ABSTRACT

The main idea behind this project is to develop an on intrusive 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 inform of alarm and pop ups.

Moreover, the warning will be deactivated manually rather than automatically. For this purpose, a de-activation dialog will be generated which will contain some simple mathematical operation which when answered correctly will dismiss the warning. Moreover, if driver feels drowsy there is possibility of incorrect response to the dialog. We can judge this by plotting a graph in time domain. If all the three input variables show a possibility of fatigue at one moment, then a Warning signal is given in form of text and sound. This will directly give an indication of drowsiness/fatigue which can be further used as record of driver performance

CONTENTS

1.INTRODUCTION

Drowsy driving is one of the major causes of deaths occurring in road accidents. The truck drivers who drive for continuous long hours (especially at night), bus drivers of long distance route or overnight buses are more susceptible to this problem. Driver real time drowsiness is an overcast nightmare to passengers in every country. Every year, a large number of injuries and deaths occur due to fatigue related road accidents. Hence, detection of driver's fatigue and its indication is an active area of research due to its immense practical applicability. The basic real time drowsiness detection system has three blocks/modules; acquisition system, processing system and warning system.

Here, the video of the driver's frontal face is captured in acquisition system and transferred to the processing block where it is processed online to detect real time drowsiness. If real time drowsiness is detected, a warning or alarm is send to the driver from the warning system. Generally, the methods to detect drowsy drivers are classified in three types; vehicle based, behavioural based and physiological based. In vehicle based method, a number of metrics like steering wheel movement, accelerator or brake pattern, vehicle speed, lateral acceleration, deviations from lane position etc. are monitored continuously.Detection of any abnormal change in these values is considered as driver real time drowsiness. This is a nonintrusive measurement as the sensors are not attached on the driver. In behavioural based method the visual behavior of the driver i.e., eye blinking, eye closing, yawn, head bending etc. are analyzed to detect drowsiness.

This is also nonintrusive measurement as simple camera is used to detect these features. In physiological based method the physiological signals like Electrocardiogram , Electrooculogram , Electroencephalogram , heartbeat, pulse rate etc. are monitored and from these metrics,real time drowsiness or fatigue level is detected.This is intrusive measurement as the sensors are attached on the driver which will distract the driver. Depending on the sensors used in the system, system cost as well as size will increase. However, inclusion of more parameters/features will increase the accuracy of the system to a certain extent.

These factors motivate us to develop a low-cost, real time driver's drowsiness detection system with acceptable accuracy. Hence, we have proposed a webcam based system to detect driver's fatigue from the face image only using image processing and machine learning techniques to make the system low-cost as well as portable.

2. LITERATURE SURVEY

There are numerous past investigates in regards to driver sleepiness location frameworks that can be utilized as a source of perspective to build up a constant framework on identifying laziness for drivers. There are likewise a few techniques which utilize various ways to deal with distinguish the sluggishness signs.

As indicated by MIROS (Malaysia Institute of Road Safety), from the time of 2007 until 2010, there were 439 instances of street mishaps that have been explored by the MIROS Crash group. The part presents the writing overview of laziness identification draws near. As per the study on driver Fatigue-Drowsiness Detection framework, yawning inclination, squint of eyes territory extraction and so forth.

There are numerous examinations finished with an open CV for android likewise which is accessible for modest cell phones too. Different analyses directed have brought about most extreme exactness when the camera was arranged at various areas. OpenCV is overwhelmingly a method for ongoing picture preparing which has liberated from cost executions on most recent PC vision calculations.

AUTHOR	YEAR OF PUBLICATION	TITLE	METHODLOGY	INFERENCE
A.Sahayadhas, K.Sundaraj, M.Murugappa	2018	Detecting driver drowsiness based on sensors	Complete information process is done under algorithm and system compares to the value.	In this calculation, the framework holds data about the edges on the grounds that the eye squinting estimations from an assortment measure of edges are utilized to screen the tiredness
Y.Dong, Z.Hu, K.Uchimura and N.Murayama	2020	Driver inattentation Monitoring system for intelligent vehicles	He uses infrared rays to detect drowsiness and more than 80% of test results were passed	This licenses continuous languor identification and empower the framework interaction a whole 720*576frames at 16.7 miniature seconds
C.Billa, F.Sivrikaya, M.A. Khan and S. Albayrak	2020	Vehicles of the future	Detects driver particularly and records data and produces alarm	This permits real time drowsiness detection and enable the syste

2.1 DROWSINESS AND FATIGUE

Antoine Picotiter, expressed that tiredness is the place where an individual is in an alert and drowsy state. The present circumstance drives the driver to not focusing on their driving. Subsequently, the vehicle can presently don't be controlled because of the driver in a semi - cognizant state. As per Gianluca Borghini etal, mental exhaustion is a factor of tiredness and it causes the individual who encounters to not have the option to perform on the grounds that it diminishes the proficiency of the mind to react towards unexpected occasions.

2.2 ELECTROENCEPHALOGRAPHY (EEG) FOR DROWSINESS DETECTION

•Electroencephalography (EEC) is a strategy that actions the mind electrical movement. It tends to be utilized to quantify the heartbeat, eye squint and surprisingly major actual development, for example, head development. It very well may be utilized on people or creatures as subjects to get the cerebrum movement. It utilizes a unique equipment that places sensors around the highest point of the head territory to detect any electrical cerebrum action.

•Authors referenced that from the technique that has been carried out by the past specialist to identify sleepiness signs, the EEG strategy is ideal to be applied for tiredness and weariness recognition. In this technique, EEG has four sorts of recurrence segments that can be examined, i.e., alpha, beta and delta. At the point when the force is expanded in alpha and delta recurrence groups it shows that the driver is confronting weariness and sleepiness.

•The hindrances of this strategy are, it is exceptionally delicate to commotion around the sensors. For instance, when the individual is doing the EEG explore; the encompassing territory should be totally quiet. The commotion will meddle with the sensor that identifies the mind action. Another weakness of this technique is that regardless of whether the outcome may be precise it isn't reasonable to use for genuine driving application. Envision when an individual is driving and he is wearing something on his head brimming with wires and when the driver moves their head, the wire may take off from their places. Despite the fact that it isn't advantageous to be utilized for ongoing driving however for try purposes and information assortment, it is probably the best strategy up until now.

2.3 DROWSINESS DETECTION USING FACE DETECTION SYSTEM

Sleepiness can be distinguished by utilizing face territory recognition. The techniques to identify sleepiness inside the face territory shift because of languor. Sign in are more noticeable and clear to be identified at the face territory, we can identify the eyes area. From eyes identification, the creator expressed that there are four kinds of eyelid development that can be utilized for laziness location. They are totally open, total close, and in the center where the eyes are from open to close and the other way around.

The calculation technique is that the size space of the eye may shift starting with one individual then onto the next. Somebody may have little eyes and appears as though it is drowsy yet some are most certainly not. Other than that, if the individual is wearing glasses there is a hindrance to distinguish eye area. The pictures that are caught should be in a specific reach from the camera since when the distance is a long way from the camera, the pictures are obscured.

FIG 2.2: Drowsiness Condition

2.4 PERCLOS (PERCENTAGE OF EYE CLOSURE)

•Drowsiness can be caught by recognizing the eye squints and level of eye conclusion (PERCLOS). For eye squint identification, propose a strategy which learns the example of span of eyelid shut. As per this proposed technique estimates the ideal opportunity for an individual shut their eyes and on the off chance that they are shut longer than the ordinary.

Eye flicker time, it is conceivable that the individual is nodding off. It is referenced that eye flicker time, it is conceivable that the individual is nodding off. It is referenced that almost 310.3ms are the normal of an ordinary individual's eye squint.

FIG 2.3 : Fatigue Condition

Subsequent to going through the exploration papers and the current techniques, this task suggested that eyes and yawning location strategies will be utilized. Eye flicker term gives the information that the more extended the individual's nearby their eyes,the drowsier it.

2.5 EXISTING SYSTEM:

Traffic congestion is one of the major modern-day crises in every big city in the world. Previously different techniques had been proposed, such as infra-red light sensor, induction loop etc. to acquire traffic date which had their fair share of demerits. In recent years, image processing has shown promising outcomes in acquiring real time traffic information using CCTV footage installed along the traffic light. Different approaches have been proposed to glean traffic data. Some of them count total number of pixels, some of the work calculate number of vehicles. These methods have shown promising results in collecting traffic data. However, calculating the number of vehicles may give false results if the intra vehicular spacing is very small (two vehicles close to each other may be counted as one) and it may not count rickshaw or auto-rickshaw as vehicles which are the quotidian means of traffic especially in South-Asian countries.

Drowsiness of the driver detected system is developed. Many existing systems require a camera which is installed in front of driver. It points straight towards the face of the driver and monitors the drivers eyes in order to identify the drowsiness. For large vehicle such as heavy trucks and buses this arrangement is not pertinent.

CONS OF EXISTING SYSTEM :

Only 40% face can be detected. Not suitable for large vehicles

Disadvantage Of Existing System:

Traffic congestion is one of the head ach. Here using infra-red light sensor to detect traffic.

2.6 Proposed System:

In this paper, a system in which density of traffic is measured by comparing captured image with real time traffic information against the image of the empty road as reference image is proposed. Each lane will have a minimum amount of green signal duration allocated. According to the percentage of matching allocated traffic light duration can be controlled.

In the proposed work, pre-existing features for facial landmark detection is implemented to identify the state of drowsiness and fatigue. 68- facial landmark predefined landmark helps in shape prediction to clearly identify the various regions of the face like eye brows, eye, mouth region etc. Various change in parameters of these distinguished points reports various expression of the person. The facial landmark recognition is carrying out as

Advantages Of Proposed System:

Minimum amount of green signal duration allocated. According to the percentage of matching allocated traffic light duration can be controlled.

3. SYSTEM ANALYSIS

3.1 MODULES:

1. **Data Acquisition**
2. **Face Detection**
3. **Facial Landmark marking**
4. **Feature Extraction**
5. **Classification**

1. Data Acquisition :

The video is recorded using webcam (Sony CMU-BR300) and the frames are extracted and processed in a laptop. After extracting the frames, image processing techniques are applied on these 2D images. Presently, synthetic driver data has been generated. The volunteers are asked to look at the webcam with intermittent eye blinking, eye closing, yawning and head bending. The video is captured for 30 minutes duration.

2. Face Detection :

After extracting the frames, first the human faces are detected. Numerous online face detection algorithms are there. In this study, histogram of oriented gradients (HOG) and linear SVM method [10] is used. In this method, positive samples of descriptors are computed on them. Subsequently, negative samples (samples that do not contain the required object to be detected i.e., human face here) of same size are taken and HOG descriptors are calculated. Usually the number of negative samples is very greater than number of positive samples. After obtaining the features for both the classes, a linear SVM is trained for the classification task. To improve the accuracy of VM, hard negative mining is used. In this method, after training, the classifier is tested on the labeled data and the false positive sample feature values are used again for training purpose.

For the test image, the fixed size window is translated over the image and the classifier

computes the output for each window location. Finally, the maximum value output is considered as the detected face and a bounding box is drawn around the face. This non-maximum suppression step removes the redundant and overlapping bounding boxes.

3. Facial Landmark marking :-

After detecting the face, the next task is to find the locations of different facial features like the corners of the eyes and mouth, the tip of the nose and so on. Prior to that, the face images should be normalized in order to reduce the effect of distance from the camera, non-uniform illumination and varying image resolution. Therefore, the face image is resized to a width of 500 pixels and converted to grayscale image. After image normalization, ensemble of regression trees [11] is used to estimate the landmark positions on face from a sparse subset of pixel intensities. In this method, the sum of square error loss is optimized using gradient boosting learning. Different priors are used to find different structures. Using this method, the boundary points of eyes, mouth and the central line of the nose are marked and the number of points for eye, mouth and nose are given in Table I. The facial landmarks are shown in Fig 2. The red points are the detected landmarks for further processing.

4. Feature Extraction

After detecting the facial landmarks, the features are computed as described below. Eye aspect ratio (EAR): From the eye corner points, the eye aspect ratio is calculated as the ratio of height and width of the eye as given by

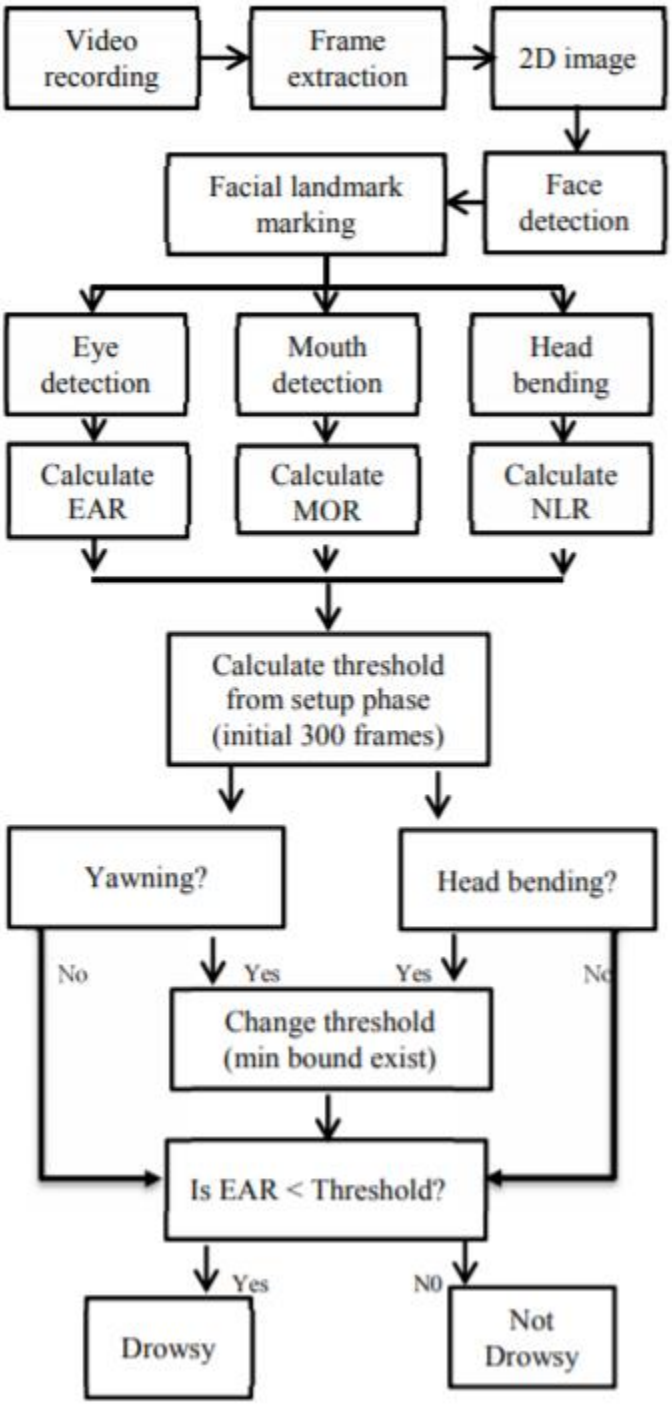
Eye Aspect Ratio Calculation

$$\mathbf{EAR} = \frac{\|\boldsymbol{P2}-\boldsymbol{P6}\|+\|\boldsymbol{P3}-\boldsymbol{P5}\|}{2\|\boldsymbol{P1}-\boldsymbol{P4}\|}$$

5. Classification :-

After computing all the three features, the next task is to detect drowsiness in the extracted frames. In the beginning, adaptive thresholding is considered for classification. Later, machine learning algorithms are used to classify the data. For computing the threshold values for each feature, it is assumed that initially the driver is in complete awake state. This is called setup phase. In the setup phase, the EAR values for first three hundred (for 10s at 30 fps) frames are recorded. Out 4of these three hundred initial frames containing face, average of 150 maximum values is considered as the hard threshold for EAR. The higher values are considered so that no eye closing instances will be present. If the test value is less than this threshold, then eye closing (i.e., drowsiness) is detected. As the size of eye can vary from person to person, this initial setup for each person will reduce this effect. Similarly, for calculating threshold of MOR, since the mouth may not be open to its maximum in initial frames (setup phase) so the threshold

3.2 System Architecture :



3.3 Software Requirement Specification :

A Software Requirements Specification (SRS) – a requirements specification for a software system is a complete description of the behavior of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. Nonfunctional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

Operation Feasibility

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization's operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. This system is targeted to be in accordance with the above-mentioned issues.

3.4 Software Requirements:

- IDE : Pycharm
- Programming Language : Python 3.7
- Framework : Flask ,Computer Vision

3.5 Hardware Requirements:

Hardware System Requirements often specify the operating system version , process type , memory , size , available disk space and additional peripherals , if any needed

- Hard Disk : 120 GB.
- Processor : Rasberry pi
- RAM : 1 GB

4. SYSTEM DESIGN

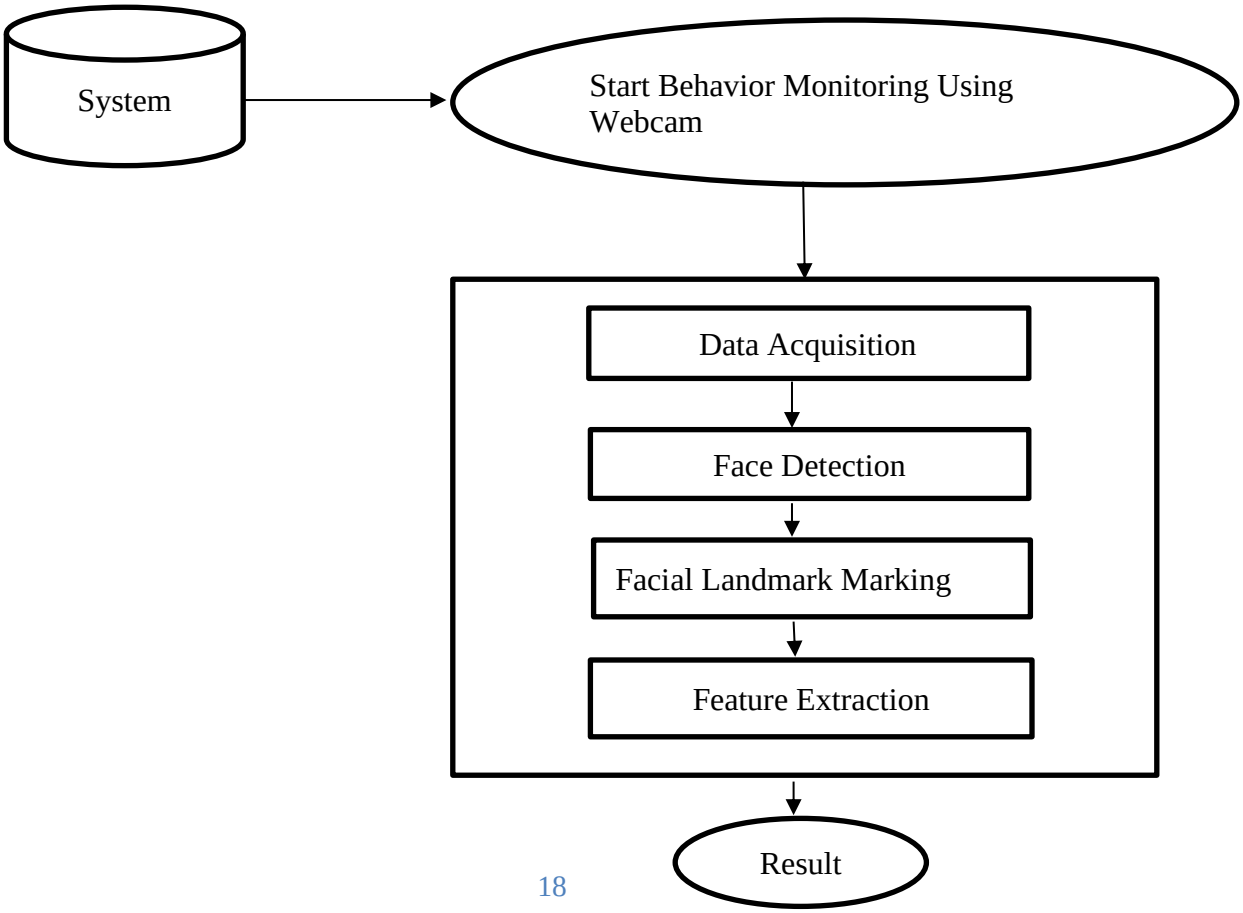
4.1 Data Flow Diagram:

The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.

The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.

DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.

DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.



4.2 Uml Diagrams:

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

Goals:

The Primary goals in the design of the UML are as follows:

Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.

Provide extendibility and specialization mechanisms to extend the core concepts.

Be independent of particular programming languages and development process.

Provide a formal basis for understanding the modeling language.

Encourage the growth of OO tools market.

Integrate best practices.

4.2.1: Use Case Diagram:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

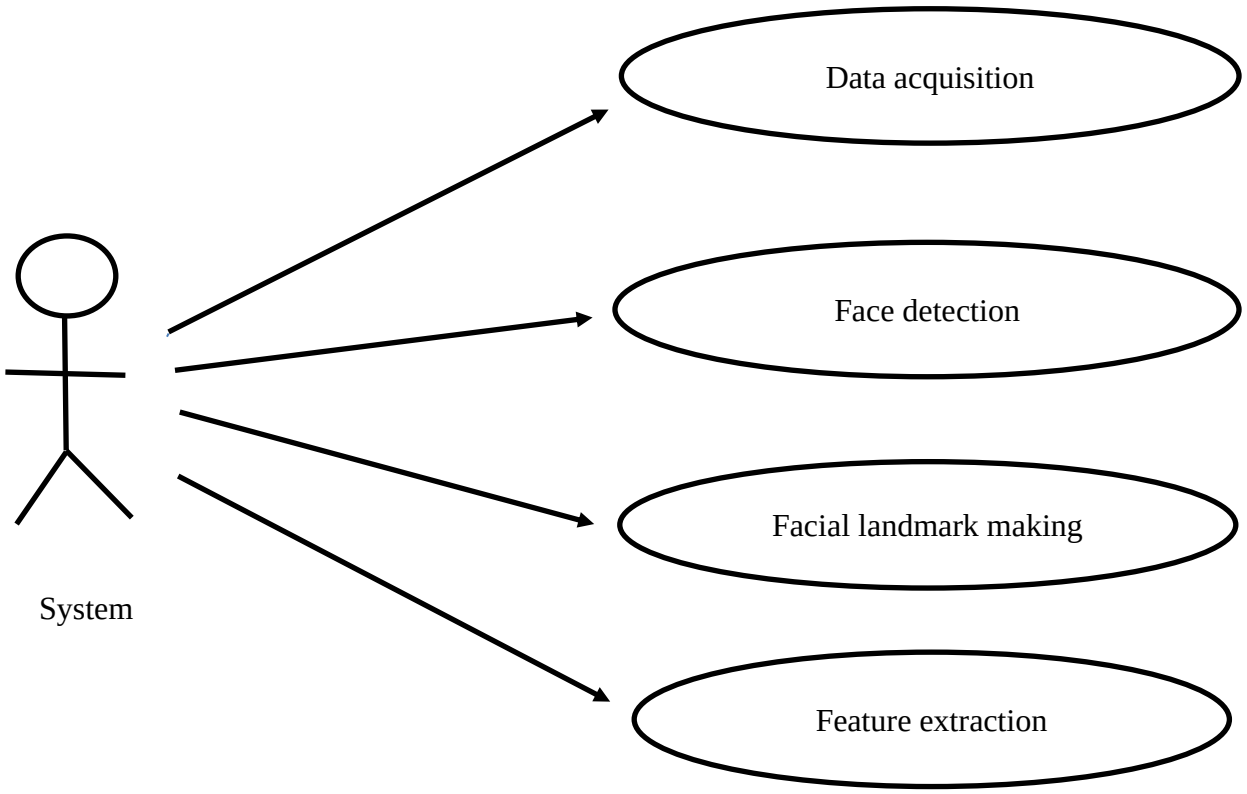


Figure: Use Class Diagram of driver drowsiness

4.2.2: Class Diagram:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

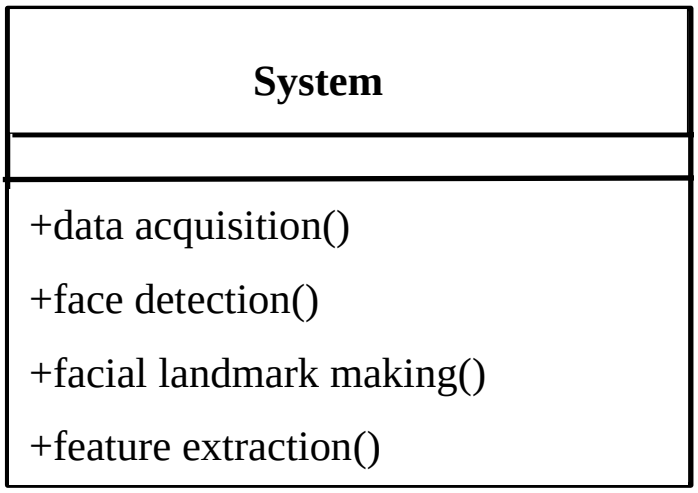


Figure: Class Diagram of driver drowsiness Detection

4.2.3: Sequence Diagram:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

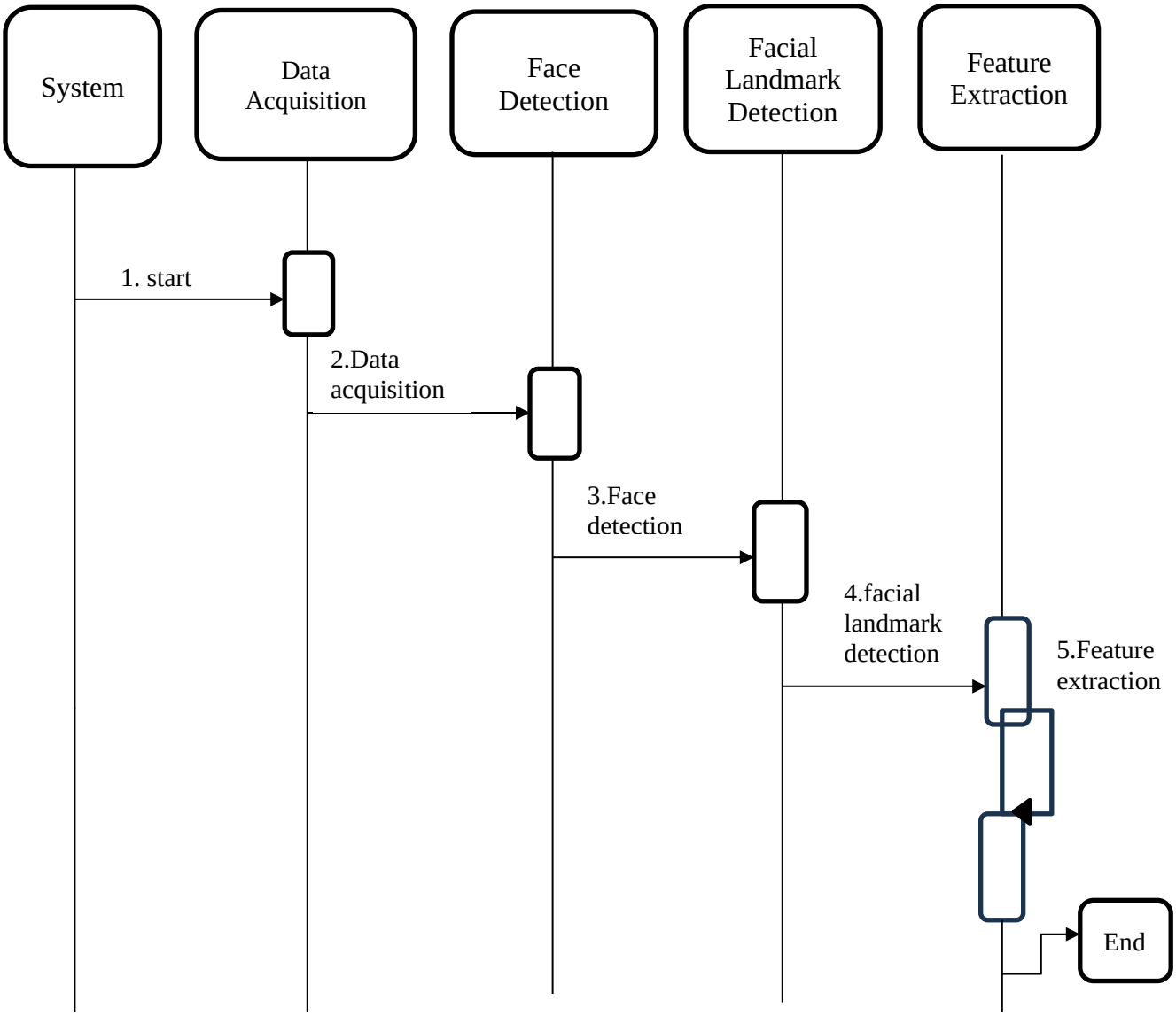


Figure: Sequence Diagram of driver drowsiness Detection

4.2.4: Activity Diagram:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

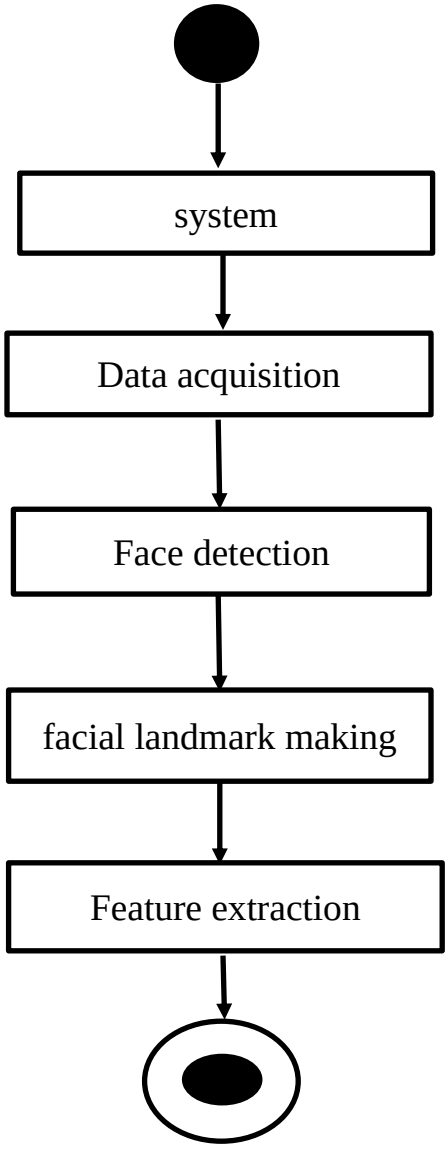


Figure: Activity Diagram of driver drowsiness Detection

5. IMPLEMENTATION

The implementation phase is less creative than system design. It is primarily concerned with user training, and file conversion. The system may be requiring extensive user training. The initial parameters of the system should be modifies as a result of a programming. A simple operating procedure is provided so that the user can understand the different functions clearly and quickly. The different reports can be obtained either on the inkjet or dot matrix printer, which is available at the disposal of the user. The proposed system is very easy to implement. In general implementation is used to mean the process of converting a new or revised system design into an operational one.

5.1 Front End Software Features

5.1.1 Python

Python is a general-purpose language. It has wide range of applications from Web development (like: Django and Bottle), scientific and mathematical computing (Orange, SymPy, NumPy) to desktop graphical user Interfaces (Pygame, Panda3D). The syntax of the language is clean and length of the code is relatively short. It's fun to work in Python because it allows you to think about the problem rather than focusing on the syntax.

Python is a good programming language for beginners. It is a high-level language, which means a programmer can focus on what to do instead of how to do it. Writing programs in Python takes less time than in some other languages.

Python drew inspiration from other programming languages like C, C++, Java, Perl, and Lisp.

Python has a very easy-to-read syntax. Some of Python's syntax comes from C, because that is the language that Python was written in. But Python uses whitespace to delimit code: spaces or tabs are used to organize code into groups.

5.1.2 History of Python:

Python is a high-level, interpreted scripting language developed in the late 1980s by Guido van Rossum at the National Research Institute for Mathematics and Computer Science in the Netherlands. The initial version was published at the alt. Sources newsgroup in 1991, and version 1.0 was released in 1994.

Python 2.0 was released in 2000, and the 2.x versions were the prevalent releases until December 2008. At that time, the development team made the decision to release version 3.0, which contained a few relatively small but significant changes that were not backward compatible with the 2.x versions. Python 2 and 3 are very similar, and some features of Python 3 have been back ported to Python 2. But in general, they remain not quite compatible.

Both Python 2 and 3 have continued to be maintained and developed, with periodic release updates for both. As of this writing, the most recent versions available are 2.7.15 and 3.6.5. However, an official End of Life date of January 1,2020 has been established for Python 2, after which time it will no longer be maintained. If you are a newcomer to Python, it is recommended that you focus on Python 3, as this tutorial will do.

Python is still maintained by a core development team at the Institute, and Guido is still in charge, having been given the title of BDFL (Benevolent Dictator For Life) by the Python community. The name Python, by the way, derives not from the snake, but from the British comedy troupe Monty Python'sFlying Circus of which Guido was, and presumably still is, a fan. It is common to find references to Monty Python sketches and movies scattered throughout the Python documentation.

5.1.3 Why Python was created?

In late 1980s, Guido Van Rossum was working on the Amoeba distributed operating system group. He wanted to use an interpreted language like ABC (ABC has simple easy-to-understand syntax) that could access the Amoeba system calls. So, he decided to create a language that was extensible. This led to design of a new language which was later named Python. 25

Python 2.0 was released in 2000, and the 2.x versions were the prevalent releases until December 2008. At that time, the development team made the decision to release version 3.0, which contained a few relatively small but significant changes that were not backward compatible with the 2.x versions. Python 2 and 3 are very similar, and some features of Python 3 have been back ported to Python 2. But in general, they remain not quite compatible.

5.1.4 Why the name Python?

No. It wasn't named after a dangerous snake. Rossum was fan of a comedy series from late seventies. The name "Python" was adopted from the same series "Monty Python's Flying Circus".

5.1.5 Features of Python:

A simple language which is easier to learn

Python has a very simple and elegant syntax. It's much easier to read and write Python programs compared to other languages like: C++, Java, C#. Python makes programming fun and allows you to focus on the solution rather than syntax. If you are a newbie, it's a great choice to start your journey with Python.

Free and open-source

You can freely use and distribute Python, even for commercial use. Not only can you use and distribute softwares written in it, you can even make changes to the Python's source code. Python has a large community constantly improving it in each iteration.

Portability

You can move Python programs from one platform to another, and run it without any changes. It runs seamlessly on almost all platforms including Windows, Mac OS X and Linux.

Extensible and Embeddable

Suppose an application requires high performance. You can easily combine pieces of C/C++ or other languages with Python code. This will give your application high performance as well as scripting capabilities which other languages may not provide out of the box.

A high-level, interpreted language

Unlike C/C++, you don't have to worry about daunting tasks like memory management, garbage collection and so on.

Large standard libraries to solve common tasks

Python has a number of standard libraries which makes life of a programmer much easier since you don't have to write all the code yourself. For example: Need to connect MySQL database on a Web server? You can use MySQLdb library using import MySQLdb . Standard libraries in Python are well tested and used by hundreds of people. So you can be sure that it won't break your application.

Object-oriented

Everything in Python is an object. Object oriented programming (OOP) helps you solve a complex problem intuitively. With OOP, you are able to divide these complex problems into smaller sets by creating objects.

5.1.6 Applications of Python:

1. Simple Elegant Syntax

Programming in Python is fun. It's easier to understand and write Python code. Why? The syntax feels natural. Take this source code for an example:

```
a = 2
b = 3
sum = a + b
print(sum)
```

2. Not overly strict

You don't need to define the type of a variable in Python. Also, it's not necessary to add semicolon at the end of the statement. Python enforces you to follow good practices (like proper indentation). These small things can make learning much easier for beginners.

3. Expressiveness of the language

Python allows you to write programs having greater functionality with fewer lines of code. Here's a link to the source code of Tic-tac-toe game with a graphical interface and a smart computer opponent in less than 500 lines of code. This is just an example. You will be amazed how much you can do with Python once you learn the basics.

4.Great Community and Support

Python has a large supporting community. There are numerous active forums online which can be handy if you are stuck.

Python is currently the most widely used multi-purpose, high-level programming language.

Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google,

Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following

Machine Learning

GUI Applications (like Kivy, Tkinter, PyQt etc.)

Web frameworks like Django (used by YouTube, Instagram, Dropbox)

Image processing (like Opencv, Pillow)

Web scraping (like Scrapy, BeautifulSoup, Selenium)

Test frameworks

Multimedia

Advantages of Python:

1. Extensive Libraries

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don't have to write the complete code for that manually.

2. Extensible

As we have seen earlier, Python can be **extended to other languages**. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

3. Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add **scripting capabilities** to our code in the other language.

4. Improved Productivity

The language's simplicity and extensive libraries render programmers **more productive** than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

4. IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

5. Simple and Easy

When working with Java, you may have to create a class to print **'Hello World'**. But in Python, just a print statement will do. It is also quite **easy to learn, understand,** and **code.** This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

6. Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and **indentation is mandatory.** This further aids the readability of the code.

7. Object-Oriented

This language supports both the **procedural and object-oriented** programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the **encapsulation of data** and functions into one.

8. Free and Open-Source

Like we said earlier, Python is **freely available.** But not only can you **download Python** for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

9. Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn't the same with Python. Here, you need to **code only once**, and you can run it anywhere. This is called **Write Once Run Anywhere (WORA)**. However, you need to be careful enough not to include any system-dependent features.

10. Interpreted

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, **debugging is easier** than in compiled languages.

Advantages of Python Over Other Languages

1. Less Coding

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don't have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

2. Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

3. Python is for Everyone

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and **machine learning**, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

Disadvantages of Python

1. Speed Limitations

We have seen that Python code is executed line by line. But since Python is interpreted, it often results in **slow execution**. This, however, isn't a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

2. Weak in Mobile Computing and Browsers

While it serves as an excellent server-side language, Python is much rarely seen on the **client-side**. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called **Carbonnelle**.

3. Design Restrictions

As you know, Python is **dynamically-typed**. This means that you don't need to declare the type of variable while writing the code. It uses **duck-typing**. But wait, what's that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can **raise run-time errors**.

4. Underdeveloped Database Access Layers

Compared to more widely used technologies like **JDBC (Java DataBase Connectivity)** and **ODBC (Open DataBase Connectivity)**, Python's database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

5. Simple

No, we're not kidding. Python's simplicity can indeed be a problem. Take my example. I don't do Java, I'm more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

5.2 SAMPLE CODE

```
from tkinter import *
import tkinter
from scipy.spatial import distance as dist
from imutils import face_utils
import numpy as np
import imutils
import dlib
import cv2
import winsound
main = tkinter.Tk()
main.title("Driver Drowsiness Monitoring")
main.geometry("500x400")

def EAR(drivereye):
    point1 = dist.euclidean(drivereye[1], drivereye[5])
    point2 = dist.euclidean(drivereye[2], drivereye[4])
    # compute the euclidean distance between the horizontal
    distance = dist.euclidean(drivereye[0], drivereye[3])
    # compute the eye aspect ratio
    ear_aspect_ratio = (point1 + point2) / (2.0 * distance)
    return ear_aspect_ratio

def MOR(drivermouth):
    # compute the euclidean distances between the horizontal
    point   = dist.euclidean(drivermouth[0], drivermouth[6])
    # compute the euclidean distances between the vertical
    point1  = dist.euclidean(drivermouth[2], drivermouth[10])
    point2  = dist.euclidean(drivermouth[4], drivermouth[8])
    # taking average
```

```
Ypoint   = (point1+point2)/2.0
# compute mouth aspect ratio
mouth_aspect_ratio = Ypoint/point
return mouth_aspect_ratio

def startMonitoring():
    pathlabel.config(text="         Webcam Connected Successfully")
    webcamera = cv2.VideoCapture(0)
    svm_predictor_path = 'SVMclassifier.dat'
    EYE_AR_THRESH = 0.25
    EYE_AR_CONSEC_FRAMES = 5
    MOU_AR_THRESH = 0.75

    COUNTER = 0
     yawnStatus = False
     yawns = 0
     svm_detector = dlib.get_frontal_face_detector()
     svm_predictor = dlib.shape_predictor(svm_predictor_path)
     (lStart, lEnd) = face_utils.FACIAL_LANDMARKS_IDXS["left_eye"]
     (rStart, rEnd) = face_utils.FACIAL_LANDMARKS_IDXS["right_eye"]
     (mStart, mEnd) = face_utils.FACIAL_LANDMARKS_IDXS["mouth"]
     while True:
         ret, frame = webcamera.read()
         frame = imutils.resize(frame, width=640)
         gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
         prev_yawn_status = yawnStatus
         rects = svm_detector(gray, 0)
        for rect in rects:
             shape = svm_predictor(gray, rect)
             shape = face_utils.shape_to_np(shape)
              leftEye = shape[lStart:lEnd]
```

```
 rightEye = shape[rStart:rEnd]
 mouth = shape[mStart:mEnd]
 leftEAR = EAR(leftEye)
 rightEAR = EAR(rightEye)
 mouEAR = MOR(mouth)
ear = (leftEAR + rightEAR) / 2.0
leftEyeHull = cv2.convexHull(leftEye)
rightEyeHull = cv2.convexHull(rightEye)
mouthHull = cv2.convexHull(mouth)
cv2.drawContours(frame, [leftEyeHull], -1, (0, 255, 255), 1)
cv2.drawContours(frame, [rightEyeHull], -1, (0, 255, 255), 1)
cv2.drawContours(frame, [mouthHull], -1, (0, 255, 0), 1)
if ear < EYE_AR_THRESH:
    COUNTER += 1
cv2.putText(frame, "Eyes Closed ", (10, 30),cv2.FONT_HERSHEY_SIMPLEX,
0 .7, (0, 0, 255), 2)
            if COUNTER >= EYE_AR_CONSEC_FRAMES:
            winsound.Beep(440, 500);
          cv2.putText(frame, "DROWSINESS ALERT!", (10,70) cv2   .FONT
         ERSHEY SIMPLEX, 0.7, (0, 0, 255), 2)
   else:
   COUNTER = 0
 cv2.putText(frame, "Eyes Open ", (10, 30),cv2.FONT_HERSHEY_SIMPLEX,
0.7, (0, 255, 0), 2)
       cv2.putText(frame, "EAR: {:.2f}".format(ear), (480,30), cv2. FONT_
     HERSHEEY_SIMPLEX, 0.7, (0, 0, 255), 2)
            if mouEAR > MOU_AR_THRESH:
cv2.putText(frame, "Yawning, DROWSINESS ALERT! ", (10,
   70),cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
            yawnStatus = True
            output_text = "Yawn Count: " + str(yawns + 1)
```

```
                cv2.putText(frame, output_text,   (10,100),cv2.FONT_HERSHEY_SIMPLEX,
                                                0.7,(255,0,0),2)
                    else:
                        yawnStatus = False
                     if prev_yawn_status == True and yawnStatus == False:
                        yawns+=1
                cv2.putText(frame, "MAR: {:.2f}".format(mouEAR), (480,
                60),cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
cv2.putText(frame,"Visual Behaviour & Machine Learning Drowsiness   Detection @
   Drowsiness",(370,470),cv2.FONT_HERSHEY_COMPLEX,0.6,(153,51,102),1)
                 cv2.imshow("Frame", frame)
                  key = cv2.waitKey(1) & 0xFF
                 if key == ord("q"):
                        break
                 cv2.destroyAllWindows()
                webcamera.release()
```

UI CODE

```
font = ('times', 16, 'bold')
 title = Label(main, text='Driver Drowsiness Monitoring System using Visual\n
 Behaviour and Machine Learning',anchor=W, justify=LEFT)
 title.config(bg='black', fg='white')
 title.config(font=font)
 title.config(height=3, width=120)
 title.place(x=0,y=5)
 font1 = ('times', 14, 'bold')
 upload = Button(main, text="Start Behaviour Monitoring Using Webcam",
 command=startMonitoring)
 upload.place(x=50,y=200)
 upload.config(font=font1)
 pathlabel = Label(main)
 pathlabel.config(bg='DarkOrange1', fg='white')
 pathlabel.config(font=font1)
 pathlabel.place(x=50,y=250)
 main.config(bg='chocolate1')
 main.mainloop()
```

6. TESTING

6.1 Test Strategy and Approach :

All functional tests will have a detailed documentation where as the field testing will be done.

Field testing will be performed manually and functional tests will be written in detail.

Test objectives:

All field entries must work properly.

Pages must be activated from the identified link.

The entry screen, messages and responses must not be delayed.

Features to be tested

Verify that the entries are of the correct format

No duplicate entries should be allowed

All links should take the user to the correct page.

TESTING:

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assembles, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

Input Design:

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

What data should be given as input?

How the data should be arranged or coded?

The dialog to guide the operating personnel in providing input.

Methods for preparing input validations &steps to follow when error occur.

Objectives:

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

Output Design:

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following

objectives.

Convey information about past activities, current status or projections of the Future.

Signal important events, opportunities, problems, or warnings.

Trigger an action.

Confirm an action.

6.2 Types of Tests

6.2.1 Unit Testing:

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

6.2.2 Functional test:

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or

special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

6.2.3 System Test:

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

6.2.4 Integration testing:

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications,

e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

6.2.5 Black Box Testing:

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

6.2.6 White Box Testing:

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

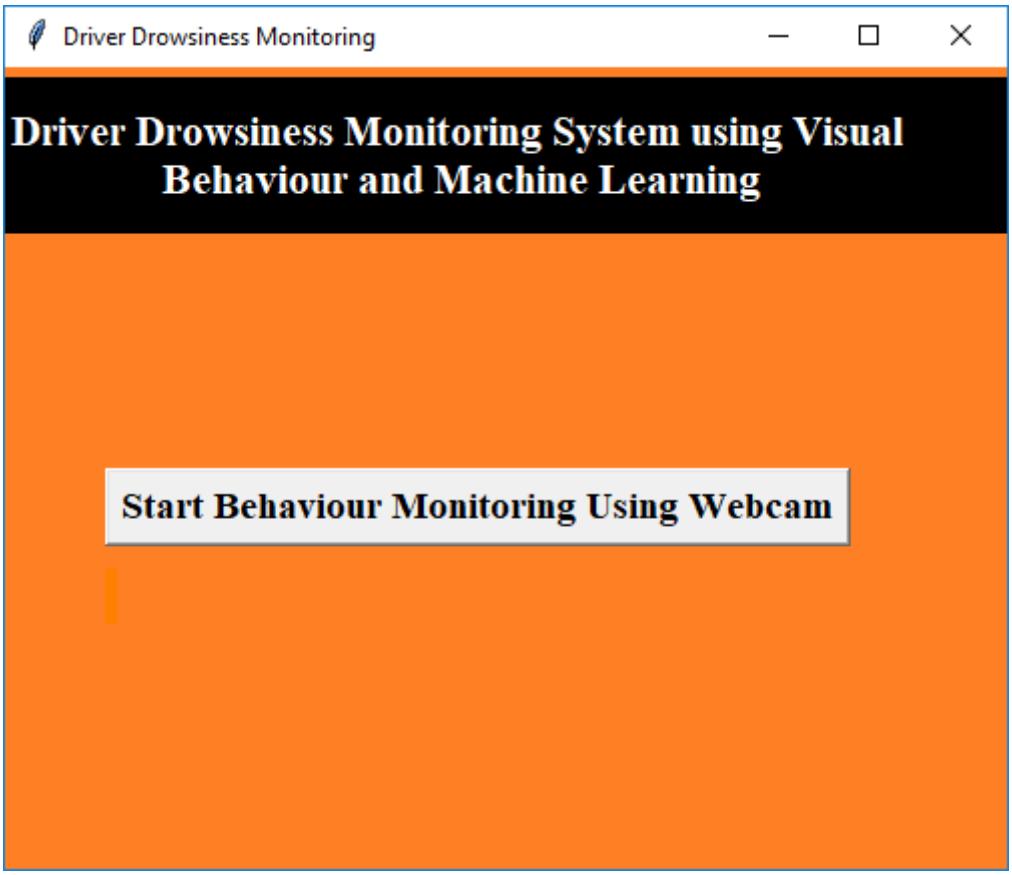
6.2.7 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

7. RESULTS

To run this project double click on ‘run.bat’ file to get below screen



In above screen click on ‘Start Behaviour Monitoring Using Webcam’ button to connect application with webcam, after clicking button will get below screen with webcam streaming

PICTURE

Fig.7.1:Start Behavior Monitoring Using Webcam

In above screen we can see web cam stream then application monitor all frames to see person eyes are open or not, if closed then will get below message

PICTURE

Fig.7.2:Person Eyes Closed

Similarly if mouth starts yawn then also will get alert message

PICTURE

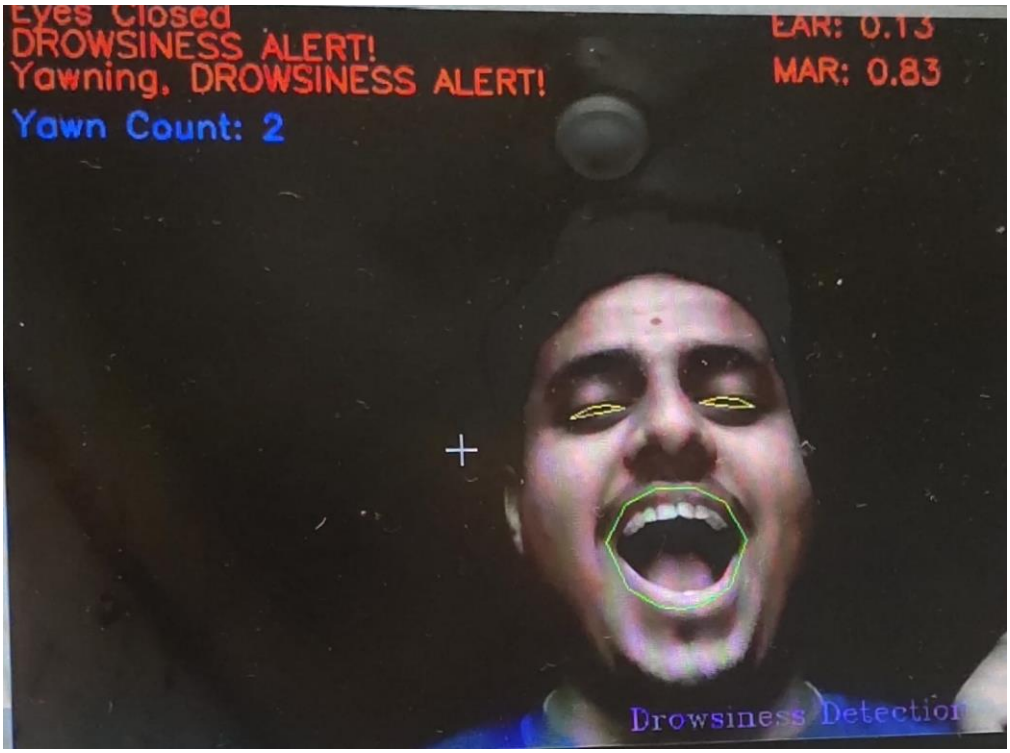


Fig.7.3:Mouth Starts Yawn

CONCLUSION

In this paper, a low cost, real time driver drowsiness monitoring system has been proposed based on visual behavior and machine learning. Here, visual behavior features like eye aspect ratio, mouth opening ratio and nose length ratio are computed from the streaming video, captured by a webcam. An adaptive thresholding technique has been developed to detect driver drowsiness in real time. The developed system works accurately with the generated synthetic data. Subsequently, the feature values are stored and machine learning algorithms have been used for classification. Bayesian classifier, FLDA and SVM have been explored here. It has been observed that FLDA and SVM outperform Bayesian classifier. The sensitivity of FLDA and SVM is 0.896 and 0.956 respectively whereas the specificity is 1 for both. As FLDA and SVM give better accuracy, work will be carried out to implement them in the developed system to do the classification (i.e., drowsiness detection) online. Also, the system will be implemented in hardware to make it portable for car system and pilot study on drivers will be carried out to validate the developed system.

BIBILOGRAPHY

- **Tony Gaddis,** Starting Out with Python (3e)
- **Kenneth A. Lambert,** Fundamentals of Python
- **James Payne,** Beginning Python using Python 2.6 and 3
- Python 3.11.4 documentation, https://docs python.org/3/

REFERENCES

[1] Y. Dong, Z. Hu, K. Uchimura, and N. Murayama, “Driver inattention monitoring system for intelligent vehicles: A review,” IEEE Trans. Transp. Syst., vol. 12, no. 2, pp. 596–614, Jun. 2020

[2] C. Bila, F. Sivrikaya, M. A. Khan, and S. Albayrak, “Vehicles of the future: A survey of research on safety issues,” IEEE Trans. Intell. Transp. Syst., vol. 18, no. 5, pp. 1046–1065, 2020.

[3] National Highway Traffic Safety Administration. “Traffic safety facts crash stats: Drowsy driving 2019,” Oct. 2017. [Online]. Available: http://www.nhtsa.gov/riskydriving/drowsy-driving

[4] W. L. Ou, M. H. Shih, C. W. Chang, X. H. Yu, C. P. Fan, "Intelligent Video-Based Drowsy Driver Detection System under Various Illuminations and Embedded Software Implementation", 2015 international Conf. on Consumer Electronics - Taiwan, 2015.

[5] W. B. Horng, C. Y. Chen, Y. Chang, C. H. Fan, “Driver Fatigue Detection based on Eye Tracking and Dynamic Template Matching”, IEEE International Conference on Networking,, Sensing and Control, Taipei, Taiwan, March 21-23, 2004.

[6] S. Singh, N. P. papanikolopoulos, “Monitoring Driver Fatigue using Facial Analysis Techniques”, IEEE Conference on Intelligent Transportation System, pp 314-318.

[7] B. Alshaqaqi, A. S. Baquhaizel, M. E. A. Ouis, M. Bouumehed, A. Ouamri, M. Keche, “Driver Drowsiness Detection System”, IEEE International Workshop on Systems, Signal Processing and their Applications, 2013.

[8] M. Karchani, A. Mazloumi, G. N. Saraji, A. Nahvi, K. S. Haghighi, B. M. Abadi, A. R. Foroshani, A. Niknezhad, “The Steps of Proposed Drowsiness Detection System Design based on Image Processing in Simulator Driving”, International Research Journal of Applied and Basic Sciences, vol. 9(6), pp 878-887, 2015.

[9] R. Ahmad, and J. N. Borole, “Drowsy Driver Identification Using Eye Blink Detection,” IJISET - International Journal of Computer Science and Information Technologies, vol. 6, no. 1, pp. 270-274, Jan. 2015.

[10] A. Abas, J. Mellor, and X. Chen, “Non-intrusive drowsiness detection by employing Support Vector Machine,” 2014 20th International Conference on Automation and Computing (ICAC), Bedfordshire, UK, 2014, pp. 188- 193.

[11] A. Sengupta, A. Dasgupta, A. Chaudhuri, A. George, A. Routray, R. Guha; "A Multimodal System for Assessing Alertness Levels Due to Cognitive Loading", IEEE Trans. on Neural Systems and Rehabilitation Engg., vol. 25 (7), pp 1037-1046, 2017.

[12] K. T. Chui, K. F. Tsang, H. R. Chi, B. W. K. Ling, and C. K. Wu, “An accurate ECG based transportation safety drowsiness detection scheme,” IEEE Transactions on Industrial Informatics, vol. 12, no. 4, pp. 1438- 1452, Aug. 2016.

[13] V. Kazemi and J. Sullivan; "One millisecond face alignment with an ensemble of regression trees", IEEE Conf. on Computer Vision and Pattern Recognition, 23-28 June, 2014, Columbus, OH, USA.