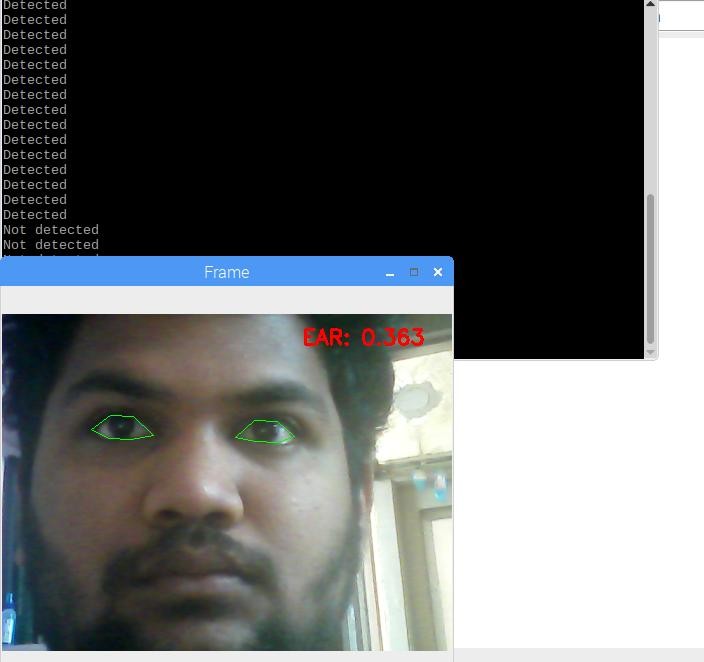
**Figure 7.5** Enter in command page



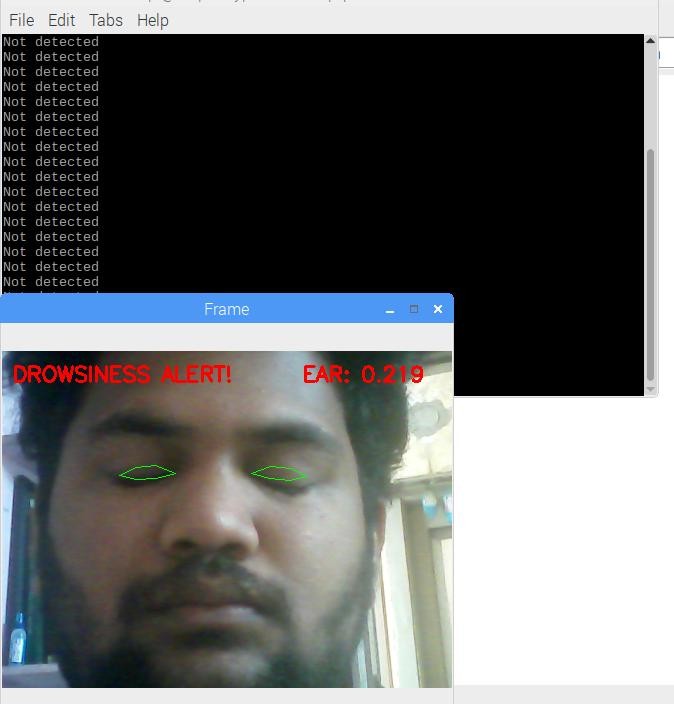
**Figure 7.6** Database folder



**Figure 7.7** Generate report



**Figure 7.8** Person recognition



**Figure 7.9** Person drowsiness alert

**CHAPTER 8**

**CONCLUSION & FUTURE SCOPE**

**8.CONCLUSION** **&** **FUTURE SCOPE**

**8.1 CONCLUSION**

The driver abnormality monitoring system developed is capable of detecting drowsiness, drunken and reckless behaviours of driver in a short time. The Drowsiness Detection System developed based on eye closure of the driver can differentiate normal eye blink and drowsiness and detect the drowsiness while driving. The proposed system can prevent the accidents due to the sleepiness while driving. The system works well even in case of drivers wearing spectacles and even under low light conditions if the camera delivers better output. Information about the head and eyes position is obtained through various self- developed image processing algorithms. During the monitoring, the system is able to decide if the eyes are opened or closed. When the eyes have been closed for too long, a warning signal is issued. processing judges the driver’s alertness level on the basis of continuous eye closures.

In this paper, a new adaptive method for symptom extraction and driver state estimation was proposed for driver hypovigilance detection. Two types of symptoms were considered:symptoms related to eye region (including PERCLOS, ELSDC, and CLOSNO) and symptom related to face region (ROT). The proposed method extracts the symptoms related to eye region using horizontal projection of top-half segment without explicit eye detection; the symptom related to face region is extracted based on face template matching. Then, the normal value of the extracted symptoms is calculated during a short training phase. According to the normal value of the extracted features, an adaptive fuzzy expert system estimates the level of fatigue and distraction.

The short training phase makes the system robust and adaptive. In other words, the proposed system may be used efficiently for different individuals with different face and eyelid behaviors. Experiments show that the accuracy of the proposed method for extracting the symptoms of driver fatigue and distraction is very good. Additionally, the system can estimate the driver fatigue and distraction very well by subjective evaluation.

The proposed method was also tested on video sequences captured in visible spectrum, but the color information was not used in any part of the system. In other words, the proposed system operates in gray-level visible spectrum. Therefore, the system may operate in IR spectrum with a few changes. The main disadvantage of our system is the face tracking method which is inaccurate and very computationally complex. Adaptive filters such as Kalman filter may reduce the complexity and increase the processing speed and accuracy of the system.

**8.2 FUTURE SCOPE**

All the drawbacks mentioned in section 2.5 have been eliminated. In future, this prototype can be extended to give alarm before sleeping by calculating the thresholdlimit measure without physical disturbance i.e., non intrusive method using modified EAR methods. Usually in EYE ASPECT RATIO method key points of Face (For example eyes,mouth etc.,) are sticked with wire. In the extended method, sticking wire may be avoided. This will lead us to a way to find out the optimum level of drowsiness.

Further, this prototype will be extended to monitor the reflect ray from eye using nano camera. If the reflection ray is absent, then eye is closed otherwise eye is opened. We believe that this will create a better opportunity to detect drowsiness.

**8.3 Applications:**

 The proposed system is used to avoidvarious road accidents caused by drowsy driving.

 This system is used for security purpose of a driver to caution the driver if any fire accident or any gas leakage occur.

 Drowsy driver detection methods can form the basis of a system to potentially reduce the number of crashes related to drowsy driving.

 In future we can implement drowsiness detection system in aircraft in order to alert pilot.

 The alcoholic sensor is also used for drunk drivers.

**CHAPTER 9**

**BIBLIOGRAPHY**

**9.BIBLIOGRAPHY**

**9.1 References made from:**

[1] Miaou, “Study of Vehicle Scrap page Rates,” Oak Ridge National Laboratory, Oak Ridge, TN,,S.P.,April 2012.

[2] Wreggit, S. S., Kim, C. L., and Wierwille, W. W., Fourth Semi-Annual Research Report”, Research on Vehicle-Based Driver Status Performance Monitoring”, Blacksburg, VA: Virginia Polytechnic Institute and State University, ISE Department, January 2013.

[3] Bill Fleming, “New Automotive Electronics Technologies”, International Conference on Pattern Recognition, pp. 484- 488,December 2012.

[4] Ann Williamson and Tim Chamberlain,“Review of on-road driver fatigue monitoring devices”, NSW Injury Risk Management Research Centre, University of New South Wales, , July 2013.

[5] E. Rogado, J.L. García, R. Barea, L.M. Bergasa, Member IEEE and E. López, February, 2013, “Driver Fatigue Detection System”, Proceedings of the IEEE International Conference on Robotics and Biometics, Bangkok, Thailand.

[6] Boon-Giin Lee and Wan-Young Chung, Member IEEE, “Driver Alertness Monitoring Using Fusion of Facial Features and Bio-Signals”, IEEE Sensors Journal, VOL. 12, NO. 7, July 2012.

[7] H. Singh, J. S. Bhatia, and J. Kaur, “Eye tracking based driver fatigue monitoring and warning system”, in Proc. IEEE IICPE, New Delhi, India, Jan. 2014.

[8] Eriksson, M and Papanikolopoulos, N.P, “Eye-tracking for Detection of Driver Fatigue”, IEEE Intelligent Transport System Proceedings, pp 314-319, September 2014.

[9] Perez, Claudio A. et al,“Face and Eye Tracking Algorithm Based on Digital Image Processing”, IEEE System, Man and Cybernetics Conference, vol. 2 ,pp 1178-1188,, February 2012.

[10] Magdalena Lindman1, JordankaKovaceva, Daniel Levin, Bo Svanberg,Lotta Jakobsson, Henrik Wiberg,“A first glance at Driver Alert Control in FOT‐data”, IRCOBI Conference,August 2012 .

[11] Singh, Sarbjit and Papanikolopoulos, N.P, “Monitoring Driver Fatigue Using Facial Analysis Techniques”, IEEE Intelligent Transport System Proceedings, pp 314-318,April 2013.

[12] T. Brand, R. Stemmer, B. Mertsching, and A. Rakotonirainy, Affordable visual DRIVER MONITOING AND ALERTING SYSTEM for fatigue and monotony;' in Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, vol. 7, pp. 6451-6456, October 2004.

[13] NHTSA, "Evaluation of techniques for ocular measurement as an index of fatigue and the basis for alertness management, " National Highway Traffic Safety Administration, Washington DC, USA, Technical Report DOT HS 808762, 1998.

[14]L.Hagenmeyer, "Development of a multimodal, universal humanmachine interface for hypovigilance-management systems;' PhD Thesis,University of Stuttgart, Stuttgart, Germany,2007.

[15] M. Sarshar and M. Rezaei, "A novel system for advanced driver assistance systems;' In Proceedings of" the IEEE International Conference on Systems, pp. 529-534, 2010