Project Report
on

Drowsiness Detection System [Web App]

by

Nayan Sharma **[Reg. No. 26100118028]**

In partial fulfillment of the requirement for the award of degree
Bachelor of Technology

Under the Supervision of

Miss Amrita Kundu
Asst. Professor, Dept. of CSE, SIEM, MAKAUT

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SURENDRA INSTITUTE OF ENGINEERING & MANAGEMENT, SILIGURI

MAY 2022

Surendra Institute of Engineering & Management, Siliguri

[Approved by AICTE & Affiliated to MAKAUT]

CERTIFICATE

This is to certify that the dissertation entitled "**Drowsiness Detection System [Web App]**" is a B. Tech project carried out by

NAYAN SHARMA **[Reg. No: 26100118028]**

as a partial fulfilment for the award of **Bachelor of Technology** in **Computer Science & Engineering Department** of **Surendra Institute of Engineering & Management, Siliguri**; affiliated to **Maulana Abul Kalam Azad University of Technology**, Kolkata in the year 2022. The above dissertation is hereby accepted and approved as a satisfactory academic requirement within prescribed period for Bachelor of Technology Degree.

Project Supervisor
(Ms. Amrita Kundu)

External

HOD, Dept. of CSE
(Mr. Santosh Shah)

ACKNOWLEDGEMENT

First and foremost, all praises to the almighty as for His mercy and grace, we were able to have progress for our final year project. We would like to seize this opportunity to thank all parties who have contributed along the process of the completion of our final year project.

We would like to express our sincere gratitude and great appreciation to our mentor and supervisor, **Miss Amrita Kundu**, who has given us this bright opportunity to engage in an Web Application development project. Thanks for her guidance through the project simulating suggestions and correcting various documents of ours with much attention and care.

Our special Thanks to our **Mr. Santosh Shah (Head of Department), Department of Computer Science and Engineering** for the endless and untiring guidance, assistances, knowledge and experiences shared during the final year period.

We are very grateful to our principal **Prof. Dr. S. Nagarajan** for providing a healthy environment in the college which helped us in concentrating in our task.

Our deepest gratitude goes to our family, for their moral support and financial assistance.. Besides that, I would like to thank **Surendra Institute of Engineering & Management [SIEM]**, especially the Computer Science & Engineering Department whereby students are given the opportunity to be trained with essential skills to excel in theoretical and practical work during this final year project period. Its well-rounded graduate philosophy has proven to be useful in the industry.

Nayan Sharma **[Reg. No. 26100118028]** Nayan Sharma.

ABSTRACT

Drowsiness and fatigue are one of the main causes leading to road accidents. They can be prevented by taking effort to get enough sleep before driving, drink coffee or energy drink, or have a rest when the signs of drowsiness occur. The main idea behind this project is to develop a nonintrusive system which can detect fatigue of any human and can issue a timely warning. Drivers who do not take regular breaks when driving long distances run a high risk of becoming drowsy a state which they often fail to recognize early enough. According to the expert's studies show that around one quarter of all serious motorway accidents are attributable to sleepy drivers in need of a rest, meaning that drowsiness causes more road accidents than drink-driving. This system will monitor the driver eyes using a camera and by developing an algorithm we can detect symptoms of driver fatigue early enough to avoid the person from sleeping. So, this project will be helpful in detecting driver fatigue in advance and will give warning output in form of alarm.

This paper proposes a way to detect the drowsiness signs among drivers by measuring the eye closing rate and yawning. This project describes on how to detect the eyes and mouth in a live video recorded. In the video, a participant will drive the driving simulation system and a webcam will be place in front of the driving simulator. The video will be recorded using the webcam to see the transition from awake to fatigue and finally, drowsy. The designed system deals with detecting the face area of the image captured from the video. The purpose of using the face area so it can narrow down to detect eyes and mouth within the face area. Once the face is found, the eyes and mouth are found by creating the lines around the eye detection and also mouth detection.

The parameters of the eyes and mouth detection are created within the face image. The video were change into images frames per second. From there, locating the eyes and mouth can be performed. Once the eyes are located, measuring the intensity changes in the eye area determine the eyes are open or closed.

If the eyes are found closed for 4 consecutive frames, it is confirm that the driver is in drowsiness condition. Also, if the mouth is detected to be open for four consecutive frames it detects that the driver is yawning and gives an alert to the driver.

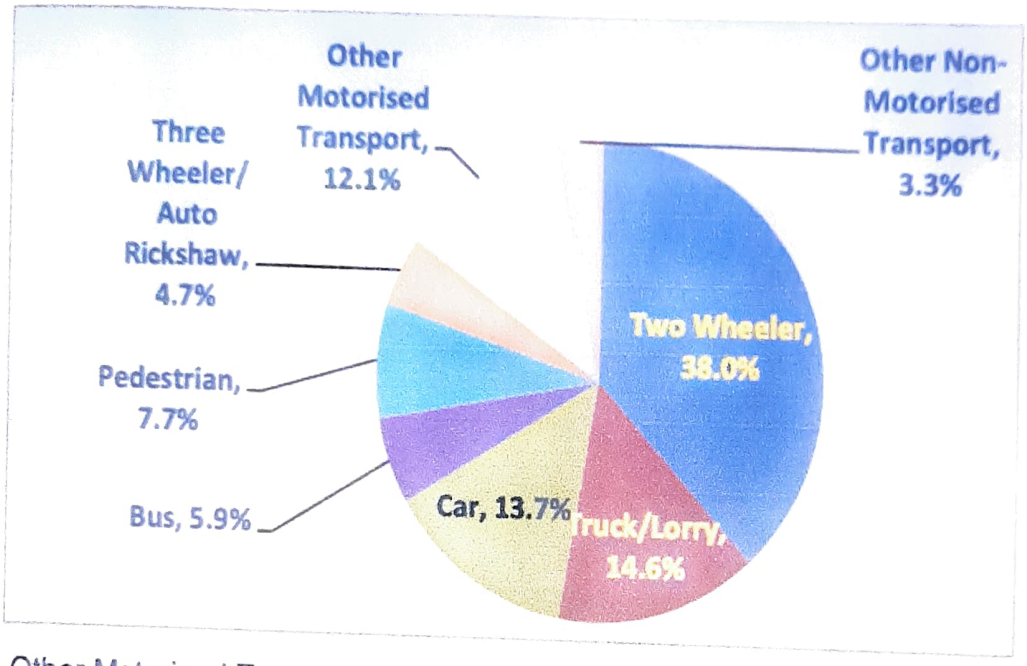
INDEX

List of Figures

CHAPTER 1

INTRODUCTION

Drowsiness is a state of near sleep, where the person has a strong desire for sleep. It has two distinct meanings, referring both to the usual state preceding falling asleep and the chronic condition referring to being in that state independent of a daily rhythm [1]. Sleepiness can be dangerous when performing tasks that require constant concentration, such as driving a vehicle. When a person is sufficiently fatigue while driving, they will experience drowsiness and this leads to increase the factor of road accident.



Other Motorised Transport includes SUV/Station Wagon, Jeep, Tractor etc.

Figure 1: Statistic of Road Accident from according to vehicle [2]

Figure 1 shows the statistic of road accident in India according to vehicle provided by NCRB(National Crime Records Bureau)[2]. The numbers of vehicles involved in road accident keep increasing each year. From Figure 1, car and taxi type of vehicles shows about nearly 13.7 % cases of road accident has been recorded. It keeps increasing every year. It shows the number of road accident were recorded by NCRB are nearly 500,000.

Figure 2 shows the drowsiness and fatigue conditions.

Figure 2: Examples of Drowsiness Condition & Fatigue

The development of technologies for detecting or preventing drowsiness while driving is a major challenge in the field of accident avoidance system. Because of the hazard that drowsiness presents on the road, methods need to be developed for counteracting its affects.

The aim of this project is to develop a simulation of drowsiness detection system. The focus will be placed on designing a system that will accurately monitor the open or closed state of the driver's eyes and mouth. By monitoring the eyes, it is believed that the symptoms of driver's drowsiness can be detected in sufficiently early stage, to avoid a car accident. Yawning detection is a method to assess the driver's fatigue. When a person is fatigue, they keep yawning to ensure that there is enough oxygen for the brain consumption before going to drowsiness state [3]. Detection of fatigue and drowsiness involves a sequence of images of a face, and the observation of eyes and mouth open or closed duration. Another method to detect eye closure is PERCLOS. This detection method is based on the time of eyes closed which refers to percentage of a specific time.

The analysis of face images is a popular research area with applications such as face recognition, and human identification and tracking for security systems. This project is focused on the localization of the eyes and mouth, which involves looking at the entire image of the face, and determining the position of the eyes and mouth, by applying the existing methods in image processing algorithm. Once the position of the eyes is located, the system is designed to determine whether the eyes and mouth are opened or closed, and detect fatigue and drowsiness.

1.1. BACKGROUND OF STUDY

Each year, there is an increase in road accidents cases involving cars and heavy vehicles like buses, lorries and trucks in Malaysia. Drowsiness and fatigue condition is one of the prime factors contributing to road accidents. Driving in this condition may result terrible causes since it affects the driver's judgment and concentration. Falling asleep on the wheel can be avoid if the drivers take efforts such as getting enough sleep before driving, taking caffeine or stop for a while to rest when the signs of fatigue and drowsiness appears.

However, in many cases, drivers refuse to take one of these steps even when they know that they are suffering from fatigue, and will continue driving. Therefore, detecting drowsiness is important as one of the steps to prevent the road accidents. This project proposed that yawning and eyes detection is the obvious signs of fatigue and drowsiness.

1.2. DEEP LEARNING [4]

Deep learning is a subset of Machine learning, which is essentially a neural network with three or more layers. These neural networks attempt to simulate the behaviour of the human brain—albeit far from matching its ability—allowing it to "learn" from large amounts of data. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy.

Deep learning drives many Artificial Intelligence applications and services that improve automation, performing analytical and physical tasks without human intervention. Deep learning technology lies behind everyday products and services (such as digital assistants,

voice-enabled TV remotes, and credit card fraud detection) as well as emerging technologies (such as self-driving cars).

1.3. PYTHON [5]

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming. Many other paradigms are supported via extensions, including design by contract and logic programming.

Python uses dynamic typing, and a combination of reference counting and a cycle-detecting garbage collector for management. It uses dynamic name resolution, which binds method and variable names during program execution.

1.3.1. PYTHON LIBRARIES [6]

- **Matplotlib** is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged. SciPy makes use of Matplotlib.

- **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 easier with OpenCV and both Python 2.7 and Python 3.

- **Dlib** is a general-purpose cross-platform software library written in the programming language C++. Its design is heavily influenced by ideas from design by contract and component-based software engineering, it is good, easy to use python bindings We have majorly used Dlib for face detection and facial landmark detection.

- **Cv2** is the module import name for opencv-pyhton, unofficial pre -built CPU-only OpenCV has many complicated steps involving building the module from scratch, which is unnecessary. I would recommend remaining with the OpenCV- python library. The library can be used to perform tasks like face detection, objection tracking, landmark detection, and much more. It supports multiple languages including python, java, C++.

- **NumPy** is a library for the python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

- **Pandas** is a software library written for the python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license.

- **OS** module in python provides functions for creating and removing a directory (folder), fetching its contents, changing and identifying the current directory, etc.

- **Tkinter** is the de facto way in python to create Graphical User interface (GUIs) and is included in all standard python Distributions. In fact, it's the only framework built into the python standard library.

- **Flask** is a micro web framework written in python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

- **Git** is a software for tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows (thousands of parallel branches running on different systems).

- **Version Control** is a system that records changes to a file or set of files over time so that you can recall specific versions later. For the examples in this book, you will use software source code as the files being version controlled, though in reality you can do this with nearly any type of file on a computer.

CHAPTER 2

LITERATURE REVIEW

There are many previous researches regarding driver drowsiness detection system that can be used as a reference to develop a real-time system on detecting drowsiness for drivers. There is also several method which use different approaches to detect the drowsiness signs. According to NCRB (National Crime Record Bureau), from the year of 2016 until 2020, they were approx. 50,000 cases of road accidents have been investigated by the NCRB crash team. [2]

2.1. DROWSINESS AND FATIGUE

Antoine Picot et al, [7] stated that drowsiness is where a person is in the middle of awake and sleepy state. This situation leads the driver to not giving full attention to their driving. Therefore, the vehicle can no longer be controlled due to the driver being in a semi-conscious state. According to Gianluca Borghini et al, [8] mental fatigue is a factor of drowsiness and it caused the person who experiences to not be able to perform because it decreases the efficiency of the brain to respond towards sudden events.

2.2. ELECTROENCEPHALOGRAPHY (EEG) FOR DROWSINESS DETECTION

Electroencephalography (EEG) is a method that measures the brain electrical activity. As shown in Figure 3, it can be used to measure the heartbeat, eye blink and even major physical movement such as head movement. It can be used on human or animal as subjects to get the brain activity. It uses a special hardware that place sensors around the top of the head area to sense any electrical brain activity.

Figure 3: Examples of EEG Data Collecting [7]

Electroencephalography (EEG) is a method that measures the brain electrical activity. As shown in Figure 3, it can be used to measure the heartbeat, eye blink and even major physical movement such as head movement. It can be used on human or animal as subjects to get the brain activity. It uses a special hardware that place sensors around the top of the head area to sense any electrical brain activity.

Authors in [9] mentioned that from the method that has been implemented by the previous researcher to detect drowsiness signs, the EEG method is best to be applied for drowsiness and fatigue detection. In the method, EEG have four types of frequency components that can be analysed, i.e. alpha (α), beta (β), theta (θ) and delta (δ). When the power is increased in alpha (α) and delta (δ) frequency bands, it shows that the driver is facing fatigue and drowsiness.

The disadvantages of this method are, it is very sensitive to noise around the sensors. For example, when the person is doing the EEG experiment, the surrounding area must be completely silent. The noise will interfere with the sensors that detect the brain activity. Another disadvantage of this method is that even if the result might be accurate, it is not suitable to use for real driving application [10]. Imagine when a person is driving and he is wearing something on his head with full of wires and when the driver moves their head, the wire may strip off from their place. Even though it is not convenient to be used for real-time driving but for experiment purposes and data collection, it is one of the best methods so far [7].

2.3. DROWSINESS DETECTION USING FACE DETECTION SYSTEM

Drowsiness can be detected by using face area detection [11], [12] and [13]. The methods to detect drowsiness within face area are vary due to drowsiness sign are more visible and clear to be detected at face area. From the face area, we can detect the eyes location. From eyes detection, author in [11] stated that there are four types of eyelid movement that can be used for drowsiness detection. They are complete open, complete close, and in the middle where the eyes are from open to close and vice versa [11]. Figure 4 is an example of the image taken for detecting eyelid movement.

a) Open eye b) Close eye c) Processed close eye

Figure 4: Examples of Eyelid Movement [11]

The algorithm processes the images captured in grey-scale method; where the colour from the images is then transformed into black and white [14] [15]. Working with black and white images is easier because only two parameters have to be measured. The author then performs the edge detection to detect the edges of eyes so that the value of eyelid area can be calculated.

The problem occurring with this method is that the size area of eye might vary from one person to another. Someone may have small eyes and looks like it is sleepy but some are not. Other than that, if the person is wearing glasses, there is obstacle to detect eye region. The images that being captured must be in certain range from the camera because when the distance is far from the camera, the images are blurred [12].

2.4. PERCLOS (PERCENTAGE OF EYE CLOSURE)

Drowsiness can be captured by detecting the eye blinks [11] and percentage of eye closure (PERCLOS). For eye blink detection, [11] propose a method which learned the pattern of duration of eyelid closed. According to [10], 'this proposed method measures the time for a person closed their eyes and if they are closed longer than the normal eye blink time, it is possible that the person is falling asleep'. In [10], the author mentioned that 'nearly 310.3ms are the average of normal person eye blink'.

PERCLOS method proposes that drowsiness is measured by calculating the percentage of the eyelid 'droops' [16]. Sets of eye open and eye closed have been stored in the software library to be used as a parameter to differentiate either the eyes is fully open or fully closed. For eyelid to droops, it happened in much slower time as the person is slowly falling asleep. Hence, the transition of the driver's drowsy can be recorded. Thus, PERCLOS method put a proportional value where when the eyes is 80% closed, which it is nearly to fully close, it assumed that the driver is drowsy [7], [10], and [17].

This method is not convenient to be used in real-time driving as it needs fix threshold value of eye opening for the PERCLOS method to perform accurately. Both methods to detect drowsiness using eye blink pattern and PERCLOS have the same problem where the camera need to be placed at a specific angle in order to get a good image of video with no disturbance of eyebrow and shadow that cover the eyes.

2.5. YAWNING DETECTION METHOD

According to [18], drowsiness of a person can be observed by looking at their face and behaviour. The author propose a method where drowsiness can be detected by mouth positioning and the images were process by using cascade of classifier that has been proposed by Viola-Jones for faces. The images were compared with the set of images data for mouth and yawning [18]. Some people will close their mouth by their hand while yawning. It is an obstacle to get good images if a person is closing their mouth while yawning but yawning is definitely a sign of a person having drowsiness and fatigue.

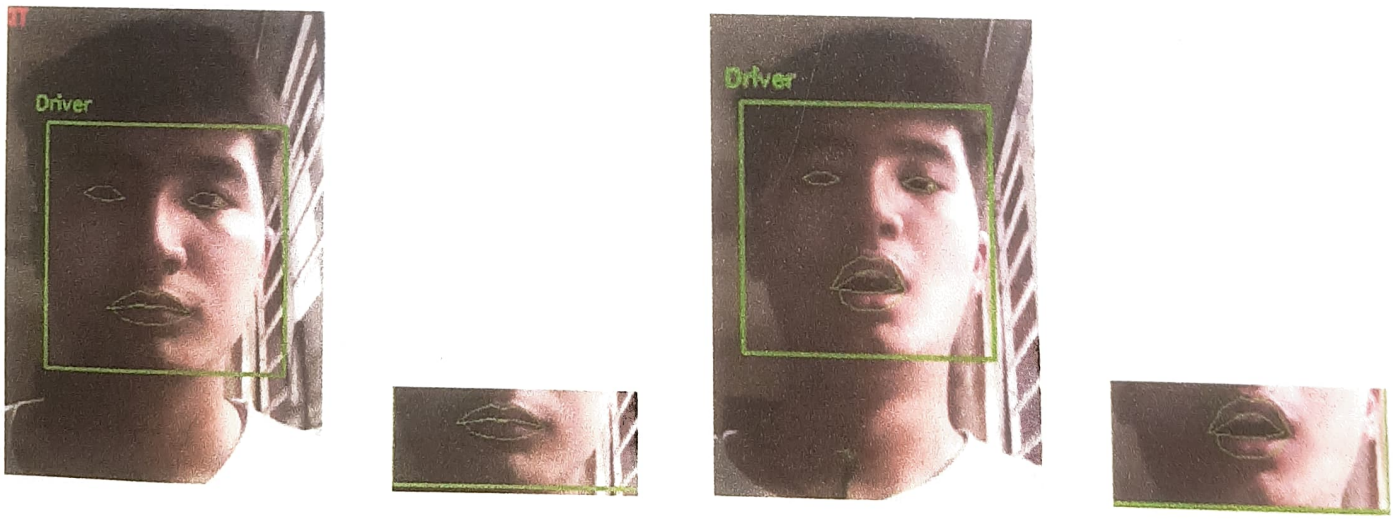


Figure 5: The examples of yawning detection method.

After gone through the research papers and the existing methods, this project proposed that eyes and yawning detection method will be used. Eye blink duration gives the data that the longer the person's close their eyes, the drowsier it will be considered. It is because when a person is in drowsy state; its eyes will be closed longer than the normal eye blink. Other than that, yawning is one of the symptoms of drowsiness where it is a normal human response when yawning is the sign that they feel drowsy or fatigue.

CHAPTER 3

PLANNING

It is essential to determine the tasks to be performed and properly manage allocation of tasks among individuals involved in the software development. Hence, planning is important as it results in effective software development.

Project planning is an organized and integrated management process, which focuses on activities required for successful completion of the project. It prevents obstacles that arise in the project such as changes in projects or organization's objectives, non-availability of resources, and so on. Project planning also helps in better utilization of resources and optimal usage of the allotted time for a project. The other objectives of project planning are listed below.

i. It defines the roles and responsibilities of the project management team members.
ii. It ensures that the project management team works according to the business objectives.
iii. It checks feasibility of the schedule and user requirements.
iv. It determines project constraints.

Several individuals help in planning the project. These include senior management and project management team. Senior management is responsible for employing team members and providing resources required for the project. The project management team, which generally includes project managers and developers, is responsible for planning, determining, and tracking the activities of the project. Table lists the tasks performed by individuals involved in the software project.

Effective project planning helps to minimize the additional costs incurred on the project while it is in progress. For effective project planning, some principles are followed. These principles are listed below.

i. **Planning is necessary:** Planning should be done before a project begins. For effective planning, objectives and schedules should be clear and understandable.

ii. **Risk analysis:** Before starting the project, senior management and the project management team should consider the risks that may affect the project. For example, the user may desire changes in requirements while the project is in progress. In such a case, the estimation of time and cost should be done according to those requirements (new requirements).

iii. **Tracking of project plan:** Once the project plan is prepared, it should be tracked and modified accordingly.

iv. **Meet quality standards and produce quality deliverables:** The project plan should identify processes by which the project management team can ensure quality in software. Based on the process selected for ensuring quality, the time and cost for the project is estimated.

v. **Description of flexibility to accommodate changes:** The result of project planning is recorded in the form of a project plan, which should allow new changes to be accommodated when the project is in progress.

Project planning comprises project purpose, project scope, project planning process, and project plan. This information is essential for effective project planning and to assist project management team in accomplishing user requirements.

3.1. ITERATIVE WATERFALL MODEL

The Software is one of the major components of management information system. In a practical software development project, the classical waterfall model is hard to use. So, Iterative Waterfall model can be thought as incorporating the necessary changes to the classical waterfall model to make it usable in practical software development projects. It is almost same as the classical waterfall model except some changes are made to increase the efficiency of the software development.

The Iterative waterfall model provides feedback paths from every phases to its preceding phases, which is the main difference from the classical waterfall model.

When errors are detected at some later phase, these feedback paths allow correcting errors committed by programmers during some phase. The feedback paths allow the phase to be reworked in which errors are committed and these changes are reflected in the later phases. But, there is no feedback path to the stage- feasibility study, because once a project has been taken, does not give up the project easily. It is good to detect errors in the same phase in which they are committed. It reduces the effort and time required to correct errors.

The simplest process model is the waterfall model, which states that the phases are organized in a linear order. The model was originally proposed by Royce, though variations of the model have evolved depending on the nature of activities and the flow of control between them. In this model, a project begins with feasibility analysis. Upon successfully demonstrating the feasibility of a project, the requirements analysis and project planning begins. The design starts after the requirements analysis is complete, and coding begins after the design is complete. Once the programming is completed, the code is integrated and testing is done. Upon successful completion of testing, the system is installed. After this, the regular operation and maintenance of the system takes place.

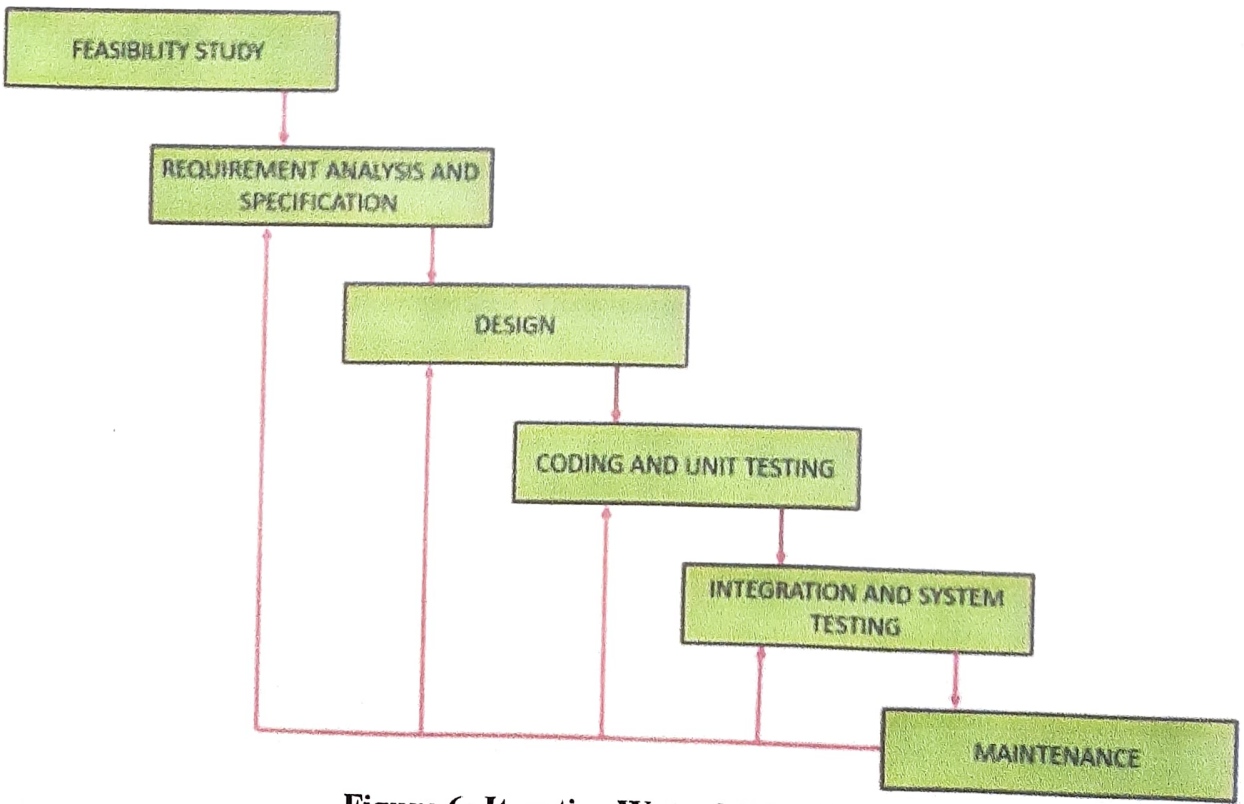


Figure 6: Iterative Waterfall Model

Feasibility study is defined as the practical extent to which a project can be performed successfully. To evaluate feasibility, a feasibility study is performed, which determines whether the solution considered to accomplish the requirements is practical and workable in the software. Information such as resource availability, cost estimation for software development, benefits of the software to the organization after it is developed and cost to be incurred on its maintenance are considered during the feasibility study. The objective of the feasibility study is to establish the reasons for developing the software that is acceptable to users, adaptable to change and conformable to established standards. Various other objectives of feasibility study are listed below.

i. To analyse whether the software will meet organizational requirements.
ii. To determine whether the software can be implemented using the current technology and within the specified budget and schedule.
iii. To determine whether the software can be integrated with other existing software.

Various types of feasibility that are commonly considered include technical feasibility, operational feasibility, and economic feasibility.

Technical feasibility assesses the current resources (such as hardware and software) and technology, which are required to accomplish user requirements in the software within the allocated time and budget. For this, the software development team ascertains whether the current resources and technology can be upgraded or added in the software to accomplish specified user requirements. Technical feasibility also performs the following tasks.

i. Analysis the technical skills and capabilities of the software development team members.
ii. Determines whether the relevant technology is stable and established.
iii. Ascertains that the technology chosen for software development has a large number of users so that they can be consulted when problems arise or improvements are required.

Operational feasibility assesses the extent to which the required software performs a series of steps to solve business problems and user requirements. This feasibility is dependent on human resources (software development team) and involves visualizing whether the software will operate after it is developed and be operative once it is installed. Operational feasibility also performs the following tasks.

i. Determines whether the problems anticipated in user requirements are of high priority.
ii. Determines whether the solution suggested by the software development team is acceptable.
iii. Analysis whether users will adapt to a new software.
iv. Determines whether the organization is satisfied by the alternative solutions proposed by the software development team.

Economic feasibility determines whether the required software is capable of generating financial gains for an organization. It involves the cost incurred on the software development team, estimated cost of hardware and software, cost of performing feasibility study, and so on. For this, it is essential to consider expenses made on purchases (such as hardware purchase) and activities required to carry out software development. In addition, it is necessary to

consider the benefits that can be achieved by developing the software. Software is said to be economically feasible if it focuses on the issues listed below.

i. Cost incurred on software development to produce long-term gains for an organization.

ii. Cost required to conduct full software investigation (such as requirements elicitation and requirements analysis).

iii. Cost of hardware, software, development team, and training.

Feasibility Study Process

Feasibility study comprises the following steps:

i. **Information assessment:** Identifies information about whether the system helps in achieving the objectives of the organization. It also verifies that the system can be implemented using new technology and within the budget and whether the system can be integrated with the existing system.

ii. **Information collection:** Specifies the sources from where information about software can be obtained. Generally, these sources include users (who will operate the software), organization (where the software will be used), and the software development team (which understands user requirements and knows how to fulfil them in software).

iii. **Report writing:** Uses a feasibility report, which is the conclusion of the feasibility study by the software development team. It includes the recommendations whether the software development should continue. This report may also include information about changes in the software scope, budget, and schedule and suggestions of any requirements in the system.

iv. **General information:** Describes the purpose and scope of feasibility study. It also describes system overview, project references, acronyms and abbreviations, and points of contact to be used. System overview provides description about the name of the organization responsible for the software development, system name or title, system category, operational status, and so on.

3.1.1. MANAGEMENT SUMMARY

Provides the following information.

i. **Environment:** Identifies the individuals responsible for software development. It provides information about input and output requirements, processing requirements of the software and the interaction of the software with other software. It also identifies system security requirements and the system's processing requirements

ii. **Current functional procedures:** Describes the current functional procedures of the existing system, whether automated or manual. It also includes the data-flow of the current system and the number of team members required to operate and maintain the software.

iii. **Functional objective:** Provides information about functions of the system such as new services, increased capacity, and so on.

iv. **Performance objective:** Provides information about performance objectives such as reduced staff and equipment costs, increased processing speeds of software, and improved controls.

v. **Assumptions and constraints:** Provides information about assumptions and constraints such as operational life of the proposed software, financial constraints, changing hardware, software and operating environment, and availability of information and sources.

vi. **Methodology:** Describes the methods that are applied to evaluate the proposed software in order to reach a feasible alternative. These methods include survey, modelling, benchmarking, etc.

vii. **Evaluation criteria:** Identifies criteria such as cost, priority, development time, and ease of system use, which are applicable for the development process to determine the most suitable system option.

viii. **Recommendation:** Describes a recommendation for the proposed system. This includes the delays and acceptable risks.

ix. **Proposed software:** Describes the overall concept of the system as well as the procedure to be used to meet user requirements. **In** addition, it provides information about improvements, time and resource costs, and impacts. Improvements are performed to enhance the functionality and performance of the existing software. Time and resource costs include the costs associated with software development from its requirements to its maintenance and staff training. Impacts describe the possibility of future happenings and include various types of impacts as listed below.

x. **Equipment impacts:** Determine new equipment requirements and changes to be made in the currently available equipment requirements.

xi. **Software impacts:** Specify any additions or modifications required in the existing software and supporting software to adapt to the proposed software.

xii. **Organizational impacts:** Describe any changes in organization, staff and skills requirement.

xiii. **Operational impacts:** Describe effects on operations such as user-operating procedures, data processing, data entry procedures, and so on.

xiv. **Developmental impacts:** Specify developmental impacts such as resources required to develop databases, resources required to develop and test the software, and specific activities to be performed by users during software development.

xv. **Security impacts:** Describe security factors that may influence the development, design, and continued operation of the proposed software.

xvi. **Alternative systems:** Provide description of alternative systems, which are considered in a feasibility study. This also describes the reasons for choosing a

particular alternative system to develop the proposed software and the reason for rejecting alternative systems.

Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed. In software engineering, such requirements are often called functional specifications. Requirements analysis is an important aspect of project management.

Requirements analysis involves frequent communication with system users to determine specific feature expectations, resolution of conflict or ambiguity in requirements as demanded by the various users or groups of users, avoidance of feature creep and documentation of all aspects of the project development process from start to finish. Energy should be directed towards ensuring that the final system or product conforms to client needs rather than attempting to mold user expectations to fit the requirements.

Requirements analysis is a team effort that demands a combination of hardware, software and human factors engineering expertise as well as skills in dealing with people.

In the design process of iteration model an iterative process starts with a simple implementation of a small subset of the developed software product requirements and iteratively enhances the evolving versions of product until the full system is implemented. In this design process in each successively iteration, the product design modifications are made and new functional capabilities are added with the new version of the product. The basic idea behind this modelling approach is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental).
The iterative model is the combination of the iterative design process and incremental build model for development. In this approach during software application developed, a number of iteration is required in a cyclic progress to develop complete software. It is also happening that more than one iteration process is done under the development of software product at the same time. This process may be described as an "evolutionary acquisition" or "incremental build" approach.

The working behaviour of iteration model is to subdivide the whole process in to various small pieces. During each successively iteration, the development module goes through the requirements, design, implementation and testing phases. Each subsequent release of the module enhances the functional capabilities to the previous release. This process of cyclic enhance in the functionality continues till the complete system is ready as per the requirement.

The key to successful use of this iterative model in the software development lifecycle is rigorous validation of requirements, and verification and testing of each version of the software against those requirements within each cycle of the model. As the software evolves through successive cycles, tests have to be repeated and extended to verify each version of the software.

Testing is an essential part of the Software Development Process. A robust and stable software product can be delivered with the use of standard testing methodologies that will help to predict the timeline of the software system.

A software application may turn even more complex with a large number of platforms and devices. More importantly, it is required to ensure whether they meet the specified requirements and can be efficiently installed and operated on the user's machine or not.

Methodologies can be considered as the set of testing mechanisms used in software development lifecycle from Unit Testing to System Testing. Selecting an appropriate testing methodology is considered to be the core of the testing process.

Basically, there are 3 testing methodologies which are used for testing. They are White Box Testing, Black Box Testing, and Grey Box Testing. These are also called as *Testing Techniques*.

White box testing technique is used to examine the program structure and business logic, it validates the code or program of an application. It is also called as *Clear Box Testing, Glass Box Testing or Open Box Testing*.
White Box Testing Techniques include:

i. **Statement Coverage:** Examines all the programming statements.
ii. **Branch Coverage:** Series of running tests to ensure if all the branches are tested.
iii. **Path Coverage:** Tests all the possible paths to cover each statement and branch.

Black Box testing method is used to test the functionality of an application based on the requirement specification. Unlike White Box Testing it does not focus on internal structure/code of the application.
Black Box Techniques include:

i. Boundary Value analysis
ii. Equivalence Partitioning(Equivalence Class Partitioning)
iii. Decision Tables
iv. Domain Tests
v. State Models
vi. Exploratory Testing (Requires less preparation and also helps to find the defects quickly).

3.2. PHASE CONTAINMENT OF ERRORS

The principle of detecting errors are close to their points of commitment as possible is known as Phase containment of errors.

3.3. ADVANTAGE OF ITERATIVE WATERFALL MODEL

3.3.1. FEEDBACK PATH

In the classical waterfall model, there are no feedback paths, so there is no mechanism for error correction. But in iterative waterfall model feedback path from one phase to its preceding phase. Allows correcting the errors which are committed and these changes are reflect in the later phase.

3.3.2. SIMPLE

Iterative waterfall model is very simple to understand and use. That's why it is one of the most widely used software development model.

3.4. DRAWBACKS OF ITERATIVE WATERFALL MODEL

3.4.1. DIFFICULT TO INCORPORATE CHANGE REQUESTS

The major drawback of the iterative waterfall model is that all the requirements must be clearly stated before starting of the development phase. Customer may change requirements after some time but the iterative waterfall model does not leave any scope to incorporate change requests that are made after development phase starts.

3.4.2. INCREMENTAL DELIVERY NOT SUPPORTED

In the iterative waterfall model, the full software is completely developed and tested before delivery to the customer. There is no scope for any intermediate delivery. So, customers have to wait long for getting the software.

3.4.3. OVERLAPPING OF PHASES NOT SUPPORTED

Iterative waterfall model assumes that one phase can start after completion of the previous phase, but in real projects, phases may overlap to reduce the effort and time needed to complete the project.

3.4.4. RISK HANDLING NOT SUPPORTED

Projects may suffer from various types of risks. But, Iterative waterfall model has no mechanism for risk handling.

3.4.5. LIMITED CUSTOMER INTERACTIONS

Customer interaction occurs at the start of the project at the time of requirement gathering and at project completion at the time of software delivery. These fewer interactions with the customers may lead to many problems as the finally developed software may differ from the customers' actual requirements.

3.5. ORGANIZATION STRUCTURE

The organizational team structure that we have followed is democratic team structure the management structure and communication path in an egoless team are illustrated in (fig: 7) democratic team is that where one team member is designated team leaders and occupies the position of the first among equal. The team leader position does not usually rotate among team members in a democratic team because a team function best when one is responsible for coordinating team activities and for making.

Final decision in situation where consensus cannot be reached. Advantage of decision. The opportunity for team member to learn from one another, and the increased satisfaction that accrues from good communication in an open, nonthreatening work environment.

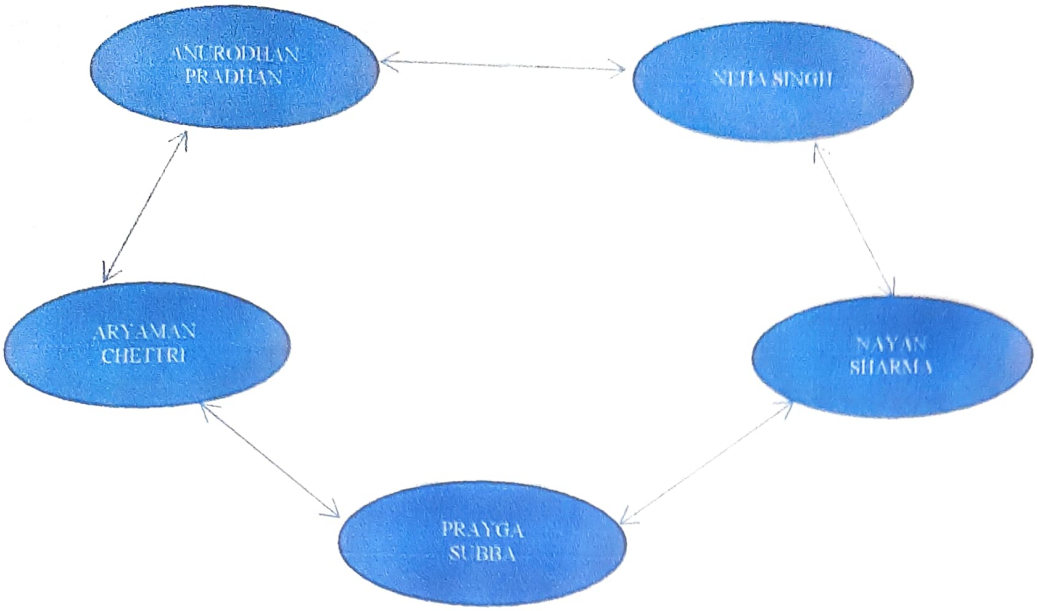


Figure 7: Communication Path

As mention in the fig 7 it shows that there is an equal administrative leadership, at different times, different members of the group provide technical leadership.
It suffers from less manpower turnover. Democratic team structure suits to less understood problems, since a group of Engineers can invent better solution than a single individual as in a chief programmer team. It encourages egoless programming as programmer can share and review one another's work.

3.6. GANTT CHART

The following Gantt chart shows the timeline of our project progress and completion.

Activity	Time Frame										
	AUG (2021)	SEP (2021)	OCT (2021)	NOV (2021)	DEC (2021)	JAN (2022)	FEB (2022)	MAR (2022)	APR (2022)	MAY (2022)	JUN (2022)
Literature Survey											
Problem Identification											
Design											
Implementation											
Testing and Validation											
Report Writing											

Proposed Achieved Ongoing

Table 1: Gantt Chart of the Project

CHAPTER 4

PROBLEM DEFINATION AND SOLUTION STRATEGY

4.1. PROBLEM DEFINATION

Driver's inattention might be the result of a lack of alertness when driving due to driver drowsiness and distraction. Driver distraction occurs when an object or event draws a person's attention away from the driving task. Unlike driver distraction, driver drowsiness involves no triggering event but, instead, is characterized by a progressive withdrawal of attention from the road and traffic demands. Both driver drowsiness and distraction, however, might have the same effects, that is decreased driving performance, longer reaction time, and an increased risk of crash involvement

4.2. SOLUTION STRATEGY

Current drowsiness detection systems monitoring the driver's condition requires complex computation and expensive equipment, not comfortable to wear during driving and is not suitable for driving conditions; for example, Electroencephalography (EEG) and Electrocardiography (ECG), i. e. detecting the brain frequency and measuring the rhythm of heart, respectively.

A drowsiness detection system which use a camera placed in front of the driver is more suitable to be use but the physical signs that will indicate drowsiness need to be located first in order to come up with a drowsiness detection algorithm that is reliable and accurate. Lighting intensity and while the driver tilt their face left or right are the problems occur during detection of eyes and mouth region.

Therefore, this project aims to analyse all the previous research and method, hence propose a method to detect drowsiness by using video or webcam. It analyses the video images that have been recorded and come up with a system that can analyse each frame of the video.

4.3. FLOW CHART

Before starting any research or project, basic information of the related topic is required to ensure that the author understands what the project is all about. In this stage, the background of study helps the author understands the relation between drowsiness and fatigue. It also helps the author in understanding the seriousness of driving a motored vehicle in drowsiness condition. It is proven that driving the vehicle in fatigue and drowsiness condition is a lead factor to road accidents.

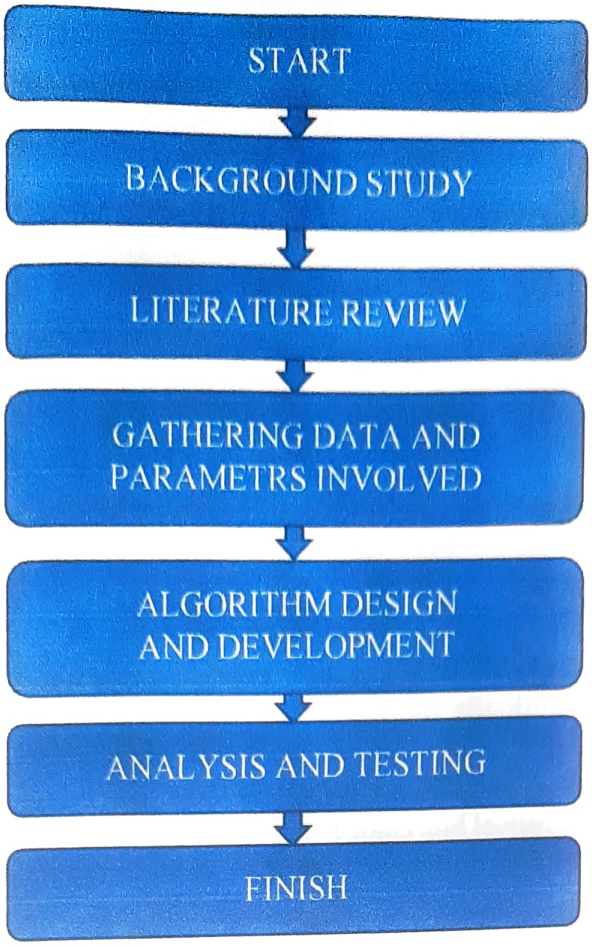


Figure 8: Flowchart of the Project

4.4. SIGNIFICANCE OF THIS PROJECT

Drowsiness and fatigue lead the cause of road accident in Malaysia. Thus, Driver Drowsiness Detection by Using Webcam is being introduced to minimize and reduce the number of accidents involving cars, lorries and trucks. It detects the drowsiness signs and alerts drivers when they are in drowsy state.

4.5. OBJECTIVES

The project focuses on these objectives, which are:

- To suggest ways to detect fatigue and drowsiness while driving.
- To study on eyes and mouth from the video images of participants in the experiment of driving simulation conducted by MIROS that can be used as an indicator of fatigue and drowsiness.
- To investigate the physical changes of fatigue and drowsiness.
- To develop a system that use eyes closure and yawning as a way to detect fatigue and drowsiness.

4.6. SCOPE OF STUDY

In this project, the author will focus on these following procedures:

- Basic concept of drowsiness detection system
- Familiarize with the signs of drowsiness
- Determine the drowsiness from these parameters
    - Eye blink
    - Area of the pupils detected at eyes
    - Yawning

- Data collection and measurement.
- Integration of the methods chosen.
- Coding development and testing.
- Complete testing and improvement.

4.7. RELEVANCY OF THE PROJECT

This project is relevant to the implementation since fatigue and drowsiness drivers contribute to the percentage of road accidents. Many researches have been conducted to implement safe driving systems in order to reduce road accidents. Detecting the driver's alertness and drowsiness is an efficient way to prevent road accidents. With this system, drivers who are drowsy will be alerted by an alarm to regulate consciousness, attention and concentration of the drivers. This will help to reduce the number of road accidents. This project is an active topic that is still being enhanced and improved by researches and can be applied in many areas such as detecting the attention-level of students in classrooms and lectures. This is also relevant to the three author's field of study since it requires the author to apply and combine the knowledge of electronics, programming and algorithms.

CHAPTER 5

PROJECT IMPLEMENTATION

5.1. MOTIVATION

According to National Highway Traffic Safety Administration, every year about 1,00,000 police-reported crashes involves drowsy driving. These crashes result in more than 1,550 fatalities and 71,000 injuries. The real number may be much higher, however, as it is difficult to determine whether a driver was drowsy at the time of a crash. So, we tried to build a system that detects whether a person is drowsy and alert him.

5.2. WORKING DETAILS

The basic thing about drowsiness detection is pretty simple. We first detect a face using dlib's frontal face detector. Once the face is detected , we try to detect the facial landmarks in the face using the dlib's landmark predictor. The landmark predictor returns 68 (x, y) coordinates representing different regions of the face, namely - mouth, left eyebrow, right eyebrow, right eye, left eye, nose and jaw. We don't need all the landmarks, here we need to extract only the eye and the mouth region [19][20].

Now, after extracting the landmarks we calculate the Eye Aspect Ratio (EAR) as:

```
def eye_aspect_ratio(eye):
        # Vertical eye landmarks
        A = dist.euclidean(eye[1], eye[5])
        B = dist.euclidean(eye[2], eye[4])
        # Horizontal eye landmarks
        C = dist.euclidean(eye[0], eye[3])

        # The EAR Equation
        EAR = (A + B) / (2.0 * C)
        return EAR
```

Figure 9: EAR Code Snippet

The eye region is marked by 6 coordinates. These coordinates can be used to find whether the eye is open or closed if the value of EAR is checked with a certain threshold value [21][22].

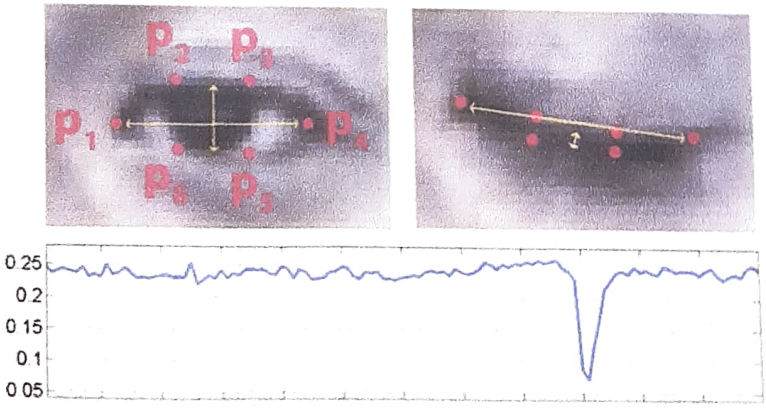


Figure 10. Calculation of EAR

In the same way we have calculated the aspect ratio for the mouth to detect if a person is yawning. Although, there is no specific metric for calculating this, so I have taken four points, 2 each from the upper and lower lip and calculated the mean distance between them as

```
def mouth_aspect_ratio(mouth):
        A = dist.euclidean(mouth[13], mouth[19])
        B = dist.euclidean(mouth[14], mouth[18])
        C = dist.euclidean(mouth[15], mouth[17])

        MAR = (A + B + C) / 3.0
        return MAR
```

Figure 11: MAR Code Snippet

5.3. RESULTS

The GUI has been created using basic HTML, CSS and JavaScript and we have used Flask to render the python code into the website. Tkinter has also been used in order to make things simpler.

5.3.1. The GUI:

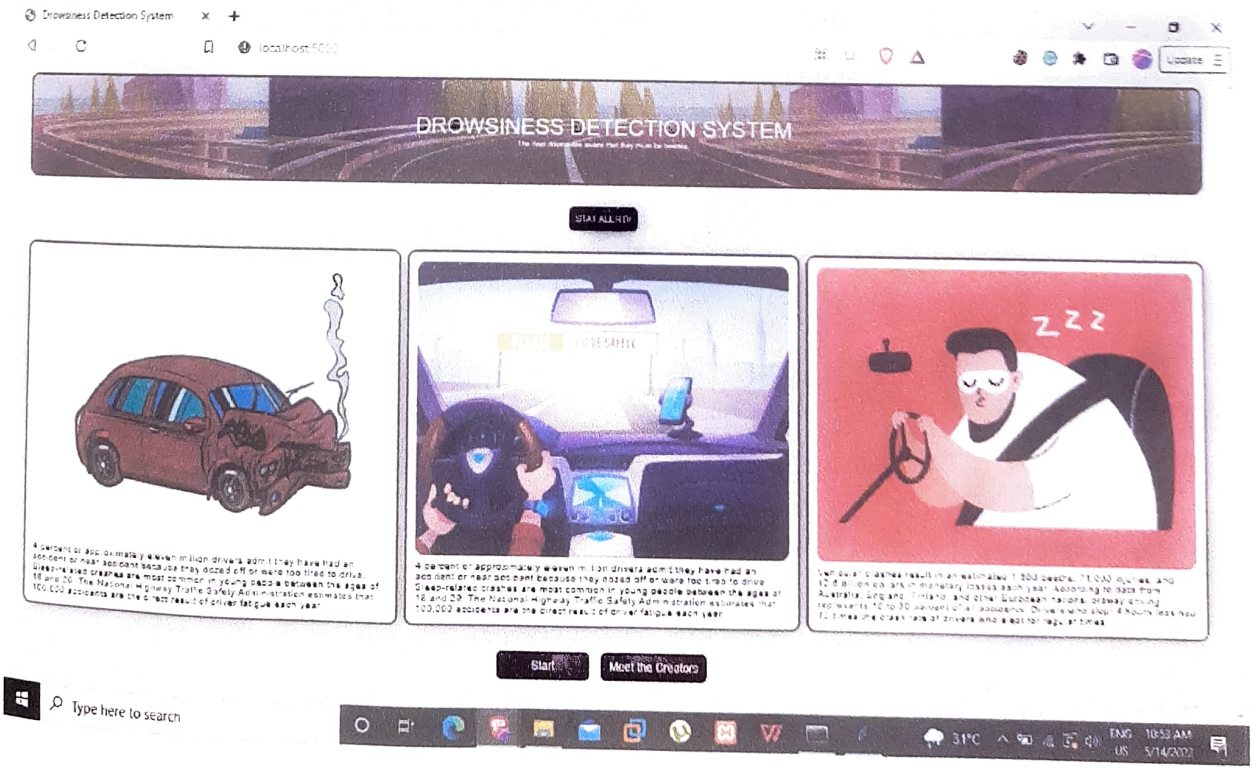


Figure 12: Main Page

5.3.1.1. Main Page

The main page of the webapp contains some basic statistics about the drowsy driving under the "Stat Alert !!" Section. Below this section there are two buttons :

- Start: To start the program.
- Meet the Creators: Navigates to another page with creators details.

5.3.1.2. Drowsiness Detection Dialogue Box

Once you click the start button at the bottom of the page, a dialogue box pops out containing 3 buttons.

- Run using web cam: It starts the webcam to detects the face for drowsiness
- Run using phone cam: It starts the camera of your phone to detect the face for drowsiness.
- Exit: It exits the program.

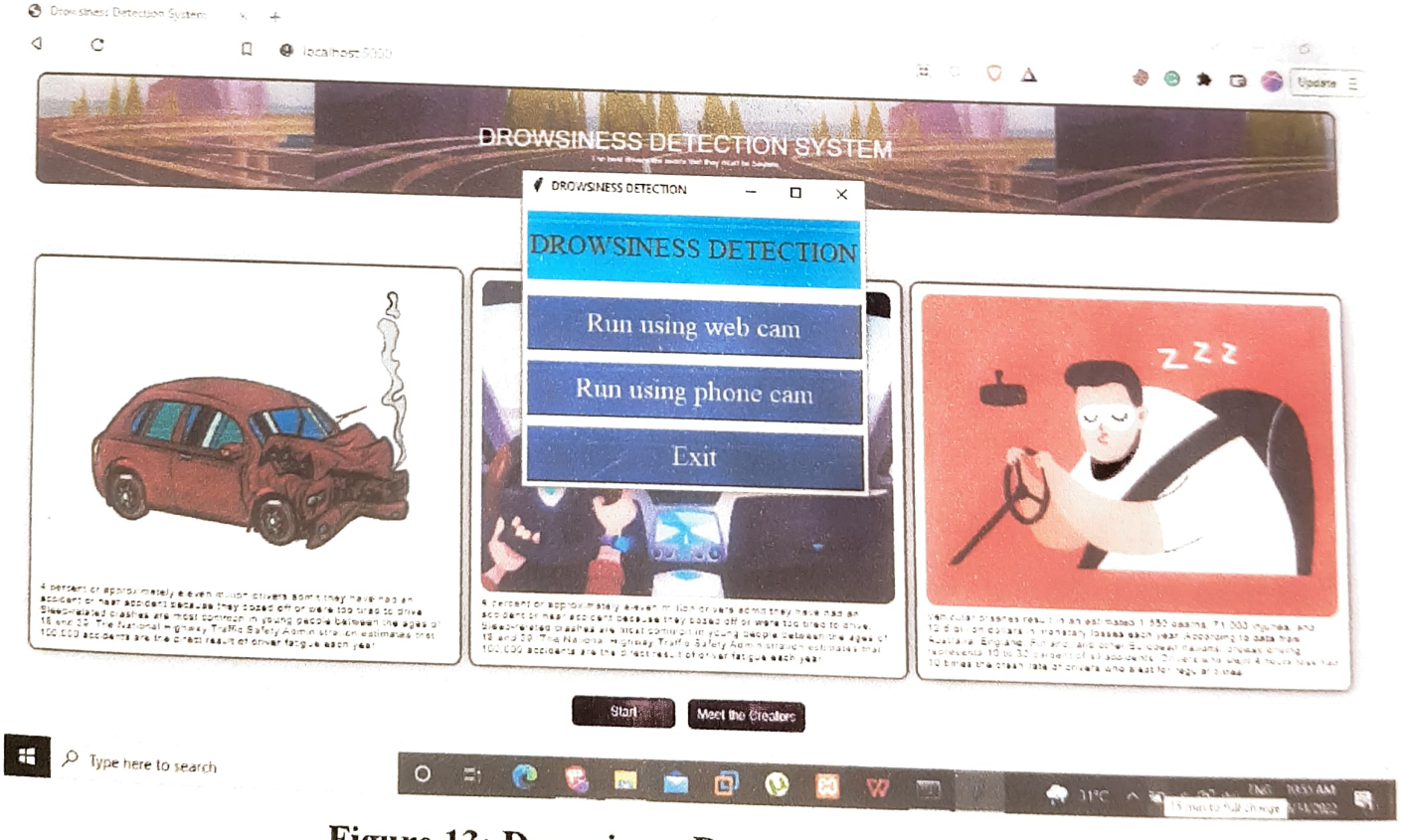


Figure 13: Drowsiness Detection Dialogue Box

5.3.1.3. Meet the Creators

This page gives the basic information about the creators of the project.

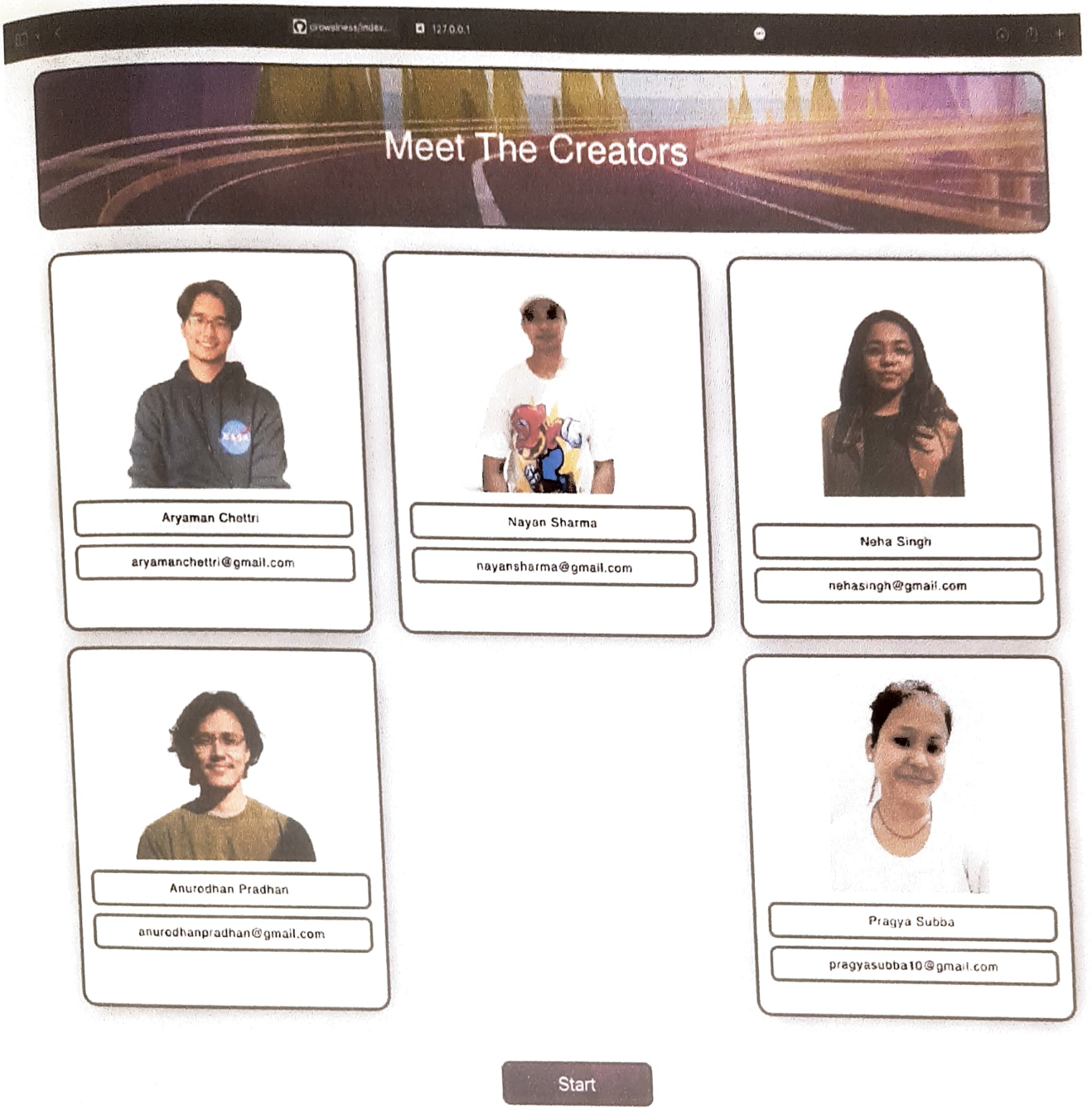


Figure 14: Meet the Creators Page

5.4. OUTPUT

The outputs of the working system detecting drowsiness is shown as

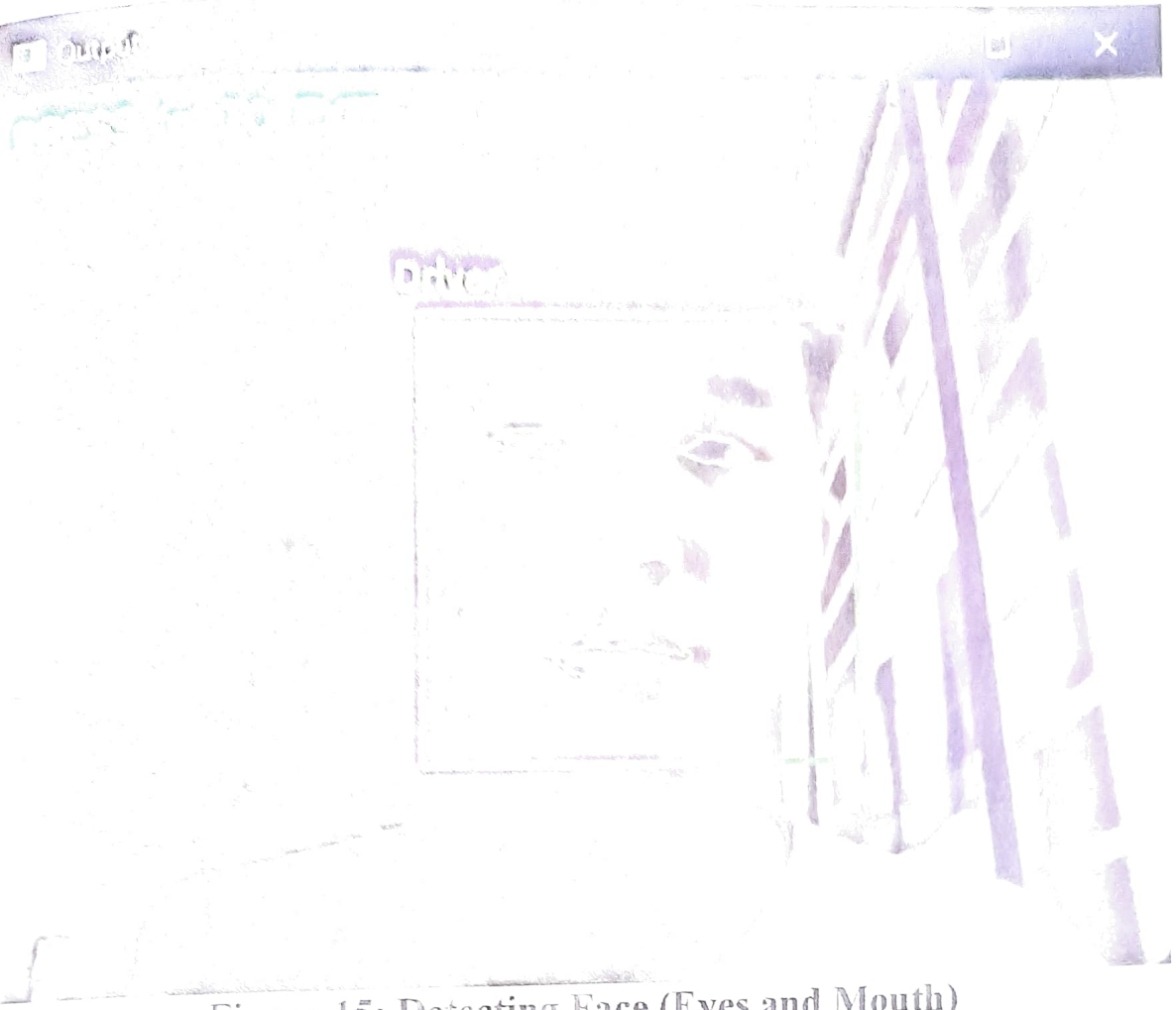


Figure 15: Detecting Face (Eyes and Mouth)

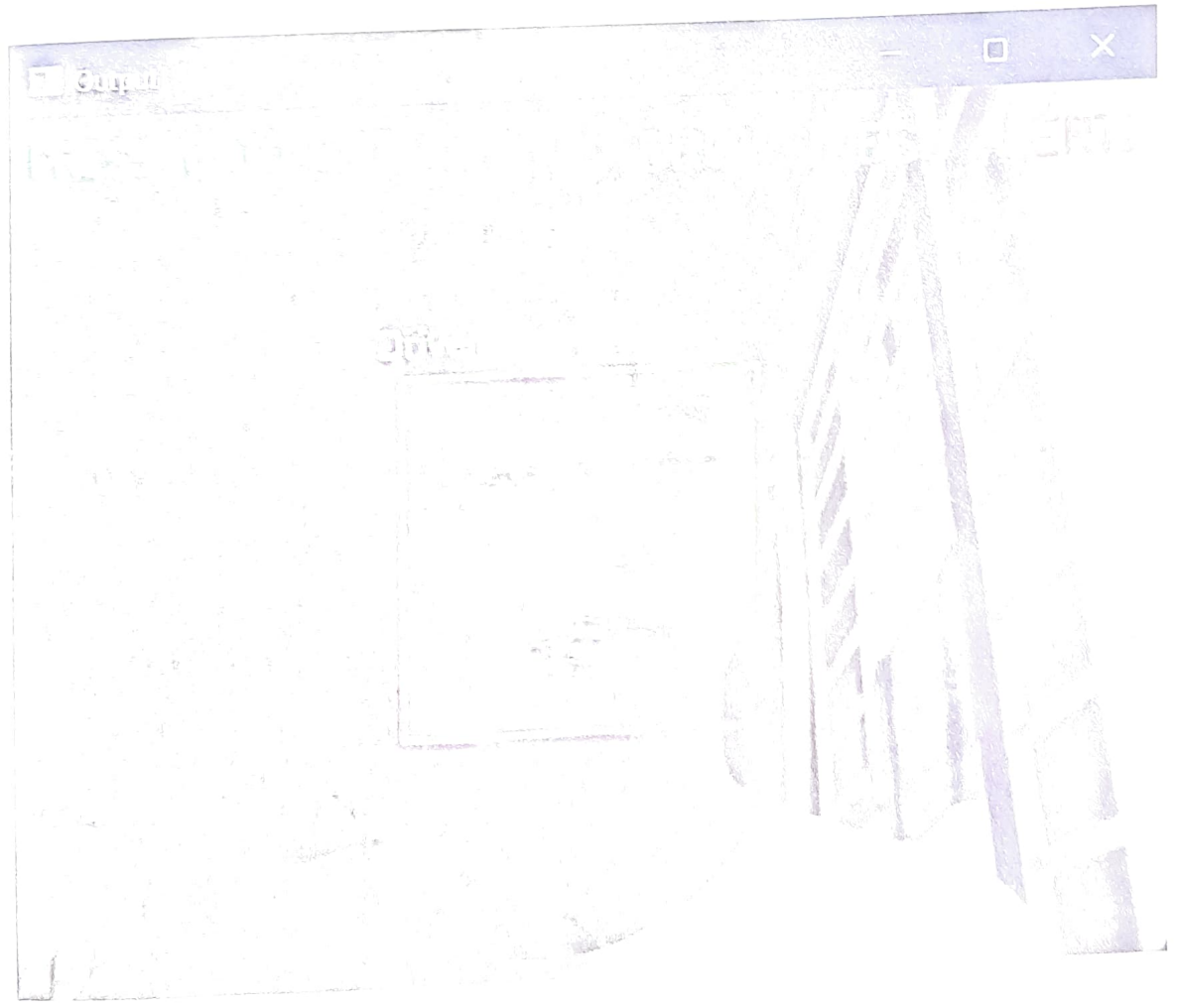


Figure 16: Detected Drowsy Eyes and gave an alert

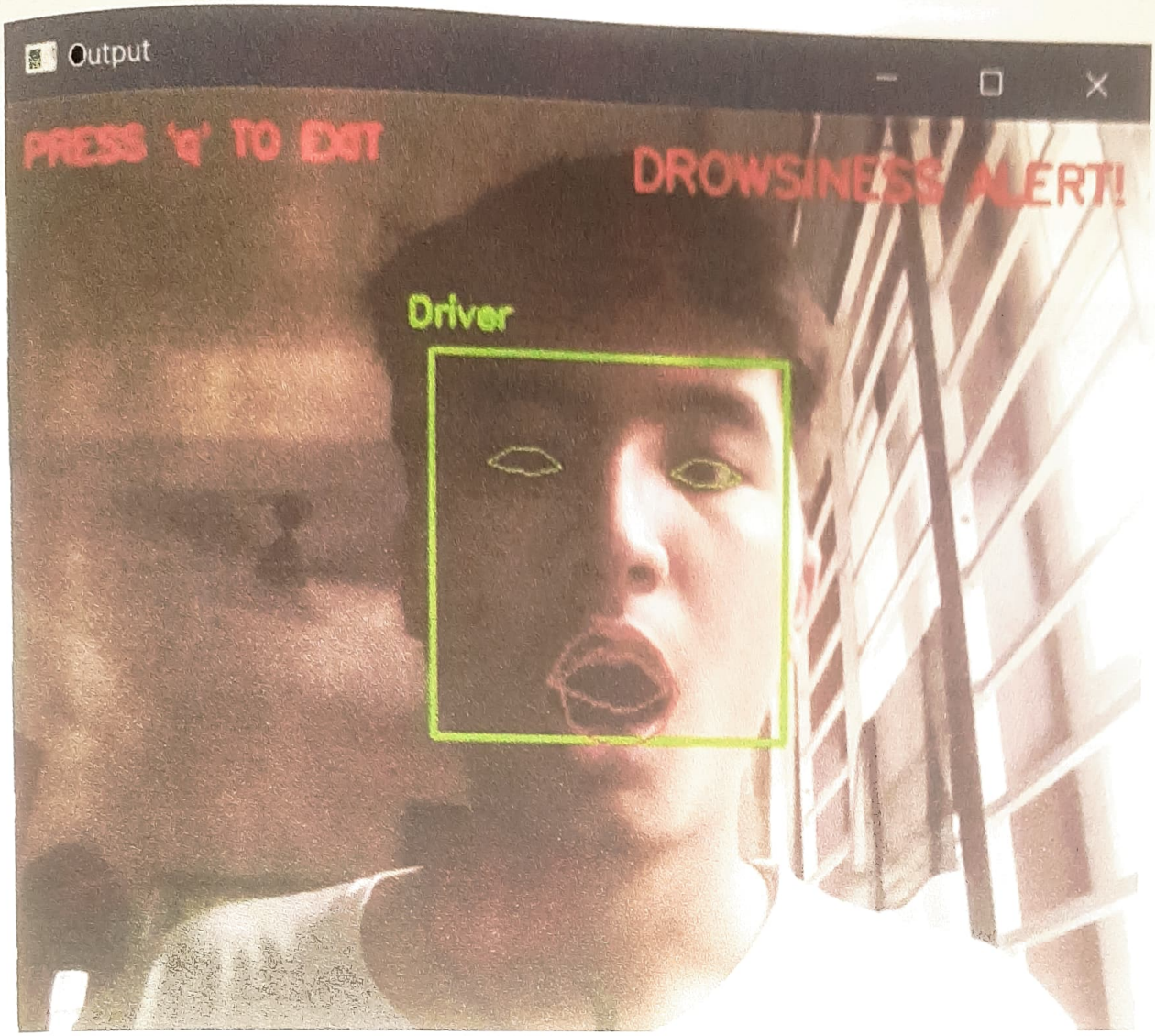
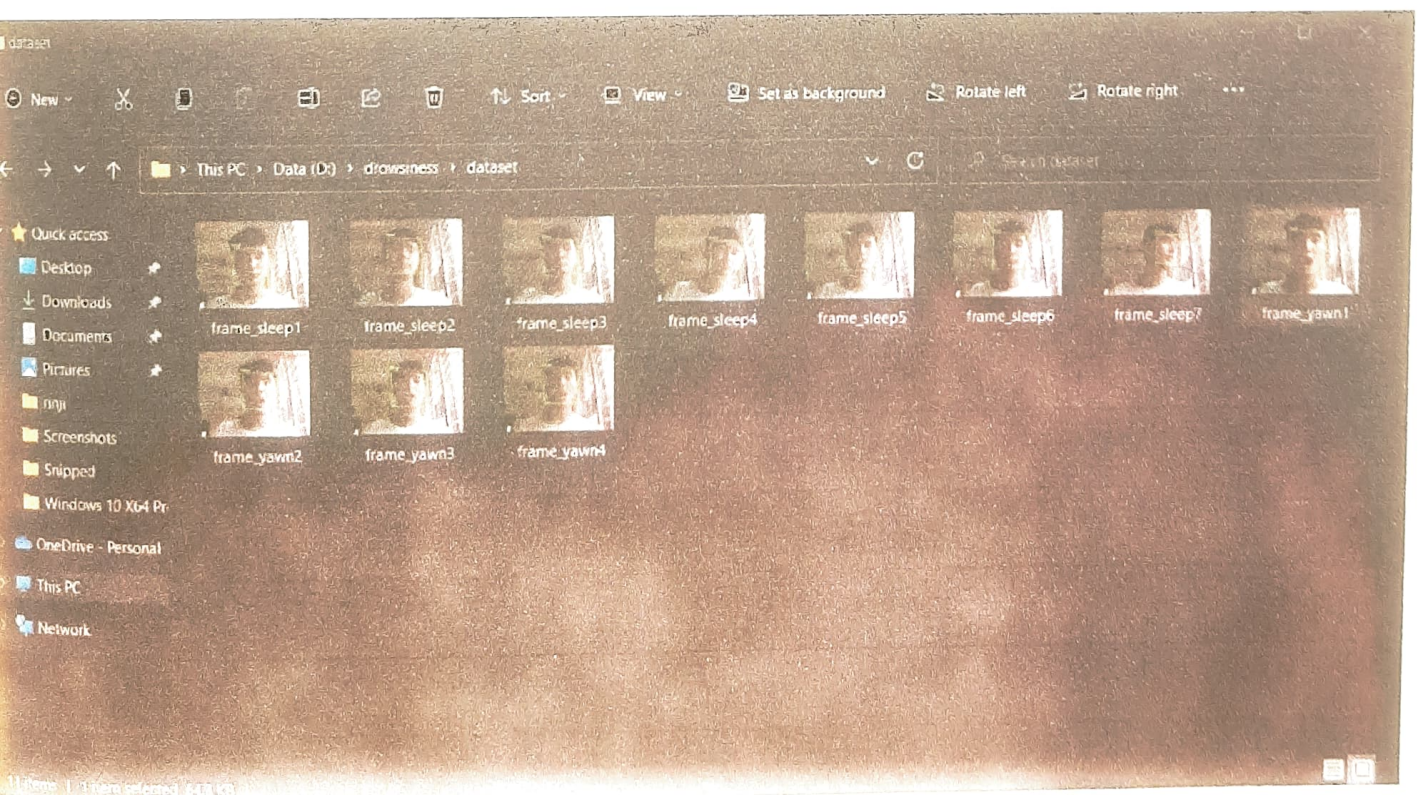


Figure 17: Detected Yawning and gave an alert

lso, in order to keep a proof of the moment when the person was sleeping or yawning, we ept a separate folder where those frames are stored as:



5.5. Streaming Using Phone Camera

We have used and Android App available for free in Play Store, named IP Webcam. After downloading it, open the app and scroll down to the option **Start Server**. It will look like:

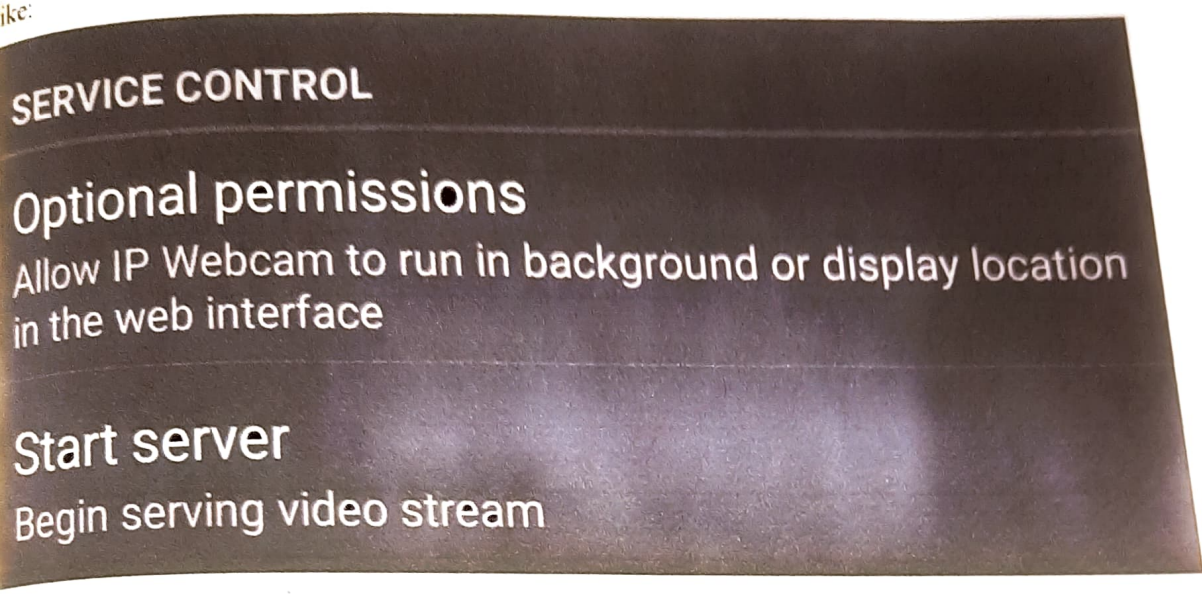


Figure 19: IP webcam Start Server

MAKE SURE THAT THE PHONE AND PC/LAPTOP IS CONNECTED TO THE SAME NETWORK.

Then, run the system in the same way as mentioned above. Click on the **Run Using Pho Cam** button to see the results:

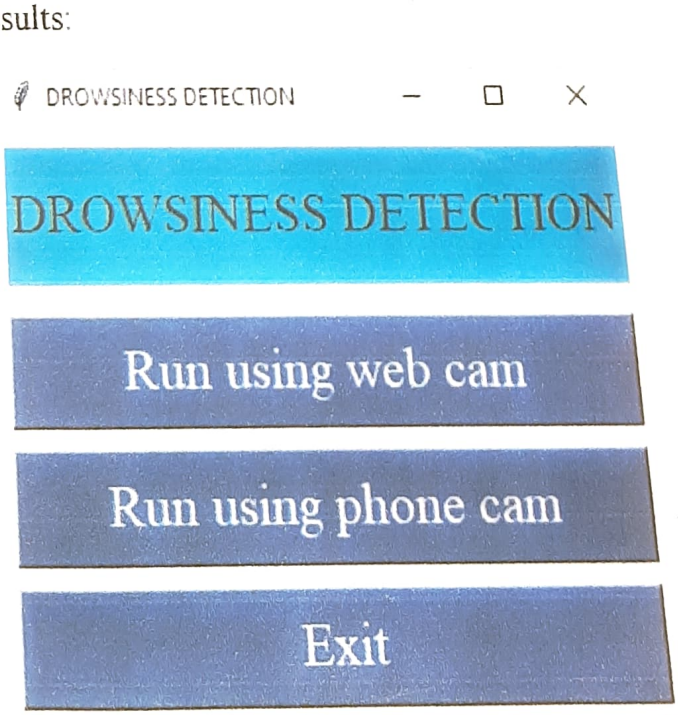


Figure 20: Drowsiness Detection Dialogue Box

Also, we have tried plotting the MAR and EAR graph Vs. Time in order to make the working clearer to the audience. The graph looks like:

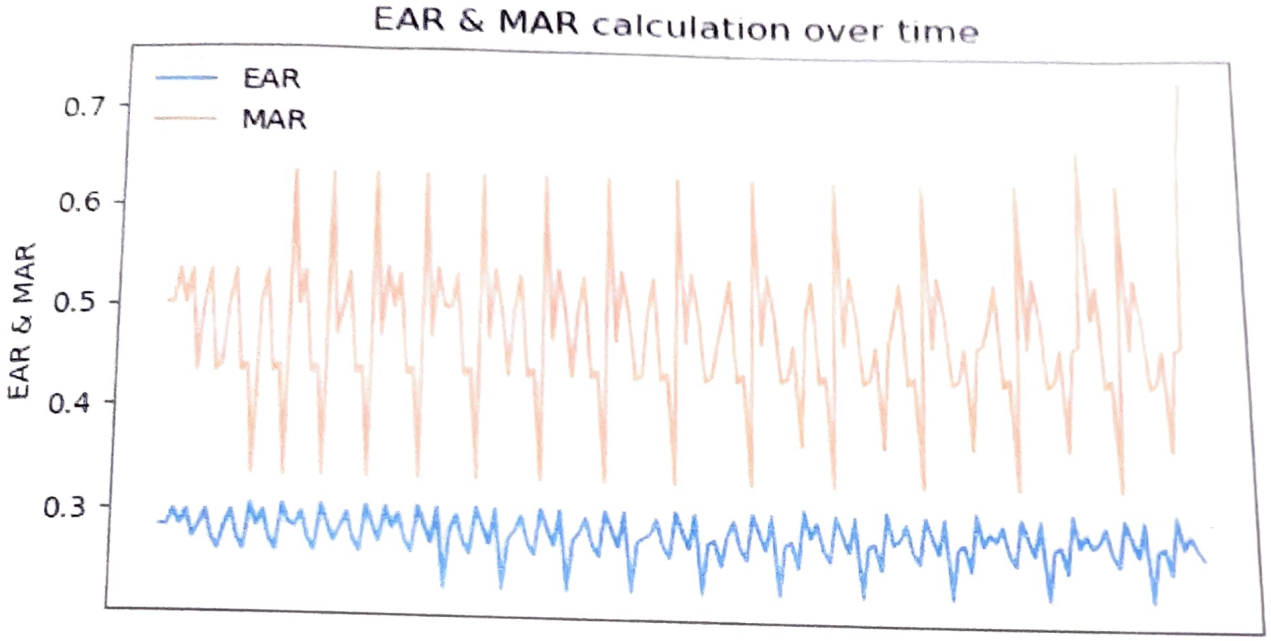


Figure 21: EAR & MAR Graph

CONCLUSION

The drowsiness detection system is an automated system that will detect the drowsiness of a driver. Here, using deep learning, the computerized system will detect the facial landmark of the driver through which we can detect the driver's drowsiness. Detect in terms of live video processing where we have fed the system with code and give the throughput as output. After the live video starts, the automated system will initiate the process. It will primarily detect the face, the eyes, and the mouth as input to the system. Whenever the driver feels drowsy, the system detects and takes the input and gives a buzzer or alert as an output to the driver. So, that accident is avoided. The driver can take a break and relax before moving ahead. So, it will reduce the risk of having accidents on the roads. We have developed this project that can be implemented on webcam and phone cameras. The project can also extend this work by implementing an IR webcam in the whole night light. It's a camera that uses infrared radiation to detect whether the person is exhausted. This research project has scope when it ultimately turns out to be developed into an application that the end-users can run on their own for their purposes on their systems. As a result, the accident ratio decreases. Hence, if commercially developed, our project will help save precious lives.

REFERENCE

[1] Bereshpolova Y, Stoelzel CR, Zhuang J, Amitai Y, Alonso JM, Swadlow HA. (2011) "Getting drowsy? Alert/no alert transitions and visual thalamocortical network dynamics". J Neurosci. 2011, 48: 17480-17487.

[2] Statistics from National Crime Record Bureau:
http://www.nhtsa.gov/staticfiles/nti/pdf/811449.pdf

[3] McKhann, G., Drachman, D., Folstein, M., Katzman, R., Price, D., & Stadlan, E. M. (1984). Clinical diagnosis of Alzheimer's disease Report of the NINCDS-ADRDA Work Group* under the auspices of Department of Health and Human Services Task Force on Alzheimer's Disease. Neurology, 34(7), 939-939.

[4] Definition of Deep learning from IBM cloud learning platform:
https://www.ibm.com/cloud/learn/deep-learning

[5] Python: https://www.tutorialspoint.com/python3/python_discussion.htm

[6] Python libraries: https://en.wikipedia.org/wiki/Main_Page

[7] Picot, S. Charbonnier, and A. Caplier, "On-line automatic detection of driver drowsiness using a single electroencephalographic channel," in Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE, 2008, pp. 3864-3867.

[8] G. Borghini, L. Astolfi, G. Vecchiato, D. Mattia, and F. Babiloni, "Measuring neurophysiological signals in aircraft pilots and car drivers for the assessment of mental workload, fatigue and drowsiness," Neuroscience & Biobehavioural Reviews, 2012.

[9] B. T. Jap, S. Lal, P. Fischer, and E. Bekiaris, "Using EEG spectral components to assess algorithms for detecting fatigue," Expert Systems with Applications, vol. 36, pp. 2352-2359, 2009.

[10] T. Danisman, I. M. Bilasco, C. Djeraba, and N. Ihaddadene, "Drowsy driver detection system using eye blink patterns," in Machine and Web Intelligence (ICMWI), 2010 International Conference on, 2010, pp. 230-233.

[11] D. Liu, P. Sun, Y. Xiao, and Y. Yin, "Drowsiness Detection Based on Eyelid Movement," in Education Technology and Computer Science (ETCS), 2010 Second International Workshop on, 2010, pp. 49-52.

[12] H. Seifoory, D. Taherkhani, B. Arzhang, Z. Eftekhari, and H. Memari, "An Accurate Morphological Drowsy Detection," ed: IEEE, 2011.

[13] Q. Wang, J. Yang, M. Ren, and Y. Zheng, "Driver fatigue detection: a survey," in Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on, 2006, pp. 8587-8591.

[14] D. J. McKnight, "Method and apparatus for displaying grey-scale or color images from binary images," ed: Google Patents, 1998.

[15] T. Welsh, M. Ashikhmin, and K. Mueller, "Transferring color to greyscale images," ACM Transactions on Graphics, vol. 21, pp. 277-280, 2002.

[16] D. F. Dinges and R. Grace, "PERCLOS: A valid psychophysiological measure of alertness as assessed by psychomotor vigilance," Federal Highway Administration. Office of motor carriers, Tech. Rep. MCRT-98-006, 1998.

[17] S. T. Lin, Y. Y. Tan, P. Y. Chua, L. K. Tey, and C. H. Ang, "PERCLOS Threshold for Drowsiness Detection during Real Driving," Journal of Vision, vol. 12, pp. 546-546, 2012.

[18] M. Saradadevi and P. Bajaj, "Driver fatigue detection using mouth and yawning analysis," IJCSNS International Journal of Computer Science and Network Security, vol. 8, pp. 183-188, 2008.

[19] Facial landmarks with Dlib, OpenCV and Python: https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-

[20] Eye blink detection with OpenCV, Python, and dlib: https://www.pyimagesearch.com/2017/04/24/eye-blink-detection-

[21] Drowsiness Detection with OpenCV: https://www.pyimagesearch.com/2017/05/08/drowsiness-detection-

[22] Real-Time Eye Blink Detection using Facial Landmarks: http://vision.fe.uni-lj.si/cvww2016/proceedings/papers/05.pdf

[23] Google: www.google.com