Drowsiness Detection using EAR

Submitted by

Charan Royston Lobo
182006
MSc Big Data Analytics
AIMIT, St. Aloysius college
Mangalore

Submitted in Partial Fulfillment of the Requirements for the Award of the Degree of

Master of Big Data Analytics (MSc)

Under the guidance of

Ms. Laveena D’Costa
HOD ,Department of MSc Big Data Analytics
AIMIT, St. Aloysius College
Mangalore

Submitted to

ALOYSIUS INSTITUTE OF MANAGEMENT AND INFORMATION TECHNOLOGY (AIMIT)
ST ALOYSIUS COLLEGE (AUTONOMOUS)
MANGALURU, KARNATAKA

2020

ALOYSIUS INSTITUTE OF MANAGEMNT AND INFORMATION TECHNOLOGY

ST ALOYSIUS COLLEGE (AUTONOMOUS)

MANGALORE, KARNATAKA

CERTIFICATE

This is to certify that the project titled

Drowsiness detection using EAR

Submitted by

Charan Royston Lobo
182006
MSc Big Data Analytics
AIMIT, St. Aloysius College
Mangalore

In partial fulfillment of the requirements for award of degree of Master of big data analytics of Mangalore University, is a bonafide record of the work carried out at

iHub IT Solutions,
Soorya Infratech Park,
Bakrabail Rd,SEZ Mudipu,
Kurnad,Karnataka 574153
During the year

2020

Ms. Laveena D’Costa
Asst Professor
HOD of department of Bigdata analytics,
AIMIT, St. Aloysius College
Mangalore-575 022.

Rev. Fr. Denzil Lobo SJ
Director,
AIMIT, St Aloysius College
Mangalore-575 022.

Prof Santhosh Rebello, Dean

PG Depts of Information Technology,

AIMIT, St. Aloysius College,
Mangalore-575 022.

Examiners

1.

2.

ALOYSIUS INSTITUTE OF MANAGEMENT AND INFORMATION TECHNOLOGY
ST. ALOYSIUS COLLEGE (AUTONOMOUS)
BEERI, MANGALORE – 575 022
P G DEPT OF COMPUTER APPLICATIONS & SOFTWARE TECHNOLOGY

PROFORMA OF PROJECT PROPOSAL

Note : The student has to fill this form and approval has to be sought from the Dean of the faculty before starting the Industry Internship.(January I week)

[1]. Student Register Number : 182006

[2]. Student Name : Charan Royston Lobo

[3]. Title of the Project : Drowsiness detection using EAR

[4]. Name & Address of the Organization :

iHub IT Solutions, First Floor, Soorya Commercials, Near Soorya Infratech Park, Mudipu, Kurnadu,Karnataka, 574153

[5]. Telephone No. : 9562781634

[6]. E-mail : info@ihubonline.co / ihubitsolutions@gmail.com

[7]. Website / URL : www.ihubonline.co

[8]. Address of the Project Site : iHub IT Solutions

[9]. Telephone No. : 9562781634

[10]. E-mail : info@ihubonline.co

[11]. Residential Address of the Student : Upper Nekkare House, Pedamale Post, Kelrai Mangaluru-575029

[12]. Telephone No : NIL

[13]. Mobile Number : 7406832946

[14]. E-mail : charanlobo@gmail.com

I hereby declare that the above furnished details are true to best of my knowledge and request you to permit me to join the Industry Internship for the Project work prescribed for MSc Big Data Analytics.

Dated : Signature of the Student

The Proposal for the project work has been accepted **Dean of the Faculty**

CERTIFICATE OF AUTHENTICATED WORK

This is to certify that the domain seminar project report entitled Drowsiness detection using EAR **submitted to Aloysius Institute of Management and Information Technology (AIMIT), St Aloysius College, Mangalore affiliated to Mangalore University in partial fulfilment of the requirement for the award of the degree of MASTER OF SCIENCE (MSC BIG DATA ANALYTICS) is an original work carried out by** Mr. Charan Royston Lobo**,** **Register no.** 182006 under my guidance. The matter embodied in this domain seminar project is authentic and is genuine work done by the student and has not been submitted whether to this University, or to any other University / Institute for the fulfilment of the requirement of any course of study.

iHub IT Solutions
First Floor, Soorya IT Park
Mudipu SEZ, Mangalore, Karnataka, India
www.ihubonline.co

Signature of the Student:	Signature of the Guide
Date: 15 / 06 / 2020	Date: 15 / 06 / 2020

Charan Royston Lobo
182006

Upper Nekkare House
Pedamale Post, Kelrai
Mangaluru, 575029

Mrs. Rekaha R

Project Manager
Soorya Infratech Park,
Mudipu,
Mangalore Karnataka 574153
www.ihubonline.co
info@ihubonline.co

*iHub IT Solutions

Soorya Infratech Park,
Bakrabail Rd, SEZ Mudipu,
Kurnad, Karnataka 574153

Ref: Ihub0122020J
13-Jun -2020

TO WHOMSOEVER IT MAY BE COCERNED

This is to certify that **Mr. Charan Royston Lobo(Reg No :182006),** student from **MSc Big Data Analytics**, **St. Aloysius Institute of Management and Information Technology (AIMIT) Beeri , Mangalore** has successfully completed the Internship Program on Data Science (Machine learning and Deep learning) from 6th January 2020 for a duration of 5 months.

During this period, he was observed to be hard working and a good team player. His association with us was very fruitful and we wish him all the best in his future endeavors.

Sincerely,

Rekha R
Project Manager
Soorya Infratech Park, Mudipu,
Mangalore Karnataka 574153
www.ihubonline.co info@ihubonline.co

iHub IT Solutions
First Floor, Soorya IT Park
Mudipu SEZ, Mangalore, Karnataka, India
www.ihubonline.co

iHub ITSolutions

Innovate Automate Transform

ABSTRACT

"1 in 25 adult drivers report that they have fallen asleep at the wheel in the past 30 days". Additionally, we believe that drowsiness can negatively impact people in working and classroom environments as well. Although sleep deprivation and college go hand in hand, drowsiness in the workplace especially while working with heavy machinery may result in serious injuries similar to those that occur while driving drowsily. Our solution to this problem is to build a detection system that identifies key attributes of drowsiness such as eye moments and the mouth and triggers an alert when someone is drowsy before it is too late.

Project Proposal / Synopsis

I. Title of the Project

Drowsiness detection using EAR

II. Statement of the Problem

A countless number of people drive on the highway day and night. Taxi drivers, bus drivers, truck drivers and people traveling long-distance suffer from lack of sleep. Due to which it becomes very dangerous to drive when feeling sleepy. The majority of accidents happen due to the drowsiness of the driver.

III. Why this particular topic chosen?

We believe that drowsiness can negatively impact people in working and classroom environments as well. Although sleep deprivation and college go hand in hand, drowsiness in the workplace especially while working with heavy machinery may result in serious injuries similar to those that occur while driving drowsily. Our solution to this problem is to build a detection system that identifies key attributes of drowsiness and triggers an alert when someone is drowsy.

IV. Objective and Scope

Driver drowsiness detection is a car safety technology which helps to save the life of the driver by preventing accidents when the driver is getting drowsy. The main objective is to first design a system to detect driver's drowsiness by continuously monitoring the drivers eyes. The system works in various lighting conditions and alert the driver on the detection of drowsiness by using buzzer or alarm.

V. Methodology

The methodology we used here is Agile, because agile software development allows the team to work together more efficiently and effectively in developing complex projects. It consists of practices that exercise iterative and incremental techniques which are easily adopted and display great results.

VI. Process Description

The driver/User basically has to turn on the system as he/she starts to drive the vehicle. It is literally a plug and play system where the hardware and software interacts with each other, enabling the computer system to recognize and adapt to hardware configuration changes with little or no intervention by a user.

.

VII. Resources and Limitations

Software Requirements:

Operating system	:	Windows 10.
Front end	:	Tkinter, flask, django.
Back end	:	Python.

Hardware Requirements:

- Processor: Intel® Core™ i3-6006U CPU
- RAM: 8 GB or above
- Hard disk: 500 GB or above(optional)

VIII. Testing Technologies used

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. Testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements. The testing techniques used in this project are:

Manual Testing.

MANUAL TESTING is a type of Software Testing where Testers manually execute test cases without using any automation tools. Manual Testing is the most primitive of all testing types and helps find bugs in the software system. Any new application must be manually tested before its testing can be automated. Manual Testing requires more effort but is necessary to check automation feasibility.Manual Testing does not require knowledge of any testing tool. The types of manual testing include:

- **Unit Testing :**
  A unit is the smallest testable part of software. It usually has one or a few inputs and usually a single output. Testing is done by checking for all the boundary conditions, path specification, error conditions, null values and whether the information is properly flowing in to the program unit (or module) and properly happen out of it or not etc.

- **Integration Testing :**
  Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements.

- **Acceptance Testing:**
  In engineering and its various subdisciplines, acceptance testing is a test conducted to determine if the requirements of a specification or contract are met. It may involve chemical tests, physical tests, or performance tests. In systems engineering it may involve black-box testing performed on a system (for example: a piece of software, lots of manufactured mechanical parts, or batches of chemical products) prior to its delivery. Acceptance testing is also known as user acceptance testing (UAT), end-user testing, operational acceptance testing (OAT), acceptance-test-driven development (ATTD) or field (acceptance) testing. Acceptance criteria are the criteria that a system or component must satisfy in order to be accepted by a user, customer, or other authorized entity. The acceptance test suite may need to be performed multiple times, as all of the test cases may not be executed within a single test iteration.

- **System Testing :**

  System testing is testing conducted on a complete integrated system to evaluate the system's compliance with its specified requirements. System testing is performed on the entire system in the context of either functional requirement specifications (FRS) or system requirement specification (SRS), or both. System testing tests not only the design, but also the behaviour and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software or hardware requirements specification(s).

- **Regression Testing :**

  Regression Testing is defined as a type of software testing to confirm that a recent program or code change has not adversely affected existing features. Regression Testing is nothing but a full or partial selection of already executed test cases which are re-executed to ensure existing functionalities work fine. This testing is done to make sure that new code changes should not have side effects on the existing functionalities. It ensures that the old code still works once the new code changes are done.

IX. Conclusion

This system focuses on improving the safety of the driver and it also can be implemented in organization or work spaces to make sure the employees don’t fall asleep.

ACKNOWLEDGEMENT

Every project big or small is successful largely due to the effort of a number of wonderful people who have always given their valuable advice or lent a helping hand. I sincerely appreciate the inspiration, support and guidance of all those people who have been instrumental in making this project a success.

Especially, please allow me to dedicate my acknowledgment of gratitude towards the following significant advisors and contributors:

I am very grateful to Ihub IT Solutions, Mangalore for having provided me this project with necessary facilities and a good environment.I would like to thank my manager Mrs. Rekha R for his guidance and the motivation provided during this project period.I would like to convey my gratitude to Mr. Vipin for building up presentation skills and confidence into a whole new level..

I would like to thank my internal guide Mrs. Laveena C Crasta who has provided me with all valuable inputs to put my efforts on the right track..

I would like to convey my gratitude to Fr. Denzil Lobo SJ for being a pillar of support in all my endeavors..

I would like to thank Rev. Dr. Praveen Martis SJ, Principal of St Aloysius College and Rev. Fr. Melwyn Pinto SJ , Director of Aloysius Institute of Management and Technology (AIMIT) for their constant support and encouragement.

I would also like to thank Prof Santhosh Rebello the Dean, PG Department of Information technology for his great support, and instruction rendered to me during my project work.

Finally, I sincerely thank my parents, family, and friends, who have been a constant source of inspiration during the preparation of this project work.

Place : Mangaluru
Date : 14/06/2020

TABLE OF CONTENTS

Table of Contents

User Interface Diagram

CHAPTER 1: INTRODUCTION

1.1 Background

"1 in 25 adult drivers report that they have fallen asleep at the wheel in the past 30 days". Additionally, we believe that drowsiness can negatively impact people in working and classroom environments as well. Although sleep deprivation and college go hand in hand, drowsiness in the workplace especially while working with heavy machinery may result in serious injuries similar to those that occur while driving drowsily. Our solution to this problem is to build a detection system that identifies key attributes of drowsiness and triggers an alert when someone is drowsy before it is too late.

1.2 Objectives

The main objective of this project on improving the safety of the driver and it also can be implemented in organization or work spaces to make sure the employees don't fall asleep.

1.3 Purpose, Scope, and Applicability

1.3.1 Purpose

A countless number of people drive on the highway day and night. Taxi drivers, bus drivers, truck drivers and people traveling long-distance suffer from lack of sleep. Due to which it becomes very dangerous to drive when feeling sleepy. The majority of accidents happen due to the drowsiness of the driver.

1.3.2 Scope

In The scope of this project is to enhance the safety of the driver as well as the passengers in the vehicle by alerting the driver when he is falling drowsy/sleep.

The main objective of this application will include are:

1) Facial Landmark detection
2) Eyes aspect ratio
3) Alerting the driver

1.3.3 Applicability

As the system works on real time data, the system does not store data permanently in any form of data storage. There is no need of internet and hence can be used anytime, anywhere by the user/driver.

1.4 Achievements

Undertaking the above project, personally has given me opportunity to learn completely new set of technologies and concepts which I was unaware before. This Project makes use of Python and opencv libraries which helps in making dynamic application. This project has introduced me to a lot of deep learning algorithms and the advanced concepts of computer vision. I'm now familiar with working on neural networks in the tensorflow environment.

1.5 Organization of Report

Chapter 1: Introduction

The system being developed is introduced in this chapter. A description of the background and context of the project is specified. Concise statement of the aims and objectives of the project along with the purpose, scope and applicability are mentioned.

Chapter 2: survey of technologies

This chapter provides a detail of all the technologies which are used to complete the project. This also demonstrates the awareness and understanding of the available technologies related to the topic of the project.

Chapter 3: Requirements and Analysis

The requirements of that lead to the development of the system are mentioned and analyzed in this chapter. This includes the problem definition, requirement specification and planning and scheduling. The details of the overall problem and the sub problems are provided in the problem definition phase. Planning for the purposes and the certain rules and constraints are specified in the planning and scheduling phase. The hardware requirements and software requirements are specified in the document. The hardware required to run the product in the system is clearly specified. The software requirements such as operating system and other software to link and install the software are listed in this section. The requirements and objectives of the new system, its functions and operations are given in the product description section.

The conceptual models are built to understand the problem domain which describes the operations that can be performed and the allowable sequences of those operations.

Chapter 4: System Design

System Design describes the features and operations in detail, including the data design, functional and interface design, process diagrams, details description of functions and the test case design.

The Basic Modules section briefly describes all the modules and the functionality of these modules. The next section is the Data Design which consists of how to organize, manage and manipulate the data. This section again has two Subsections-Schema design and Data Integrity and constraints. Schema design defines the structure and explanation of schemas. Data Integrity and constraints specifies all the validity checks and constraints.

Chapter 5: Implementation and testing

This chapter defines the plan of implementation and standards used in implementation, code efficiency, and testing approach, integrated testing, Modifications and improvements of project.

Chapter 6: Results and Decisions

Here the test reports explains the test results and reports based on the test cases thereby showing that the software is capable of facing any problematic situation and that it works fine in different conditions. And the user documentation defines the working of the software i.e. explaining its different functions, components with screen shots.

Chapter 7: Conclusions

This chapter summarizes the entire work. It explains all the points that have been made in the earlier chapters. It also includes the limitations of the system, listing all the criticism encountered during the demonstration of the software. It also proposes the future scope of the project i.e. new areas of investigation prompted by developments. It also explains the limitations of the system and the future possible enhancements of the system.

CHAPTER 2: SURVEY OF TECHNOLOGIES

- **Python -** Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library. Python interpreters are available for many operating systems. A global community of programmers develops and maintains CPython, an open source[34] reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development.

- **Open CV -** MySQL OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel). The library is cross-platform and free for use under the open-source BSD license. OpenCV is written in C++ and its primary interface is in C++, but it still retains a less comprehensive though extensive older C interface. All of the new developments and algorithms appear in the C++ interface. There are bindings in Python, Java and MATLAB/OCTAVE. The API for these interfaces can be found in the online documentation. Wrappers in other languages such as C#, Perl, Ch,Haskell,and Ruby have been developed to encourage adoption by a wider audience. Since version 3.4, OpenCV.js is a JavaScript binding for selected subset of OpenCV functions for the web platform. OpenCV runs on the following desktop operating systems: Windows, Linux, macOS, FreeBSD, NetBSD, OpenBSD. OpenCV runs on the following mobile operating systems: Android, iOS, Maemo, BlackBerry 10.The user can get official releases from SourceForge or take the latest sources from GitHub. OpenCV uses CMake.

CHAPTER 3: REQUIREMENTS AND ANALYSIS

3.1 Problem Definition

A countless number of people drive on the highway day and night. Taxi drivers, bus drivers, truck drivers and people traveling long-distance suffer from lack of sleep. Due to which it becomes very dangerous to drive when feeling sleepy. The majority of accidents happen due to the drowsiness of the driver.

3.2 Requirements Specification

3.2.1 System Features

3.2.1.1 Facial Landmark detection:

Facial Landmark detection is the process of localizing key facial structures on a face, including the eyes, eyebrows, nose, mouth, and jawline, and from these features, in the context of drowsiness detection, we only extract the eye and mouth regions.

3.2.1.2 Eye Aspect Ratio:

After the eye regions have been extracted, we apply the eye aspect ratio to determine if the eyes are closed. If the eyes have been closed for a sufficiently long enough period of time, we can assume the user is at risk of falling asleep and sound an alarm to grab their attention.

3.2.1.3 Alerting system:

A system that can automatically detect driver drowsiness in a real-time video stream, using the above mentioned modules and then alert the driver with alarm or sound if the driver appears to be drowsy.

3.2.2 External Interface Requirements

User interface

The Since we have used python for developing the drowsiness detection system, Python provides various options for developing graphical user interfaces (GUIs) such as Tkinter, wxPython and JPython. We have used tkinter for the user interface. Tkinter is the Python interface to the Tk GUI toolkit shipped with Python. Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit..

Hardware Interface

- Minimum of 5 MP Camera
- A laptop with 4 GB RAM
- A USB cable

Software Interface

- Python 2.7 or above versions
- Anaconda software

3.2.3 Performance Requirements

- The system only works well when the camera is able to get full view of the face of the driver.
- The system's initial load time depend on the device configuration on which it is running.
- The system must be interactive and the delays involved must be less.
- The overall performance of the system depends on the performance of the device.

3.2.4 Safety Requirements

As the system works on real time data, the system does not store data permanently in any form of data storage. Therefore there is minimum risk in terms of privacy of the driver or the user of the system.

3.2.5 Security Requirements

User Since the system works on real time drowsiness detection, the camera needs to be placed in full view of the driver’s face in order to accurately detect the eyes and mouth regions.

.

3.2.6 Software Quality Attributes

I. Reliability

It’s The solution should reliably detect drowsiness so that it can serve its purpose as a system for promoting driver safety.

II. Real time response

The operation of a vehicle can involve relatively high speeds, a system that cannot detect drowsiness and warn that driver promptly can lead to serious consequences.

III. Unobtrusive

It is very important that the solution is as transparent to the driver as possible.

IV. Flexible

To be effective, the solution should be designed so as to accommodate for all types of users, in terms of physical attributes.

3.3 Planning and Scheduling

Planning, Scheduling and completion of different milestones of the project is displayed in the below table: Table 1.

Sl. No.	Milestone Name	Milestone Description	TimeLine (weeks)	Remarks
1	Requirement Specification	Completed Specification	1	Attempt should be made to add some more relevant functionalities
2	Technology Familiarization	Understanding of the technology needed to implement the Project	2	The presentation should be from the point of view of being able to apply it to the project, rather than from a theoretical perspective.
3	High level & detailed design	Listing down all possible scenarios & then combining up with flow charts	3	The scenarios should match to the requirement specification.
4	Implementation of the front end of the system	Implementation of the main screen, screen that follows sharing giving various options	2	During this milestone period it would be a good idea for developer to start working on a test plan of entire system. This test plan can be updated as & when new scenarios come to mind
5	Integration Testing	The system should be thoroughly tested by running all test cases written for the system.	2	Another one week should have been there to handle any issues found during testing of the system

6	Final Review	Issues found during the previous milestone are fixed & the system is ready for the final review	1	During the final review of the project, it should be checked that the entire requirement specified during the milestones are fulfilled.

3.4 Software and Hardware Requirements

Software Requirements:

Operating system	:	Windows 10
Front end	:	Tkinter
Back end	:	Python.

Hardware Requirements:

Processor	:	Dual Core or above
RAM	:	4GB
Hard disk	:	1TB

3.5 Preliminary Product Description

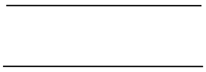
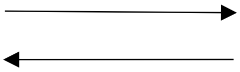
The main objective of this project/product is on improving the safety of the driver and it also can be implemented in organization or work spaces to make sure the employees don’t fall asleep.

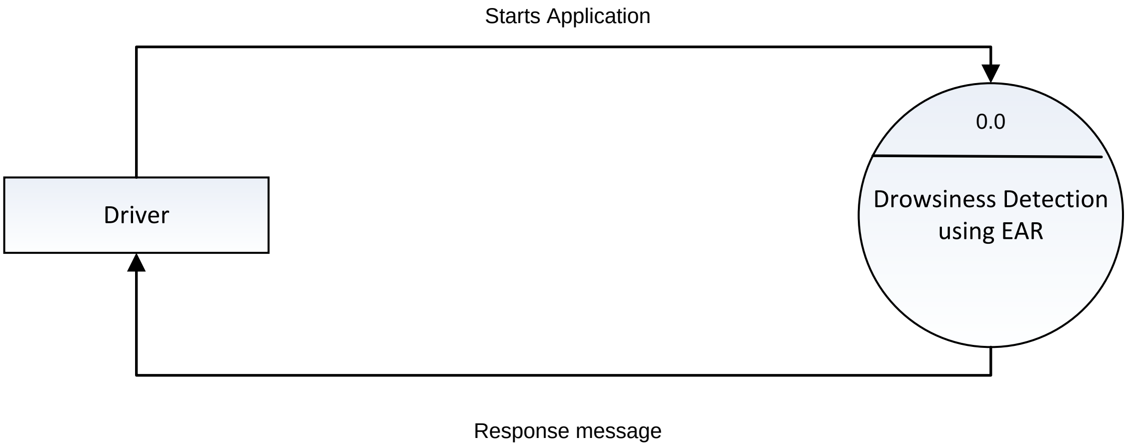
3.6 Conceptual Models

3.6.1 Data Flow Diagrams

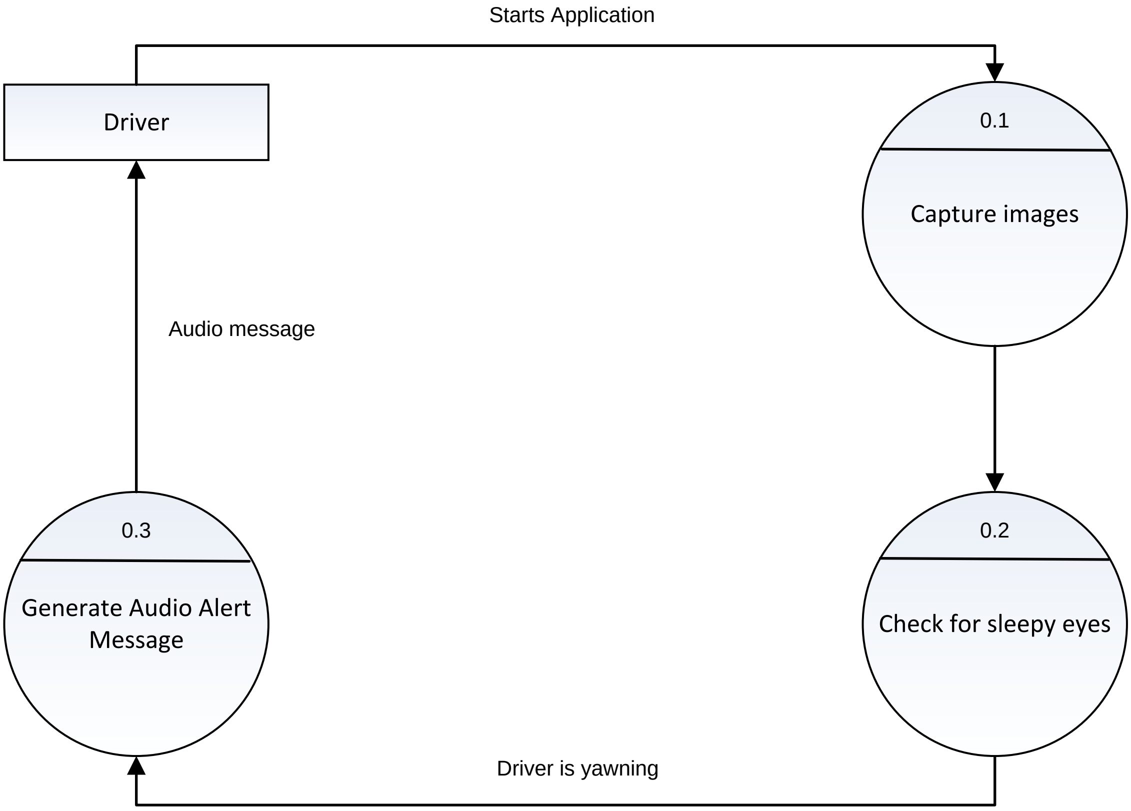
Data flow diagram (DFD) is a graphical representation of the “flow” of data through an information system; it differs from the flowchart as it shows the data flow instead of the control flow of the program.

A data flow diagram can also be used for the visualization of data processing. DFDs show the flow data from external entities into the system, showed how the data moved from one process to another, as well as its logical storage. The basic notation used to create a DFD makes it easy to analyze and understand.

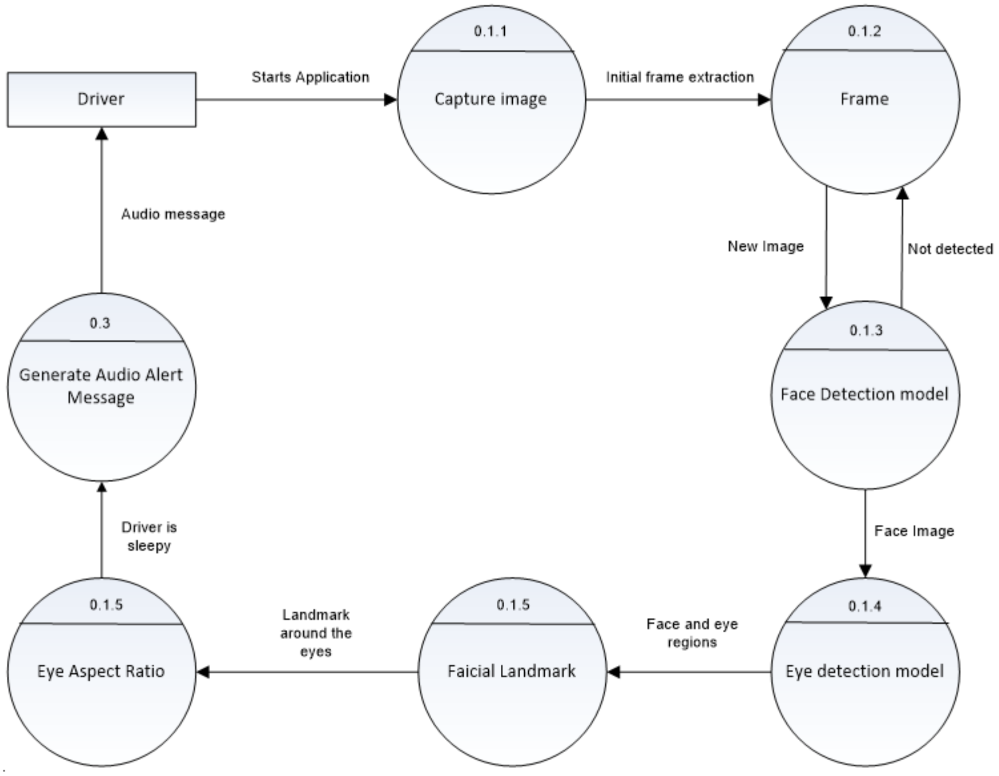
SYMBOL	NOTATION	DESCRIPTION
	Source/Destination	Sources and destinations (sink) define the system’s boundaries. It is represented by a rectangle.
	Process	Represents an activity that changes, moves or otherwise transforms data. It is shown as a round-cornered rectangle/ oval.
	Data store	Represents data at rest and implies that the data are held between processes. It is shown as an open-ended, horizontal rectangle.
	Flow of activity or control	Represents data in motion. It is depicted with an arrow.



Data Flow Diagram-Level 0



Data Flow Diagram-Level 1



Data Flow Diagram-Level 2

CHAPTER 4: SYSTEM DESIGN

4.1 Main Modules

4.1.1 Facial Landmark detection

Facial Landmark detection is the process of localizing key facial structures on a face, including the eyes, eyebrows, nose, mouth, and jawline, and from these features, in the context of drowsiness detection, we only extract the eye and mouth regions. Facial landmarks have been successfully applied to face alignment, head pose estimation, face swapping, blink detection and much more. Detecting facial landmarks is therefore a two step process:

- Localize the face in the image.
- Detect the key facial structures on the face ROI.

4.1.2 Eye aspect ratio (EAR)

Updating After the eye regions have been extracted, we apply the eye aspect ratio to determine if the eyes are closed. If the eyes have been closed for a sufficiently long enough period of time, we can assume the user is at risk of falling asleep and sound an alarm to grab their attention.

$$\mathrm{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

Where *p1, …, p6* are 2D facial landmark locations.

4.1.3 Alerting system

Registration A system that can automatically detect driver drowsiness in a real-time video stream, using the above mentioned modules and then alert the driver with sound or alarm if the driver appears to be drowsy.

4.2 Data Design Project Structure

4.2.1 Data Integrity and Constraints

A The method we have selected here is called as the behavioral methods. These methods are based on detecting specific behavioral clues exhibited by a driver while in a drowsy state. A typical focus is on facial expressions that might express characteristics such as: rapid, constant blinking, nodding or swinging of the head, or frequent yawning. These are all tell-tale signs that a person might be sleep deprived and/or feeling drowsy. Typically, systems based on this methodology use a video camera for image acquisition and rely on a combination of computer vision and machine learning techniques to detect events of interest, measure them, and make a decision on whether the driver may be drowsy or not. If the sequence of captured images and measured parameters (e.g., pattern of nodding or time lapsed in "closed eye state") suggest that the driver is drowsy, an action — such as sounding an audible alarm — might be warranted.

• **Head or eye position.** When a driver is drowsy, some of the muscles in the body begin to relax, leading to nodding. This nodding behavior is what researchers are trying to detect. Research exploiting this feature has started just recently. Detecting head or eye position is a complex computer vision problem which might require stereoscopic vision or 3D vision cameras.

•**Eye state**. To move forward with this, we'll be computing a metric called the eye aspect ratio (EAR). Unlike traditional image processing methods for computing blinks which typically involve some combination of:

- Eye localization.
- Thresholding to find the whites of the eyes.
- Determining if the "white" region of the eyes disappears for a period of time (indicating a blink).

The eye aspect ratio is instead a much more elegant solution that involves a very simple calculation based on the ratio of distances between facial landmarks of the eyes.

•**Yawning** Frequent yawning is a behavioral feature that tells that the body is fatigued or falling into a more relaxed state, leading towards sleepiness. Detecting yawning can serve as a preemptive measure to alert the driver. It should be noted, however, that yawning does not always occur before the driver goes into a drowsy state. Therefore it cannot be used as a stand-alone feature; it needs to be backed up with additional indicators of sleepiness

4.3 Procedural Design

4.3.1 Logic Diagrams

4.3.1.1 Sequence Diagrams

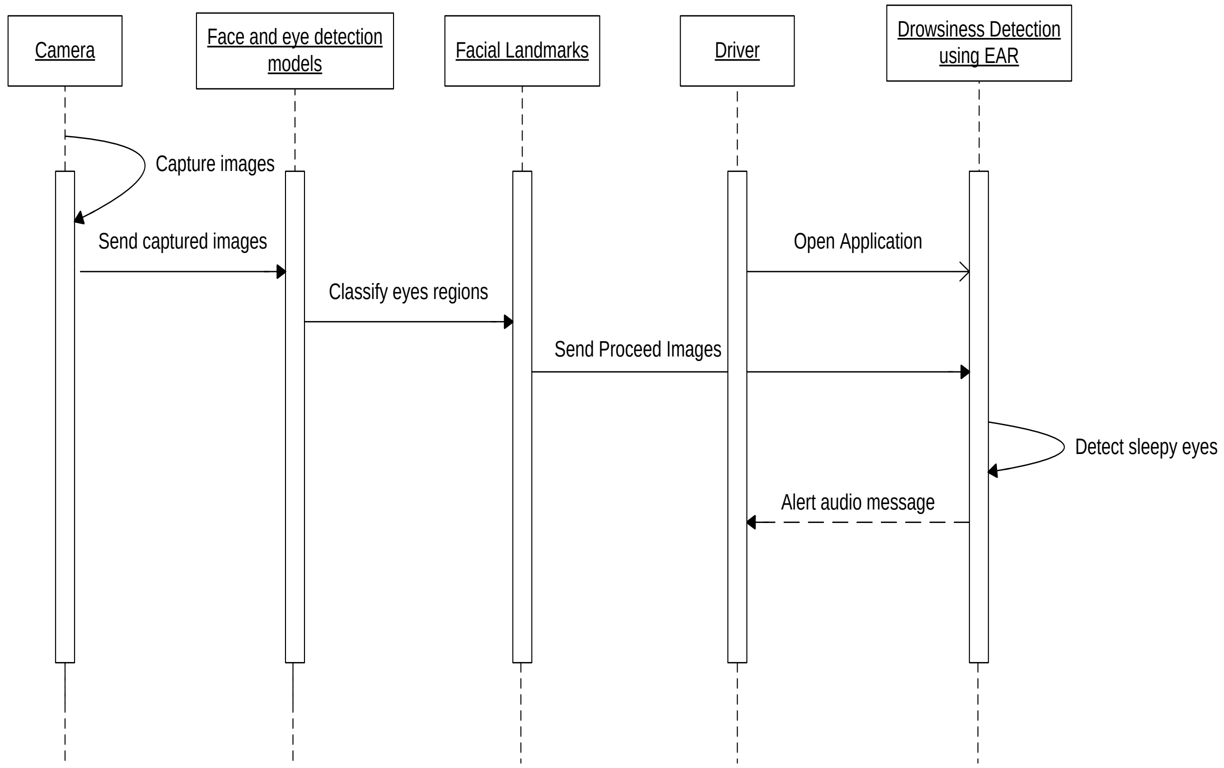
A Sequence Diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. UML sequence diagrams model the flow of logic within your system in a visual manner, enabling you both to document and validate your logic, and are commonly used for both analysis and design purposes. Sequence diagrams are the most popular UML artefact for dynamic modelling, which focuses on identifying the behaviour within your system.

•User or another system that will interact with the objects. Actors are placed in columns.

•The objects in the system that is being modelled. Represented in rectangular boxes.

•Dashed line is lifeline which indicates the existence of the object over time. Rectangular box is Activation which indicates that the object is performing an action.

•To display the interaction, messages are used. These are horizontal arrows with the message name written above them



Sequence diagram for Drowsiness detection

4.3.1.2 Use Case Diagrams

Activity Use Case Diagram Symbols and Notations. Use case diagrams model the functionality of system using actors and use cases. Use case diagrams depict:

System: Draw your system's boundaries using a rectangle that contains use cases. Place actors outside the system's boundaries.

Use Case: Draw use cases using ovals. Label the ovals with verbs that represent the system's functions. A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse.

Actors: Actors are the users of a system. An actor is a person, organization, or external system that plays a role in one or more interactions with your system. Actors are drawn as stick figures. When one system is the actor of another system, label the actor system.

Scenario: A scenario is one hypothetical instance of how a particular use case might play out. A single use case thus inspires many different scenarios, in the same way that planning a driving trip from one city to another can involve many different routes.

Associations: Associations between actors and use cases are indicated in use case diagrams by solid lines. An association exists whenever an actor is involved with an interaction described by a use case.

System boundary boxes (optional): You can draw a rectangle around the use cases, called the system boundary box, to indicate the scope of your system.

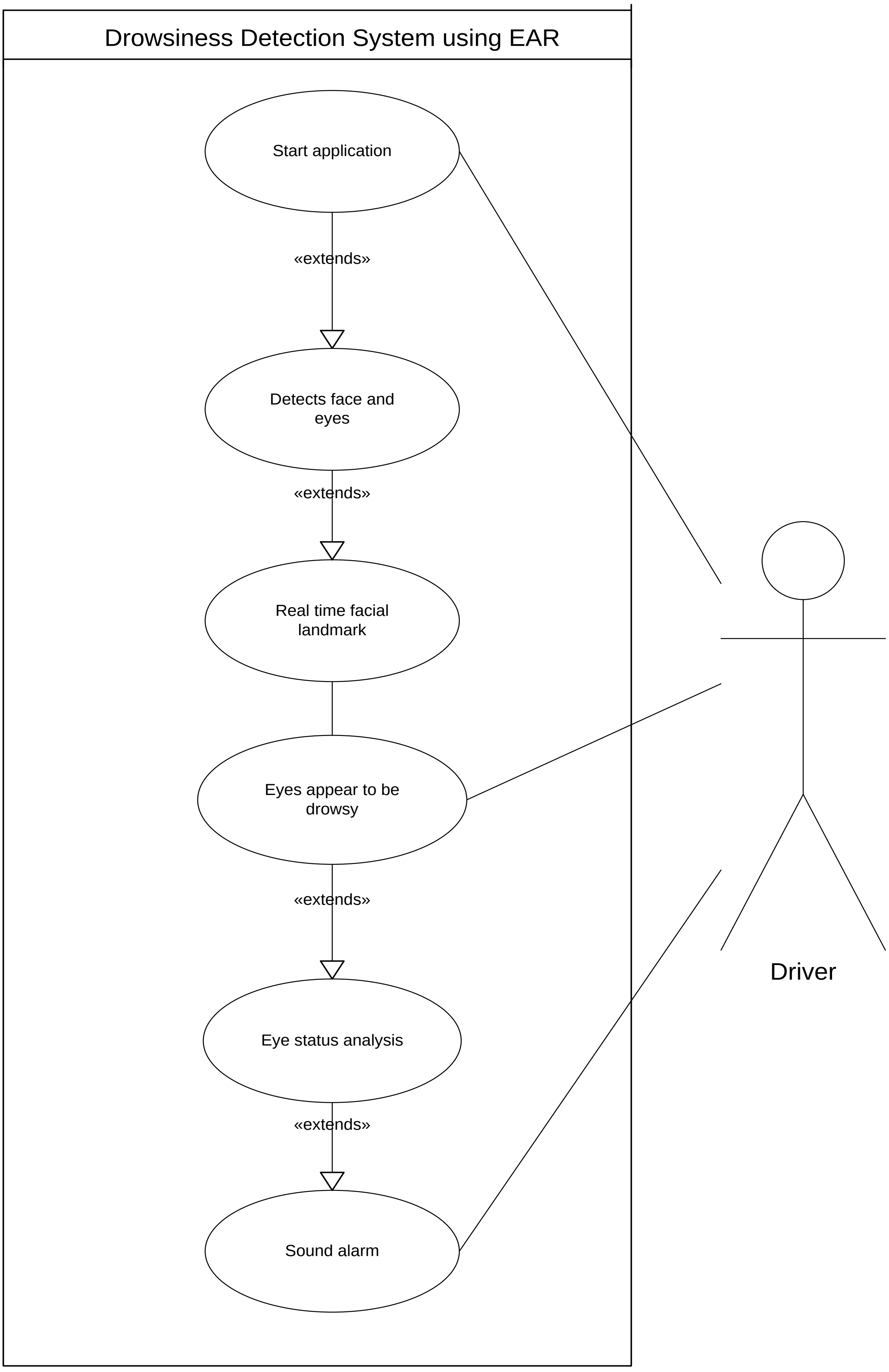
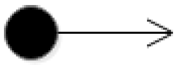
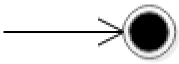
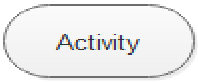
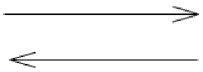
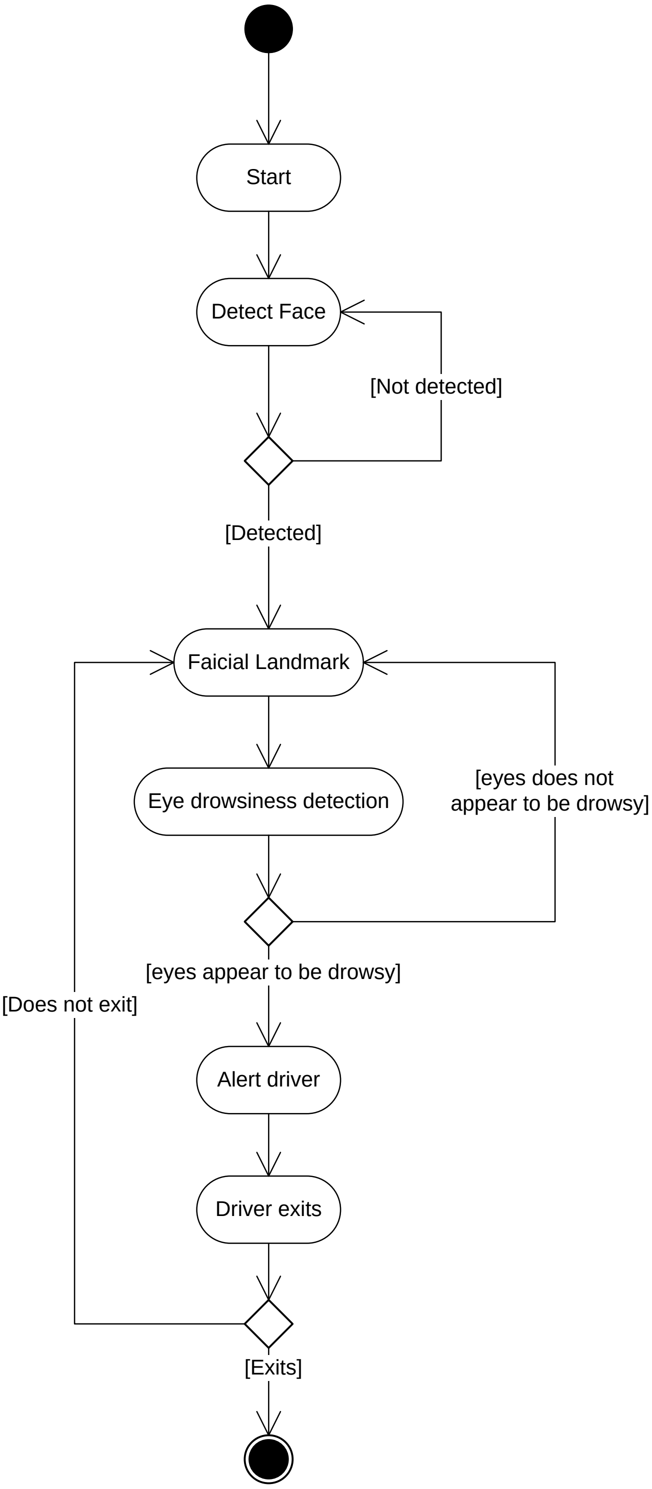


Figure 8: Use case diagram for drowsiness detection

4.3.1.3 Activity Diagram:

Activity diagrams are used to describe the operations of the system components in step-by-step workflows. It is a dynamic diagram that shows the activity and the event that causes the object to be in a particular state. It is a simple and intuitive illustration of what happens in a workflow, what activities can be done in parallel and whether there are alternative paths through the workflow. Rounded rectangles represent activities, Diamonds represent decisions, a black circle represents the start (initial state) of the work flow, and an encircled black circle represents the end (final state). Arrows run from the start towards the end and represent the order in which activities happen.

Shape	Notation	Description
	Initial Activity	Represents the starting point or first activity of the flow. It is denoted by a solid circle.
	Final Activity	Represents the end of activity diagram, also called as final activity. It is shown by a bull’s eye symbol.
	Activity	Represents an activity. It is shown by a rectangle with rounded (almost oval) edges.
	Flow of Activity or Control	Shows the direction of the workflow in the activity diagram. It is depicted with an arrow.



Activity diagram for drowsiness detection

4.3.1.4 State Chart Diagram:

A state diagram shows the behavior of classes in response to external stimuli. Specifically a state diagram describes the behavior of a single object in response to a series of events in a system.

- **States**

  State represent situations during the life of an object. You can easily illustrate a state in Smart Draw by using a rectangle with rounded corners.

- **Transition**

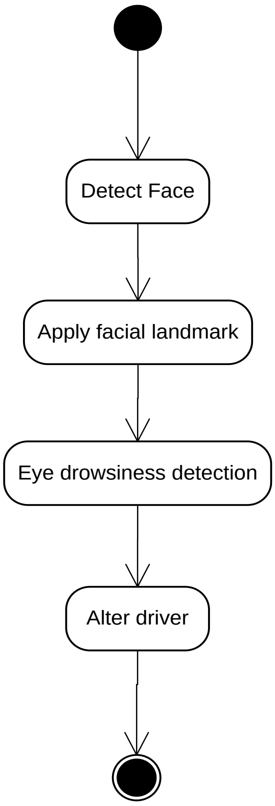
  A solid arrow represents the path between different states of an object. Label the transition with the event that triggered it and the action that results from it. A state can have a transition that points back to itself.

- **Initial State**

  A filled circle followed by an arrow represents the object's initial state.

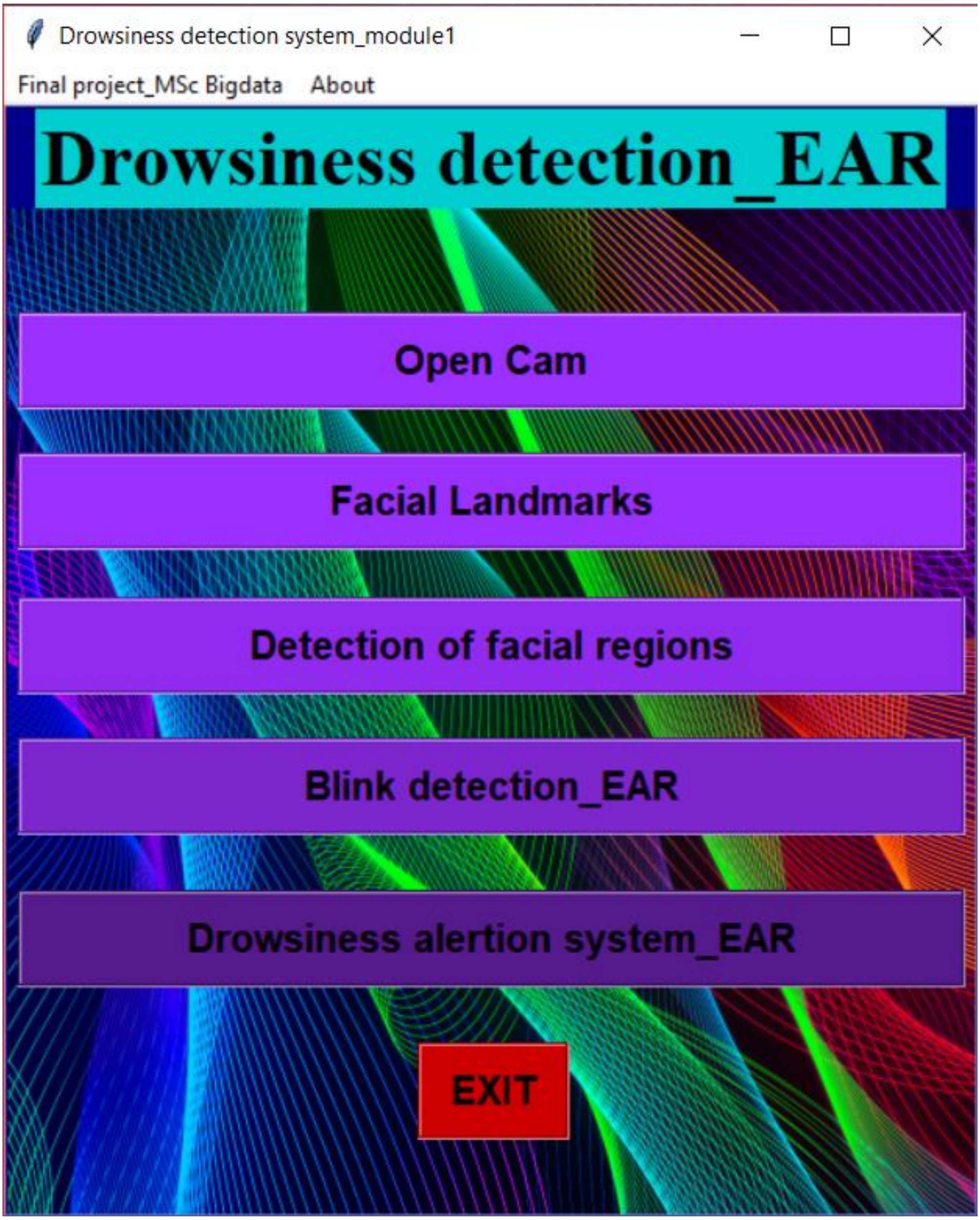
- **Final State**

  An arrow pointing to a filled circle nested inside another circle represents the object's final state.



State chart diagram for drowsiness detection

4.4 User interface Design:



Drowsiness detection system_module1
Final project_MSc Bigdata
About
Drowsiness detection_EAR
Open Cam
Facial Landmarks
Detection of facial regions
Blink detection_EAR
Drowsiness alertion system_EAR
EXIT

Fig: User interface design

Applying Facial landmark model to the live video stream:

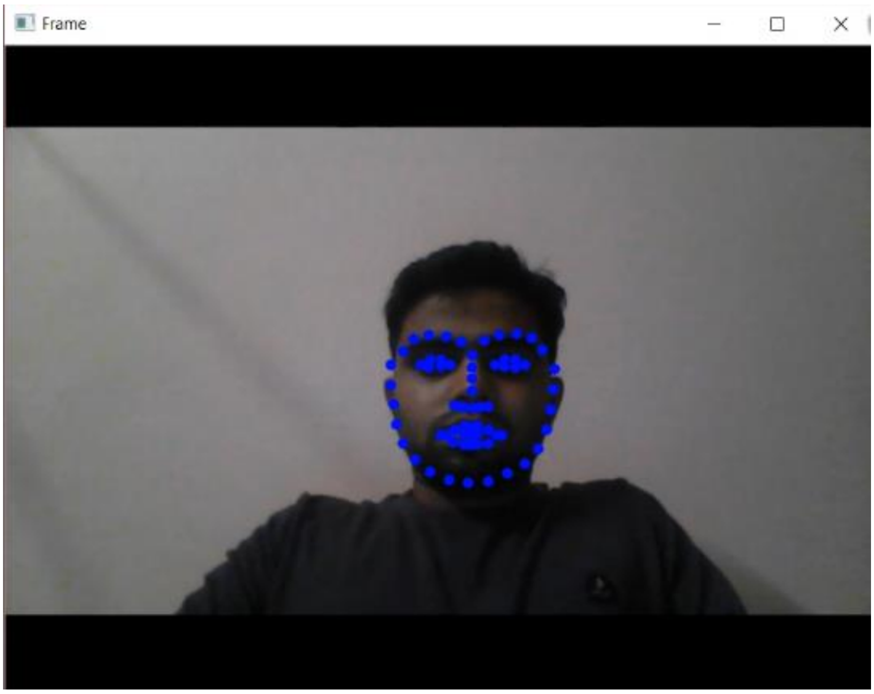


Fig shows the landmark model applied on the user's face

Detection of the facial regions

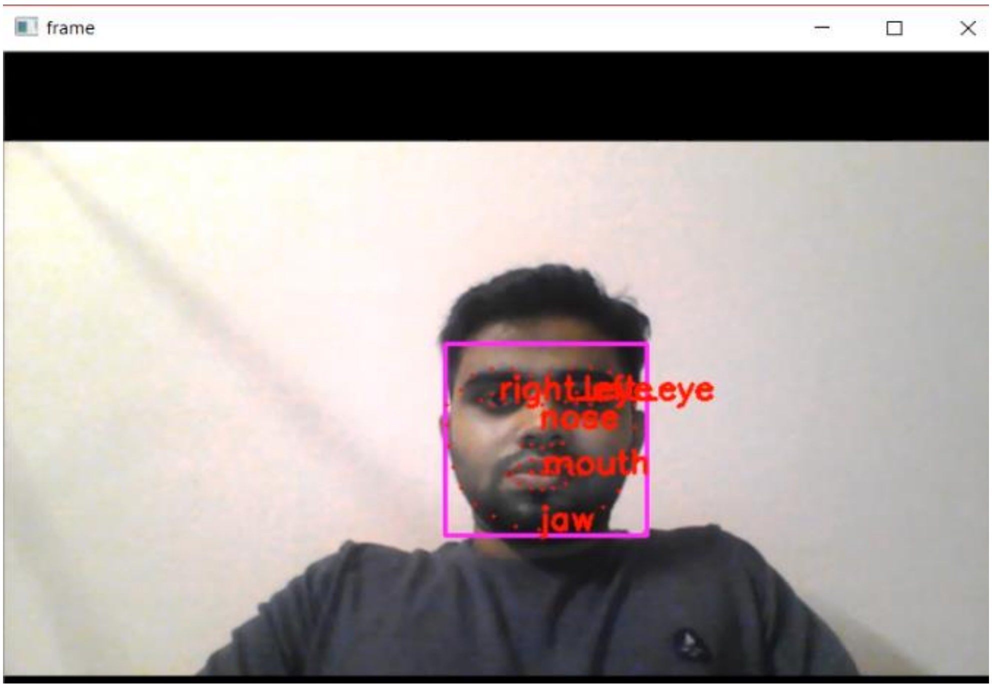


Fig shows the model extraction the facial regions from user's face

Blink detection by applying the EAR algorithm (eye aspect ratio):

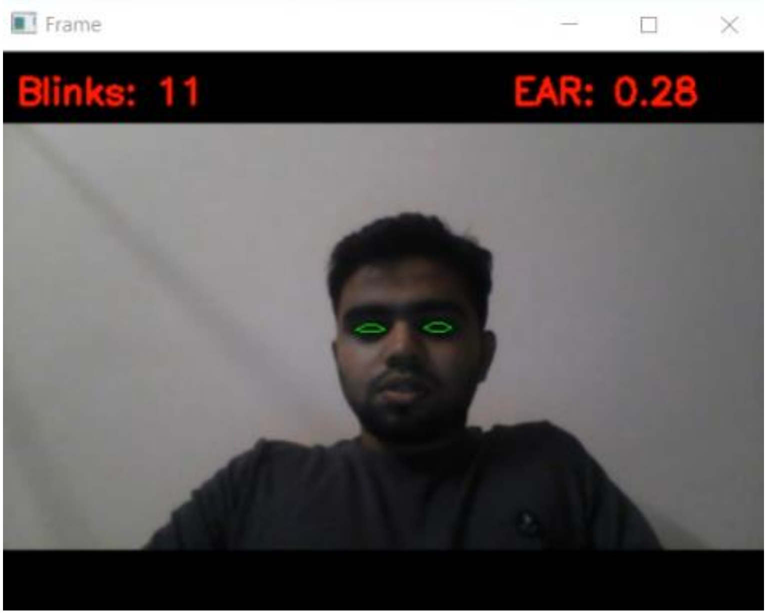


Fig shows the number of times the user blinked by using the EAR algorithm

Drowsiness detection using the EAR algorithm:

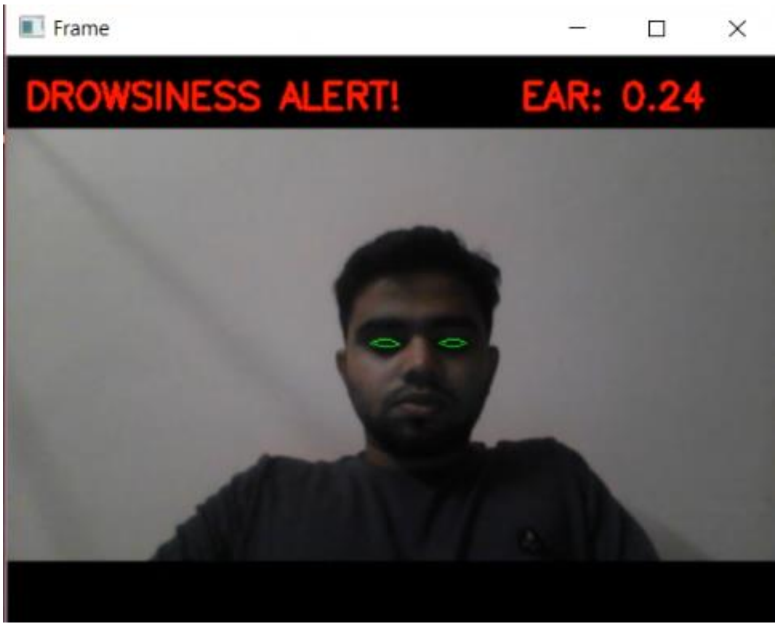


Fig shows the system alerting the user when he closes his eyes

4.5 Security Issues

As the system works on real time data, the system does not store data permanently in any form of data storage. Therefore there is minimum risk in terms of privacy of the driver or the user of the system. And the camera needs to be placed in full view of the driver's/user face in order to accurately detect the eyes and mouth regions.

4.6 Test Cases Design

Software testing:

It is a process used to identify the correctness, completeness and quality of developed computer software i.e., software testing is to execute a software program with the intent of finding bugs.

Software quality assurance:

It involves the entire software development process monitoring and improving the process, making sure that any agreed upon standards and procedures are followed and ensuring that problems are found and dealt with. It is oriented to 'Prevention'.

SOFTWARE DEVELOPMENT LIFE CYCLE:

□ Requirements

□ Analysis

□ Design

□ Coding

□ Testing

□ Maintenance

APPROACHES TO TESTING:

There are two basic approaches to testing:

- Black-Box Testing
- White-Box Testing

BLACK-BOX TESTING:

In black box testing the structure of the program is not considered. Test cases are decided solely on the requirements or specification of the program or module, and the internals of the module or the program are not considered for selection of test cases.

In black-box testing, the tester only knows the inputs that can be given to the system and what output the system should give. In other words, the basis for deciding test cases in functional testing is the requirements or specifications of the system or module. This form of testing is also called functional or behavioural testing.

WHITE-BOX TESTING:

White box testing is concerned with the function that the tested program is supposed to perform and does not deal with the internal structure of the program responsible for actually implementing that function thus black-box testing is concerned with the functionality rather than implementation of the program. White-box testing, on the other hand is concerned with testing the implementation of the program. The intent of this testing is not to exercise all the different input or output conditions (although that may be a by- product) but to exercise the different programming structures and data structures used in the program. White-box testing is also called structural testing.

Chapter 5: Implementation and Testing

5.1 Implementation Approaches

Implementation is the stage in the project where theoretical design is turned into a working system and is giving confidence on the new system for the users that will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the changeover, an evaluation, of change over methods. Apart from planning major task of preparing the implementation are education and training of users.

The objective of the system implementation phase is to implement a fully functional system, which deliverables met in order of priority and with effective training. The end result of this phase is an accurately functioning system with properly trained users. The more complex system being implemented, the more involved will be the system analysis and the design effort required just for implementation.

The implementation phase deals with issues of quality, performance, baselines, libraries, and debugging. The end deliverable is the product itself. During the implementation phase, the system is built according to the specifications from the previous phases. This includes writing code, performing code reviews, performing tests, selecting components for integration, configuration, and integration.

The implementation includes the following things.

- Careful planning
- Investigation of system and constraints.
- Design the methods to achieve the charge over.
- Training the staff in the changed phase.

5.2 Coding Details and Code Efficiency

5.2.1 Code Efficiency

- The following were eliminated using the inbuilt code efficiency tools provided by the IDE and also by subsequent code reviews
- Unused local variables
- Empty catch blocks

- Unused parameters
- Empty 'if' statements
- Unused private methods
- Short/long variable and method names

Moreover to check the efficiency of the coding methods used Code Coverage tools provided with the IDE was used to check the following branch coverage, loop coverage and strict condition coverage (decision coverage)

5.3 Testing Approach

Testing is the important step of s/w development. An elaborate testing of the data is prepared and the system is using the test data. While testing, errors are noted and correction is made. The users are trained to operate the developed system.

Characteristics of a Good Test:

•Tests are likely to catch bugs

•No redundancy

•Not too simple or too complex

5.3.1 Unit Testing

Unit testing focuses on testing the smallest unit of the module. To check whether each module in the software works properly so that it gives desired outputs to the given inputs. All validations and conditions are tested in the module level in the unit test. Control paths are tested to ensure the information properly flows into, and output of the program unit and out of the program unit under test. Boundary conditions are tested to ensure that the modules operate at boundaries.

5.3.2 User Acceptance Testing

The This is arguably the most important type of testing, as it is conducted by the Quality Assurance Team who will gauge whether the application meets the intended specifications and satisfies the client's requirement. The QA team will have a set of pre-written scenarios and test cases that will be used to test the application.

5.4 Modifications and Improvements

The bugs & defects identified at each of the stages were fixed as and when they are found. The regression testing is not performed completely as the product is not yet reached the released stage. Some enhancements have been given by the customer when they saw a demo of the product. Proper measures are taken to incorporate these enhancements in the product.

CHAPTER 6: RESULTS AND DISCUSSION

6.1 Test Reports:

SL.No	USER ACTION	EXPECTED RESULTS	TEST RESULTS
1	Face of user not facing the camera.	Facial landmark model doesn’t pick up the face from the user.	Successful
2	Face of user placed right in front of the cam.	Model detects face and extracts region of interest from user.	Successful

6.2 User Documentation

The system itself is very user friendly which will guide the end users about how to go about using the system. The system shall not provide specific guidelines to the users for using the system. It is has easy navigation process wherein the user can smoothly navigate through the process, so there won’t be necessary to provide any user documentations.

CHAPTER 7: CONCLUSIONS

7.1 Conclusion

The drowsiness detection System is mainly useful to the drivers as well as the users. This helps the drivers to reduce the risk of accidents and also the users in this case the students /employees to stay alert. The "Drowsiness detection System " is an easy and also an efficient way to manage the safety of the drivers.

.

7.2 Limitations of the System

- May require high quality camera for better results.
- Distance of the face from the camera may also play a vital role.
- It might be annoying if the system rings the alarm when the user is not drowsy.

7.3 Future Scope of the Project

- We can add more features to detect drowsiness such as head moment and so on.
- Can also integrate with raspberry pie for better results and efficiency.

CHAPTER 8: REFERENCES

Website Referred

- https://www.pantechsolutions.net/driver-drowsiness-detection-using-opencv-and-python
- https://www.pyimagesearch.com/2017/05/08/drowsiness-detection-opencv/
- https://blog.goodaudience.com/real-time-face-and-eyes-detection-with-opencv-54d9ccfee6a8
- https://towardsdatascience.com/a-guide-to-face-detection-in-python-3eab0f6b9fc1
- https://towardsdatascience.com/face-detection-for-beginners-e58e8f21aad9
- https://datascienceplus.com/face-and-eye-detection-with-opencv/

CHAPTER 9: GLOSSARY

Abbreviation	Description
SDD	Software Design Document
GUI	Graphical User Interface
SRS	Software Requirement Specification
PDF	Portable Document Format

ALOYSIUS INSTITUTE OF MANAGEMENT AND INFORMATION TECHNOLOGY
ST. ALOYSIUS COLLEGE (AUTONOMOUS)
BEERI, MANGALORE – 575 022
P G DEPT OF COMPUTER APPLICATIONS & SOFTWARE TECHNOLOGY

FEEDBACK FORM for DISSERTATION-I

Student Register Number : ____

Student Name : ____

Title of the Project : ____

Name of the company : ____

Date of Dissertation – I : ____

I. Strengths of the System Requirement Specification

1.

2.

3.

II. Weakness of the System Requirement Specification

1.

2.

3.

III. Suggestions for improving the System Requirement Specification

1.

2.

3.

4.

5.

6.

Name & Signature	Name & Signature	Name & Signature
Examiner -I	Examiner -II	Internal Guide

ALOYSIUS INSTITUTE OF MANAGEMENT AND INFORMATION TECHNOLOGY
ST. ALOYSIUS COLLEGE (AUTONOMOUS)
BEERI, MANGALORE – 575 022
P G DEPT OF COMPUTER APPLICATIONS & SOFTWARE TECHNOLOGY

FEEDBACK FORM for DISSERTATION-II

Student Register Number : ____

Student Name : ____

Title of the Project : ____

Name of the company : ____

Date of Dissertation – II : ____

I. Strengths of the System Design Document

1.
2.
3.

II. Weakness of the System Design Document

1.
2.
3.

III. Suggestions for improving the System Design Document

1.
2.
3.
4.
5.
6.

Name & Signature	Name & Signature	Name & Signature
Examiner -I	Examiner -II	Internal Guide

ALOYSIUS INSTITUTE OF MANAGEMENT AND INFORMATION TECHNOLOGY
ST. ALOYSIUS COLLEGE (AUTONOMOUS)
BEERI, MANGALORE – 575 022
P G DEPT OF COMPUTER APPLICATIONS & SOFTWARE TECHNOLOGY

FEEDBACK FORM for DISSERTATION-III

Student Register Number : ______________________________

Student Name : ______________________________

Title of the Project : ______________________________

Name of the company : ______________________________

Date of Dissertation – III : ______________________________

I. Strengths of the Mid Term Evaluation

1.

2.

3.

II. Weakness found during Mid-Term Evaluation

1.

2.

3.

III. Suggestions for improving the System Development process

1.

2.

3.

4.

5.

6.

Name & Signature Examiner -I	Name & Signature Examiner -II	Name & Signature Internal Guide